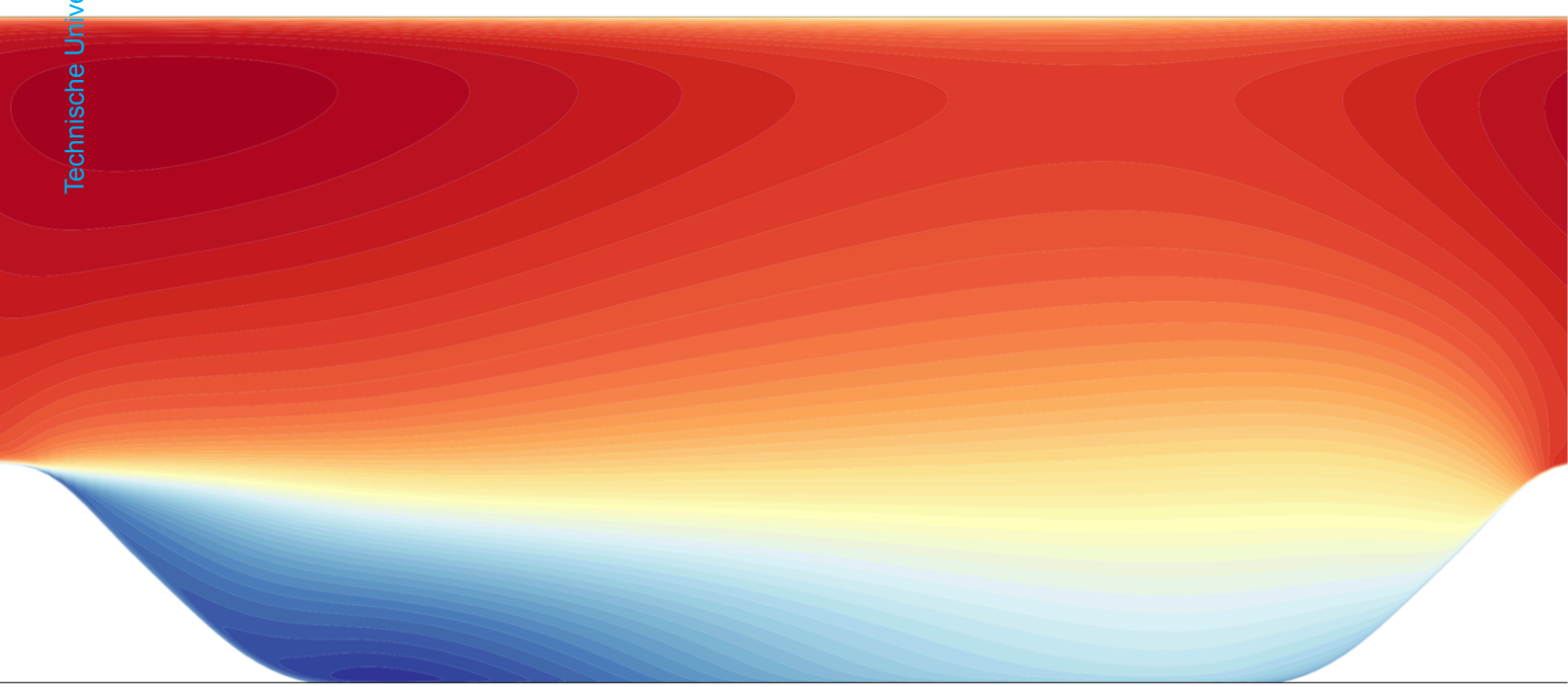
Inferring algebraic stress models from high-fidelity data and applying machine learning to enhance and generalize turbulence lag models

Master of Science Thesis

K.B.E. Nieuwenhuisen

Technische Universiteit Delft



TU Delft

Inferring algebraic stress models from high-fidelity data and applying machine learning to enhance and generalize turbulence lag models

Master of Science thesis

by

K.B.E. Nieuwenhuisen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday June 16, 2020 at 2:30 PM.

Student number:	4205405	
Project duration:	June 25, 2019 – June 16, 2020	
Supervisor:	Dr. R.P. Dwight	TU Delft
Thesis committee:	Dr. S.J. Hulshoff	TU Delft
	Dr.ir. M. Pini	TU Delft
	Ir. J. Steiner	TU Delft

TU Delft

Abstract

Adverse pressure gradients, separation and other forms of non-equilibrium flows are often encountered in flows of interest. In these type of flows, the Boussinesq hypothesis does not hold and often leads to erroneous predictions by eddy viscosity models. In an attempt to capture these non-equilibrium effects, lag parameter models introduce a lag parameter, which is derived from an elliptic blending Reynolds stress model. A novel deterministic machine learning algorithm, referred to as Sparse Regression of Turbulent Stress Anisotropy (SpaRTA), has been used with the objective of developing a data-driven turbulence model based on the elliptic blending $k-\omega$ lag parameter model and evaluating its performance in terms of generalizability, interpretability and its ability to infer the quantities of interest. Corrective terms are introduced and computed directly from high-fidelity data to account for the model-form error by using the k-corrective-frozen-RANS approach. SpaRTA is then used to infer algebraic stress models for these corrective terms using a Galilean invariant integrity basis. It was shown that the k-corrective-frozen-RANS framework has the ability of representing the mean flow features by propagating the corrective terms through a CFD model in OpenFOAM and comparing its result to high-fidelity data. Cross-validation is used to test the performance of the models on unseen data using three flow cases that involve separation, namely periodic hills ($Re = 10595$), converging-diverging channel ($Re = 12600$) and curved backward-facing step ($Re = 13700$). In order to assess the impact of the additional transport equation of the lag parameter, the same data-driven approach was applied to the conventional two-equation $k-\omega$ model. Utilizing an additional transport equation for the lag parameter in this data-driven approach did not result in any significant improvements in terms of predictive capability or generalizability, as both data-driven approaches showed a similar performance, although the data-driven $k-\omega$ models were more numerically stable. It was found that corrective terms formulated using a reduced integrity basis yields data-driven models that have a similar predictive capability compared to models that used the full integrity basis to construct the corrective terms. A significant portion of the resulting data-driven models showed an improvement in predictive capability over the standard (non-data-driven) $k-\omega$ model. Furthermore, most of the models were able to generalize their predictions to two-dimensional flow cases that had different complexity and showed a significant improvement over the baseline $k-\omega$ model.

Preface

This thesis report concludes my master programme in Aerodynamics at the Faculty of Aerospace Engineering at the Delft University of Technology. In this study, I combined machine learning with turbulence modelling, in an attempt to enhance and generalize existing turbulence models. I selected this thesis topic because of my interest in data-driven turbulence modeling and its underlying mathematical theory.

First of all, I would like to express my gratitude to my supervisor, Dr. Richard Dwight. Your willingness to go in-depth to explain the fundamental theory, your comprehensive feedback and the freedom I received throughout this project are much appreciated. I liked your enthusiasm and that we could have a laugh every now and then in between the serious conversations. Secondly, I would like to thank Yuyang Luan and Julia Steiner for getting me started with OpenFOAM. I would like to thank my thesis buddies for the enjoyable coffee and lunch breaks. Furthermore, I would like to thank my friends from around Delft and Den Helder for the laughter, conversations and support. Last but not least, I would like to thank my parents and sister for their support throughout the years.

K.B.E. Nieuwenhuisen
Delft, June 2020

List of Figures

List of Tables

Acronyms

BFS Backward-Facing Step.

CBFS Curved Backward-Facing Step.

CFD Computational Fluid Dynamics.

DNS Direct Numerical Simulation.

EB Elliptic Blending.

ER Expansion Ratio.

GEP Gene Expression Programming.

LES Large Eddy Simulation.

LEVM Linear Eddy Viscosity Model.

NASA National Aeronautics and Space Administration.

NLEVM Non-Linear Eddy Viscosity Model.

OLS Ordinary Least Squares.

OpenFOAM Open Source Field Operation and Manipulation.

PH Periodic Hills.

RANS Reynolds-Averaged Navier-Stokes.

ReLU Rectified Linear Unit.

RNC Random Numerical Constant.

RQ Research Question.

RSM Reynolds Stress Model.

SO Sub-Objective.

SpaRTA Sparse Regression of Turbulent Stress Anisotropy.

TMR Turbulence Modelling Resource.

List of Symbols

A_i Output vector.

B Set of input features.

C_f Skin friction coefficient.

I_i Invariants.

J Cost function.

L Turbulent length scale.

N_L Number of grid points.

N_t Number of time steps.

N Total computational cost.

P Production.

Re Reynolds number.

R Residual of the k equation.

T_{ij} Base tensor.

W_{ij} Weights matrix.

Ω_{ij} Rotation tensor.

Θ Coefficient vector.

α Elliptic parameter.

δ_{ij} Kronecker delta.

η Kolmogorov length scale.

η Invariant of the Reynolds stress anisotropy tensor.

λ_i Eigenvalues of the Reynolds stress anisotropy tensor.

λ_r Tikhonov-regularisation parameter.

λ Regularisation weight.

$\mathcal{C}_R$ Library of candidate functions for R.

$\mathcal{C}_{b_{ij}^{\Delta}}$ Library of candidate functions for b_{ij}^{Δ}.

$\mathcal{D}_{\Delta}$ Unique abstract model forms.

$\mathcal{D}_{ij}$ Spatial redistribution tensor.

$\mathcal{P}_{ij}$ Production tensor.

ν_t Eddy viscosity.

ν Kinematic viscosity.

ω Specific turbulence dissipation rate.

$\overline{S}_{ij}$ Mean rate-of-strain tensor.

$\overline{p}$ Mean pressure.

$\overline{u_i}$ Mean velocity.

ρ Density.

ρ Mixing parameter.

III_b Invariant of the Reynolds stress anisotropy tensor.

II_b Invariant of the Reynolds stress anisotropy tensor.

σ Activation function.

τ_{ij} Reynolds stress tensor.

ε_{ij} Dissipation tensor.

ε Turbulence dissipation rate.

φ^* Lag parameter.

φ_{ij} Pressure-rate-of-strain tensor.

φ Lag parameter.

ξ Invariant of the Reynolds stress anisotropy tensor.

a_{ij} Reynolds stress anisotropy tensor.

b_i Bias vector.

b_{ij}^{Δ} Residual for constitutive relation.

b_{ij} Non-dimensionalised Reynolds stress anisotropy tensor.

k Specific turbulence kinetic energy.

l_m Mixing-length.

p Pressure.

t Time.

u_i' Velocity fluctuation.

u_i Velocity.

x Cartesian coordinate.

y^+ Wall-distance.

Contents

1 Introduction

1.1. Background

In classical physics, turbulence remains an unsolved problem to this day. Many efforts have been made by outstanding scientists in the field of turbulence research, but understanding the nature and the precise prediction of turbulence still remains a challenge [11]. Analytical solutions to the Navier-Stokes equations, which govern the behaviour of turbulent flows, do not exist [12]. Exact solutions only exist for very simplified conditions. Therefore, numerical simulations of turbulent flows are used to obtain a complete description of the flow. Despite the enormous growth in computing power over the years, the Direct Numerical Simulation (DNS) of turbulent flows is limited to low-to-moderate Reynolds number flows, since the computational costs increases rapidly with the Reynolds number. Nevertheless, DNS is a useful tool for research [12] that leads to satisfactory results [13]. In industrial applications, simplified engineering approximations such as Reynolds-Averaged Navier-Stokes (RANS) and Large Eddy Simulation (LES) are popular and widespread [14]. Although LES is a powerful method, its application is still limited by the required grid resolution [15] or timestep for wall-modelled LES. In the next few decades computing power will even further increase, which could pave the way for LES. However, it is believed that this increase will be used more for advanced RANS models rather than LES and that RANS will still have an important role in the foreseeable future [15].

The most common RANS models are based on the Boussinesq hypothesis, where it is assumed that the stresses are proportional to the rate-of-strain. In general, flow is said to be in equilibrium when the time scale of the turbulence is much smaller than the time scale of the mean flow [16]. In such circumstances, the turbulence is able to react quickly to changes in the mean flow, and therefore, the assumption of Boussinesq is reasonable. However, in non-equilibrium flows, there is a delay in the response of the turbulence with respect to changes in the mean flow. Standard eddy viscosity models are unable to account for this lag, and consequently, the predictions of these models are often erroneous. These flows entail a so-called 'stress-strain misalignment'. Instead of assuming a linear relation between the stresses and strains, Reynolds Stress Models (RSM) solve modelled transport equations for the Reynolds stresses, thereby improving the ability to capture such misalignment [17]. However, these improvements come with a cost, as additional transport equations have to be solved. As a result, these models are more computationally expensive, can suffer from numerical instability and converge slowly [18]. Consequently, RSMs have not seen a widespread application in the industry.

In an attempt to capture the effects of the so-called stress-strain misalignment, lag parameter models introduce a lag parameter, which is used to scale the eddy viscosity. Revell et al. [17] proposed a transport equation for a lag parameter C_{as}, in which the expression for the total derivative of a_{ij} is obtained from a Reynolds stress transport equation. As a result, the ability of a Reynolds stress model to address non-equilibrium is incorporated at the expense of a relatively small increase in computational cost [17]. Lardeau & Billard [19] proposed a model, referred to as the lag EB $k-\varepsilon$ model, based on the elliptic blending Reynolds-stress model of Manceau & Hanjalic [20] to overcome the dependency on a high Reynolds number formulation, such that the model is applicable in close proximity to the wall. Using a similar approach to [19], Biswas et al. [21] developed a lag EB $k-\omega$ model using the underlying two equation $k-\omega$ model.

With the availability of high-fidelity data sets, developments in machine learning techniques and ever increasing computational power, a data-driven approach to turbulence modeling has been introduced. In an attempt to improve existing RANS models, high-fidelity data from DNS and data from experiments has been used to calibrate closures [14]. As this field is emerging, there are many possibilities to further develop the data-driven approach and hence it is truly relevant in the field of turbulence modeling. Duraisamy et al. [22–24] use high-fidelity data to infer a spatially varying multiplier of the production term in the Spalart-Allmaras model. A different approach by Emory et al. [25, 26] introduces a perturbation to the eigenvalues of the anisotropy tensor to represent a local injection of uncertainty. Building upon this approach, Edeling et al. [16, 27] introduce two additional transport equation such that the perturbations are made representative of local flow features. Ling et al. [28] find that it would be desirable, for the accuracy of the model and physical realizability, that machine learning algorithms respect invariance properties in case the corresponding physical system obeys one. Domain knowledge is used to embed these properties into the machine learning model. Wang et al. [29] directly predict the Reynolds stress discrepancy from high-fidelity data. Weatheritt & Sandberg [30, 31] use symbolic regression and Gene Expression Programming to model the anisotropy of the Reynolds stress tensor. Schmelzer et al. [5] introduce a deterministic machine learning method referred to as SpaRTA, which stands for Sparse Regression of Turbulent Stress Anisotropy. The model-form error is captured by using the k-corrective-frozen-RANS approach, in which two corrective terms are computed from high-fidelity data, namely a corrective term to the stress-strain relationship b_{ij}^{Δ} and a correction R to the turbulent transport equations. SpaRTA is used to construct separate models for these corrections and involves building a library of candidate functions, selection of models using sparse regression and inference to obtain a model with the correct units. Cross-validation is used to assess the performance of these models for unseen flow cases, which allows for a selection of models with the best predictive capability.

In this work, the SpaRTA approach by Schmelzer et al. [5] and the k-corrective-frozen-RANS approach is extended to the elliptic blending $k-\omega$ lag parameter model by Biswas et al. [21]. The novelty of the work is twofold. It researches whether the additional transport equation for the lag parameter allows the data-driven model to incorporate more physics, and thereby, improve the predictive capability and the ability to generalize. Secondly, it investigates whether a full integrity basis leads to improvements compared to a reduced integrity basis, in terms of predictive capability, generalizability and ability to infer the quantities of interest.

1.2. Research Objective and Questions

The main objective of this research is defined as:

> *"To develop a data-driven turbulence model based on the k-corrective-frozen-RANS approach and machine learning, utilizing the elliptic blending lag parameter $k-\omega$ model from [21] and to evaluate the model in terms of generalizability, interpretability and its ability to infer the quantities of interest."*

Sub-Objectives (SOs) that follow from the main research objective are defined as

SO1 Develop and verify the data-driven elliptic blending lag parameter $k-\omega$ model in OpenFOAM.

SO2 Select suitable flow cases to test the model.

SO3 Collect high-fidelity data.

SO4 Setup k-corrective-frozen-RANS in OpenFOAM.

SO5 Determine feature importance of data set.

SO6 Assemble training and test data to assess generalizability, interpretability and ability to infer quantities of interest.

The two main Research Questions (RQs) and their subquestions are listed below. Collectively, the subquestions answer the main research questions.

RQ1 **What is the ability of the data-driven elliptic blending lag parameter model in terms of generalizing its predictions to other flow cases?**

RQ1.1 **Is the model able to generalize its predictions to two-dimensional flow cases that have different complexity?**
Flow cases that differ in complexity have different flow phenomena and characteristics. Therefore, the aim is to find out how well the model generalizes to these flow cases.

RQ1.2 **Which set of features has the best performance in terms of generalizability?**
The third subquestion concerns the set of features that is used to find patterns by the machine learning algorithm. It is important to find out what features have a significant impact on the predictions and on the generalizability of the algorithm.

RQ2 **What are the strengths and weaknesses of applying the k-corrective-frozen-RANS approach to the lag parameter model?**

RQ2.1 **What is the behaviour of the model in terms of convergence and stability?**
The assessment of convergence and stability is of high importance especially for industrial applications, but also for applications in the field of research.

RQ2.2 **Can the model physically be interpreted in a tractable manner?**
The novelty of the model is encapsulated in the additive term, which is applied in the additional transport equation of the lag parameter, thereby providing the model the ability to incorporate more physics than the previous models that do not have an additional transport equation.

RQ2.3 **How does the model compare to other high-fidelity simulations with regard to computational cost?**
In order to assess the applicability of the model, it is important to know where this model ranks among other simulations such as RANS, LES and DNS in terms of computational cost.

1.3. Thesis Outline

This report is structured in three parts as follows. The first part consists of Chapters 2 and 3, in which the necessary background information is provided to develop the motivation more comprehensively. First, Chapter 2 gives an overview of the field of turbulence modeling, introduces lag parameter models and provides their historical perspective together with the state-of-the-art models. Chapter 3 provides a brief introduction to machine learning and background information to understand commonly used machine learning algorithms in RANS turbulence modeling. Building upon this knowledge, various state-of-the-art data-driven approaches to RANS turbulence modeling are discussed. The second part consists of two chapters that provide a description of the methodology and test cases that are used to conduct this research and required to provide an answer to the research questions. Chapter 4 discusses the model-form error of RANS models, which is followed by an introduction of the k-corrective-frozen-RANS approach. In addition, an overview of the two frameworks is provided and an introduction to Sparse Regression of Turbulent Stress Anisotropy is provided. Test cases used for the purpose of verification, comparison and assessment of the different frameworks are shown and discussed in Chapter 5. The final part consists of Chapters 6 and 7 and provides answers to the research questions. Chapter 6 verifies the OpenFOAM implementation of the elliptic blending lag parameter model. Then, the identification of the model-form error is validated. Results of the SpaRTA method are shown and discussed and the performance of the two different frameworks is assessed. Finally, conclusions and recommendations for future work are given in Chapter 7.

2

Turbulence Modeling

The purpose of this chapter is to provide the reader with an overview of the theory that is related to the field of turbulence modelling. It aims to give insight into the different numerical approaches, along with their strengths, drawbacks and assumptions, thereby focusing on RANS turbulence modeling, in order to understand the motivation behind this study. Furthermore, it aims to introduce the concept and reasoning behind the use of lag parameter models to the reader.

It starts by introducing the governing equations that are relevant for this study in Section 2.1. Then, Section 2.2 gives a description of the different numerical approaches in computational fluid dynamics to solve the Navier-Stokes equations. In addition, it describes some of the most widely used RANS turbulence closure models and their strengths and limitations. Section 2.3 describes the relevance of the anisotropy tensor, its representation, the requirement of Galilean invariance and the non-linear constitutive stress-strain relation, which is used later to construct the functional forms of the corrective terms. Finally, Section 2.4 provides a historic perspective on lag parameter models, as well as recent advances and justifies the choice of the turbulence model considered in this research.

2.1. Governing Equations

The motion of constant-property Newtonian fluids is governed by the Navier-Stokes equations. The Navier-Stokes equations are based on the continuum hypothesis in order to obtain continuous fields. The Navier-Stokes equations are derived from conservation of momentum, which is based on Newton's second law and relates the acceleration of fluid particles to the surface and body forces. The Navier-Stokes equations for incompressible flow are given by

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\,\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} \tag{2.1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.2}$$

where $\boldsymbol{u}$ are the velocity components, ρ is the density, p is the pressure and ν is the kinematic viscosity. Analytical solutions to the Navier-Stokes equations do not exist [12]. Exact solutions only exist for very simplified conditions. Therefore, numerical simulations of turbulent flows are used to obtain a complete description of the flow.

2.2. Numerical Modeling of Turbulent Flows

Turbulent flows are three-dimensional, unsteady, random and have a wide range of length and time scales. These properties make it difficult to develop an approach or model that is tractable [10]. The nonlinear convective term and the pressure gradient term give rise to difficulties. Nevertheless, the Navier-Stokes equations can be solved numerically in various ways, of which three widespread and common approaches will be discussed in this section. Furthermore, the most widely used RANS-based turbulence models are discussed.

2.2.1. Direct Numerical Simulation and Large-Eddy Simulation

In DNS, (2.1) and (2.2) are solved directly, resolving all scales of motion, meaning that no (modelling) assumptions are needed. If applicable, DNS is the most accurate among the various approaches. However, DNS can become very expensive in terms of computational cost, as a vast range of scales has to be resolved. The total computational cost of a DNS scales as $N_t \sim \mathrm{Re}^3$. Despite the enormous growth in computing power over the years, DNS of turbulent flows is limited to low-to-moderate Reynolds number flows, since the computational costs increases rapidly with the Reynolds number. Nevertheless, DNS is a useful tool for research [12] that leads to satisfactory results [13].

The vast range of scales that are encountered in a turbulent flow can be separated by a filter into large and small turbulence scales. LES is based on this principle, where the large scales are resolved and the influence of the smaller subgrid-scales on the larger scales is modelled. These subgrid-scales have a universal character and are modelled by subgrid-scale models. In terms of computational expense, LES is in between DNS and RANS.

Although LES is a powerful method, its application is still limited by the required grid resolution [15] or timestep for wall-modeled LES. In the next few decades the computing power will even further increase, which could pave the way for LES. However, it is believed that RANS will still have an important role in the foreseeable future [15].

2.2.2. Reynolds-Averaged Navier-Stokes

A statistical approach to the numerical solution of the Navier-Stokes equations is employed by the RANS procedure. In RANS, the velocity is decomposed into its ensemble mean and a fluctuation as

$$\boldsymbol{u} = \overline{\boldsymbol{u}} + \boldsymbol{u'} \,. \tag{2.3}$$

This decomposition was first proposed by Osborne Reynolds, and thus, is referred to as the Reynolds decomposition. The RANS equations are obtained by substituting the decomposition into the incompressible Navier-Stokes equations and applying the averaging operator. In tensor notation, the RANS equations are given by

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0 \tag{2.4}$$

$$\frac{\bar{D}\overline{u_i}}{\bar{D}t} = \nu \frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j} - \frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j} \tag{2.5}$$

where the mean substantial derivative is defined as

$$\frac{\bar{D}}{\bar{D}t} = \frac{\partial}{\partial t} + \overline{u_j} \cdot \nabla \,. \tag{2.6}$$

Although the RANS equations look similar to the Navier-Stokes equation, there is one important difference, namely the term that includes the Reynolds stresses $\overline{u_i' u_j'}$. It is this term that plays a crucial role in the RANS approach.

The RANS equations ((2.4) and (2.5)) constitute four equations, namely the three components of the mean momentum equations and one equation for the mean continuity equation. However, the three components of the mean velocity $\overline{\boldsymbol{u}} = \{\overline{u}, \overline{v}, \overline{w}\}^T$ and the mean pressure $\overline{p}$ already form four unknowns. In addition, there are also the unknown Reynolds stresses. Transport equations for the Reynolds stresses do exist, but introduce additional unknowns. Thus, there are more unknowns than equations. Consequently, the set of equations cannot be solved and is termed unclosed. This problem is often referred to as the closure problem. In order to solve the RANS equations, it is necessary to model the Reynolds stresses.

2.2.3. RANS Turbulence Closure Models

Typically, eddy viscosity models make the intrinsic assumption that a_{ij} is only a function of local mean flow quantities. Furthermore, some widespread models are based on the Boussinesq hypothesis, where it is assumed that the stresses are proportional to the mean rate-of-strain as follows

$$a_{ij} \simeq -2\nu_t \bar{S}_{ij} \tag{2.7}$$

where $\nu_t(\boldsymbol{x})$ represents the eddy or turbulent viscosity and the anisotropy stress tensor a_{ij} is given by

$$a_{ij} \equiv \overline{u_i' u_j'} - \frac{2}{3} k \delta_{ij} \tag{2.8}$$

The mean rate-of-strain tensor is defined as

$$\bar{S}_{ij} \equiv \frac{1}{2} \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) . \tag{2.9}$$

The turbulent kinetic energy k is defined as half the trace of the Reynolds stress tensor, i.e.:

$$k \equiv \frac{1}{2} \overline{u_i' u_i'} . \tag{2.10}$$

The anisotropy tensor can be non-dimensionalized as

$$b_{ij} \equiv \frac{a_{ij}}{2k} . \tag{2.11}$$

There are a variety of possible expressions for the eddy viscosity ν_t. These eddy viscosity models are often categorized by the amount of transport equations that are solved. The most simple models do not solve additional transport equations and are referred to as algebraic models or zero-equation models. Most of these models calculate the eddy viscosity based on some prescription for the turbulent mixing-length. Consequently, algebraic models are often unable to account for history effects of the turbulence. For that reason, their application is often limited to simple flow geometries, where algebraic models can be quite useful. In terms of implementation, zero-equation models are of little difficulty. An example of an algebraic model is Prandtl's mixing-length model, which calculates the eddy viscosity as

$$\nu_t = l_m^2 \left| \frac{\partial \overline{u_i}}{\partial x_j} \right| \tag{2.12}$$

where l_m is the mixing-length.

An improvement in terms of accuracy is obtained by including a modelled transport equation for the turbulent kinetic energy k. Models that are based on this principle are referred to as one-equation models. Despite the improved performance, the turbulent mixing-length still has to be prescribed, which is a major drawback. Prandtl's one-equation model calculates the eddy viscosity as

$$\nu_t = l_m \sqrt{k} . \tag{2.13}$$

The exact transport equation for the turbulent kinetic energy is given by

$$\frac{\partial k}{\partial t} + \overline{u_j} \frac{\partial k}{\partial x_j} = \tau_{ij} \frac{\partial \overline{u_i}}{\partial x_j} - \nu \overline{\frac{\partial u_i'}{\partial x_k} \frac{\partial u_i'}{\partial x_k}} + \frac{\partial}{\partial x_j} \left[\nu \frac{\partial k}{\partial x_j} - \frac{1}{2} \overline{u_i' u_i' u_j'} - \frac{1}{\rho} \overline{p' u_j'} \right] \tag{2.14}$$

where the Reynolds stress tensor for incompressible flow is defined as

$$\tau_{ij} \equiv -\overline{u_i' u_j'} . \tag{2.15}$$

The first term on the right-hand side of (2.14) is the production term, which is responsible for the production of turbulent kinetic energy, or more precisely, the rate at which this energy is transferred from the mean flow to the fluctuating velocity field. Dissipation is described by the second term on the right-hand side, which acts as a sink in (2.14). The three terms in between the brackets are molecular diffusion, turbulent transport and pressure diffusion, respectively. The exact transport equation for the turbulent kinetic energy is unclosed, as the production, dissipation, turbulent transport and pressure diffusion are unknown. Therefore, these terms require modelling. The modelled terms below are part of Prandtl's one-equation model. The production term is modelled as

$$\tau_{ij}^m \frac{\partial \overline{u_i}}{\partial x_j} \simeq \left(2 \nu_t \bar{S}_{ij} - \frac{2}{3} k \delta_{ij} \right) \frac{\partial \overline{u_i}}{\partial x_j} \tag{2.16}$$

where the Boussinesq hypothesis has been used. The dissipation is modelled as

$$-\nu\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_k}} \simeq -C_D\frac{k^{3/2}}{l_m} \tag{2.17}$$

where C_D is a model constant. In turbulent flows, the representation of the turbulent transport of a scalar quantity ϕ' is approximated as gradient-diffusion, i.e., $\overline{u_j'\phi'} \sim \mu_t\partial\Phi/\partial x_j$, which is analogous to processes of molecular transport [32]. However, such a straightforward analog does not exist for the pressure diffusion term. Therefore, generally, the pressure diffusion term is grouped with the turbulent transport term or is neglected. For simple flows, DNS results by Mansour et al. [33] have indicated that the pressure diffusion term is quite small. Thus, the turbulent transport and pressure diffusion terms are approximated by

$$-\frac{1}{2}\overline{u_i'u_i'u_j'} - \frac{1}{\rho}\overline{p'u_j'} \simeq -\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x_j} \tag{2.18}$$

where σ_k is a closure coefficient and in general has a value of $\sigma_k = 1.0$ [10]. The modelled transport equation for the turbulent kinetic energy then becomes

$$\frac{\partial k}{\partial t} + \overline{u_j}\frac{\partial k}{\partial x_j} = \tau_{ij}^m\frac{\partial\overline{u_i}}{\partial x_j} - C_D\frac{k^{3/2}}{l_m} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{2.19}$$

where τ_{ij}^m is given by (2.16). The one-equation model is highly efficient in terms of computational effort, as only one transport equation has to be solved. It is more complex than the zero-equation model, but still straightforward. However, these models do not account for transport effects on the turbulence length scales [32].

A class that take the transport effects into account is the class of two-equation models. These models solve modelled transport equation for two turbulence quantities, such that the model is complete. Complete in this sense refers to the fact that a length-scale or time-scale can be formed with the two aforementioned turbulence quantities. Consequently, the requirement of prescribing a turbulence length-scale is no longer needed. In general, the Boussinesq hypothesis lies at the basis of these models. The most widespread two-equation model is the $k-\varepsilon$ model, which solves a modelled transport equation for both the turbulent kinetic energy k and the dissipation ε, as shown below

$$\frac{\partial k}{\partial t} + \overline{u_j}\frac{\partial k}{\partial x_j} = \tau_{ij}^m\frac{\partial\overline{u_i}}{\partial x_j} - \varepsilon + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{2.20}$$

$$\frac{\partial \varepsilon}{\partial t} + \overline{u_j}\frac{\partial \varepsilon}{\partial x_j} = C_{\varepsilon 1}\frac{\varepsilon}{k}\tau_{ij}^m\frac{\partial\overline{u_i}}{\partial x_j} - C_{\varepsilon 2}\frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] \tag{2.21}$$

where the model constants are

$$C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3\,. \tag{2.22}$$

The eddy viscosity is determined as

$$\nu_t = C_\mu\frac{k^2}{\varepsilon} \tag{2.23}$$

where $C_\mu = 0.09$. At the wall, the boundary condition are given as $k = 0$ and $\varepsilon = 0$. In terms of accuracy, the performance of the standard $k-\varepsilon$ model is acceptable for simple flows. The models' performance is reasonable for flows that have a small mean pressure gradient and mean streamline curvature, such as two-dimensional thin shear flows [10]. However, for more complex flows, such as most of the non-equilibrium flows, the model performs poorly and can be quite inaccurate. Over a period of time, many adjustments have been proposed to improve the performance of the $k-\varepsilon$ model. These adjustments are often ad-hoc and applicable to a specific flow case.

Another well-known and widespread two-equation model is the $k-\omega$ model, in which two modelled transport equations are solved for the turbulent kinetic energy k and the specific rate of dissipation ω, respectively, as shown below

$$\frac{\partial k}{\partial t} + \overline{u_j}\frac{\partial k}{\partial x_j} = \tau_{ij}^m\frac{\partial\overline{u_i}}{\partial x_j} - \beta^* k\omega + \frac{\partial}{\partial x_j}\left[\left(\nu + \sigma^*\nu_t\right)\frac{\partial k}{\partial x_j}\right] \tag{2.24}$$

$$\frac{\partial \omega}{\partial t} + \overline{u_j}\frac{\partial \omega}{\partial x_j} = \alpha \frac{\omega}{k}\tau_{ij}^m \frac{\partial \overline{u_i}}{\partial x_j} - \beta\omega^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma\nu_t)\frac{\partial \omega}{\partial x_j}\right] \tag{2.25}$$

where the model constants are given by

$$\alpha = 5/9, \quad \beta = 3/40, \quad \beta^* = 9/100, \quad \sigma = 1/2, \quad \sigma^* = 1/2\,. \tag{2.26}$$

The eddy viscosity is determined as

$$\nu_t = \frac{k}{\omega}\,. \tag{2.27}$$

The boundary conditions at solid walls for the turbulence kinetic energy is set to $k = 0$. Menter [34] suggests the following boundary condition for ω at the wall

$$\omega = 10\frac{6\nu}{\beta\left(\Delta y_1\right)^2} \tag{2.28}$$

in which Δy_1 denotes the distance to the first grid point away from the wall. The $k-\omega$ model performs well in boundary layer flows and flows with adverse pressure gradients. In many flows, its performance is superior to the $k-\varepsilon$ model [13].

To overcome the limitations of eddy viscosity models, Reynolds stress models solve modelled transport equations for the unknown Reynolds stresses $\overline{u_i'u_j'}$. These models are often referred to as second-order closure models and do account for effects of streamline curvature, changes in strain rate and secondary motions [32], but at the cost of increased complexity and computational effort. The exact transport equation of the Reynolds stress for incompressible flow is given by

$$\begin{aligned}\frac{\partial \overline{u_i'u_j'}}{\partial t} + \overline{u_k}\frac{\partial \overline{u_i'u_j'}}{\partial x_k} = &-\left(\overline{u_i'u_k'}\frac{\partial \overline{u_j}}{\partial x_k} + \overline{u_j'u_k'}\frac{\partial \overline{u_i}}{\partial x_k}\right) - 2\nu\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_j'}{\partial x_k}} + \overline{\frac{p'}{\rho}\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right)} \\ &+\frac{\partial}{\partial x_k}\left(\nu\frac{\partial \overline{u_i'u_j'}}{\partial x_k} - \overline{u_i'u_j'u_k'} - \overline{\frac{p'}{\rho}\left(u_i'\delta_{jk} + u_j'\delta_{ik}\right)}\right)\end{aligned} \tag{2.29}$$

where the first term on the right-hand side is responsible for the production of Reynolds stresses by the mean velocity gradients. The second term resembles dissipation of the Reynolds stresses. The correlation between the fluctuating pressure and the fluctuating strain rate is described by the third term. This term, often referred to as the pressure-rate-of-strain tensor, does not produce or dissipate turbulent kinetic energy, but it serves to redistribute energy between the different components of the Reynolds stress $\overline{u_i'u_j'}$. The pressure-rate-of-strain tensor has received a lot of interest of turbulence modellers, as it is of the same order as the production term, and therefore, plays an important role in most flows. The spatial redistribution between components of the Reynolds stresses is described by the last term, which is a combination of multiple diffusion terms. Equation (2.29) is often written in a compact form as

$$\frac{\partial \overline{u_i'u_j'}}{\partial t} + \overline{u_k}\frac{\partial \overline{u_i'u_j'}}{\partial x_k} = \mathcal{P}_{ij} - \varepsilon_{ij} + \varphi_{ij} + \mathcal{D}_{ij} \tag{2.30}$$

where the left-hand side, $\mathcal{P}_{ij}$ and $\nu\dfrac{\partial \overline{u_i'u_j'}}{\partial x_k}$ in $\mathcal{D}_{ij}$ are in closed form and the remaining terms require modelling.

2.3. Anisotropy Tensor

This section starts by providing the relevance and importance of the anisotropy tensor in Section 2.3.1. Its representation is discussed in Section 2.3.2. Stemming from the fact the Navier-Stokes equations are Galilean invariant, any model for the anisotropy tensor also has the requirement of being frame invariant, which is discussed in Section 2.3.3. Finally, Section 2.3.4 presents the integrity basis used to model the anisotropy tensor in the most general way.

2.3.1. Relevance

The components of the Reynolds stress tensor can be distinguished between shear stresses and normal stresses, where the former indicates the off-diagonal components and the latter denotes the diagonal

components. However, this distinction depends on the coordinate system. Therefore, an intrinsic distinction that does not depend on the coordinate system would be between isotropic and anisotropic stresses. Such a decomposition for the Reynolds stress tensor is given in (2.31).

$$\tau_{ij} = a_{ij} + \frac{2}{3}k\delta_{ij} = 2kb_{ij} + \frac{2}{3}k\delta_{ij}\,. \tag{2.31}$$

Substituting this expression for the Reynolds stress into the mean momentum equation ((2.5)) leads to

$$\frac{\bar{D}\overline{u_i}}{\bar{D}t} = \nu\frac{\partial^2\overline{u_i}}{\partial x_j \partial x_j} - \frac{\partial a_{ij}}{\partial x_j} - \frac{1}{\rho}\frac{\partial\left(\overline{p} + \frac{2}{3}\rho k\right)}{\partial x_j} \tag{2.32}$$

from which can be seen that only the anisotropic term is effective in the transport of momentum, and therefore, important. Furthermore, the isotropic part can be absorbed in a modified pressure term [10]. In terms of accuracy, correctly predicting the amount and type of anisotropy is critical [1].

2.3.2. Representation

Visualization of data can be a useful tool to have a better understanding of the problem. In turbulent flows, the anisotropic behaviour is a quantity that is visualized to compare various turbulence models or to inform the development of a turbulence model. The non-dimensionalized anisotropy tensor is given by (2.11) and is as follows

$$b_{ij} = \frac{a_{ij}}{2k} = \frac{\overline{u_i' u_j'}}{2k} - \frac{1}{3}\delta_{ij}\,.$$

The non-dimensionalized anisotropy tensor is a second-order tensor, for which the powers are given by

$$b^2_{ij} = b_{ik}b_{kj}, \quad b^3_{ij} = b_{ik}b_{kl}b_{lj}\,. \tag{2.33}$$

The anisotropy tensor consists of six components that characterize the anisotropy of the Reynolds-stress tensor. However, it can be simpler characterized by using the variables ξ and η. By using the Cayley-Hamilton theorem, the characteristic polynomial of a tensor $\boldsymbol{b}$ and the corresponding invariants are given by

$$\boldsymbol{b}^3 - \mathrm{I}_b\boldsymbol{b}^2 + \mathrm{II}_b\boldsymbol{b} - \mathrm{III}_b\boldsymbol{I} = 0 \tag{2.34}$$

where the invariants are as follows

$$\mathrm{I}_b = b_{ii}\,, \tag{2.35}$$

$$\mathrm{II}_b = -\frac{1}{2}b_{ij}b_{ji}\,, \tag{2.36}$$

$$\mathrm{III}_b = \det\left(b_{ij}\right). \tag{2.37}$$

The anisotropy tensor has two independent invariants, as the trace is zero. The invariants II_b and III_b are functions of the eigenvalues λ_i, which can be found by diagonalizing the anisotropy tensor. The invariants are related to the eigenvalues in the following way

$$\mathrm{II}_b = \lambda_1^2 + \lambda_1\lambda_2 + \lambda_2^2\,, \tag{2.38}$$

$$\mathrm{III}_b = -\lambda_1\lambda_2\left(\lambda_1 + \lambda_2\right). \tag{2.39}$$

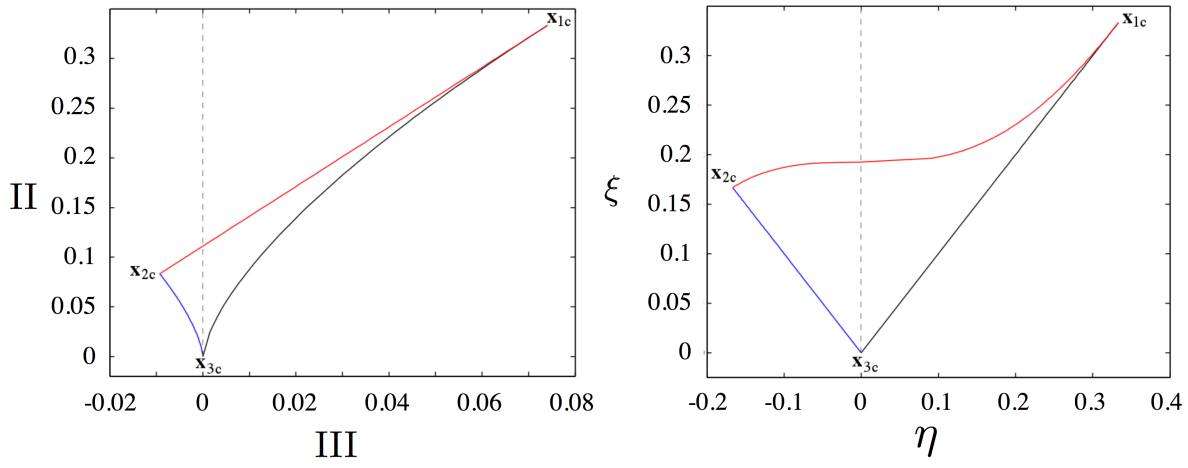
Because of convenience, Pope [10] suggests to use the variables ξ and η as the two invariants. This way, the lower left quadrant of the Lumley triangle is stretched, which results in a more detailed view of the region near x_{3c} [1], making the view on the non-linear behaviour in the return to isotropy more clear [35]. These invariants were introduced in order to evaluate the trajectories of the return to isotropy of homogeneous turbulence [35]. The variables η and ξ are defined as follows

$$6\eta^2 = -2\mathrm{II}_b = b^2_{ii} = b_{ij}b_{ji}\,, \tag{2.40}$$

$$6\xi^3 = \ 3\mathrm{III}_b = b^3_{ii} = b_{ij}b_{jk}b_{ki}\,. \tag{2.41}$$

Anisotropy invariant maps are one of the techniques used to visualize the anisotropy tensor. Invariant maps represent two-dimensional plots and can be linear or non-linear. Figure 2.1 shows two non-linear

invariant maps, where Fig. 2.1a was originally proposed by Lumley & Newman [36]. The magnitudes of the anisotropy are indicated by the eigenvalues, while the eigenvectors indicate the directions. The second invariant map (Fig. 2.1b) is a transformation from the Lumley triangle and is used for the evaluation of the return-to-isotropy trajectories in homogeneous turbulence [35].



(a) Lumley triangle

(b) Turbulence triangle

Figure 2.1: Non-linear anisotropy invariant maps [1].

Both invariant maps show various states of turbulence. In every turbulent flow, the invariants can be determined from the anisotropy tensor and plotted in the invariant maps. Points inside the triangle correspond to realizable Reynolds stresses, while points outside the triangle correspond to non-realizable Reynolds stresses, in which the eigenvalues are negative or complex [10]. Realizability was introduced by Schumann [37] as he proposed three conditions for the Reynolds stress tensor $\boldsymbol{\tau}$ to ensure realizability. These conditions are given by

$$\tau_{\alpha\alpha} \geq 0 \quad \forall \alpha \in \{1,2,3\}\,, \tag{2.42}$$

$$\tau_{\alpha\beta}^2 \leq \tau_{\alpha\alpha}\tau_{\beta\beta} \quad \forall \alpha \neq \beta, \tag{2.43}$$

$$\det(\boldsymbol{\tau}) \geq 0, \tag{2.44}$$

and are trivial consequences of the fact that $\boldsymbol{\tau}$ is positive semi-definite. One speaks of a positive semi-definite matrix $\boldsymbol{A}$ if $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \geq 0$, $\forall \boldsymbol{x} \in \mathbb{R}^N$. As the Reynolds stress tensor is the arithmetic average of $\boldsymbol{u}'\otimes\boldsymbol{u}'$, the outer product of $\boldsymbol{u}'$ with itself (which is positive semi-definite), the arithmetic average of such tensors is also positive semi-definite, and thus, $\boldsymbol{\tau}$ is positive semi-definite. The first condition requires that the components of the turbulence kinetic energy are non-negative. The second condition is the consequence of the Schwarz' inequality, which states that every off-diagonal component of the Reynolds stress tensor is bounded by the magnitude of autocorrelations [37]. The last condition requires the Reynolds stress tensor to be real [36]. In general, the differential equations corresponding to a turbulence model will not satisfy these conditions in all circumstances [38]. In cases where these conditions are not satisfied, the negative eigenvalues could lead to a negative eddy viscosity, which in turn will lead to numerical instability.

The invariant maps show three points that are labeled x_{1c}, x_{2c} and x_{3c}, which refer to one-, two- and three-component turbulence, respectively. Componentality indicates the relative strengths of the velocity fluctuations. Table 2.1 is obtained from Pope [10] and provides the characteristics of the different turbulence states in terms of invariants and eigenvalues of the anisotropy tensor $\boldsymbol{b}$.

Table 2.1: Characteristics of turbulence states [10]

State of turbulence	Invariants	Eigenvalues	Label in Fig. 2.1
Isotropic	$\xi = \eta = 0$	$\lambda_1 = \lambda_2 = \lambda_3 = 0$	x_{3c}
Two-component	$\xi = -\frac{1}{6}$, $\eta = \frac{1}{6}$	$\lambda_1 = \lambda_2 = 0$	x_{2c}
One-component	$\xi = \frac{1}{3}$, $\eta = \frac{1}{3}$	$\lambda_1 = \frac{2}{3}$, $\lambda_2 = \lambda_3 = -\frac{1}{3}$	x_{1c}

The linear invariant maps are shown in Fig. 2.2. The first subfigure shows an invariant map that uses the eigenvalues λ_1 and λ_2 as coordinate system. Figure 2.2b shows the barycentric map, in which the coordinates of the corners x_{1c}, x_{2c} and x_{3c} are given by (1, 0), (0, 0) and $\left(1/2, \sqrt{3}/2\right)$, respectively. The coordinate system is defined as

$$x_B = C_{1c}x_{1c} + C_{2c}x_{2c} + C_{3c}x_{3c}\,, \tag{2.45}$$

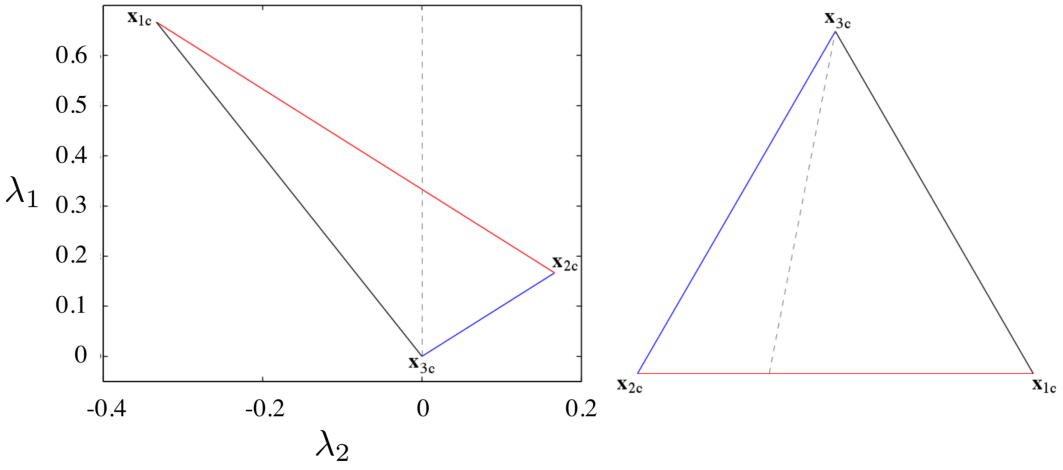
$$y_B = C_{1c}y_{1c} + C_{2c}y_{2c} + C_{3c}y_{3c}\,, \tag{2.46}$$

where the coefficients C_{1c}, C_{2c} and C_{3c} are determined by the eigenvalues as

$$C_{1c} = \lambda_1 - \lambda_2\,, \tag{2.47}$$

$$C_{2c} = 2\left(\lambda_2 - \lambda_3\right)\,, \tag{2.48}$$

$$C_{3c} = 3\lambda_3 + 1\,. \tag{2.49}$$

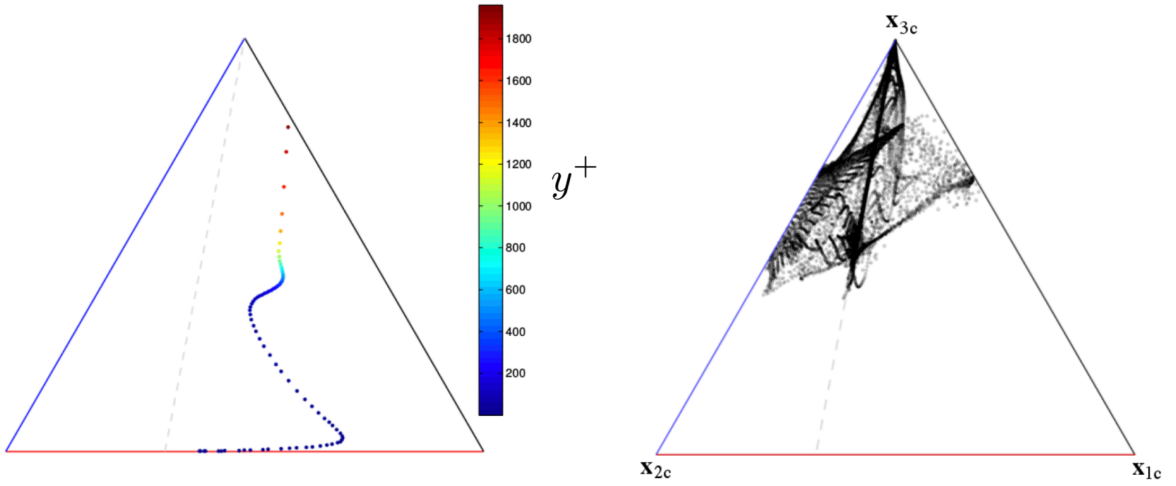


(a) Eigenvalue triangle

(b) Barycentric map

Figure 2.2: Linear anisotropy invariant maps [1].

While the invariant maps are able to provide useful information regarding the componentality of the turbulence, it comes at the cost of the loss of spatial information. In an attempt to retrieve some of the spatial information in a qualitative way, the wall-distance y^+ can be colored, as is shown in Fig. 2.3a. Another drawback becomes apparant when one maps a whole domain onto an invariant map, which then becomes difficult to interpret, as can be seen from Fig. 2.3b. This difficulty is due to the fact that different locations in the domain that have a nearly similar anisotropy tensor are mapped onto the same location in the invariant map.



(a) (b)

Figure 2.3: Barycentric maps from [1]. Figure (a) presents the DNS data of a turbulent channel flow for a range of points normal to the wall. The results from a converging-diverging nozzle flow are shown in figure (b) for the entire domain.

Emory and Iaccarino [1] introduced a new visualization technique to show the componentality while maintaining the spatial information. Using the red-green-blue (RGB) color system and the componentality coefficients C_{1c}, C_{2c} and C_{3c}, a color map is constructed. The colors are determined as follows

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = C_{1c} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + C_{2c} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + C_{3c} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} . \tag{2.50}$$

As a result of this color map, one-component turbulence is red, two-component turbulence is green and three-component or isotropic turbulence is blue. Any other state in the invariant map is a combination of the aforementioned colors. An example of what this looks like is given in Fig. 2.4.

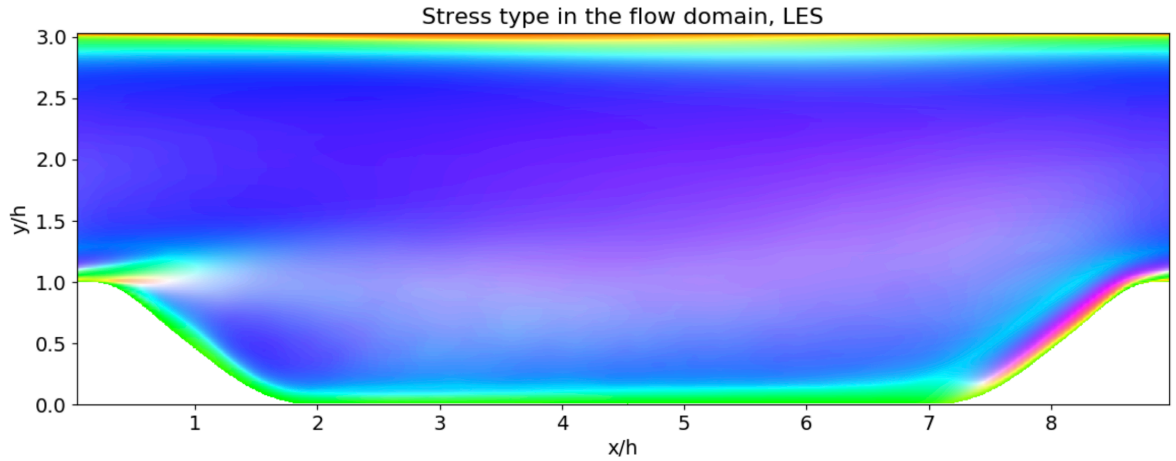


Figure 2.4: Componentality indicated by RGB color system for a Large Eddy Simulation of the periodic hill flow case. $Re = 12595$. Obtained from [2].

2.3.3. Invariance of Tensor-Based Functionals

The Navier-Stokes are known to be Galilean invariant, which means that they remain the same in different inertial frames. Stemming from this fact, any model for the anisotropy tensor has the physical requirement of also being frame invariant, and thus, should not depend on the choice of inertial frame. A scalar function f is said to be frame invariant if

$$f(\boldsymbol{S}, \boldsymbol{v}, c) = f\left(\boldsymbol{Q}\boldsymbol{S}\boldsymbol{Q}^T, \boldsymbol{Q}\boldsymbol{v}, c\right), \quad \forall \boldsymbol{Q}, \boldsymbol{S}, \boldsymbol{v}, c \tag{2.51}$$

where $\boldsymbol{S} \in \mathbb{R}^{3\times3}$ denotes a tensor argument, $\boldsymbol{v} \in \mathbb{R}^3$ denotes a vector argument, $c \in \mathbb{R}$ denotes a scalar argument and $\boldsymbol{Q} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ denotes an orthogonal transformation matrix. In a similar fashion, a tensor-based functional $\boldsymbol{h}$ is frame invariant if

$$\boldsymbol{Q}\boldsymbol{h}\left(\boldsymbol{S},\boldsymbol{v},c\right)\boldsymbol{Q}^T = \boldsymbol{h}\left(\boldsymbol{Q}\boldsymbol{S}\boldsymbol{Q}^T,\boldsymbol{Q}\boldsymbol{v},c\right), \quad \forall \boldsymbol{Q},\boldsymbol{S},\boldsymbol{v},c\,. \tag{2.52}$$

2.3.4. Non-Linear Constitutive Stress-Strain Relation

In RANS turbulence modelling, the set of equations is unclosed and the Reynolds stresses require modelling. Pope [10] proposed to relate the anisotropy tensor b_{ij} with the normalized mean rate-of-strain tensor S_{ij} and normalized mean rotation tensor Ω_{ij}, which are given by

$$S_{ij} = \tau\frac{1}{2}\left(\frac{\partial\overline{u_i}}{\partial x_j} + \frac{\partial\overline{u_j}}{\partial x_i}\right), \tag{2.53}$$

$$\Omega_{ij} = \tau\frac{1}{2}\left(\frac{\partial\overline{u_i}}{\partial x_j} - \frac{\partial\overline{u_j}}{\partial x_i}\right) \tag{2.54}$$

in which $\tau = \frac{1}{\omega}$ denotes the time scale. The Cayley-Hamilton theorem is then used to derive an integrity basis, which consists of ten nonlinear base tensors $\boldsymbol{T}^{(n)}$. It is termed an integrity basis, since any symmetric deviatoric second-order tensor formed from $\boldsymbol{S}$ and $\boldsymbol{\Omega}$ can be represented by a linear combination of the ten base tensors. This integrity basis is then used to express the anisotropy tensor in the most general $\boldsymbol{h}$ as

$$\boldsymbol{b}\left(\boldsymbol{S},\boldsymbol{\Omega}\right) = \boldsymbol{h}\left(\boldsymbol{S},\boldsymbol{\Omega}\right) = \sum_{n=1}^{10}\boldsymbol{T}^{(n)}\alpha_n\left(I_1,\ldots,I_5\right). \tag{2.55}$$

The nonlinear base tensors read

$$\begin{aligned}
\boldsymbol{T}^{(1)} &= \boldsymbol{S}, & \boldsymbol{T}^{(6)} &= \boldsymbol{\Omega}^2\boldsymbol{S} + \boldsymbol{S}\boldsymbol{\Omega}^2 - \frac{2}{3}\boldsymbol{I}\cdot\operatorname{trace}\left(\boldsymbol{S}\boldsymbol{\Omega}^2\right),\\
\boldsymbol{T}^{(2)} &= \boldsymbol{S}\boldsymbol{\Omega} - \boldsymbol{\Omega}\boldsymbol{S}, & \boldsymbol{T}^{(7)} &= \boldsymbol{\Omega}\boldsymbol{S}\boldsymbol{\Omega}^2 - \boldsymbol{\Omega}^2\boldsymbol{S}\boldsymbol{\Omega},\\
\boldsymbol{T}^{(3)} &= \boldsymbol{S}^2 - \frac{1}{3}\boldsymbol{I}\cdot\operatorname{trace}\left(\boldsymbol{S}^2\right), & \boldsymbol{T}^{(8)} &= \boldsymbol{S}\boldsymbol{\Omega}\boldsymbol{S}^2 - \boldsymbol{S}^2\boldsymbol{\Omega}\boldsymbol{S},\\
\boldsymbol{T}^{(4)} &= \boldsymbol{\Omega}^2 - \frac{1}{3}\boldsymbol{I}\cdot\operatorname{trace}\left(\boldsymbol{\Omega}^2\right), & \boldsymbol{T}^{(9)} &= \boldsymbol{\Omega}^2\boldsymbol{S}^2 + \boldsymbol{S}^2\boldsymbol{\Omega}^2 - \frac{2}{3}\boldsymbol{I}\cdot\operatorname{trace}\left(\boldsymbol{S}^2\boldsymbol{\Omega}^2\right),\\
\boldsymbol{T}^{(5)} &= \boldsymbol{\Omega}\boldsymbol{S}^2 - \boldsymbol{S}^2\boldsymbol{\Omega}, & \boldsymbol{T}^{(10)} &= \boldsymbol{\Omega}\boldsymbol{S}^2\boldsymbol{\Omega}^2 - \boldsymbol{\Omega}^2\boldsymbol{S}^2\boldsymbol{\Omega},
\end{aligned} \tag{2.56}$$

and the corresponding invariants are given by

$$I_1 = \operatorname{trace}\left(\boldsymbol{S}^2\right), \quad I_2 = \operatorname{trace}\left(\boldsymbol{\Omega}^2\right), \quad I_3 = \operatorname{trace}\left(\boldsymbol{S}^3\right), \quad I_4 = \operatorname{trace}\left(\boldsymbol{\Omega}^2\boldsymbol{S}\right), \quad I_5 = \operatorname{trace}\left(\boldsymbol{\Omega}^2\boldsymbol{S}^2\right). \tag{2.57}$$

2.4. Elliptic Blending Lag Parameter Model

The concept and reasoning behind the use of of elliptic blending and lag parameters is introduced in Section 2.4.1. Lag parameter models are placed in a historical perspective in Section 2.4.2. Then, two different lag parameter models are introduced that utilize the widely used k-ε and k-ω models as underlying two-equation model in Sections 2.4.3 and 2.4.4, respectively.

2.4.1. Brief Introduction

Adverse pressure gradients, separation and other forms of non-equilibrium flow are often encountered in flows of interest. In general, flow is said to be in equilibrium when the time scale of the turbulence is much smaller than the time scale of the mean flow [16]. In such circumstances, the turbulence is able to react quickly to changes in the mean flow, and therefore, the assumption of Boussinesq is reasonable. However, in non-equilibrium flows, there is a delay in the response of the turbulence with respect to changes in the mean flow. Standard eddy viscosity models are unable to account for this lag, and consequently, the predictions of these models are often erroneous. These flows entail a so-called 'stress-strain misalignment'. The elliptic blending lag parameter models try to capture the effects of the stress-strain misalignment by introducing a lag parameter φ. The approach that is used for this research is based on an underlying two-equation model, such as the $k-\varepsilon$ or $k-\omega$ model, in combination with a

transport equation for the lag parameter. This transport equation is derived from an elliptic blending Reynolds stress model, thereby incorporating the ability of a RSM to address non-equilibrium [17]. In addition, an elliptic blending equation is solved for α, which is required to make the model applicable near the wall, since early lag models were derived from high Reynolds number Reynolds stress models [39].

2.4.2. Historical Perspective

RANS models have seen many developments over the years. The development of two-equation models has especially received some effort, as these models are often used in practical applications [21]. In an attempt to improve the accuracy of near-wall turbulence predictions, the region near the wall was modelled by introducing wall functions [40]. Another patch for near-wall turbulence is the application of damping functions, which are used to introduce damping effects. Durbin [41] proposed a method in which the ad hoc damping functions were no longer required. He introduced the so-called 'wall-normal' velocity scale $\overline{v^2}$ to scale the eddy viscosity. With the introduction of the $\overline{v^2}-f$ model, Durbin initiated the concept of elliptic relaxation, as he introduced an elliptic equation for the relaxation function f. Building upon this concept, Hanjali et al. [42] proposed the $\zeta-f$ model in which a transport equation for the velocity scale $\zeta=\overline{u^2}/k$ is solved, thereby increasing numerical robustness over Durbin's model. Lien & Kalitzin [43] altered the elliptic relaxation equation to enhance the numerical stability of the model. However, some non-vanishing terms are neglected [44]. To overcome this problem, Laurence et al. [44] propose a different model, referred to as the $\varphi-f$ model, in which homogeneous boundary conditions are maintained. Incorporating the principles of the $\overline{v^2}-f$ model, Billard & Laurence [45] proposed a refined $k-\varepsilon-\overline{v^2}/k$ model, referred to as the BL-$\overline{v^2}/k$. This model uses the non-dimensional coefficient α, which is a solution of an elliptic equation, to blend quasi-homogeneous and near-wall terms, such that a smooth transition is ensured. The model is compared to the $\overline{v^2}-f$ and k-ω SST models for flow over periodic hills and an asymmetric plane diffuser, which both exhibit separated flow and a recirculation region. Both the BL-$\overline{v^2}/k$ and $\overline{v^2}-f$ models accurately predicted the recirculation zone in the periodic hill case, whereas the k-ω SST model excessively predicted the size of the recirculation zone. All models underpredict the shear stress profiles in both flow cases consistently. In the recovery regions in both flow cases, the $\overline{v^2}-f$ shows a slight improvement in the prediction of the velocity profiles over the BL-$\overline{v^2}/k$ model.

In an attempt to address non-equilibrium, Revell et al. [17] proposed a transport equation for a lag parameter C_{as}. In this transport equation, the total derivative of a_{ij} is obtained from the transport equation for the Reynolds stresses, thereby incorporating the ability of a Reynolds stress model to address non-equilibrium at the expensive of a relatively small increase in computational cost [17]. As some terms in this model are based on a high Reynolds number formulation, the model is not applicable in close proximity to the wall [19]. Lardeau & Billard [19] proposed a model, referred to as the lag EB $k-\varepsilon$ model, based on the elliptic blending Reynolds-stress model of Manceau & Hanjalic [20] to overcome this problem. This model consists of three transport equations and its formulation looks similar to the $k-\varepsilon-\overline{v^2}/k$ model in [45], but involves extra production terms in the lag parameter transport equation. In order to validate their model, Lardeau and Billard compared their results with DNS data, and with the results from the k-ω SST model, for flow through a channel with varying Reynolds numbers in the range of $Re_\tau=180-2000$. The predicted profiles of velocity, turbulence kinetic energy and eddy viscosity showed an improvement compared to the predictions by the k-ω SST model. Especially the predicted turbulence kinetic energy in the near-wall region and in the log-layer were in close agreement with the DNS data, which can be an important property when high-fidelity data is used as boundary condition and propagated through the RANS model [19]. In addition, the lag model predicted the decay of the eddy viscosity more accurately in the region where the strain-rate is small, i.e. in the defect layer, although linear eddy viscosity models tend to overpredict the eddy viscosity in this region. In order to assess the performance of the model on pressure-induced separation, the periodic hill, the curved backward-facing step and the NASA 2D hump were also selected as validation cases. For the periodic hill case at a Reynolds number of $Re=10595$, the lag EB $k-\varepsilon$ model predicts the onset of separation accurately, and the reattachment location is predicted close to that of the LES data. The turbulence kinetic energy profiles, especially near the edge of the recirculation zone, show a significant improvement over the k-ω SST model, although RANS models often predict the turbulence kinetic energy erroneously for this specific flow case. The predicted recirculation zone was in close agreement with LES data for the curved backward-facing step at $Re=13700$. Furthermore, the velocity profiles were accurately

predicted by the lag model. The 2D hump by NASA had a much higher Reynolds number than the other two flow cases, namely $Re = 936000$. The onset of separation was accurately predicted, but the reattachment point was predicted too far downstream. More complex test cases included a rotating channel expansion, a NACA0012 wing-tip vortex and a high-lift experiment by NASA.

Using a similar approach to [19], Biswas et al. [21] developed a lag EB $k-\omega$ model using the underlying two equation $k-\omega$ model. A simple channel flow with Reynolds numbers of $Re_\tau = 550$, 1000, 2000 and 5200 was considered as a first validation case. The predicted velocity profiles show close agreement with DNS data. In the logarithmic region, the predictions by the lag model show an improvement over those of the $k-\omega$ and $k-\omega$ SST models, which is the result of the eddy viscosity being better scaled near the wall due to the lag parameter [21]. Flow over a backward-facing step at $Re_H = 37500$ was used to assess the performance of the model for separated flows. All models predicted the recirculation zone accurately. In addition, a discrepancy in the velocity profiles was observed for all models, as the predicted recovery was more slowly than that from the experiments. The flow over periodic hills and over a curved backward-facing step have been used to assess the performance of the model for flows that involve separation from a curved surface. The lag parameter model predicts a recirculation zone that is approximately 10% larger in comparison with the LES data. The velocity profiles are in reasonable agreement with the LES data. The shear velocity profiles show large discrepancies with the LES data, except for the profile downstream of the reattachment location. In the curved backward-facing step case, the prediction of the reattachment location was improved compared to other models. Large discrepancies were observed in the shear velocity profiles, but this did not significantly affect the mean velocity profiles. As a final validation case, flow over the wall mounted hump by NASA at $Re = 936000$ was considered. The lag model overpredicted the recirculation zone by 8% in comparison with the experimental results, which is an improvement compared to the prediction of the k-ω SST model that overpredicted it by 14%. In the work of [40], additional flow cases with varying complexity were considered as validation cases, such as a series of two-dimensional bumps, a swept bump, three-dimensional diffusers and a NACA0020 wing body junction flow. In the series of two-dimensional bumps, the predicted mean velocity profiles were in closer agreement with the LES data the other models. The performance of the model for flows with transverse pressure gradients was assessed in the swept bump flow case. All models predicted spanwise velocity profiles reasonably well. It should be noted that the magnitude of the spanwise velocities is considerably lower than that of the streamwise velocities. The three-dimensional diffusers were used to assess the performance of the lag model for three-dimensional separated flow, which are often a challenge for LEVMs [40]. Two diffusers with an aspect ratio of $AR = 1.5$ and $AR = 4$ were used as test case. For the lower aspect ratio flow case, high-fidelty data showed that the flow separates on the upper flared wall, while for the higher aspect ratio case the flow separates on the side flared wall. The lag parameter model accurately predicted the pressure distributions, showing a significant improvement over the other models. Furthermore, the location of separation, its pattern and the magnitude of backflow were predicted accurately by the lag model, thereby showing an improvement over the other models. The most complex validation case was the wing body junction flow, in which no extraordinary improvements were observed for the lag parameter model.

2.4.3. Underlying k-ε Model

The lag model by Lardeau and Billard [19] uses a low Reynolds number version of the $k-\varepsilon$ model as underlying two-equation model. The transport equations for the turbulence kinetic energy k and the rate of dissipation of turbulence kinetic energy ε are given by

$$\frac{Dk}{Dt} = P - \varepsilon + \frac{\partial}{\partial x_j}\left[\left(\frac{\nu}{2} + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{2.58}$$

$$\frac{D\varepsilon}{Dt} = \frac{1}{\tau}\left(C_{\varepsilon 1}P - C_{\varepsilon 2}\varepsilon\right) + E + \frac{\partial}{\partial x_j}\left[\left(\frac{\nu}{2} + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] \tag{2.59}$$

where E represents the viscous effects by the wall and is defined as

$$E = C_k \nu \nu_t (1-\alpha)^3 \left(\frac{\partial |2S_{ij}n_j| n_k}{\partial x_k}\right)^2 . \tag{2.60}$$

This E term represents a damping function and is only active near solid walls, as the elliptic blending variable α goes to zero as the wall is approached. Away from the wall, α reaches unity and the E term reduces to zero. Fixes to popular turbulence models often come at the cost of impaired model robustness or unintended activity in certain flow regions [19]. In order to overcome these limitations, many recent developments in the RANS community have seen the introduction of additional transport equations [19], such that more information about turbulent behaviour can be introduced into the flow field [17, 45, 46]. In addition to the transport equations of the $k-\varepsilon$ model, Lardeau and Billard [19] derived a transport equation for the lag parameter φ. Its derivation starts with the definition for the eddy viscosity, which is given as

$$\nu_t = C_\mu \varphi \frac{k^2}{\varepsilon} \tag{2.61}$$

in which C_μ is a constant and φ represents the lag effect on the eddy viscosity. This definition for the lag parameter shows similarity in many ways to the definition used in Billard and Laurence [45], in which $\varphi \equiv \overline{v^2}/k$ is introduced to model the damping of turbulence near the wall [19]. In its exact form, the production term P is given by

$$P = -\overline{u_i' u_j'} \frac{\partial \overline{u_i}}{\partial x_j} \tag{2.62}$$

which for incompressible flow can also be written as

$$P = -k a_{ij} S_{ij} \,. \tag{2.63}$$

Please note that only in this section a_{ij} indicates the normalized anisotropy tensor, while usually the normalized isotropy tensor is indicated by b_{ij}. This is done to follow the notation used in the work of [17, 19]. Linear eddy viscosity models are based on the Boussinesq hypothesis, for which the production term becomes

$$P = 2\nu_t S_{ij} S_{ij} = \nu_t S^2 \tag{2.64}$$

where S denotes the strain-rate magnitude ($S = \sqrt{2 S_{ij} S_{ij}}$). In order to get an expression for φ, it is assumed that (2.61) and (2.63) are equal definitions of the eddy viscosity [19]:

$$-\frac{k a_{ij} S_{ij}}{S^2} = C_\mu \varphi \frac{k^2}{\varepsilon} \,. \tag{2.65}$$

Equation (2.65) provides an expression for the lag parameter as function of the resolved quantities, which is given by

$$\varphi = \underbrace{-\frac{a_{ij} S_{ij}}{S}}_{\xi} \frac{1}{C_\mu} \underbrace{\frac{\varepsilon}{kS}}_{\eta^{-1}} = \frac{\xi}{C_\mu \eta} \,. \tag{2.66}$$

The term denoted with ξ is equivalent to the definition of the parameter C_{as}, which is used in the lag model of Revell et al. [17]. It indicates the degree of alignment between the strain and stresses. The ratio of the mean-velocity strain to the turbulent time scales is indicated by the term η. The transport equation of φ follows from the total derivative of φ, which is given by

$$\frac{D\varphi}{Dt} = \frac{1}{C_\mu \eta} \frac{D\xi}{Dt} - \frac{\xi}{C_\mu \eta^2} \frac{D\eta}{Dt} \,. \tag{2.67}$$

Revell et al. [17] used a Reynolds Stress Model to exactly derive the transport equation of ξ, which is given as

$$\frac{D\xi}{Dt} = -\frac{S_{ij}}{kS} \left[P_{ij} - \varepsilon_{ij} + \phi_{ij}^* - a_{ij} \left(P - \varepsilon \right) \right] - \frac{1}{S} \frac{DS_{ij}}{Dt} \left(a_{ij} + \frac{2 S_{ij} \xi}{S} \right) \tag{2.68}$$

The Elliptic-Blending Reynolds Stress model by Manceau [47] is used to obtain expressions for the first four terms on the right-hand side of (2.68). Please refer to the work of [47] for the fully elaborated terms. In order to derive the final expression for the transport equation of φ, the transport equation of η needs to be defined, which is obtained from

$$-\frac{\varphi}{\eta} \frac{D\eta}{Dt} = -\varphi \frac{P}{k} \left(1 - C_{\varepsilon_1} \right) + \left(1 - C_{\varepsilon_2} \right) \frac{\varphi}{\tau} \,. \tag{2.69}$$

Finally, the transport equation for the lag parameter φ is obtained and is given by

$$\begin{aligned}\frac{D\varphi}{Dt} = &-\left(1-\alpha^3\right) C_w^* \frac{\varphi}{\tau} - \alpha^3 \left(\tilde{C}_1 + C_1^* \frac{P}{\varepsilon}\right) \frac{\varphi}{\tau} + \alpha^3 \frac{1}{S^2\tau} \left(C_4^* a_{ik} S_{kj} - C_5^* a_{ik} W_{kj}\right) S_{ij} \\ &- C_{P1} \frac{\varphi P}{k} + \alpha^3 C_{P2} \varphi S + \alpha^3 \frac{C_{P3}}{\tau} + \frac{\partial}{\partial x_j} \left[\left(\frac{\nu}{2} + \frac{\nu_t}{\sigma_\varphi}\right) \frac{\partial \varphi}{\partial x_j}\right] .\end{aligned} \tag{2.70}$$

The eddy viscosity is calculated as

$$\nu_t = C_\mu \varphi k \min\left(\tau_{lim},\ \frac{C_T}{C_\mu \sqrt{3} \varphi S}\right) \tag{2.71}$$

where the turbulent time scales are

$$\tau_{lim} = \sqrt{\tau^2 + C_t^2 \frac{\nu}{\varepsilon}}, \quad \tau = \frac{k}{\varepsilon} . \tag{2.72}$$

The elliptic equation that is solved for α is given by

$$\alpha - L^2 \frac{\partial}{\partial x_j} \left(\frac{\partial \alpha}{\partial x_j}\right) = 1 \tag{2.73}$$

where the turbulent length scale L is defined as

$$L = C_L \sqrt{\frac{k^3}{\varepsilon^2} + C_\eta^2 \sqrt{\frac{\nu^3}{\varepsilon}}} . \tag{2.74}$$

The constants are as follows

$$\begin{aligned} C_w^* = C_{\varepsilon 2} - 1 + C_w - \frac{1}{C_\mu}, \quad \tilde{C}_1 = C_1 + C_{\varepsilon 2} - 2, \quad C_{P1} = 2 - C_{\varepsilon 1}, \quad C_{P2} = \frac{C_3^*}{\sqrt{2}}, \\ C_{P3} = \frac{f_\mu}{C_\mu} \left(\frac{2}{3} - \frac{C_3}{2}\right), \quad C_4^* = \frac{2}{C_\mu} \left(1 - C_4\right), \quad C_5^* = \frac{2}{C_\mu} \left(1 - C_5\right) \end{aligned} \tag{2.75}$$

where

$$C_1 = 1.7, \quad C_1^* = 0.9, \quad C_3 = 0.8, \quad C_3^* = 0.65, \quad C_4 = 0.625, \quad C_5 = 0.2, \quad C_w = 5 . \tag{2.76}$$

2.4.4. Underlying k-ω Model

Biswas et al. [21] recently developed a lag model where the $k-\omega$ model was the underlying two-equation model. The $k-\omega$ lag model is given by

$$\frac{Dk}{Dt} = P - \beta^* k\omega + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right) \frac{\partial k}{\partial x_j}\right] \tag{2.77}$$

$$\frac{D\omega}{Dt} = \gamma_{new} \frac{P}{k} \omega - \beta \omega^2 + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\omega}\right) \frac{\partial \omega}{\partial x_j}\right] \tag{2.78}$$

where

$$\gamma_{new} = \left(1 - \alpha^3\right) \gamma_1 + \alpha^3 \gamma_2 , \tag{2.79}$$

which is used to make sure that the production of turbulence kinetic energy beyond the wake is not spurious [21]. During the development of the model, Biswas et al. followed an approach that was similar to the development of the LAG EB $k-\varepsilon$ by Lardeau and Billard [19]. The expression for the lag parameter φ^* and its transport equation, respectively, are given by

$$\varphi^* = -\frac{a_{ij} S_{ij}}{S} \frac{\omega}{S} \tag{2.80}$$

$$\begin{aligned}\frac{D\varphi^*}{Dt} = &-\left(1-\alpha^3\right)C_w^*\varphi^*\omega - \alpha^3\left(\tilde{C}_1 + C_1^*\frac{P}{\beta^* k\omega}\right)\varphi^* k\omega - C_{p1}\frac{P}{k}\varphi^* + \alpha^3 C_{p2}\varphi^* S \\ &+\alpha^3\frac{C_{p3}}{\varphi_h}\beta^*\omega + \alpha^3\frac{\beta^*\omega}{\varphi_h}\left(C_4^* a_{ik}S_{kj} - C_5^* a_{ik}W_{kj}\right)S_{ij} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\phi}\right)\frac{\partial\varphi^*}{\partial x_j}\right]\end{aligned} \tag{2.81}$$

where the expression for the normalized anisotropy tensor, here denoted as a_{ij}, is only used to close the penultimate term and is defined as

$$a_{ij} \simeq -2\frac{\nu_t}{k}\left[S_{ij} + \beta_2\frac{2\left(S_{ik}W_{kj} - W_{ik}S_{kj}\right)}{\left|\left(S_{kl}+W_{k}l\right)\left(S_{kl}+W_{kl}\right)\right|}\right]. \tag{2.82}$$

The eddy viscosity is determined as

$$\nu_t = k\min\left(\frac{\varphi^*}{\omega},\, \frac{\alpha_s}{|S|}\right). \tag{2.83}$$

The elliptic equation is similar to that of the $k-\varepsilon$ model, which is shown in (2.73), but with a different turbulent length scale that is defined as

$$L^2 = C_L^2\left(\frac{k^3}{\left(\beta^* k\omega\right)^2} + C_\eta^2\sqrt{\frac{\nu^3}{\beta^* k\omega}}\right). \tag{2.84}$$

The model constants are given by

$$\begin{aligned}\beta^* = 0.09, \quad \sigma_k = 2, \quad \gamma_1 = 0.5, \quad \gamma_2 = 0.6, \quad \beta = 0.075, \quad \sigma_\omega = 2, \quad C_w^* = 0.05, \quad \tilde{C}_1 = 1.6, \\ C_1^* = 0.9, \quad C_{p1} = 0.4, \quad C_{p2} = 0.46, \quad \varphi_h = 0.41, \quad C_4^* = 3.41, \quad C_5^* = 7.27, \quad \sigma_\varphi = 1, \\ C_L = 0.164, \quad C_\eta = 75, \quad C_1 = 1.7, \quad C_5 = 0.2\,.\end{aligned} \tag{2.85}$$

Biswas et al. [21] bounded the anisotropy tensor shown in (2.82), which is only used to close the penultimate term in (2.81), to make it realizable. It was given the following bounds:

$$\begin{aligned} a_{ij} &= \max\left(\min\left(a_{ij},\, 1\frac{1}{3}\right),\, -\frac{2}{3}\right) && \text{for } i = j, \\ a_{ij} &= \max\left(\min\left(a_{ij},\, 1.0\right),\, -1.0\right) && \text{for } i \neq j. \end{aligned}$$

3

Data-Driven RANS Turbulence Modeling

The purpose of this chapter is to provide the reader with the necessary background information to understand the current developments in data-driven turbulence modeling and to place the method used in this study into perspective. It also aims to provide the reader insight into the most commonly used machine learning algorithms, and provide a motivation for the machine learning method used in this study.

This chapter starts by giving a very brief introduction to machine learning algorithms in Section 3.1. A historical perspective and state-of-the-art developments in data-driven RANS turbulence modeling approaches are given in Section 3.2. In addition, the choice for the machine learning method considered in this study is justified. A more in-depth background is given in Sections 3.3 to 3.5 for the most common machine learning algorithms in data-driven RANS turbulence modeling.

3.1. Brief Introduction

Machine learning involves the study of algorithms that are used to discover patterns in data, such that a prediction or decision can be made based upon those patterns. Within the field of machine learning, several distinctions can be made between algorithms, which are typically categorized as follows

- Supervised learning
- Unsupervised learning
- Reinforcement learning

In supervised learning, algorithms are given a data set that contains inputs and the corresponding desired outputs. The algorithm is trained to map these inputs to the desired outputs. After the training phase, the algorithm is fed with new unseen data. The algorithms map these new inputs to outputs based on the patterns found in the training phase. In unsupervised learning, no outputs are provided to the inputs. Therefore, the algorithm has to find a structure in the data by itself, and thus, the learning is unsupervised. In the last category, reinforcement learning, a software agent has a specific task, but it doesn't know yet how to perform this task in the best way possible. The agent interacts with its environment and learns by trial and error.

In this study, the focus will be on supervised learning algorithms. As there are many different algorithms, some of the most commonly used algorithms in data-driven turbulence modeling will be discussed below.

3.2. Historical Perspective and State-of-the-Art

In 1895, Reynolds published a paper in which he presented the findings of his research on turbulence. In this paper, Reynolds decomposed the velocity into a mean and fluctuating component. Nowadays, this decomposition is referred to as the Reynolds decomposition, showing the importance of his work on the field of turbulence modeling. In 1877, years before Reynolds published his paper, Boussinesq introduced the concept of an eddy viscosity [32]. Prandtl introduced the mixing length model in 1925,

nowadays referred to as an algebraic model, which provided a way to compute the eddy viscosity. In the years after, the field of RANS modeling has seen many developments, such as one-equation models, two-equation models and second-order closures to mention a few. However, in the past decades, the development of RANS models has been stagnant, as can be seen from the fact that models such as the $k-\varepsilon$ [48], $k-\omega$ [49] and Spalart-Allmaras [50] model are still currently used, but were developed decades ago [51]. Although the development of new models with increased complexity is an ongoing process [52–54] and has shown to result in accurate predictions in some cases, it can be argued that over the past 15 years there has been no truly significant improvement in the accuracy of the predictions [24]. Despite all the effort, RANS models that are universally applicable with predictive capabilities are still lacking [29]. High-fidelity approaches such as DNS or LES have seen developments over the years that open up new prospects. However, DNS has been seen as a research tool and LES has proven to be a powerful method, but is still limited to low-to-moderate Reynolds number flow and relatively simple geometries [15].

Nowadays, the field of RANS modelling still remains an active field with the introduction of data-driven turbulence modelling. Recent advances include the use of machine learning algorithms and high-fidelity data. With the focus on RANS models that utilize the Boussinesq hypothesis, Duraisamy et al. [22–24] use data from high-fidelity simulations and experiments to infer a spatially varying multiplier $\boldsymbol{\beta}(\boldsymbol{x})$ of the production term in the Spalart-Allmaras model. Bayesian inversion is used to infer the spatial distribution of these discrepancies. In [22], Duraisamy and Parish use Gaussian processes to extract the functional relationship $\boldsymbol{\beta}(\boldsymbol{\eta})$ between the corrective field and the non-dimensional input features to enhance the models' capability to generalize. Their proposed modeling paradigm, which is termed field inversion and machine learning (FIML), is applied to a turbulent channel flow. The spatially varying multiplier was applied to the production term in the transport equation of the turbulence kinetic energy of the k-ω model by Wilcox [55]. This work was mainly of illustrative nature and requires a wider class of problems to improve the predictions in practical situations [22]. In the work of [24], the spatial distributions $\boldsymbol{\beta}(\boldsymbol{x})$ from a large number of inverse problems are transformed into model forms $\boldsymbol{\beta}(\boldsymbol{\eta})$ using an artificial neural network. In their work, three wind turbine airfoils with varying thickness that involve flow separation were used as demonstration. The predictions of lift and drag coefficients were significantly improved in comparison to the baseline model. They succeeded in generalizing the models' predictions to airfoil shapes and flow conditions that the model had not seen before. However, future work is required to make the models applicable to a wide range of problems. Singh [56] observed that in some problems it was not possible to accurately develop the functional relationship $\boldsymbol{\beta}(\boldsymbol{\eta})$. In order to ensure that the inferred field is learnable, Duraisamy et al. [57] introduced two methods that embed a neural network within the inversion procedure. Both methods were tested on a two-dimensional wind turbine airfoil and proved to generate learnable model discrepancy functions. The capability to generalize of the two methods is to be explored in future work.

A different approach by Emory et al. [25, 26] focuses on the the Boussinesq assumption and its associated model-form uncertainty. In their work, a perturbation to the eigenvalues of the anisotropy tensor is introduced in order to represent a local injection of uncertainty. For each cell, the Reynolds stress predicted by the k-ω SST model is mapped onto the barycentric map, after which it is pulled towards one of the corners of the barycentric map [25], thereby perturbing the eigenvalues. As a result, the anisotropy tensor, and consequently the Reynolds stress tensor, are modified. The altered Reynolds stress tensor is then fed to the RANS solver. In [25], this framework is demonstrated for steady transonic flow a two-dimensional bump, which is a flow case that involves streamline curvature and separation. In the work of [26], the applied framework is twofold. First, flow regions in which the predictions are plausibly erroneous are identified and the development of these so-called marker functions is part of ongoing work. Then, uncertainty is injected by perturbing the eigenvalues. The proposed framework is demonstrated for turbulent channel flow, square duct flow and transonic flow over a two-dimensional bump. Emory et al. were able to generate secondary recirculation in the square duct flow, which can not predicted by any original eddy-viscosity closure. Edeling et al. [16, 27] use an approach that is build upon the work of Emory et al. [26]. By introducing two additional transport equations, the perturbations are made representative of local flow features. These transport equations involve linear combinations of the eigenvalues of the anisotropy tensor (C_{1c} and C_{2c}) and are inspired by the lag model of Olsen and Coakley [58]. The difference between these coefficients determine the amount of perturbation $\Delta\lambda_\alpha$ from the baseline state. Furthermore, the sign of this difference determine in which direction of the barycentric map the perturbations are made. In the study of [16], the model was applied

to flow over a turbulent flat plate and it was found that the bounds on the quantities of interest were determined by the same a priori known coefficient values. Following up on this work, Edeling et al. [27] introduced two strategies to quantify uncertainty in predictions. One of these methods involves an approach in which no reference data is used, also referred to as the data-free approach. The second approach is referred to as the data-driven approach, which uses Bayesian inference to infer the posterior distribution of the two lag coefficients. Both approaches have been applied to a subsonic jet and a backward-facing step flow case. Edeling et al. showed that the lag equations have the ability to bring the turbulence back to a realizable state, even if the baseline model violates the realizability constraints [27].

The application of data-driven techniques to physical systems has raised questions with regard to the incorporation of domain knowledge such as invariances or symmetries, and more specifically, to what extent it can and should be done. For example, the Navier-Stokes equations obey Galilean invariance, which means that any scalar flow variable is invariant to rotations, reflections or translations of the reference frame. Ling et al. [28] find that it would be desirable, for the accuracy of the model and physical realizability, that machine learning algorithms respect invariance properties in case the corresponding physical system obeys one. Therefore, they explore two approaches that address the question of how a machine learning model can be taught an invariance or symmetry property. The first approach, in which a machine learning algorithm is trained on a functional basis of invariant inputs, uses domain knowledge to embed the invariance property into the machine learning model. In the second approach, the machine learning model was trained on several transformed versions of the data, such that it was able to learn the invariance property itself. Two machine learning algorithms, namely the Random Forest and the Neural Network, were tested on a wall-mounted cube in crossflow, for which RANS and DNS data was available. This flow case involves regions in which the Reynolds stress anisotropy is significant [59], three-dimensionality, flow curvature, stagnation and separation [28]. For the first approach, the performance by both algorithms was very similar. However, the neural network proved to have a better performance for the second approach, as they were better able to learn the invariance property, which does not come as a surprise due to their capability of deep learning [28]. A notable result was the fact that the models that were trained on the basis of invariant inputs yielded higher accuracy. In the study of [3], deep neural networks with embedded variance are used to predict the Reynolds stress anisotropies and to demonstrate the capability of the model to generalize to flows with a vastly different Reynolds number. The predicted Reynolds stress anisotropy tensor is propagated through the velocity field for a turbulent duct flow and for flow over a wavy wall. The tensor basis neural network (TBNN) was able to predict secondary flow, but slightly overpredicted the strength of the corner vortices. In the wavy wall flow case, the TBNN was able to predict separation, but incorrectly predicted the size of the separated region. Compared to the linear eddy viscosity model (k-ε) and the quadratic eddy viscosity model (based on [60]), the TBNN showed an improved prediction of the mean velocity. A broader set of flows is required for the train and test phase of the model in order to reach its full potential.

RANS models are limited in their predictive accuracy, for which the main source of error lays in the large discrepancies in the modeled Reynolds stresses. Wang et al. [29] propose an approach in which these discrepancies $\Delta\tau_\alpha$ are reconstructed using DNS databases, i.e., $\Delta\boldsymbol{\tau} = \boldsymbol{\tau}^{DNS} - \boldsymbol{\tau}^{RANS}$. Instead of reconstructing these discrepancies in the space of physical coordinates $\boldsymbol{x}$, as in the work of Wu et al. [61], Wang et al. reconstructs them in a space of mean flow features $\boldsymbol{q}$ that are Galilean invariant, thereby overcoming the limitation of the spatial dependency. This allows for extrapolation to a much broader range of flows that share the same characteristics such as flow curvature and separation. Random forests are used to construct the regression function $f_\alpha : \boldsymbol{q} \mapsto \Delta\tau_\alpha$. This approach has been tested by using two classes of flows, namely fully developed turbulent flow in a square duct at various Reynolds numbers and flows that involve massive separation. The first prediction test case is the flow through a square duct ($Re = 3500$), in which secondary flow occurs. DNS data for the same square duct geometry, but at lower Reynolds numbers, have been used as training data. For the class of flows that involve massive separation, the periodic hill flow case ($Re = 10595$) was selected as test case, which features separation, mean flow curvature and a nonparallel shear layer. Two different ways of training were explored. First, the algorithm was trained on DNS data from the same geometry, but at lower Reynolds numbers. The second training scenario involved DNS and LES data from two different geometries and Reynolds numbers, namely the wavy channel ($Re = 360$) and the curved backward-facing step ($Re = 13200$). In comparison to the baseline predictions by the RANS model, significant improvements were observed

for both test cases. However, these improvements were less renounced when different geometries were used for the training and test cases, thereby showing the difficulty of generalization. In the work of [62], Wu et al. demonstrate a systematic approach for the selection of features as input to the machine learning algorithm, thereby improving the ad-hoc approach to selecting features in [29]. Furthermore, they address the problem of having ill-conditioned RANS equations, which can be a consequence of substituting predicted Reynolds stresses explicitly into the RANS equations, especially in flows with high Reynolds numbers [63, 64]. Wu et al. split the anisotropy tensor into a linear and nonlinear part as $\boldsymbol{b} = \nu_t^L \boldsymbol{S} + \boldsymbol{b}^\perp$ and build regression functions to predict ν_t^L and $\boldsymbol{b}^\perp$ using random forest regression and DNS data. This splitting allows the enhancement of the conditioning of the RANS equations, as the linear term can be treated implicitly [62]. Flow in a square duct at $Re = 2200$ is used to train the algorithm, after which it is tested on the same geometry, but with increased Reynolds numbers of $Re = 3500$ and $Re = 125000$. As a final scenario, the periodic hill flow case at $Re = 5600$ was used as a training case and predictions were made for the flow over periodic hills at the same Reynolds number, but with a modified geometry, as the training case had a steeper hill profile. The proposed framework demonstrated its capability to satisfactory predict the mean velocity field, and furthermore, it successfully predicted a mean flow pattern that the machine learning algorithm had encountered in the training phase.

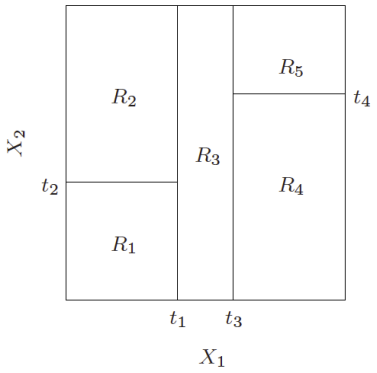
Weatheritt & Sandberg [30, 31] use symbolic regression and Gene Expression Programming (GEP) to model the anisotropy of the Reynolds stress tensor. As the algorithm is non-deterministic, there solution is non-unique, and therefore, a sample is taken from multiple runs of the machine learning algorithm. GEP is used to construct a functional $a_{ij} = a_{ij}(V_{ij}^1, V_{ij}^2, \dots, I^1, I^2, \dots)$, which uses a linearly independent basis of four basis functions and two scalar invariants, derived through the Cayley-Hamilton theorem [30]. DNS results behind a backward-facing step at the reattachment point are used for the training phase. The Reynolds number, which is based on the step height h, is $Re_h = 3000$. The algorithm is tested on two cases, namely the backward-facing step flow case with an increased Reynolds number, i.e., $Re_h = 5100$. In addition, the algorithm is assessed on its performance to generalize to a different flow field, as flow over periodic hills is used as a test case, in which the Reynolds number is $Re_h = 10595$. The resulting models proved to be very effective when the test case was similar to the training case. However, as the model was specifically trained on one flow field, the predictions in the favorable pressure gradient region were poor. In the study of [31], statistics from a hybrid RANS/LES database [65] are used to make regression models, which means that a full flow field of DNS data is not required. The models are trained on several duct flows with varying aspects ratios and a Reynolds number of $Re_b = 10000$. Realizability has not been incorporated into their framework, but all converged models were realizable. The capability to generalize to a different flow field is assessed by including an asymmetric diffuser as test case. An improved performance of the new models over the baseline models was observed for flow through the asymmetric diffuser. Furthermore, their approach showed excellent numerical stability. Recently, Weatheritt & Sandberg [66] used this approach to approximate the Reynolds stress tensor in a junction body flow.

Schmelzer et al. [67] used a deterministic framework to learn the form of models for the additive term b_{ij}^Δ, which accounts for the model-form error. High-fidelity data from a DNS and LES of flow over periodic hills at different Reynolds number has been used to learn the model and for validation purposes. Bayesian inversion is used to identify the coefficient fields $\boldsymbol{a}$ of the base tensor series given the data $\boldsymbol{b}^\Delta$. Rough coefficient fields are discarded as a lack of smoothness might lead to overfitting. A library of candidate functions is build and deterministic symbolic regression is used to identify the most relevant ones. In order to do so, an optimisation problem for the vector of coefficients $\Theta^{(n)}$ is solved, which includes a regularisation term to promote sparsity. The model forms can then be retrieved from $M := b_{ij}^\Delta = \sum_{n=1}^{N} \mathcal{B}\boldsymbol{\Theta}^{(n)} T_{ij}^{(n)}$. In this study, two models $\mathrm{M}^{(1)}$ and $\mathrm{M}^{(2)}$ were developed and trained on the periodic hill flow case. The first model was trained on flow over periodic hills with a Reynolds number of $Re_H = 10595$, while the second model was trained at $Re_H = 2800$. In general, both models had an improved performance compared to the baseline k-ω model for predictions of the streamwise velocity at $Re_H = 2800$ and $Re_H = 10595$. In addition, both models predicted the reattachment point more accurately than the baseline LEVM for the case at $Re_H = 10595$. In the work of [5], a deterministic machine learning method referred to as SpaRTA is introduced, which is an abbreviation for Sparse Regression of Turbulent Stress Anisotropy. As an extension to the method in [31], k-corrective-frozen-RANS is introduced to account for the model-form error using high-fidelity data. This approach utilizes a correction b_{ij}^Δ to the stress-strain relationship, as well as a correction R to

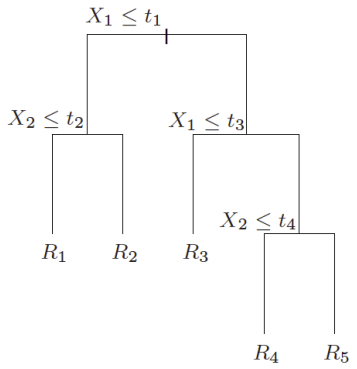
the turbulent transport equations. Schmelzer et al. validated that the model-form error was successfully captured for the periodic hill, the curved backward-facing step and the converging-diverging channel flow cases. The k-corrective-frozen-RANS is a cost-effective approach, as no inversion procedures are required. However, full-field data is necessary for the computation of the corrective terms. SpaRTA is used to construct separate models for these corrections and involves building a library of candidate functions, selection of models using sparse regression and inference to obtain a model with the correct units. The training cases included flow over periodic hills, over a curved backward-facing step and through a converging-diverging channel. Per training case, several models were constructed using the SpaRTA approach. Cross-validation was used to assess the performance of these models for unseen flow cases, which allowed for a selection of models with the best predictive capability. Overall, an improved performance in predictive capability was observed when corrections to both the stress-strain relationship and the transport equations were applied. In addition, an assessment of the effect of applying only a correction to the transport equations was performed, in which it was shown that sufficient results could be achieved for the velocity field. The approach of Schmelzer et al [68] has several advantages. First of all, it allows for a robust and quick way of enhancing RANS turbulence models. The corresponding computational cost is low [68] and the implementation of additional (algebraic) terms is straightforward [69]. Secondly, this method is an open-box machine learning approach, as the resulting corrective terms are constructed using the integrity basis and are interpretable. The SpaRTA approach, which is a symbolic regression method, will be used in this work and is elaborated upon in Section 4.5.

3.3. Tree-Based Methods

Partitioning the feature space into a number of regions R_m and fitting a model in each of these regions is the main idea of tree-based methods. A common algorithm for tree-based regression is the CART algorithm, which is an abbreviation for Classification And Regression Tree. Figures 3.1a and 3.1b show a visualization of the segmentation of the feature space (X_1, X_2) and the corresponding decision tree, respectively.



(a) Segmentation of two-dimensional feature space by recursive binary splitting [70].



(b) Corresponding decision tree [70].

Figure 3.1: CART principle.

As can be seen, a tree consists of several nodes that are connected by branches. The nodes at the end of the branches are referred to as terminal nodes or leaves and correspond to the regions R_1, $R_2 \ldots, R_5$ in Fig. 3.1b. Each junction in the tree has an assigned condition. The regression model that describes a tree is given by

$$f(x) = \sum_{m=1}^{M} c_m I\left(x \in R_m\right) \tag{3.1}$$

in which I is a function that indicates whether the observed inputs x fall into region R_m. As example, c_m could be chosen to be the average of all the response y_i in the region R_m, which is equal to minimizing the sum of squares $\sum(y_i - f(x_i))^2$. However, the computational costs corresponding to the search for the best segmentation that leads to a minimum sum of squares can become excessive. Therefore, a greedy algorithm is used [70]. At each interior node, the feature space is split for feature X_j at the

point s, which in formula form is written as

$$R_1(j,s) = \{X | X_j \leq s\} \quad \text{and} \quad R_2(j,s) = \{X | X_j > s\} \,. \tag{3.2}$$

Then, the following optimisation problem is solved for the splitting variable j and the split point s

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right] . \tag{3.3}$$

Trees are high-variance estimators, which is a major problem [70]. Consequently, a small change in the input could lead to a completely different tree. A possibility to reduce the variance of an estimator is to use bootstrap aggregation or bagging. In bagging, samples are taken from the data set and the same regression model is fit to each one of these. Each bootstrap tree corresponds to a different sample, and typically, contains different splitting features, a different number of leaves and different cut-points. Then, in regression, the results by the individual models are averaged to create the prediction. In classification problems, the majority of the votes by the individuals models would be the final prediction.

Trees can also be used for the ensemble-technique boosting, in which a set of weak classifiers is used to create a strong classifier. The most popular boosting algorithm is AdaBoost, which can be used on different machine learning algorithms, but works well for decision trees. AdaBoost uses a procedure in which multiple trees are trained sequentially on a data set with certain weights. Initially, each data point in the set is given the same weight $w_i = \frac{1}{N}$, and thus, the first tree is trained on an unmodified data set. The weights of the data set are then modified, i.e., weights of data points that were misclassified are increased, while the weights of data points that were correctly classified are decreased. This modified data set is then used to train the next tree, such that the successive tree is forced to focus on the data points that were missed by its predecessors. This iterative process is repeated until a certain number of trees is generated. The final prediction is then a weighted majority vote of all the individual predictions.

In random forests, which is an ensemble of trees, each tree is trained on a different subset of the data set. During the construction of a tree, the splitting features are randomly selected (and are a subset) from the complete set of features. This mechanism of random feature selection helps to decorrelate the trees, which in turn reduces the prediction variance of the random forest.

3.4. Artificial Neural Networks

Artificial neural networks are used in both regression as classification problems. Such networks consist of several neurons or nodes, which are organised in layers. Within a layer, the neurons are not interconnected, but the neurons between adjacent layers are. A schematic view of the architecture of a neural network is given in Fig. 3.2.

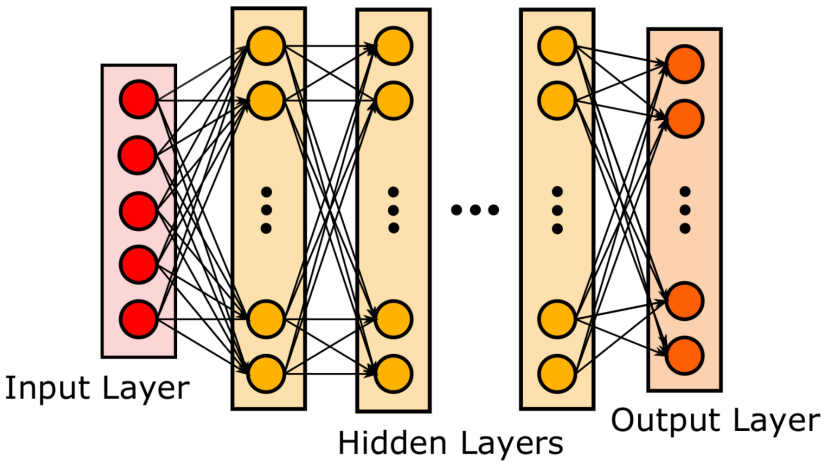


Figure 3.2: Schematic of the architecture of a neural network [3].

As can be seen, this network has an input layer, multiple hidden layers and an output layer, which all consist of several neurons. Each neuron takes in the inputs $\boldsymbol{x}$ with corresponding weights $\boldsymbol{w}$ including

a bias term b as well. Finally, the output is created by transforming the result through an activation function σ. Several activation functions exist, such as linear functions, sigmoids and the rectified linear unit (ReLU). Activation functions are used such that an artificial neural network has the ability to represent complex non-linear behaviour. The output vector A_i^l of a layer l is given by

$$A_i^l = \sigma\Big(\underbrace{W_{ij}^{l-1}A_j^{l-1} + b_i^{l-1}}_{z_i^l}\Big) \tag{3.4}$$

in which W_{ij}^{l-1} denotes the matrix of weights of the connections between layers l and $l-1$, A_j^{l-1} denotes the output of the previous layer and b_i^{l-1} are the bias terms. The amount of hidden layers, which is referred to as the depth of the neural network, the amount of neurons in a layer and the type activation function are important hyperparameters that require tuning. In general, it is better to have too many neurons in a hidden layer than too few, such that the nonlinearities in the data can be captured by the algorithm. A possible consequence of using too many neurons is that one overfits the model. However, using appropriate regularization, the extra weights can be shrunk towards zero [70]. During the training phase of a neural network, in which its weights and biases are refined, the predicted output is computed through a series of matrix-vector products and matrix manipulations in a forward pass and compared to the true data. In order to evaluate the algorithm's prediction, a cost function is used, which measures the error between the predicted output and the true data. As mentioned before, to prevent overfitting, regularization is needed, which can be achieved by stopping early or by introducing a penalty term to the cost function. Then, this cost function J is typically minimized by gradient descent, referred to as back-propagating. In this procedure, the gradients of the cost function with respect to the weights and biases are calculated via the chain rule as

$$\frac{\partial J}{\partial W_{ij}^l} = \frac{\partial J}{\partial z_i^l}\frac{\partial z_i^l}{\partial W_{ij}^l} = \frac{\partial J}{\partial z_i^l}A_j^{l-1} \tag{3.5}$$

$$\frac{\partial J}{\partial b_i^l} = \frac{\partial J}{\partial z_i^l}\frac{\partial z_i^l}{\partial b_i^l} = \frac{\partial J}{\partial z_i^l} \tag{3.6}$$

The weight and biases are then updated at the $(t+1)^{st}$ training iteration according to

$$W_{ij}^{t+1} = W_{ij}^t - \gamma_t\frac{\partial J}{\partial W_{ij}^t} \tag{3.7}$$

$$b_i^{t+1} = b_i^t - \gamma_t\frac{\partial J}{\partial b_i^t} \tag{3.8}$$

in which y_t denotes the learning rate. Commonly, the approach of mini-batches gradient descent [71] is used, in which the training data set is split in mini-batches. This approach provides a compromise between computational expenses and the efficiency of the memory and its use is widespread in the training of artificial neural networks [72]. A part of the training set is kept separate during the training phase, such that it can be used as a validation set, in which the ability of the model to generalize is assessed.

3.5. Symbolic Regression

Traditional methods such as linear or nonlinear regression methods assume a functional of a given form, and subsequently, fit parameters to that functional. Instead of assuming a certain (set of) functional(s), symbolic regression algorithms search the space of mathematical expressions to find a functional that best matches a given data set. Furthermore, symbolic regression methods simultaneously perform a search for the parameters. Two subsets of symbolic regression will be discussed here, which are Gene Expression Programming (GEP) and Sparse Regression.

3.5.1. Gene Expression Programming

The main philosophy behind Gene Expression Programming is the evolution of its candidate solutions by survival of the fittest [30]. It is a non-deterministic process, as the changes that are made at every iteration are random. The formation of a candidate solution is performed in the following steps:

1. Randomly select a top-level node, which can only be an operator node.
2. Pick the next lower-level nodes, which can be operator, constant or variable nodes.
3. Continue the above step until only nodes from the terminal set (constant or variables) are at the extremities of the tree.
4. Repeat the above three steps, such that more trees are build until a pre-determined population size of trees has been reached.

Each tree is evaluated for its performance on predicting the desired output value given the data set. Furthermore, the trees are ranked according to this performance, after which a certain set is used in the formation of new trees for the following iteration. An example of such an evaluation of the performance, or fitness, is given by Weatheritt and Sandberg [30] and is defined as

$$\mathrm{Fit}(f^{\mathrm{guess}}) = 1 - \frac{1}{n}\sum_{k=1}^{n}\frac{||f^{\mathrm{guess}}(x_k,\, y_k) - f_k||}{||f_k||} \tag{3.9}$$

where a score of 1 defines the highest fitness. In general, the set of constants is not predefined, and therefore, Random Numerical Constants (RNCs) are introduced. The RNCs have values that lie between the two predefined numbers r_{min} and r_{max}. The GEP algorithm starts by randomly creating N candidate solutions such that a population P is reached. The next step is the selection process, in which the search is directed away from poorer functionals by having a tournament selection. This tournament selection, which acts as a filter on the population, puts a larger weight on the chance of survival compared to more fit solutions [30]. Then, the genetic operator modifies the candidate solutions, thereby exploring new solutions that combine certain aspects of existing solutions. As a final step, poor candidate solutions are filtered out in the selection process.

3.5.2. Sparse Regression

Models that are based on the principle of sparse regression are formulated using the form of an optimization problem [73], which is given as

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}\left(\mathcal{D};\boldsymbol{\Theta}\right) + \lambda\Omega\left(\boldsymbol{\Theta}\right) \tag{3.10}$$

in which $\mathcal{L}$ denotes a loss function, λ is a regularization control parameter and Ω denotes a regularization term. In addition, $\mathcal{D}$ indicates the training data set, which includes the features $\boldsymbol{X}$ and the target values $\boldsymbol{T}$, and $\boldsymbol{\Theta}$ denotes the parameter or coefficient set. For example, the loss function could be defined as a least square error function, which would become

$$\mathcal{L}\left(\mathcal{D};\boldsymbol{\Theta}\right) = \frac{1}{2}\left\|\boldsymbol{T} - \boldsymbol{X}\boldsymbol{\Theta}\right\|^2 . \tag{3.11}$$

There are two reasons to introduce a regularisation term, namely an improvement in prediction accuracy and interpretation. Using only the Ordinary Least Squares (OLS) estimates can lead to a prediction in which the bias is low and the variance high [74] and often leads to a poor performance in prediction and and interpretation [75]. The problem of high variance could be reduced by shrinking the coefficients with the goal of improving the predictive capability. However, this comes at the cost of an increase in bias. A possible approach to shrink coefficients is by using Ridge regression, in which a penalty is imposed on the size of the coefficients. The regularisation term is given by $\Omega(\Theta_j) = \lambda\sum_{j=1}^{p}\Theta_j^2$. The optimization problem that is solved in Ridge regression then reads

$$\Theta^{\mathrm{ridge}} = \arg\min_{\Theta}\left\{\sum_{i=1}^{N}\left(T_i - \Theta_0 - \sum_{j=1}^{p}X_{ij}\Theta_j\right)^2 + \lambda\sum_{j=1}^{p}\Theta_j^2\right\} \tag{3.12}$$

in which $\lambda \geq 0$. Its value determines the amount of shrinkage, e.g., a larger regularization parameter leads to more shrinkage. However, the coefficients are not shrunk to zero and for that reason, models can be hard to interpret. Subset selection can be used to improve the interpretability, but comes with the risk of variability in the models [74].

A different approach is Lasso regression, in which the regularisation term is given by $\Omega(\Theta_j) = \lambda \sum_{j=1}^{p} |\Theta_j|$. The corresponding optimization reads

$$\Theta^{\text{lasso}} = \arg\min_{\Theta} \left\{ \sum_{i=1}^{N} \left(T_i - \Theta_0 - \sum_{j=1}^{p} X_{ij}\Theta_j \right)^2 + \lambda \sum_{j=1}^{p} |\Theta_j| \right\} \tag{3.13}$$

in which the added penalty is equal to the sum of the absolute value of the coefficients. In contrast to Ridge regression, the coefficients can actually be shrunk to zero, and therefore, be eliminated from the model and improve interpretability. As the regularisation parameter increases, more and more coefficients are set to zero. Determining the value of this parameter is a trade-off between bias and variance, e.g., increasing λ leads to an increase in bias and a reduction in variance and vice versa. A limitation of the Lasso is that when groups of highly correlated variables exist, lasso tends to select only one variable from each group [75]. However, in many applications, a grouping effect is considered desirable. One speaks of a grouping effect when the coefficients corresponding to a group of highly correlated variables are approximately equal.

Another approach, referred to as elastic net regression [75], combines the aforementioned l_1- and l_2-norm regularisation terms. By applying this blending of the l_1- and l_2-norm, the elastic net has the ability to identify sparse models and has a good predictive capability. In addition, the elastic net is able to select groups of variables that are highly correlated [75]. The optimisation problem that is solved in elastic net regression is given by

$$\Theta^{\text{elastic net}} = \arg\min_{\Theta} \left\{ \sum_{i=1}^{N} \left(T_i - \Theta_0 - \sum_{j=1}^{p} X_{ij}\Theta_j \right)^2 + (1-\alpha)\,\lambda \sum_{j=1}^{p} |\Theta_j| + \alpha\lambda \sum_{j=1}^{p} \Theta_j^2 \right\} \tag{3.14}$$

in which α denotes the parameter that determines the mix of the l_1- and l_2-norm regularisation terms.

4

Methodology

This chapter provides a description of the methodology that is used in this research. Its aim is to provide the reader with a description of the implementation, motivation and assessment of the k-corrective-frozen-RANS approach in combination with the lag parameter model. In addition, it aims to provide insight into the SpaRTA method, which is used to infer functionals for the corrective terms.

First, the model-form error of the RANS equations is introduced in Section 4.1. Capturing this model-form error is the aim of the frozen-RANS approach, which is discussed in Section 4.2. In Section 4.3, extending upon the idea of frozen-RANS, an additional corrective term is introduced in the k-corrective-frozen-RANS approach, which is the method used in this study to capture the model-form error. Then, Section 4.4 describes the two frameworks that are used to assess the effect of the additional transport equation for the lag parameter with regard to generalizability. The machine learning approach, which is utilized to identify models for the two corrective terms that are used to capture the model-form error, is presented in Section 4.5. Finally, the computational cost of this approach is discussed in Section 4.6.

4.1. Model-Form Error of RANS Equations

In the RANS approach, many modeling assumptions are made, which in turn leads to a solution that differs from the true solution. An introduction to the model-form error starts from the RANS equations (2.4 and 2.5), which, for the ease of reading, are shown again and are defined as

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0\,,$$

$$\frac{\bar{D}\overline{u_i}}{\bar{D}t} = \nu \frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j} - \frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j}\,.$$

As this system of equations remains unclosed, the Reynolds stress tensor $\tau_{ij} = \overline{u_i' u_j'}$ requires modeling, which is also referred to as Reynolds stress closure modeling. Various closures exist, such as linear eddy viscosity models, nonlinear eddy viscosity models and Reynolds stress models, which have been subject of discussion in Section 2.2.3. All of these closures introduce sources of inaccuracies, such as model-form and model-parameter errors, which are also referred to as structural and parametric errors, respectively. Figure 4.1 provides a schematic overview of these inaccuracies for RANS turbulence modelling.

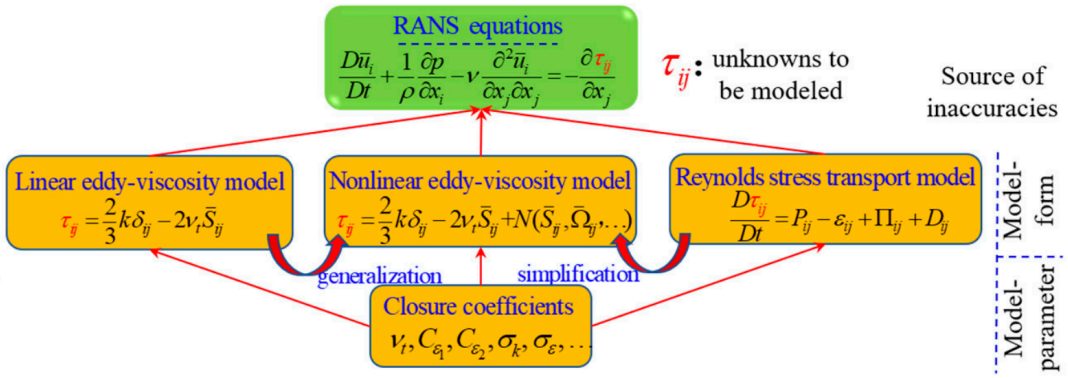


Figure 4.1: Schematic overview of structural (model-form) and parametric (model-parameter) inaccuracies [4].

In this study, the goal is to capture the model-form error $\tau_{ij}^{\Delta} = \tau_{ij}^{\text{DNS/LES}} - \tau_{ij}^{\text{RANS}}$ by utilizing a general form of NLEVMs, based on the integrity basis discussed in Section 2.3.4. In order to do this, the k-corrective-frozen-RANS approach will be used, which is the subject of discussion in the next section.

4.2. Frozen-RANS

In the frozen-RANS approach [31], high-fidelity data is used to enhance a RANS turbulence model. In contrast to the field inversion and adjoint approach (Appendices A and B), the frozen-RANS approach does not depend on an optimization procedure. For that reason, it is a very cost-efficient method. It is, however, also limited to full-field data. In this approach, the residuals of the baseline turbulence model given the high-fidelity data are computed, such that the model-form error can be extracted. The baseline turbulence model utilizes a linear relation between the anisotropy tensor b_{ij} and the mean rate-of-strain S_{ij}, which is given by the Boussinesq assumption as

$$b_{ij}^{0} \simeq -\frac{\nu_t}{k} S_{ij} \,. \tag{4.1}$$

Let high-fidelity data be denoted as $\{\cdot\}^{\star}$, so that $b_{ij}^{\star}$ is the normalized anisotropy tensor, $k^{\star}$ is the turbulence kinetic energy, and $\overline{u_i}^{\star}$ is the mean velocity field, all from DNS or LES. The computation of the residuals is equivalent to adding an additive term b_{ij}^{Δ} as

$$b_{ij}^{\star} \simeq -\frac{\hat{\nu}_t}{k^{\star}} S_{ij}^{\star} + b_{ij}^{\Delta} \,. \tag{4.2}$$

In order to evaluate b_{ij}^{Δ}, an estimation of the eddy viscosity is necessary. In frozen-RANS, this is done by solving the ω transport equation using high-fidelity data, i.e.

$$\frac{D\hat{\omega}}{Dt} = \gamma \frac{P^{\star}}{k^{\star}} \hat{\omega} - \beta \hat{\omega}^2 + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\hat{\nu}_t}{\sigma_\omega}\right)\frac{\partial \hat{\omega}}{\partial x_j}\right] , \tag{4.3}$$

such that the obtained ω can be used for the computation of the eddy viscosity ν_t. The quantities that are solved for are indicated using $\{\hat{\cdot}\}$. The term frozen-RANS [31] stems from the fact that the remaining variables are frozen as one equation is solved. A schematic overview of this approach is shown in Fig. 4.2.

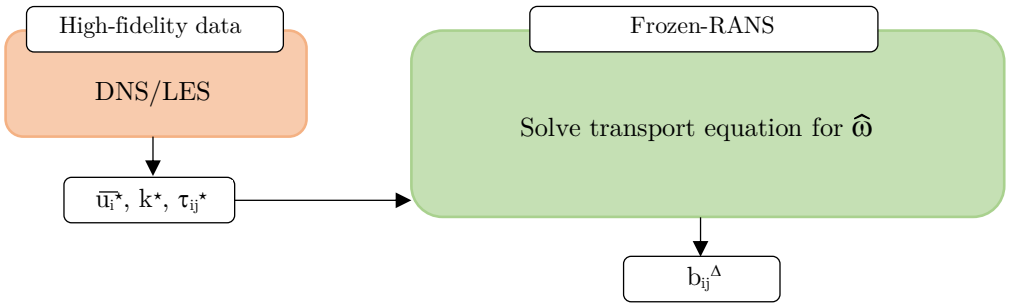


Figure 4.2: Schematic overview of frozen-RANS approach.

4.3. Theory of k-Corrective-Frozen-RANS

Extending on the idea of the frozen-RANS approach [31], the k-corrective-frozen-RANS [5] also attempts to capture the model-form error. In the work of [5], in addition to the use of the additive term b_{ij}^Δ, a corrective term $\hat{\boldsymbol{R}}$ is introduced to the transport equations of the $k-\omega$ SST model, which leads to the following set of augmented equations

$$\frac{Dk^\star}{Dt} = P^\star + \hat{\boldsymbol{R}} - \beta^* k^\star \hat{\omega} + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_k \hat{\nu}_t)\frac{\partial k^\star}{\partial x_j}\right], \tag{4.4}$$

$$\frac{D\hat{\omega}}{Dt} = \gamma \frac{P^\star + \hat{\boldsymbol{R}}}{k^\star}\hat{\omega} - \beta \hat{\omega}^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \hat{\nu}_t)\frac{\partial \hat{\omega}}{\partial x_j}\right] + CD_{k\omega}\,. \tag{4.5}$$

The corrective term $\hat{\boldsymbol{R}}$ is the equivalent of the residual of the k-equation, which is coupled with the transport equation for ω. The reason for using this corrective term $\hat{\boldsymbol{R}}$ is that the additive term b_{ij}^Δ also changes the production term $P = -\overline{u_i' u_j'}\frac{\partial \overline{u_i}}{\partial x_j}$ in the transport equations of k and ω, and it is not evident that solving the transport equation of k while using the high-fidelity data, will result in the same k as that from high-fidelity data [68].

Now that all necessary variables are specified, (4.2) can be utilised for the computation of b_{ij}^Δ. When the iterative process of k-corrective-frozen-RANS has converged, the additive terms b_{ij}^Δ and $\hat{\boldsymbol{R}}$ are added as static fields in OpenFOAM, after which a simulation is performed for the flow cases in Chapter 5 using the same initial conditions. A schematic overview of the approach used by Schmelzer et al. [5] is given in Fig. 4.3.

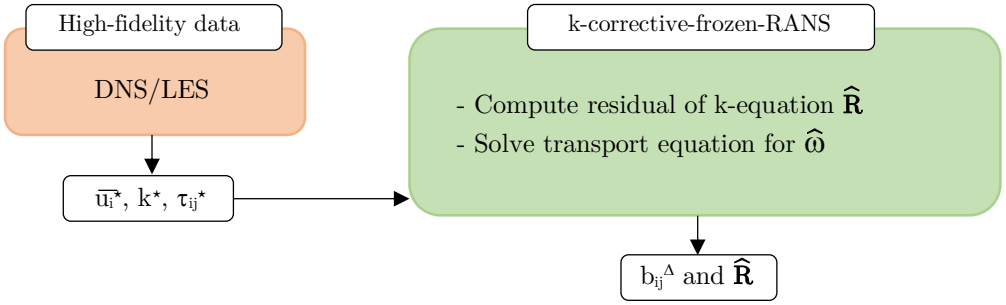
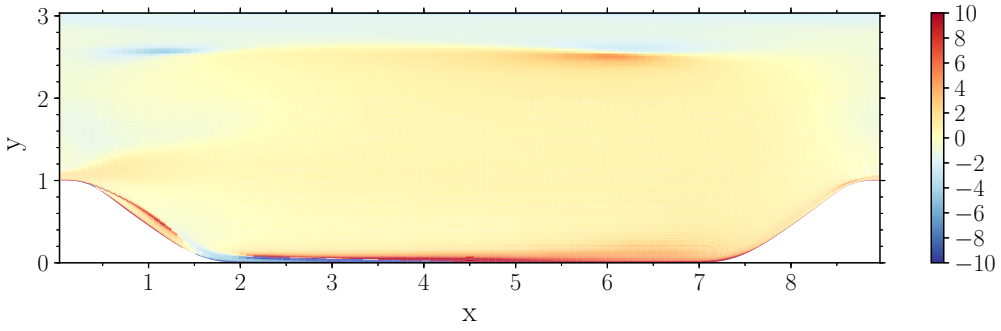


Figure 4.3: Schematic overview of the k-corrective-frozen-RANS approach used by Schmelzer et al. [5].

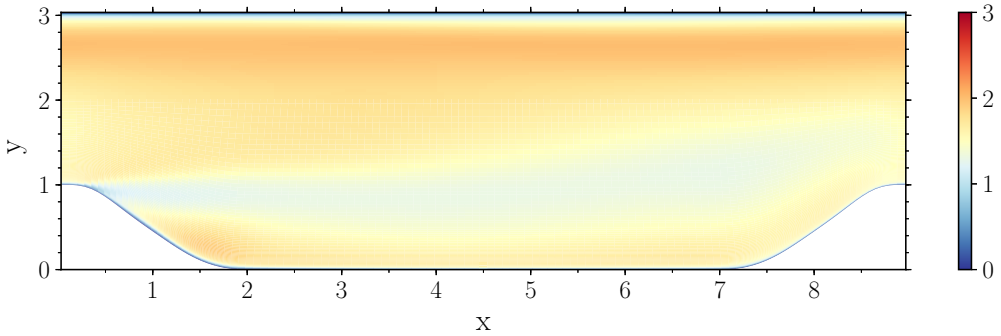
In this study, the k-corrective-frozen-RANS is used in combination with the lag parameter model by Biswas et al. [21]. Since this model has an additional transport equation for the lag parameter φ^*, there are many possible generalizations of the k-corrective-frozen-RANS approach, which raises questions:

- Should φ^* be computed directly from high-fidelity data using its definition or should its transport equation be used?

  In order to research this possibility, first a comparison between the lag parameter $\varphi^{*\star}$ from high-fidelity data and the lag parameter φ^* from RANS is made.



(a) Distribution of the lag parameter $\varphi^{*\star}$ computed from LES data for the periodic hill flow case.



(b) Distribution of the lag parameter φ^* obtained from RANS for the periodic hill flow case.

Figure 4.4: Comparison of lag parameter distribution.

As can be seen from Figs. 4.4a and 4.4b, the distributions are significantly different, so computing the lag parameter directly from high-fidelity data would not be wise. The definition given by (2.80) has k and ω as variables, which are solutions of a RANS approximation, and for that reason, do not correspond to a high-fidelity solution. Furthermore, even if k and ω were to be computed accurately from high-fidelity data, the assumptions taken to derive the transport equation of the lag parameter are too large to allow a comparison [76]. Therefore, it has been decided to compute φ^* from its transport equation.

- If the transport equation is used, should a corrective term be included?

  As aforementioned, the solutions from high-fidelity simulations and from RANS are fundamentally different, and therefore, it has been decided that a single additive corrective term, which would account for the deficit $\varphi^*_{deficit}$ between the high-fidelity solution and the RANS solution, is not required.

  For the same reason as in the transport equations of k and ω, one could argue to use a corrective term $\hat{\boldsymbol{R}}$ as an addition to the production terms in the transport equation of φ^*. The effect of this has been tested by comparing the results that are obtained when the standard transport equation of φ^* is used, i.e. (4.6) with the results that are obtained when an augmented version of the transport equation of φ^* is used, in which the corrective term $\hat{\boldsymbol{R}}$ is introduced, i.e. (4.7).

$$\begin{aligned}\frac{D\hat{\varphi}^*}{Dt} = &-\left(1-\alpha^3\right) C^*_w \hat{\varphi}^* \hat{\omega} - \alpha^3 \left(\tilde{C}_1 + C^*_1 \frac{P^\star}{\beta^* k^\star \hat{\omega}}\right) \hat{\varphi}^* k^\star \hat{\omega} - C_{p1} \frac{P^\star}{k^\star} \hat{\varphi}^* + \alpha^3 C_{p2} \hat{\varphi}^* S^\star \\ &+ \alpha^3 \frac{C_{p3}}{\varphi_h} \beta^* \hat{\omega} + \alpha^3 \frac{\beta^* \hat{\omega}}{\varphi_h} \left(C^*_4 a_{ik} S^\star_{kj} - C^*_5 a_{ik} W^\star_{kj}\right) S^\star_{ij} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\hat{\nu}_t}{\sigma_\phi}\right) \frac{\partial \hat{\varphi}^*}{\partial x_j}\right],\end{aligned} \tag{4.6}$$

$$\frac{D\hat{\varphi}^*}{Dt} = -\left(1-\alpha^3\right) C_w^* \hat{\varphi}^* \hat{\omega} - \alpha^3 \left(\tilde{C}_1 + C_1^* \frac{P^\star + \hat{\boldsymbol{R}}}{\beta^* k^\star \hat{\omega}}\right) \hat{\varphi}^* k^\star \hat{\omega} - C_{p1} \frac{P^\star + \hat{\boldsymbol{R}}}{k^\star} \hat{\varphi}^* + \alpha^3 C_{p2} \hat{\varphi}^* S^\star + \alpha^3 \frac{C_{p3}}{\varphi_h} \beta^* \hat{\omega} + \alpha^3 \frac{\beta^* \hat{\omega}}{\varphi_h} \left(C_4^* a_{ik} S_{kj}^\star - C_5^* a_{ik} W_{kj}^\star\right) S_{ij}^\star + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\hat{\nu}_t}{\sigma_\phi}\right) \frac{\partial \hat{\varphi}^*}{\partial x_j}\right]. \tag{4.7}$$

The resulting mean velocity field for the periodic hill case is shown in Fig. 4.5, in which the labels 'Without $\hat{\boldsymbol{R}}$' and 'with $\hat{\boldsymbol{R}}$' indicate the usage of (4.6) and (4.7), respectively. It can be seen that the model-form error is better captured when the corrective term $\hat{\boldsymbol{R}}$ is used in the transport equation of the lag parameter φ^*. Without this corrective term, the streamwise velocity profiles are overestimated for $y/h \geq 1.5$ and underestimated for $y/h \leq 1.5$. The resulting turbulence kinetic energy profiles show an even more significant difference, as can be seen from Fig. 4.6. By including the corrective term, an improvement of all predicted streamwise turbulence kinetic energy profiles is observed. Therefore, the corrective term is included in the transport equation of the lag parameter.

- In case of a corrective term, what type of correction is best to use and where should it be placed?

There are many possibilities when it comes to applying a corrective term. A separate term $\varphi^*_{deficit}$ that accounts for the difference between the high-fidelity solution and the RANS solution has not been applied, as the solutions are fundamentally different. Instead, an approach similar to that of Schmelzer et al. [68] has been used, in which a corrective term is added to the production term. Furthermore, this corrective term is also modelled in a similar way to the production term. Therefore, to conclude the k-corrective-frozen-RANS approach for the lag parameter model, the corrective term $\hat{\boldsymbol{R}}$ will be included as is shown in (4.7).

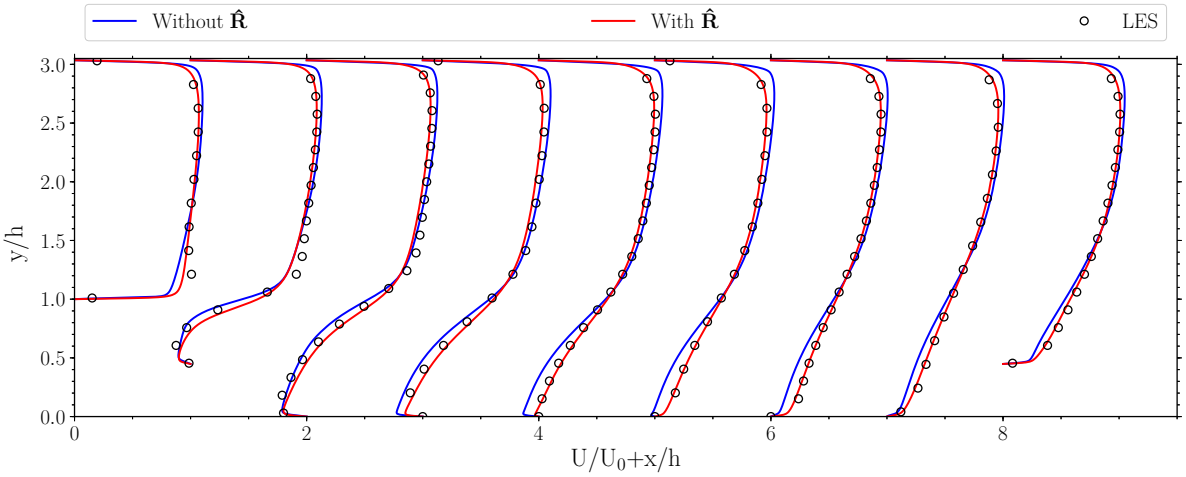


Figure 4.5: Effect of a corrective term $\hat{\boldsymbol{R}}$ in the transport equation of φ^* on the resulting mean velocity field for the periodic hill flow case.

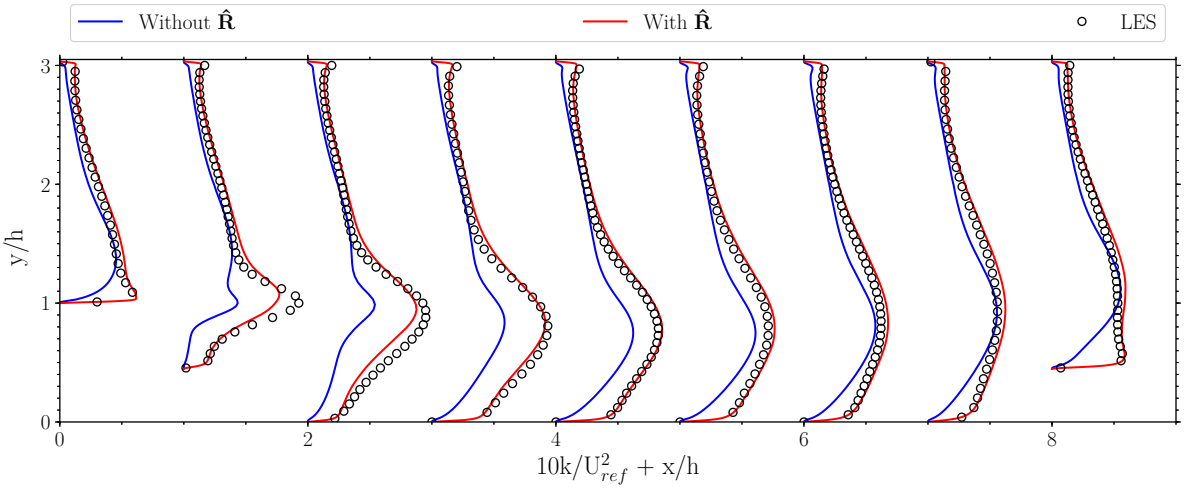


Figure 4.6: Effect of a corrective term $\hat{\boldsymbol{R}}$ in the transport equation of φ^* on the resulting turbulence kinetic energy for the periodic hill flow case.

4.4. Overview of Frameworks

The k-corrective-frozen-RANS approach will be used in two different frameworks, in order to assess the effect of the lag parameter. Therefore, the first framework uses a baseline model, which is discussed in Section 4.4.1. The second framework uses the elliptic blending lag parameter model and is showed in Section 4.4.2.

4.4.1. Framework 1 - Standard k-ω Model

The first framework uses an augmented version of the standard k-ω model, which is given by

$$\frac{Dk^\star}{Dt} = P^\star + \hat{\boldsymbol{R}}_\mathbf{1} - \beta^* k^\star \hat{\omega} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\hat{\nu}_t}{\sigma_k}\right)\frac{\partial k^\star}{\partial x_j}\right] \tag{4.8}$$

$$\frac{D\hat{\omega}}{Dt} = \gamma\frac{P^\star + \hat{\boldsymbol{R}}_\mathbf{1}}{k^\star}\hat{\omega} - \beta\hat{\omega}^2 + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\hat{\nu}_t}{\sigma_\omega}\right)\frac{\partial \hat{\omega}}{\partial x_j}\right] \tag{4.9}$$

in which the eddy viscosity is computed as

$$\hat{\nu}_t = \frac{k^\star}{\hat{\omega}}\,. \tag{4.10}$$

Alongside the computation of ω, the variables $\overline{u_i}$, k and b_{ij} are kept frozen and the residual R is computed from (4.8) and fed back into (4.9). The outputs are the additive terms b_{ij}^{Δ} and R. The main purpose of this framework is to assess the performance of the SpaRTA approach on a standard turbulence model and to provide a baseline for comparison with the second framework.

4.4.2. Framework 2 - Elliptic Blending Lag Parameter Model

An augmented version of the elliptic blending lag parameter model (Section 2.4.4) is used in the second framework. Equations (4.11) to (4.13) show the augmented equations of the elliptic blending $k-\omega$ lag parameter model.

$$\frac{Dk^\star}{Dt} = P^\star + \hat{\boldsymbol{R}}_{\boldsymbol{2}} - \beta^* k^\star \hat{\omega} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\hat{\nu}_t}{\sigma_k}\right)\frac{\partial k^\star}{\partial x_j}\right] \tag{4.11}$$

$$\frac{D\hat{\omega}}{Dt} = \gamma_{new}\frac{P^\star + \hat{\boldsymbol{R}}_{\boldsymbol{2}}}{k^\star}\hat{\omega} - \beta\hat{\omega}^2 + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\hat{\nu}_t}{\sigma_\omega}\right)\frac{\partial \hat{\omega}}{\partial x_j}\right] \tag{4.12}$$

$$\begin{aligned}\frac{D\hat{\varphi}^*}{Dt} = &-\left(1-\alpha^3\right)C_w^*\hat{\varphi}^*\hat{\omega} - \alpha^3\left(\tilde{C}_1 + C_1^*\frac{P^\star + \hat{\boldsymbol{R}}_{\boldsymbol{2}}}{\beta^* k^\star \hat{\omega}}\right)\hat{\varphi}^* k^\star \hat{\omega} - C_{p1}\frac{P^\star + \hat{\boldsymbol{R}}_{\boldsymbol{2}}}{k^\star}\hat{\varphi}^* + \alpha^3 C_{p2}\hat{\varphi}^* S^\star \\ &+ \alpha^3\frac{C_{p3}}{\varphi_h}\beta^*\hat{\omega} + \alpha^3\frac{\beta^*\hat{\omega}}{\varphi_h}\left(C_4^* a_{ik}S_{kj}^\star - C_5^* a_{ik}W_{kj}^\star\right)S_{ij}^\star + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\hat{\nu}_t}{\sigma_\phi}\right)\frac{\partial \hat{\varphi}^*}{\partial x_j}\right]\end{aligned} \tag{4.13}$$

$$\hat{\nu}_t = k^\star \min\left(\frac{\hat{\varphi}^*}{\hat{\omega}}, \frac{\alpha_s}{|S|^\star}\right). \tag{4.14}$$

While U_i, k and τ_{ij} are kept frozen, the transport equations for ω and φ^* are solved, in which the computed residual from the k transport equation is fed back into (4.12) and (4.13). Furthermore, the elliptic equation for α is solved, as the elliptic blending parameter α is used in the transport equations of (4.12) and (4.13). In this framework, SpaRTA is used in combination with the elliptic blending lag parameter model and the main purpose of this framework is to be able to assess its performance in terms of generalizability, interpretability and ability to infer the quantities of interest. A schematic overview of this framework is presented in Fig. 4.7.

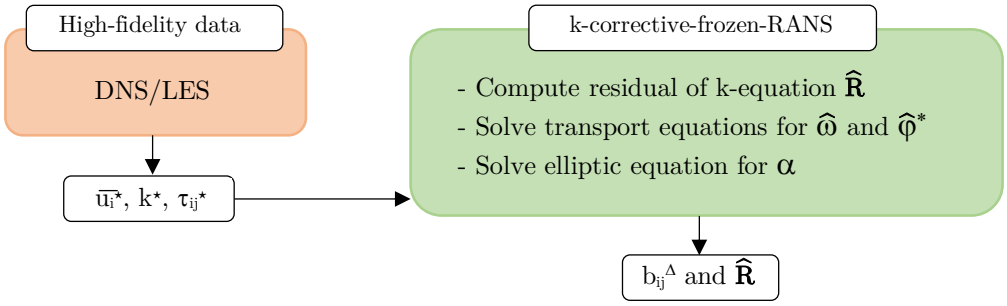


Figure 4.7: Schematic overview of framework 2.

4.5. Sparse Regression of Turbulent Stress Anisotropy

Opposed to the non-deterministic Gene Expresson Programming algorithm, SpaRTA is a deterministic symbolic regression algorithm. It constructs a set of functionals to perform regression. As a first step, a library of candidate functions have to be build.

4.5.1. Construction of Candidate Functions

In the work of [5], the input features I_1 and I_2 are squared to form the following set:

$$\boldsymbol{\mathcal{B}}_1 = \left[I_1,\, I_2,\, I_1^2,\, I_2^2\right]^T. \tag{4.15}$$

The candidates in (4.15) are multiplied with each other to form the set $\boldsymbol{\mathcal{B}}_2$, which is given by

$$\boldsymbol{\mathcal{B}}_2 = \left[I_1 I_2,\, I_1^2 I_2,\, I_1^3,\, I_2^2 I_1,\, I_2^3,\, I_1^2 I_2^2\right]^T. \tag{4.16}$$

As a final step in the process of the construction of candidate functions, the candidates in $\boldsymbol{\mathcal{B}}_1$ are multiplied with the candidates in $\boldsymbol{\mathcal{B}}_2$, which leads to a maximum degree of 6. Duplicates are removed and a constant is added, which leads to the following final set of input features:

$$\boldsymbol{\mathcal{B}} = \left[c,\, I_1,\, I_2,\, I_1^2,\, I_2^2,\, I_1^2 I_2^3,\, I_1^4 I_2^2,\, I_1 I_2^2,\, I_1 I_2^3,\, I_1 I_2^4,\, I_1^3 I_2,\, I_1^2 I_1^4,\, I_1^2 I_2,\, I_1 I_2,\, I_1^3 I_2^2,\, I_1^2 I_2^2\right]^T \tag{4.17}$$

which has a total of 16 elements, i.e. $|\boldsymbol{\mathcal{B}}| = 16$.

In this study, a similar approach is followed in which, opposed to Schmelzer et al [5], the full integrity basis is used. The set of input features then consists of 207 features:

$$\boldsymbol{\mathcal{B}} = \left[c,\, I_1,\, I_2,\, I_3,\, I_4,\, I_5, \ldots,\, I_3^2 I_4^2 I_5^2,\, I_4^3 I_5^2,\, I_4^4 I_5^2\right]^T . \tag{4.18}$$

Functionals for b_{ij}^{Δ} and R are explored in the form of

$$b_{ij}\left(S_{ij},\, \Omega_{ij}\right) = \sum_{n=1}^{10} T_{ij}^{(n)} \alpha_n \left(I_1, \ldots, I_5\right) , \tag{4.19}$$

$$R = 2k b_{ij}^{R} \frac{\partial \overline{u_i}}{\partial x_j} , \tag{4.20}$$

in which α_n denotes the coefficients that can be dependent upon the five invariants.

Therefore, in order to create a library of tensorial candidate functions to regress models for b_{ij}^{Δ}, each of the input features in $\boldsymbol{\mathcal{B}}$ is multiplied with each base tensor $T_{ij}^{(n)}$, resulting in the following library

$$\boldsymbol{\mathcal{C}}_{b_{ij}^{\Delta}} = \left[cT_{ij}^{(1)},\, cT_{ij}^{(2)}, \ldots,\, I_4^4 I_5^2 T_{ij}^{(10)}\right]^T . \tag{4.21}$$

The library of candidate functions to identify models for R is created by taking the double dot product of each term in the library of $\boldsymbol{\mathcal{C}}_{b_{ij}^{\Delta}}$ with the mean velocity gradient tensor $\frac{\partial \overline{u_i}}{\partial x_j}$, which results in the following library

$$\boldsymbol{\mathcal{C}}_R = \left[cT_{ij}^{(1)} \frac{\partial \overline{u_i}}{\partial x_j},\, cT_{ij}^{(2)} \frac{\partial \overline{u_i}}{\partial x_j}, \ldots,\, I_4^4 I_5^2 T_{ij}^{(10)} \frac{\partial \overline{u_i}}{\partial x_j}\right]^T . \tag{4.22}$$

The terms in the libraries of $\boldsymbol{\mathcal{C}}_{b_{ij}^{\Delta}}$ and $\boldsymbol{\mathcal{C}}_R$ can then be evaluated for each validation case by using the corresponding high-fidelity data. After evaluation, the libraries read

$$\boldsymbol{\mathcal{C}}_{b_{ij}^{\Delta}} = \begin{bmatrix} cT_{xx}^{(1)}|_{k=0} & cT_{xx}^{(2)}|_{k=0} & \ldots & I_4^4 I_5^2 T_{xx}^{(10)}|_{k=0} \\ cT_{xy}^{(1)}|_{k=0} & cT_{xy}^{(2)}|_{k=0} & \ldots & I_4^4 I_5^2 T_{xy}^{(10)}|_{k=0} \\ cT_{xz}^{(1)}|_{k=0} & cT_{xz}^{(2)}|_{k=0} & \ldots & I_4^4 I_5^2 T_{xz}^{(10)}|_{k=0} \\ cT_{yy}^{(1)}|_{k=0} & cT_{yy}^{(2)}|_{k=0} & \ldots & I_4^4 I_5^2 T_{yy}^{(10)}|_{k=0} \\ cT_{yz}^{(1)}|_{k=0} & cT_{yz}^{(2)}|_{k=0} & \ldots & I_4^4 I_5^2 T_{yz}^{(10)}|_{k=0} \\ cT_{zz}^{(1)}|_{k=0} & cT_{zz}^{(2)}|_{k=0} & \ldots & I_4^4 I_5^2 T_{zz}^{(10)}|_{k=0} \\ \vdots & \vdots & & \vdots \\ cT_{zz}^{(1)}|_{k=K} & cT_{zz}^{(2)}|_{k=K} & \ldots & I_4^4 I_5^2 T_{zz}^{(10)}|_{k=K} \end{bmatrix} \in \mathbb{R}^{6K \times |\boldsymbol{\mathcal{C}}_{b_{ij}^{\Delta}}|} , \tag{4.23}$$

$$\boldsymbol{\mathcal{C}}_R = \begin{bmatrix} cT_{ij}^{(1)} \frac{\partial \overline{u_i}}{\partial x_j}|_{k=0} & \ldots & I_4^4 I_5^2 T_{ij}^{(10)} \frac{\partial \overline{u_i}}{\partial x_j}|_{k=0} \\ \vdots & & \vdots \\ cT_{ij}^{(1)} \frac{\partial \overline{u_i}}{\partial x_j}|_{k=K} & \ldots & I_4^4 I_5^2 T_{ij}^{(10)} \frac{\partial \overline{u_i}}{\partial x_j}|_{k=K} \end{bmatrix} \in \mathbb{R}^{K \times |\boldsymbol{\mathcal{C}}_R|}, \tag{4.24}$$

in which k denotes the number of grid points.

4.5.2. Model Selection using Elastic Net Regression

Following the construction of the library with candidate functions, the next step in the SpaRTA approach is the model selection. In order to to regress the target data $\boldsymbol{\Delta}$, which consists of $\boldsymbol{b}^{\Delta}$ and $\boldsymbol{R}$, a linear model is formed. This model is formed by finding the coefficients $\boldsymbol{\Theta}$ in

$$\boldsymbol{\Delta} = \boldsymbol{C}_{\Delta} \boldsymbol{\Theta} . \tag{4.25}$$

Equation (4.25) presents a system of equations that is overdetermined. Standard least-squares regression on (4.25) would result in a dense coefficient vector Θ [5]. Consequently, this would result in models

that are too complex with the risk of overfitting. Although the candidate functions are constructed not to be collinear, they can be approximately multi-collinear, and as a result, the libraries $\boldsymbol{C}_\Delta$ can be ill-conditioned. This in turn would have the consequence of having coefficients that significantly vary in magnitude. Resulting models would be unsuitable for CFD solvers as their implementation leads to additional numerical stiffness of the problem. Furthermore, it would hinder the convergence of the solution [5]. In SpaRTA, the selection of models is constrained such that the selected models have an optimal balance between error and complexity, and do not overfit the target data. A combinatoric study could be performed for the two libraries, but due to the exponential growth of the possible models with the amount of candidate functions I, this would already become unfeasible for the simple libraries mentioned in (4.21) and (4.22). Therefore, SpaRTA utilizes the elastic net formulation

$$\Theta = \arg\min_{\hat{\boldsymbol{\Theta}}} \left\| \boldsymbol{C}_\Delta \hat{\boldsymbol{\Theta}} - \boldsymbol{\Delta} \right\|_2^2 + \lambda\rho \left\| \hat{\boldsymbol{\Theta}} \right\|_1 + 0.5\lambda\left(1-\rho\right) \left\| \hat{\boldsymbol{\Theta}} \right\|_2^2 \tag{4.26}$$

which promotes the sparsity of the coefficients $\boldsymbol{\Theta}$ by blending the l_1-norm (Lasso regression) and l_2-norm (Ridge-regression). In (4.26), the mixing parameter is denoted by ρ and the regularisation weight by λ. Lasso regression only allows for a few nonzero coefficients, while setting the others coefficients to zero, thereby promoting sparsity. In contrast, Ridge-regression does not shrink the coefficients to zero, but does enforce the coefficients to be relatively small. In addition, it has the ability to identify the candidate functions that are correlated. The regularisation weight λ effects the model in such a way that for a very large regularisation weight, all the coefficients $\boldsymbol{\Theta}$ will be zero. By decreasing λ, the number of nonzero coefficients increase, which in turn allows the construction of a sparse model. It is necessary to specify the regularisation weight and the mixing parameter in order to solve the optimisation problem shown in (4.26). In [77], a range of mixing parameters is chosen, see (4.27), such that varying regularisation types are covered.

$$\boldsymbol{\rho} = [0.01,\, 0.1,\, 0.2,\, 0.5,\, 0.7,\, 0.9,\, 0.95,\, 0.99,\, 1.0]^T \,. \tag{4.27}$$

In total, 100 entries are used for the regularisation vector λ. These entries are uniformly spaced on a log-scale between $\lambda_0 = \varepsilon\lambda_{max}$ and λ_{max}, where $\varepsilon = 10^{-3}$. Now that the search space of $(\boldsymbol{\lambda}, \boldsymbol{\rho})$, also referred to as elastic net, is constructed, the coordinate descent algorithm is used to find the solution $\boldsymbol{\Theta}_\Delta^{(i,j)}$ at each point (λ_i, ρ_j) of the optimisation problem shown in (4.26). This could result in solution vectors $\boldsymbol{\Theta}_\Delta^{(i,j)}$ that have the same zero entries, i.e., non-unique model forms, and therefore, SpaRTA filters out the unique abstract model forms $\boldsymbol{\mathcal{D}}_\Delta = \left\{ \bar{\boldsymbol{\Theta}}_\Delta^d \,\middle|\, d = 1, \ldots, D \right\}$ where D denotes the amount of unique model forms.

4.5.3. Inference of Model Coefficients

The final step of SpaRTA involves the model inference, in which a model with the correct units is obtained. The candidate functions in $\boldsymbol{\mathcal{D}}_\Delta$ are standardised, such that the relevance of each function was not based on its magnitude. Therefore, an additional regression has to be performed that uses unstandardised candidate functions, such that the correct units are obtained. The convergence of solutions by CFD solvers is affected by large coefficients [30, 31, 66]. For that reason, the required additional regression will be performed by a Ridge regression

$$\boldsymbol{\Theta}_\Delta^{s,d} = \arg\min_{\hat{\boldsymbol{\Theta}}_\Delta^{s,d}} \left\| \boldsymbol{\mathcal{C}}_\Delta^S \hat{\boldsymbol{\Theta}}_\Delta^{s,d} - \boldsymbol{\Delta} \right\|_2^2 + \lambda_r \left\| \hat{\boldsymbol{\Theta}}_\Delta^{s,d} \right\|_2^2 , \tag{4.28}$$

in which s indicates the selected columns or elements in $\boldsymbol{C}_\Delta$ and $\boldsymbol{\Theta}_\Delta^d$, respectively, and λ_r denotes the Tikhonov-regularisation parameter. Finally, the models are obtained by taking the dot product of the library of candidate functions $\boldsymbol{\mathcal{C}}_\Delta$ and the coefficients $\boldsymbol{\Theta}_\Delta^d$ as

$$M_\Delta^d := \boldsymbol{\mathcal{C}}_\Delta^d \boldsymbol{\Theta}_\Delta^d \,. \tag{4.29}$$

4.6. Computational Cost

As the methodology can be divided in several phases, the computational cost of each phase will be discussed in this section. As this approach depends on full field data from high-fidelity simulations such

as DNS or LES, the corresponding computational cost can be quite significant if the high-fidelity data is not readily available. The required number of grid points for a DNS N_L^3 in three dimensions scales with $N_L^3 \sim L/\eta \sim \mathrm{Re}^{9/4}$, in which L is the integral length scale and η is the Kolmogorov length scale. The computational cost N_t in terms of time steps scales with $N_t \sim T/\Delta t \sim \mathrm{Re}^{3/4}$. Using the above estimates, the total computational cost N of a DNS scales as $N \sim N_L^3 \cdot N_t \sim \mathrm{Re}^3$.

The computational cost of a LES depends on the type of turbulent flow that is being solved. For homogeneous isotropic turbulence and free shear flows the amount of required modes is in the order of 40^3, which is satisfactory if one wants to resolve 80% of the energy [10]. In wall bounded flows, the costs of a LES scale differently, as the resolution requirements are different. Chapman [78] estimated that (in aerodynamic applications) a LES with near wall resolution scales with the Reynolds number as $\mathrm{Re}^{1.8}$, which makes its application in high-Reynolds number flow infeasible [10]. Alternatively, near-wall modeling can be used to overcome the problem of resolving the scales near the wall. Then, depending on the implementation, the computational expenses are not dependent on the Reynolds number or increase weakly with the Reynolds number as $\ln(\mathrm{Re})$ [10]. Generally speaking, complete databases that are tailored for the purpose of data-driven turbulence modeling can be said to be lacking [79]. However, new databases are being created for the sole purpose of data-driven turbulence modeling. For example, Xiao et al. [51] generated a database of DNS data for a family of flows over periodic hills that vary in slope.

The computational cost associated with the k-corrective-frozen-RANS approach is not significant, as the computation of the additive correction terms b_{ij}^{Δ} and R is performed within the order of a couple of minutes on a normal personal computer. Propagating these additive terms through a CFD solver results in a computational time that is in the order of tens of minutes. Furthermore, the computational costs associated with the machine learning phase are not significant, as SpaRTA utilizes sparse regression to identify the models, which is computationally inexpensive. The cross-validation phase is the most expensive part in terms of time, as the models have to be propagated through a CFD solver, which can take up to tens of minutes each time.

5

Test Cases

The purpose of this chapter is to provide the reader with a description of the four flow cases that are selected for the purpose of verification, comparison and assessment of the machine learning framework applied to the lag parameter model. For each of the cases, a description is given with regard to the boundary conditions, flow characteristics and source of the high-fidelity data. In addition, it aims to explain to the reader the relevance of the flow cases used in this study.

First, Section 5.1 describes the periodic hill flow case. Section 5.2 presents the converging-diverging channel. Then, the backward-facing step is shown in Section 5.3. Finally, the curved backward-facing step is presented in Section 5.4.

5.1. Periodic Hills

The periodic hill geometry is shown in Fig. 5.1. The distance between the two hill crests is $L_x = 9h$ and the channel height is $L_y = 3.035h$, where h denotes the height of the hill crest. The Reynolds number, which is based on the bulk velocity U_b and the hill crest height h, is $Re = 10595$. No-slip boundary conditions are imposed on the upper- and lower walls. Periodicity is assumed for the flow in the streamwise direction and thus cyclic boundary conditions are imposed on the inlet and outlet. In the streamwise direction, a source term is added to the momentum equation, because of the periodicity of the flow in this direction. High-fidelity data for this flow case was obtained from Breuer et al. [6]. In their research, the flow was solved using DNS for Reynolds numbers of $Re = 700$, $Re = 1400$, $Re = 2800$ and $Re = 5600$. In addition, high-resolved LES were done to solve the flow for Reynolds numbers of $Re = 5600$ and $Re = 10595$.

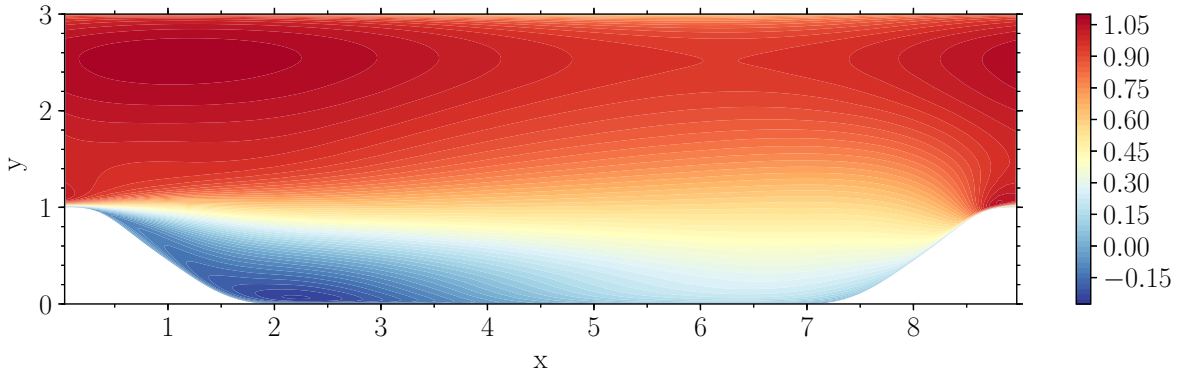


Figure 5.1: Contours of mean velocity magnitude $|U|$. $Re = 10595$. Data from a LES by Breuer et al [6].

Originally proposed by Mellen et al. [80], the periodic hill flow case is used as a test case that is computationally affordable and one that involves separation and reattachment. The flow case makes an interesting test case for the benchmarking of turbulence models, due to several properties, such as

mean flow curvature, non-parallel shear layers, separation from a curved surface and reattachment at a flat plate [6]. These properties pose a challenge for RANS models and are often encountered in flows of engineering interest [29].

5.2. Converging-Diverging Channel

The geometry of the converging-diverging channel is shown in Fig. 5.2. The hill crest is located near $x = 5.21564$ and has a height of approximately $y = 2/3$. The channel half-height is $h = 1$. The Reynolds number, which is based on the channel half-height h and the maximum velocity U_{max} at the inlet, is $Re = 12600$. On the upper- and lower wall, no-slip boundary conditions are imposed. Furthermore, the flow is fully developed at the inlet of the channel. Laval and Marquillie [7] used DNS to solve this flow case and made the results available through the TMR by NASA. In their research, two DNS were simulated for a friction Reynolds number of $Re_\tau = 395$ and $Re_\tau = 617$.

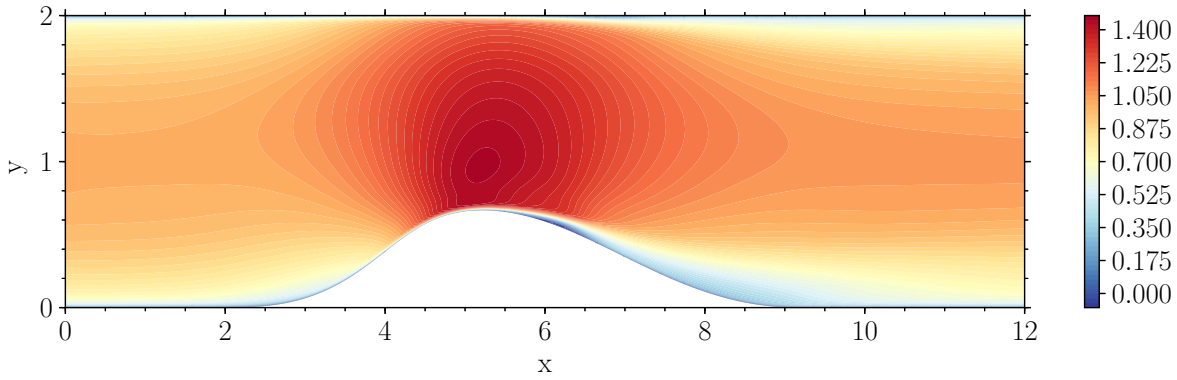


Figure 5.2: Contours of mean velocity magnitude $|U|$. $Re = 12600$. Data from a DNS by Laval and Marquillie [7].

This flow case involves curvature and an adverse pressure gradient, thereby making it an interesting test case for the evaluation of RANS models. The results indicate that separation occurs on the lower wall, and more specifically, on the leeward side of the hill. Separation of the flow starts at $x = 5.8$ on the lower wall and continues till $x = 6.6$, after which the flow reattaches.

5.3. Backward-Facing Step

The backward-facing step flow case is shown in Fig. 5.3. This flow case features a turbulent boundary layer that suddenly encounters a backward-facing step, which leads to flow separation at a fixed location. The entry section in the streamwise direction, measured from the inlet to the step, has a length of $10h$, where h is the step height. The section behind the expansion has a length of $20h$ in the streamwise direction. The expansion ratio is $ER = 1.2$, as the vertical height is $W_1 = 5h$ before the step and $W_2 = 6h$ post expansion. No-slip boundary conditions are imposed on the lower wall. Inlet boundary conditions is given by a boundary layer simulation. The Reynolds number, which is based on the step height h and the mean inlet freestream velocity U_0, is $Re = 5100$. DNS data by Le et al. [8] is available for this flow case.

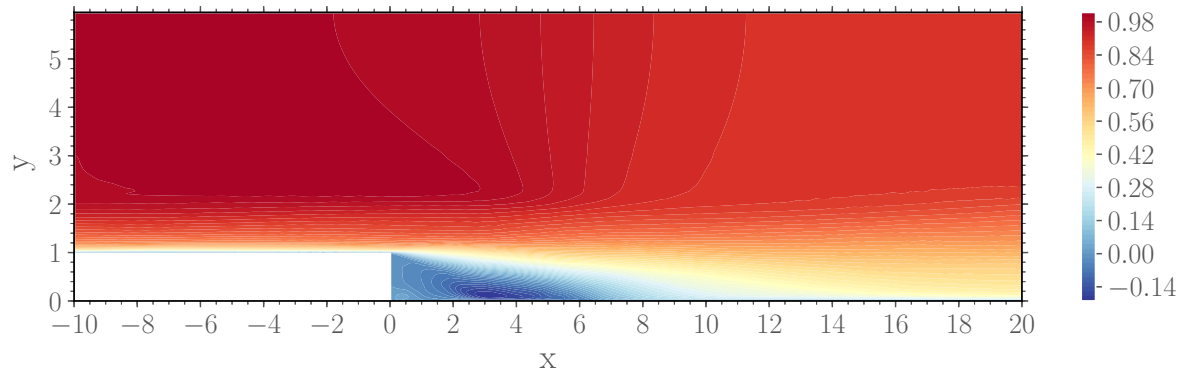


Figure 5.3: Contours of mean velocity magnitude $|U|$. $Re = 5100$. Data from a DNS by Le et al. [8].

The backward-facing step flow cases features separated flow in which two circulatory regions exist. A smaller circulatory region rotating counter-clockwise is located in the corner, followed by a larger region rotating clockwise. In addition, the separation location is fixed, which alltogether makes it an interesting test case. Results from the DNS study by Le et al. [8] show that the flow reattaches again around $x = 6.0$.

5.4. Curved Backward-Facing Step

Figure 5.4 shows the geometry of the curved backward-facing step flow case. The height of the duct upstream is $8.52h$, where h is the height of the step. The Reynolds number, which is based on the step height h and the inlet free-stream velocity U_{in}, is $Re = 13700$. No-slip boundary conditions are enforced on the upper- and lower walls. The inlet boundary conditions were taken from a turbulent channel flow simulation. This precursor simulation was in close agreement with DNS results [9], and thus, leads to a high-fidelity representation of the turbulent boundary layer as it approaches the curved backward-facing step. Bentaleb et al. [9] presented the results of a highly-resolved LES, which was used to investigate separation of the boundary layer from a curved backward-facing step.

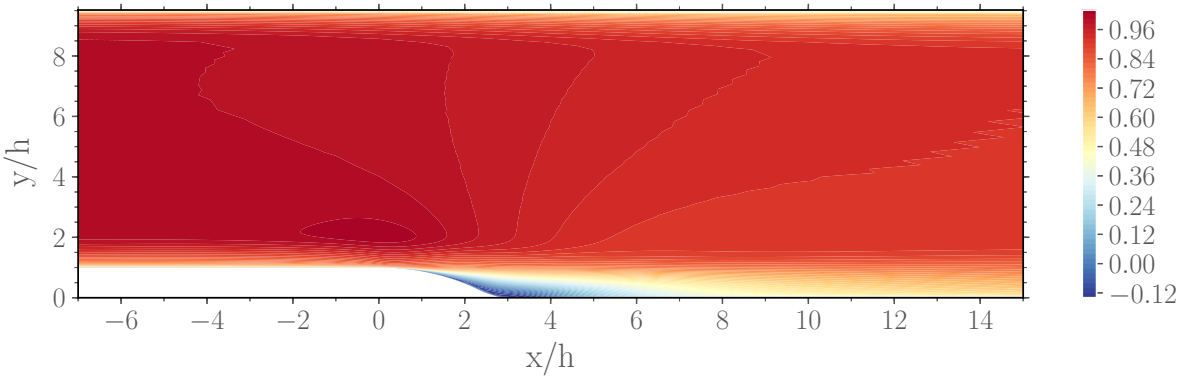


Figure 5.4: Contours of mean velocity magnitude $|U|$. $Re = 13700$. Data from a LES by Bentaleb et al. [9].

The study of Bentaleb et al. [9] focuses on the separation process and the corresponding properties in this region, making it an interesting test case to benchmark the performance of a RANS models in terms of predicting the onset of separation and the reattachment location. As RANS models often perform poorly in the prediction of separation from curved surfaces, this high-fidelity data cannot only be used to assess the performance, but also to identify the origins of defects in turbulence closure approximations [9]. The time-averaged results reveal that the flow separates at $x/h = 0.83$ and reattaches at $x/h = 4.36$.

6

Results

This chapter presents the results of the k-corrective-frozen-RANS approach and the SpaRTA method. The aim is to provide an answer to whether an additional transport equation leads to an improvement in performance with regard to predictive capability, generalizability and the ability to infer quantities of interest. In addition, the question to whether a full integrity basis shows advantages over a reduced integrity basis is answered.

First, the correct implementation of the elliptic blending lag parameter model in OpenFOAM is verified in Section 6.1. In Section 6.2, the ability of the frameworks to capture the model-form error is validated. The results of the SpaRTA approach are presented in Sections 6.3 and 6.4, in which the discovered models and the process of cross-validation is shown, respectively. The predictive capability and generalizability are assessed in Section 6.5. Finally, the usage of a reduced integrity basis versus a full integrity basis for the construction of the functionals of the corrective terms is discussed in Section 6.6.

6.1. Verification of Model Implementation

This section verifies the correct implementation in OpenFOAM of the elliptic blending lag parameter model. These results are compared to high-fidelity data such as DNS, LES or experiments. In addition, the OpenFOAM results from the various RANS models are compared to the results from Biswas et al. [21]. Section 6.1.1 presents the results for the periodic hill flow case. The verification process for the backward-facing step is shown in Section 6.1.2. Finally, Section 6.1.3 presents the verification results from the curved backward-facing step flow case.

6.1.1. Periodic Hill

The implementation of the elliptic blending $k-\omega$ lag parameter model in OpenFOAM is verified against the results of Biswas et al. [21]. Figure 6.1 shows a comparison of the skin friction coefficients for the periodic hill flow case, which is described in Section 5.1. The OpenFOAM results for the skin friction coefficient are in close agreement with the results from Biswas et al. [21]. Small discrepancies can be seen around the peaks at $x/h = 0.1$ and $x/h = 8.6$, as the OpenFOAM simulation predicts a slightly larger peak for the $k-\omega$ lag EB model. A similar observation can be done for the $k-\omega$ model, which also predicted a larger skin friction coefficient compared to the $k-\omega$ results of Biswas. A possible explanation for this difference could be that there is a small difference in the bulk velocity, which is used to normalize the wall shear stress τ_w.

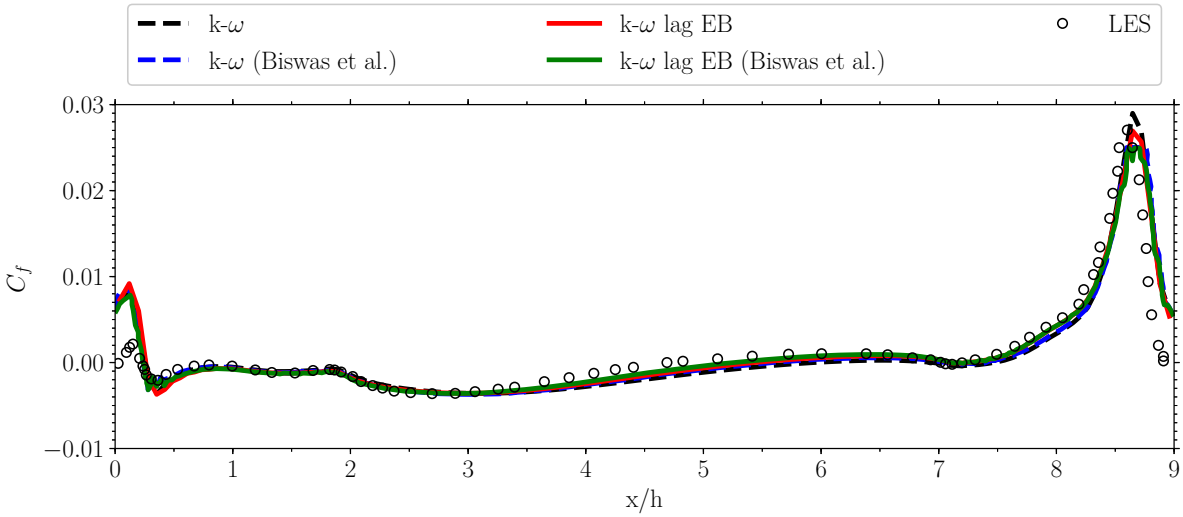


Figure 6.1: Comparison of skin friction coefficients.

The comparison of the velocity profiles for the periodic hill flow case is shown in Fig. 6.2. It can be observed that these profiles predicted by the various RANS models and high-fidelity data are in close agreement.

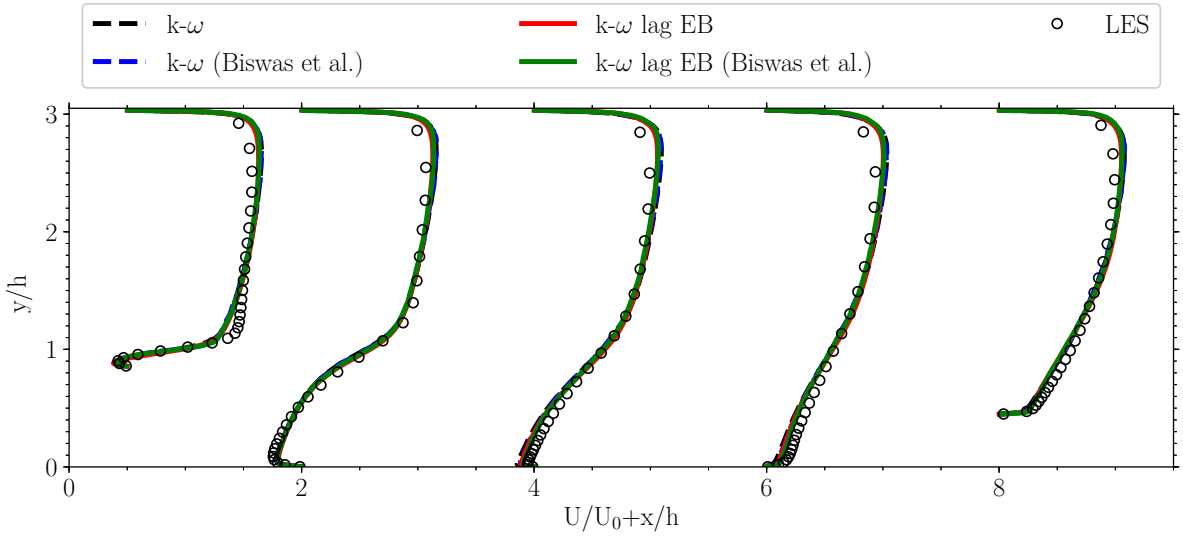


Figure 6.2: Comparison of velocity profiles.

One of the best ways to assess the performance of RANS models is to compare the eddy viscosity ν_t, but computing it from DNS/LES data can be ambiguous for non-equilibrium flows, which is why Biswas et al. [21] suggested to use shear stress profiles instead. Significant differences between the RANS models and the high-fidelity data are observed. Close to the hills, none of the RANS models predict a shear profile that is in close agreement with the high-fidelity data. Overall, the $k-\omega$ lag EB model proves to be the most accurate in terms of representing the high-fidelity data. Furthermore, the shear stress profiles predicted by the $k-\omega$ lag EB model and those by Biswas et al. [21] are in close agreement. A similar observation can be made for the $k-\omega$ results.

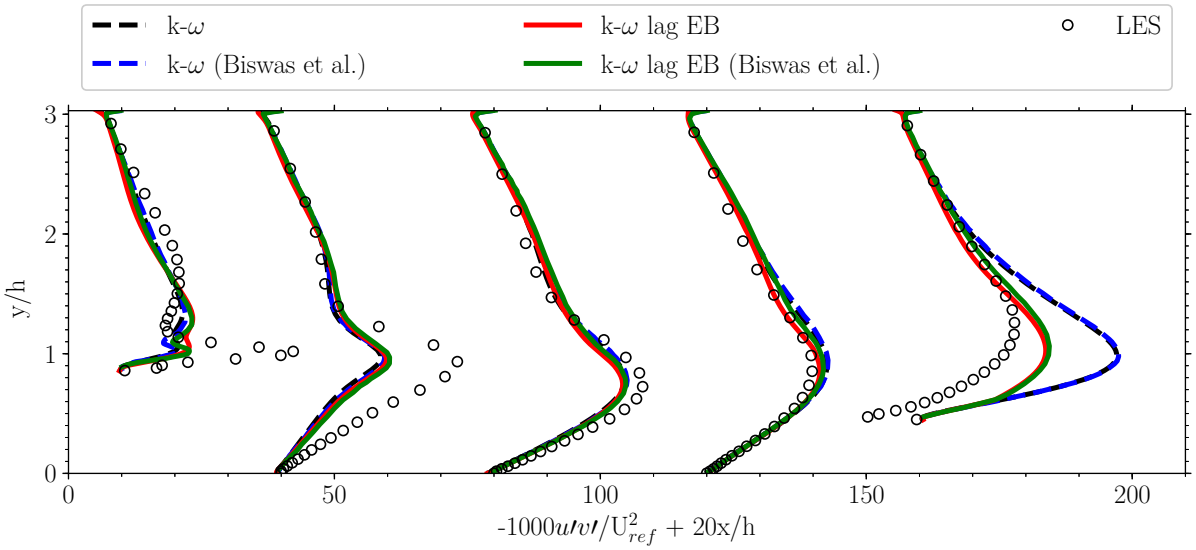


Figure 6.3: Comparison of shear stress profiles.

6.1.2. Backward-Facing Step

The second case that has been used to verify the OpenFOAM implementation of the model is the backward-facing step flow case. Please note that the flow case used for verification is different than the one described in Section 5.3. In the current flow case, the Reynolds number, which is based on the step height h, is approximately $Re = 36000$. Experimental data from Driver and Seegmiller [81] has been used to validate the various RANS models. Figure 6.4 gives a comparison of the skin friction coefficients. All RANS models accurately predict the location where the separated flow reattaches. Comparing the OpenFOAM results with the results by Biswas et al. [21], it can be seen that the results are similar behind the step up to a point of $x/h \approx 8$. After this point, the OpenFOAM results for the lag EB $k-\omega$ model follow more closely. However, the OpenFOAM results for the $k-\omega$ model also show some discrepancies in comparison with the results of Biswas et al. [21]. A possible explanation for this could be a small difference in the boundary conditions for k, ω, U or p, as similar discrepancies are observed for both $k-\omega$ models and both lag models. Unfortunately, it was not possible to check this further and (possibly) resolve these discrepancies, as the complete description of boundary conditions used in [21] is unavailable.

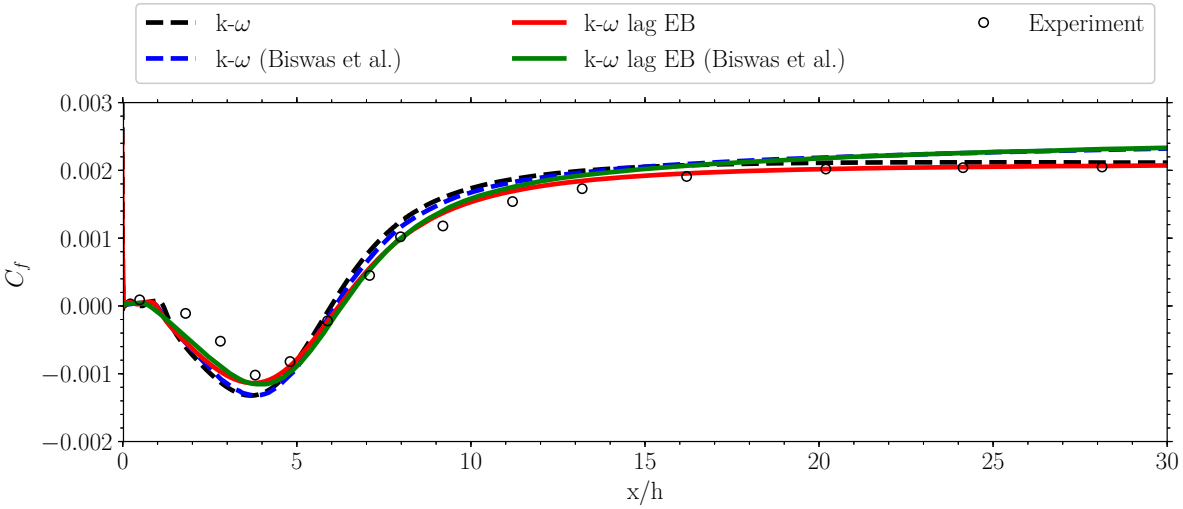


Figure 6.4: Comparison of skin friction coefficients.

The velocity profiles are compared in Fig. 6.5. It can be seen that the velocity profiles predicted by the RANS models recover more slowly than those from the experiments. The predictions by the lag EB $k-\omega$ model and those of Biswas et al. [21] correspond reasonably well. Discrepancies between the OpenFOAM implementation and the results by Biswas et al. [21] are observed around $y/h = 2$ for all streamwise profiles. However, this is true for both the $k-\omega$ model and the $k-\omega$ lag EB model.

The shear stress profiles are shown in Fig. 6.6 for the backward-facing step flow case. In comparison with the experimental data, the RANS models accurately predict the shear stress profiles. The shear stress profiles predicted by the $k-\omega$ lag EB model in OpenFOAM, are truly similar to those from the work of Biswas et al. [21].

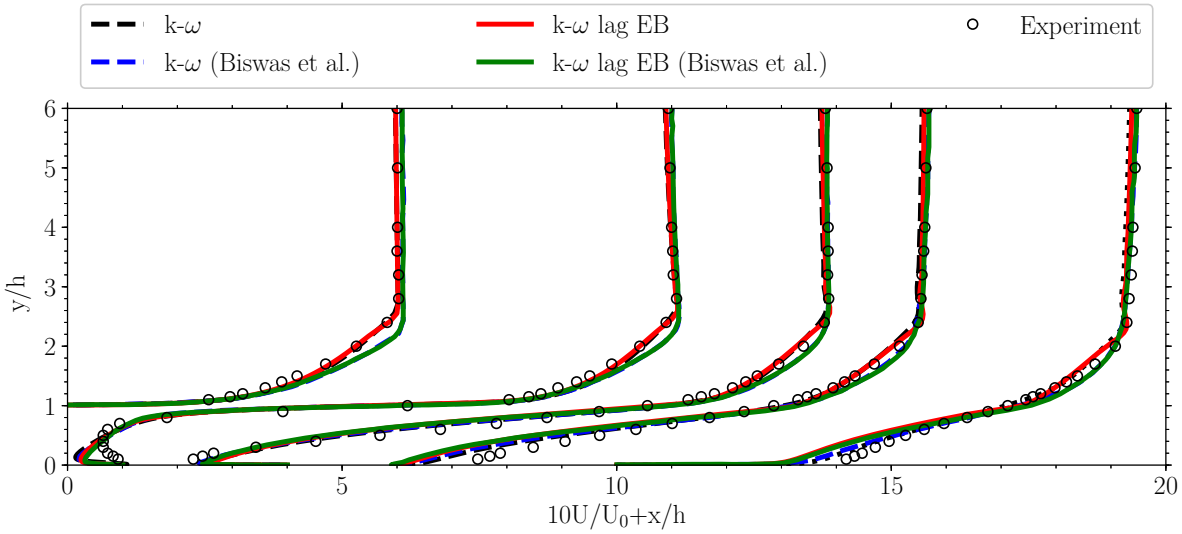


Figure 6.5: Comparison of velocity profiles.

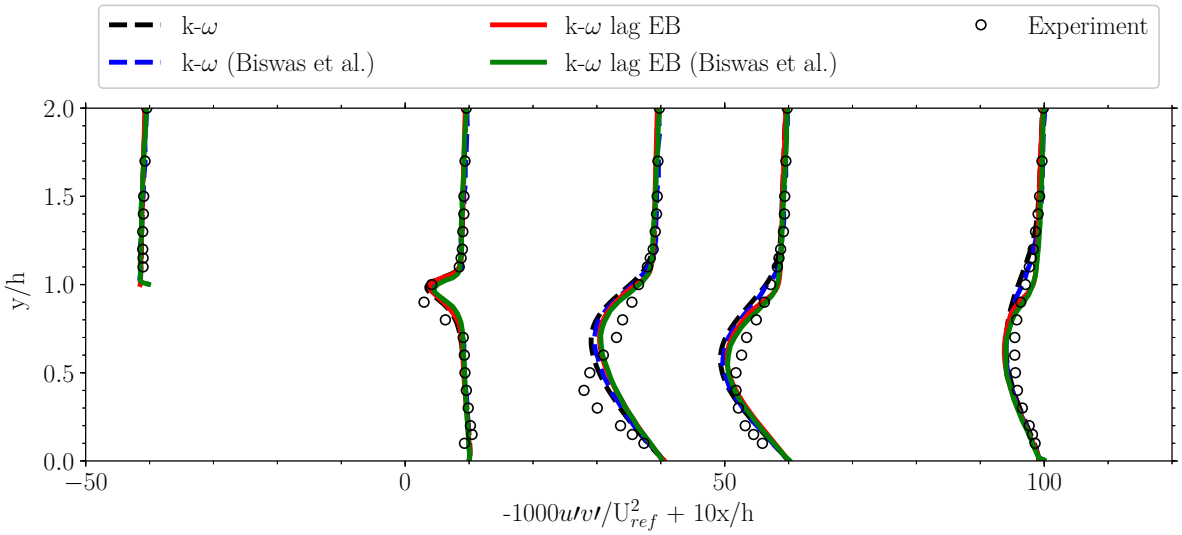


Figure 6.6: Comparison of shear stress profiles.

6.1.3. Curved Backward-Facing Step

The curved backward-facing step, as described in Section 5.4, is the final case that has been used to verify the implementation of the model. The high-fidelity data used for this comparison is a LES by Bentaleb et al. [9]. A comparison between predicted skin friction coefficients by the various RANS models and high-fidelity data is given in Fig. 6.7. Out of the three verification cases, the differences

between the OpenFOAM results of the lag EB $k-\omega$ model and that of Biswas et al. [21] are the most significant for the current flow case. However, this is also true for the other RANS models. Therefore, it is likely that it is a difference in the case setup in OpenFOAM, rather than a difference in the model implementation. A possible explanation is that there is a difference in the boundary conditions for k, ω, U or p, as similar discrepancies are observed for both $k-\omega$ models and both lag models.

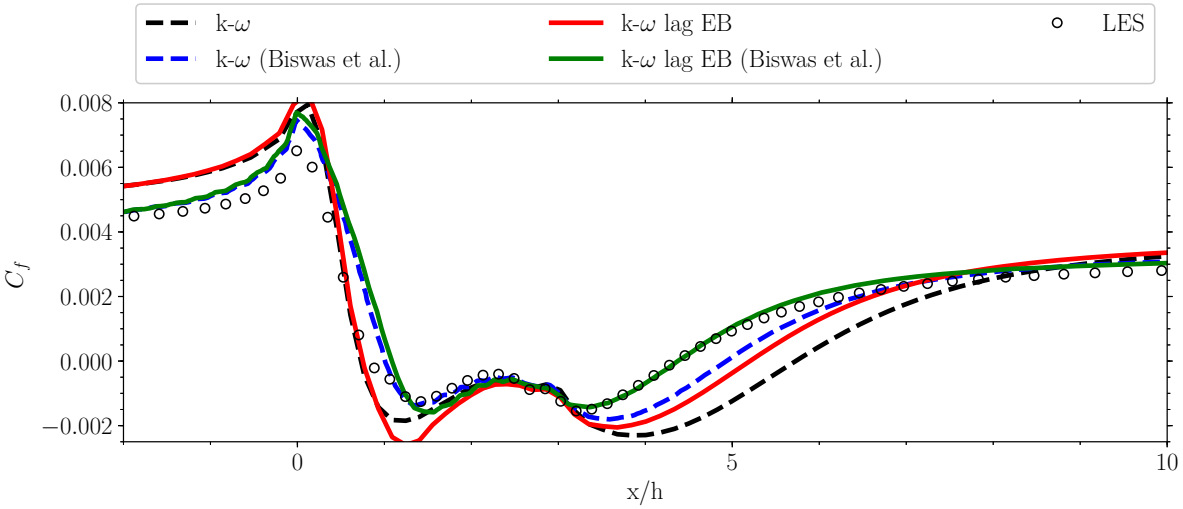


Figure 6.7: Comparison of skin friction coefficients.

Figure 6.8 provides a comparison of the velocity profiles for the curved backward-facing step. Qualitatively, the differences are less apparent than those in the skin friction results. The velocity profiles predicted by the lag EB $k-\omega$ model show close agreement with the LES data. The most significant difference between the predictions of the lag EB $k-\omega$ model and the results of Biswas et al. [21] can be found at a streamwise location of $x/h = 2$, as the velocity profile predicted by the OpenFOAM simulation is in closer agreement with the LES results. Results by both models in [21] predict a larger normalized mean velocity compared to the LES data for the streamwise profiles $x/h = 2$ and $x/h = 4$, whereas the results by the OpenFOAM implementation do not.

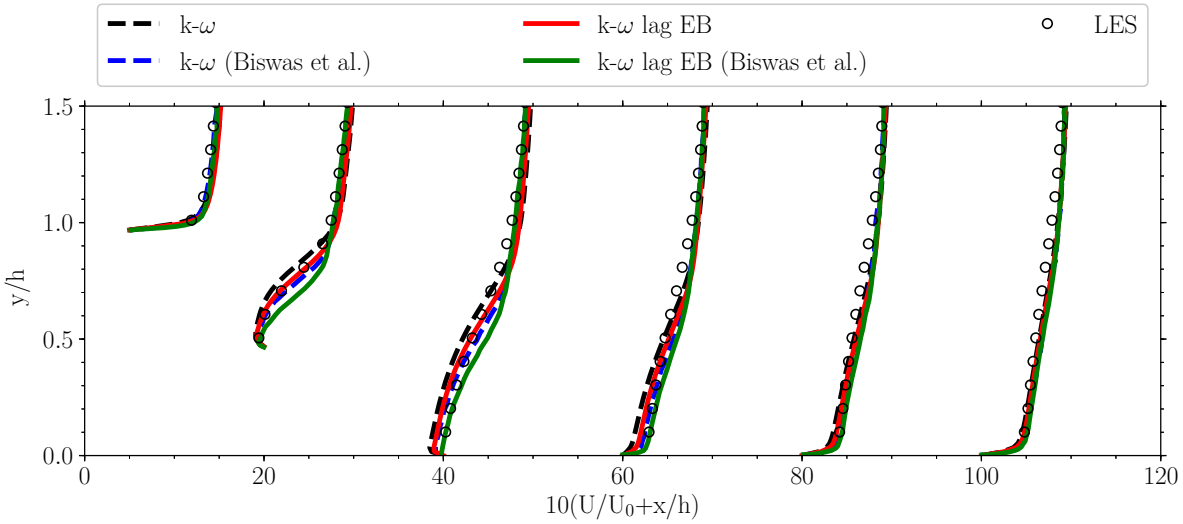


Figure 6.8: Comparison of velocity profiles.

The turbulence kinetic energy profiles are shown in Fig. 6.9. In the near-wall region ($y/h \leq 1.0$), the differences between the k profiles by the RANS models and the LES data is the most significant.

This difference is largest after the curved backward-facing step, but decreases as one progresses in the streamwise direction. Furthermore, among all the RANS models, the largest discrepancy with the LES data is shown by the lag EB $k-\omega$ model, which is true for both implementations. An exception is the OpenFOAM implementation of the $k-\omega$ lag EB model, which shows an improvement over the standard $k-\omega$ model for streamwise profiles downstream of $x/h = 6$.

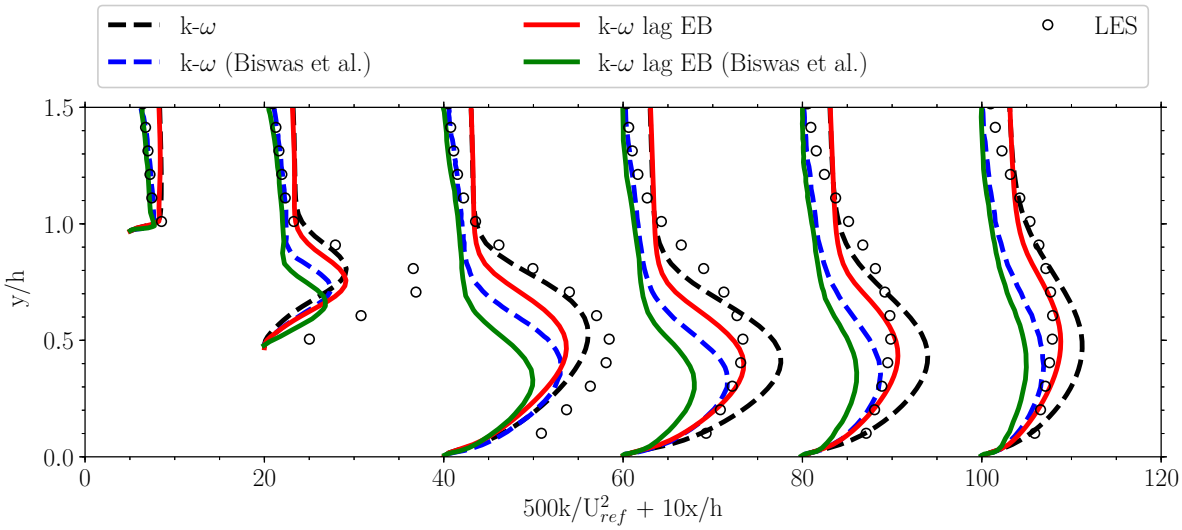


Figure 6.9: Comparison of turbulence kinetic energy profiles.

6.2. Validation of Model-Form Error Identification

When the process of frozen-RANS has converged, the additive terms b_{ij}^{Δ} and R are added as static fields in OpenFOAM, after which a simulation is performed for the flow cases in Chapter 5 using the same initial conditions. The ability of both frameworks to capture the model-form error is validated in this section. First, the results for the periodic hill flow case are shown in Section 6.2.1. Then, Section 6.2.2 presents the results for the converging-diverging channel. Finally, the results for the curved backward-facing step are given in Section 6.2.3.

6.2.1. Periodic Hills

The streamwise velocity profiles for the periodic hill flow case are shown in Fig. 6.10. Framework 1 is indicated by b_{ij}^{Δ} and R (k-ω) and Framework 2 is denoted by b_{ij}^{Δ} and R (lag EB). It can be seen that the mean flow velocity is essentially reproduced by both approaches.

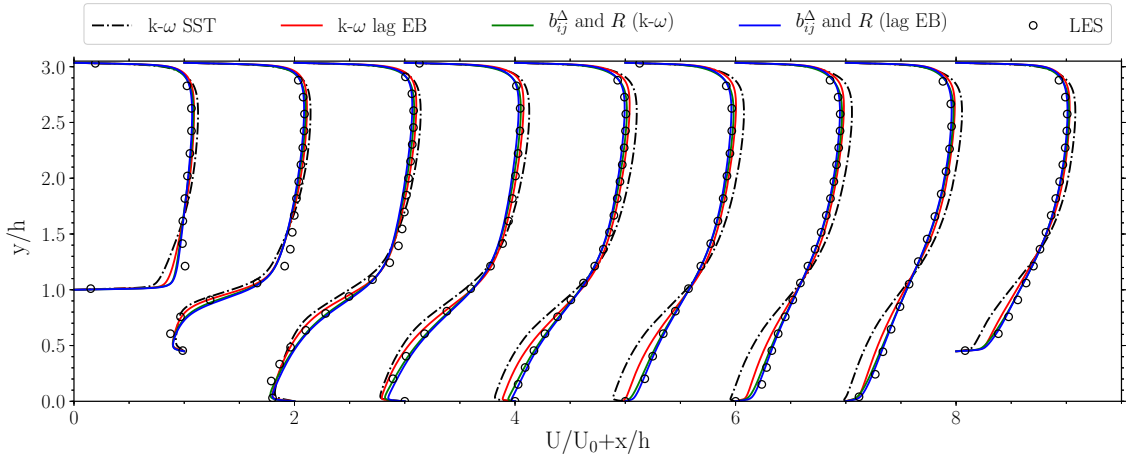


Figure 6.10: Predictions of streamwise velocity profiles.

The turbulence kinetic energy profiles are shown in Fig. 6.11. For the profiles at $x/h = 1$ and $x = h2$, there is a small discrepancy between the reproduced results and the LES results, which is located downstream of the point where flow separation occurs. The k-ω SST and k-ω lag EB model are unable to capture the large scale motion behind the first hill crest, which is inherent to RANS turbulence modeling and therefore not suprising. In general, the turbulence kinetic energy profiles are well reproduced by both approaches.

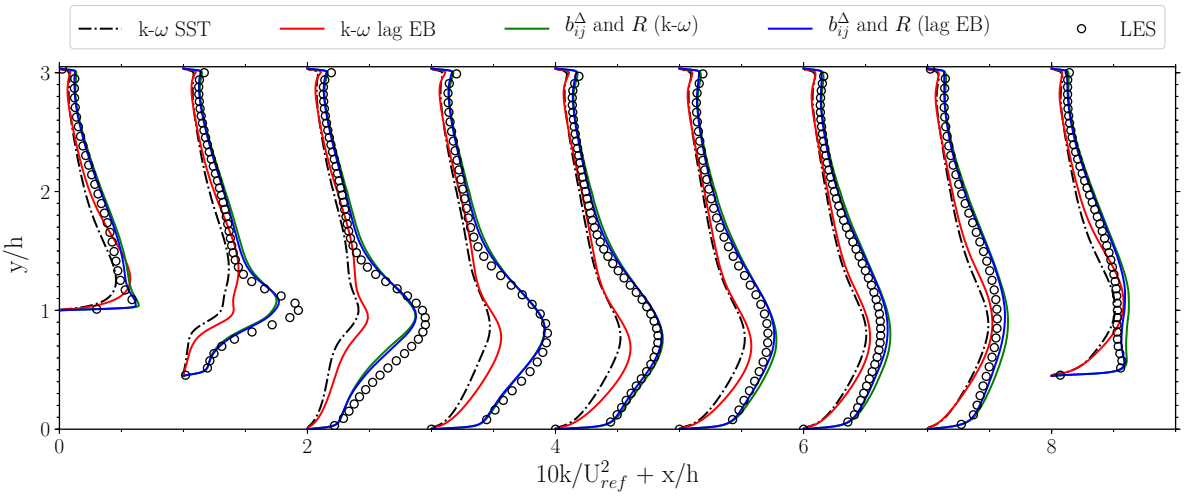


Figure 6.11: Predictions of streamwise turbulence kinetic energy profiles.

The shear velocity profiles are presented in Fig. 6.12. Near the hills, the discrepancy with the LES data is the most significant, although improvements are shown compared to the predictions by the k-ω SST and k-ω lag EB models.

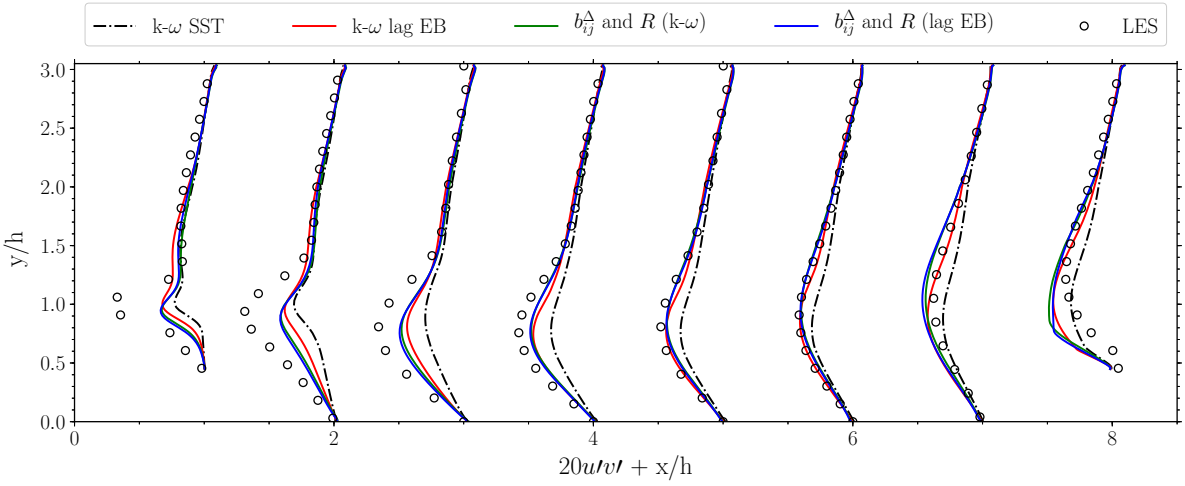


Figure 6.12: Predictions of streamwise shear velocity profiles.

The skin friction coefficient results are shown in Fig. 6.13. The spike near $x/h = 0$ is overestimated by all models, but this overestimation is most significant for the predictions of framework 1 and framework 2. From $x/h = 3$ onwards, the predictions by both frameworks show an improvement over the other models. Both frameworks accurately predict the reattachment location. The peak between $x/h = 8$ and $x/h = 9$ is slightly overestimated by both frameworks, whereas the other models underpredict this peak.

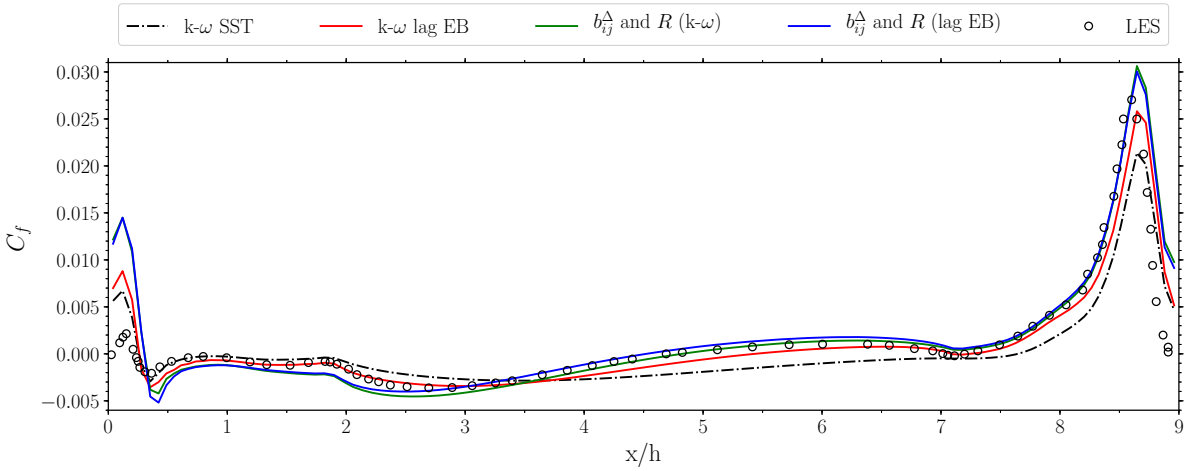


Figure 6.13: Predictions of skin friction coefficients.

6.2.2. Converging-Diverging Channel

This subsection presents the resuls for the converging-diverging channel. Figure 6.14 shows the predicted streamwise velocity profiles. It can be seen that all the predictions are similar to the DNS data, with the exception of the predictions by the k-ω SST model. On the lee-side and downstream of the bump, the k-ω SST model predicts a too slow recovery.

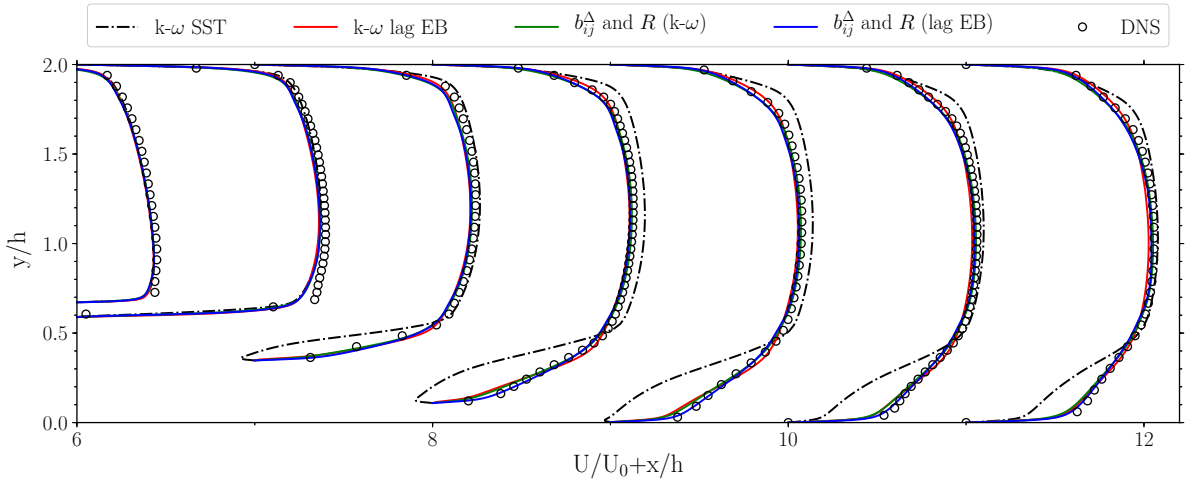


Figure 6.14: Predictions of streamwise velocity profiles.

The predictions of the streamwise turbulence kinetic energy profiles are shown in Fig. 6.15. Both approaches predicted accurate profiles compared to the DNS data. The largest discrepancy is observed at $x/h = 7$, located downstream of the small separation bubble, in which the turbulence kinetic energy is underestimated by both frameworks compared to the DNS data. However, an improvement is shown over the other models. Both approaches seem to have difficulty in fully recovering the DNS profile downstream of the small separation bubble on the lee-side of the bump, which is in line with the observation for the periodic hill flow case.

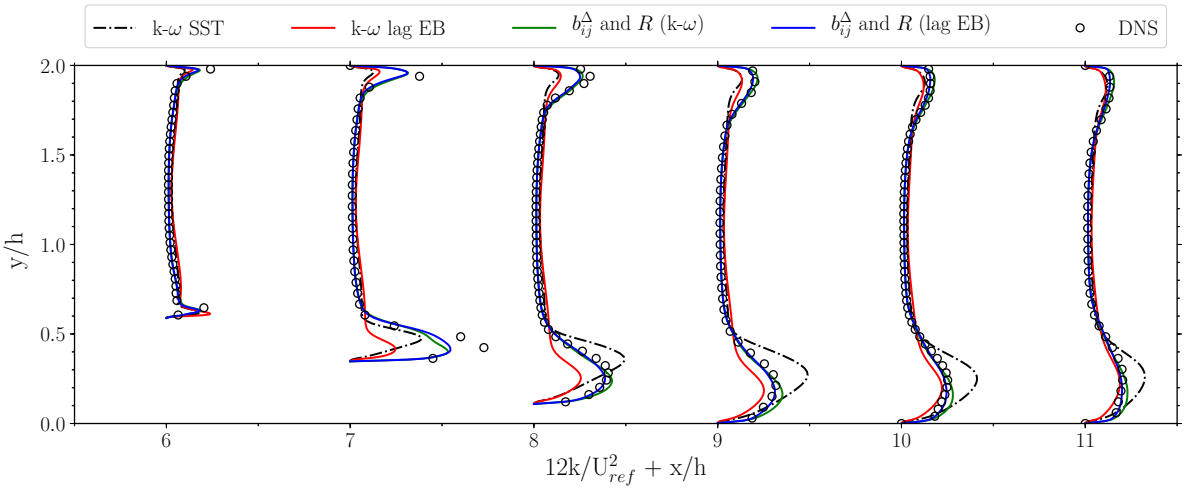


Figure 6.15: Predictions of streamwise turbulence kinetic energy profiles.

Figure 6.16 presents the streamwise shear velocity profiles. In general, the predictions follow the DNS profiles closely, with the exception of the predictions downstream of the separation bubble on the lee-side of the bump. Both frameworks show a minor improvement over the other models.

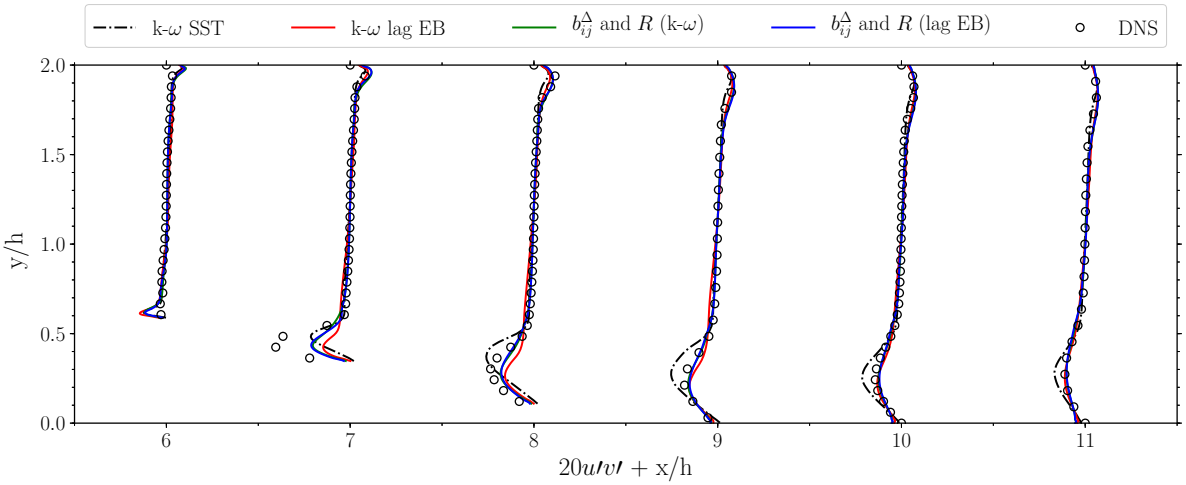


Figure 6.16: Predictions of streamwise shear velocity profiles.

The results for the skin friction coefficient are shown in Fig. 6.17. Both frameworks fail to predict the small separation bubble, although it seems that both frameworks captured the correct form with a wrong magnitude between $x/h = 6$ and $x/h = 7$, which is a remarkable observation. Out of all models, the k-ω SST model is the only one that predicts separation. However, the length of the separation region is significantly overpredicted.

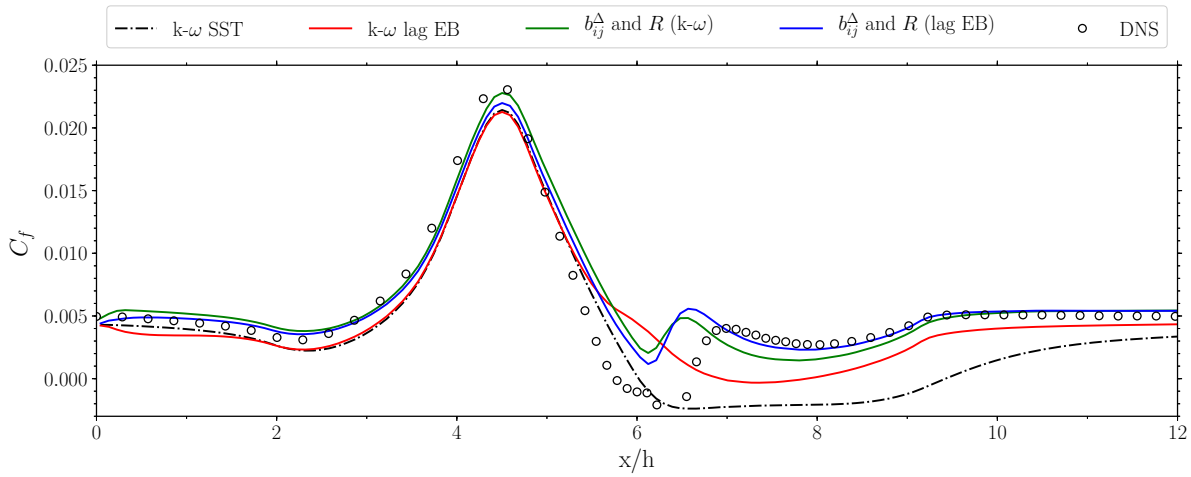


Figure 6.17: Predictions of skin friction coefficients.

6.2.3. Curved Backward-Facing Step

This subsection presents the validation results for the curved backward-facing step flow case. The predicted streamwise velocity profiles are given in Fig. 6.18. From all the flow cases that are used for validation, the mean velocity predictions for the curved backward-facing step showed the most discrepancies with the high-fidelity data. The velocity profiles are systematically underestimated above $y/h \approx 1$. Both frameworks predict an accurate recovery and thereby show an improvement compared to the other models, which fail to do so as the predicted recovery is too slow.

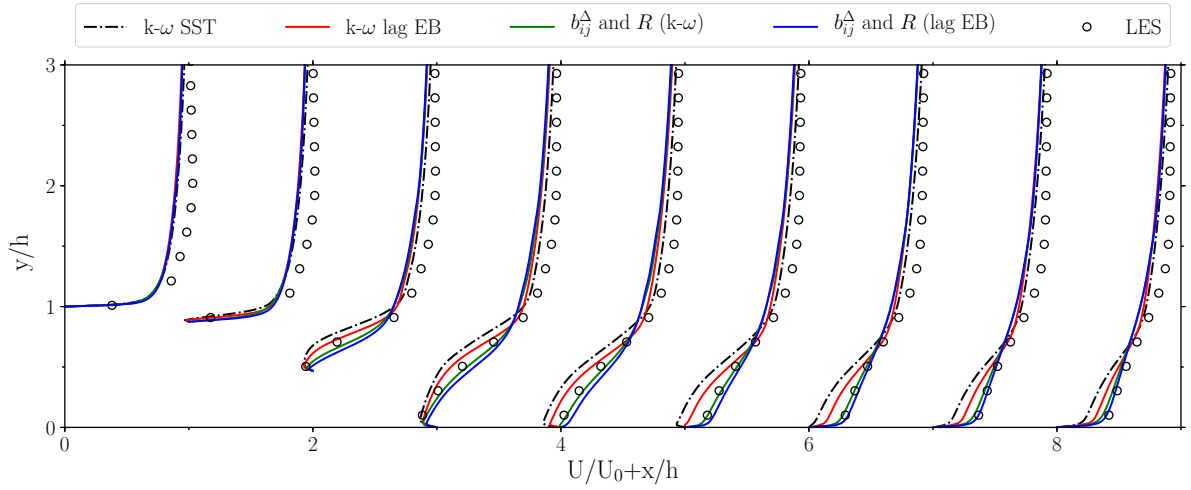


Figure 6.18: Predictions of streamwise velocity profiles.

The turbulence kinetic energy profiles are shown in Fig. 6.19. The largest discrepancies are observed between $x/h = 2H$ and $x/h = 4$, as both frameworks underestimate the turbulence kinetic energy compared to the LES data. However, a significant improvement is observed in comparison with the other models.

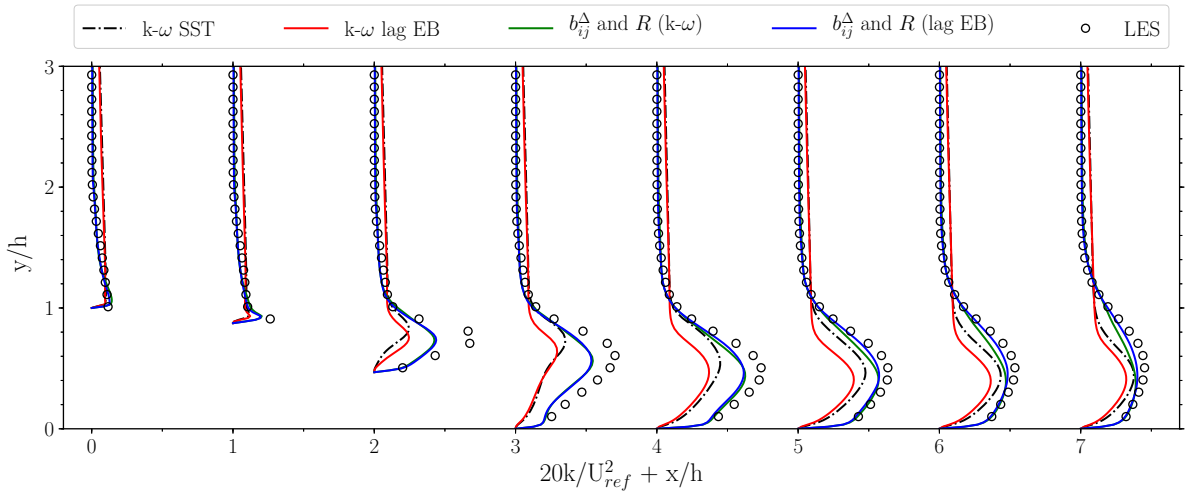


Figure 6.19: Predictions of streamwise turbulence kinetic energy profiles.

Figure 6.20 present the predicted streamwise shear velocity profiles. Both frameworks show a minor improvement over the other models, but the discrepancy with the LES data is still significant, especially between $x/h = 2$ and $x/h = 4$.

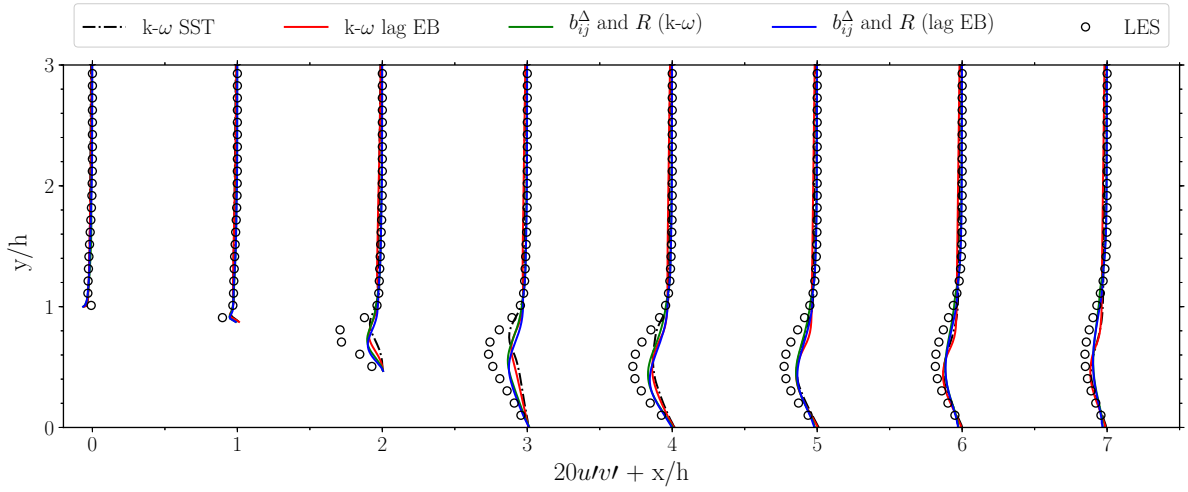


Figure 6.20: Predictions of streamwise shear velocity profiles.

The predictions for the skin friction coefficient are shown in Fig. 6.21. It can be seen that both approaches predict the separation location too far downstream, whereas the other models predict this location accurately. However, the point of reattachment is predicted more accurately by b_{ij}^{Δ} and R (k-ω). Furthermore, the initial peak is represented accurately by both frameworks in comparison with the LES data.

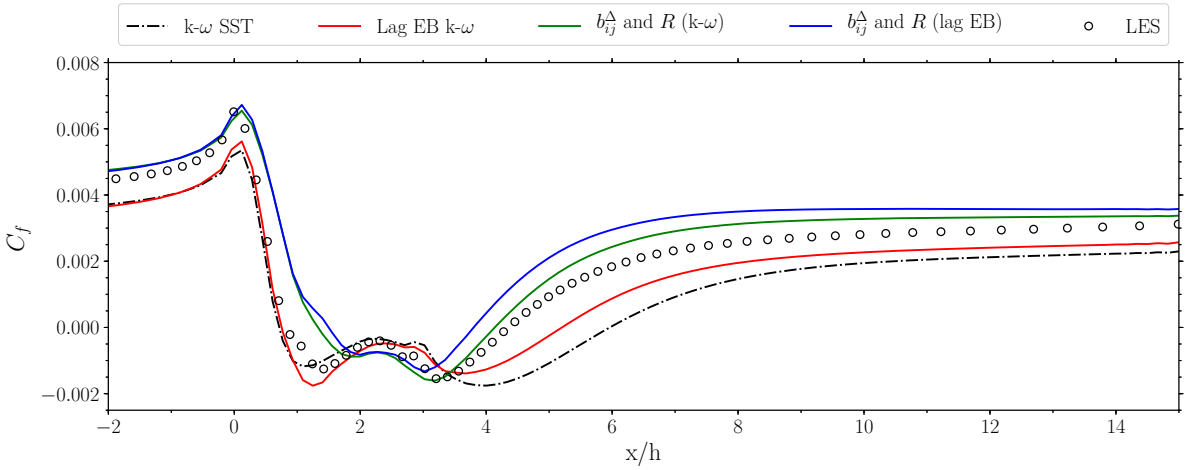


Figure 6.21: Predictions of skin friction coefficients.

To conclude this section, both frameworks are able to capture the model-form error. Furthermore, these frameworks give essentially the same results in terms of predictive capability and the ability to infer quantities of interest. Any improvements of the data-driven lag parameter models, that have an additional transport equation, has to come from their ability to generalize, which will be the topic of discussion in Sections 6.4 and 6.5.

6.3. Discovery of Models

Following the approach of Schmelzer et al. [5], the goal of this step in the SpaRTA method is to find a set of different models, i.e., that vary in complexity and accuracy. The reason is that such a set is better suited for the following step, the cross-validation. Instead of selecting the best models based on only the training data, the ensemble of models will be tested on unseen test cases and the final selection will be based on the lowest mean-squared error on the velocity field $\varepsilon(U)$. For the model inference step, several regularisation parameters ($\lambda_r = [0,\ 0.0001,\ 0.01]$) have been tested. Selecting the final regularisation parameter comes down to a trade-off between the numerical stability of the model, predictive capability and its accuracy. In this study, the Ridge regression is performed with $\lambda_r = 0.01$ as regularisation parameter, which is mainly chosen due to the resulting numerical stability of the models. Selecting a lower regularisation parameter leads to increased coefficients, which in turn led to unstable models and possibly overfitted models. For each flow case, a representative subset is hand-selected and used for the cross-validation step. Five models for b_{ij}^{Δ} and five models for $\hat{\boldsymbol{R}}$ are selected. These models are visualized in Fig. 6.30 for the various flow cases. First, Section 6.3.1 presents the discovered and selected models using the baseline approach, which is referred to as framework 1. Then, Section 6.3.2 shows the discovered and selected models for b_{ij}^{Δ} and $\hat{\boldsymbol{R}}$ found using the approach with the lag parameter, referred to as framework 2.

6.3.1. Framework 1 - Standard k-ω Model

The SpaRTA approach that is described in Section 4.5 has been used on the flow cases mentioned in Chapter 5 and a visualization of all the discovered models for b_{ij}^{Δ} for the periodic hill flow case is given in Fig. 6.22. As can be seen, 193 models have been discovered. The matrix shows the model-structure and the mean-squared error of the discovered models for b_{ij}^{Δ}. It indicates the active (coloured) candidate functions (x-axis) for each model M_i with index i (y-axis). Furthermore, the values of the coefficients are indicated by the colour.

The discovered models for b_{ij}^{R} are visualized in Fig. 6.23. As can be seen, 39 different models have been discovered for b_{ij}^{R} for the periodic hill flow case. Similar to Schmelzer et al. [5], $T_{ij}^{(1)}$ and $T_{ij}^{(1)}I_1$ have been identified as relevant candidates. Although $T_{ij}^{(1)}I_2$ is a relevant candidate, it is not the third most relevant candidate as in the work of [5]. Instead, $T_{ij}^{(6)}$ seems to be a more relevant term and is also seen in $T_{ij}^{(6)}I_2$.

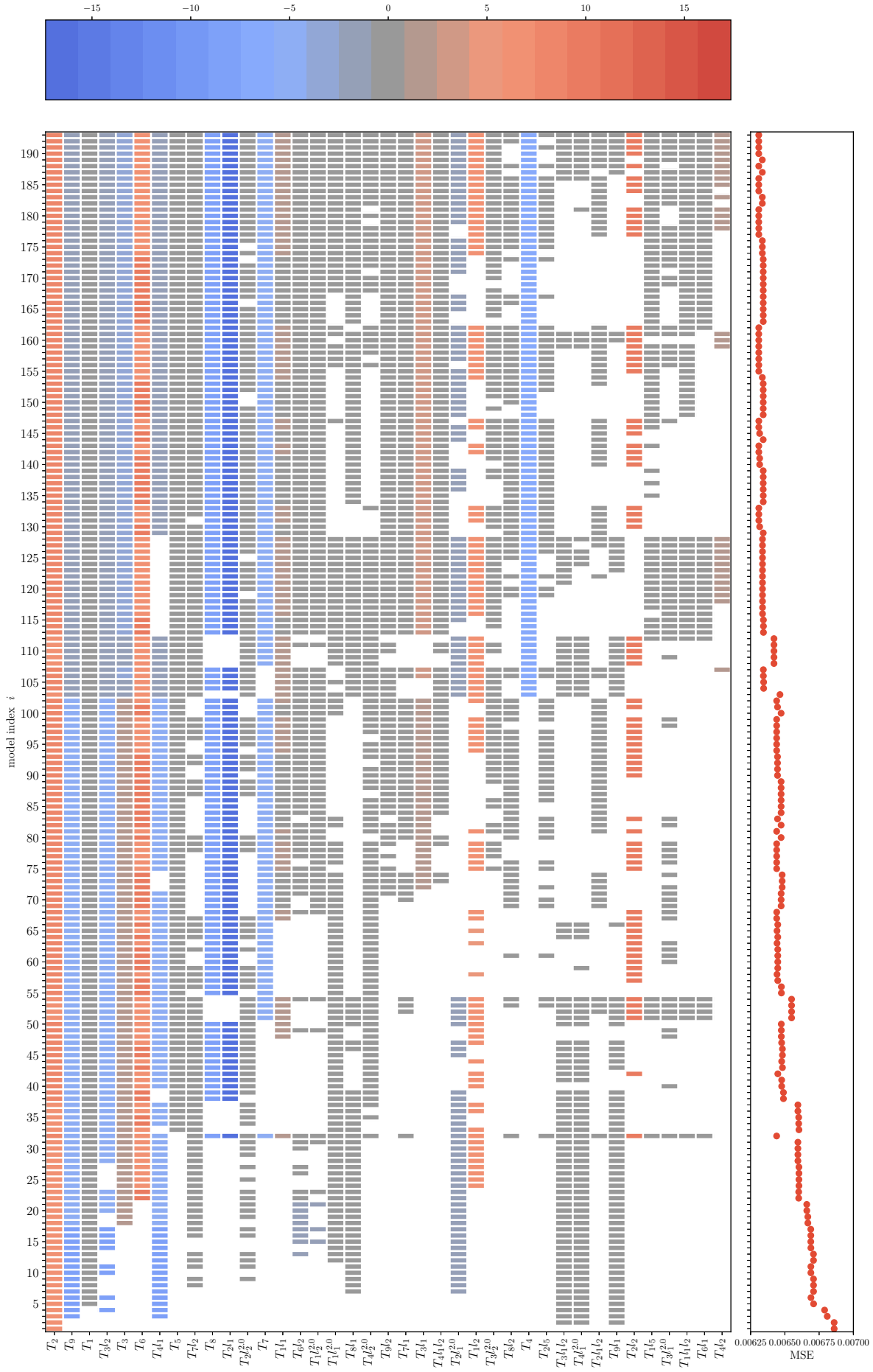


Figure 6.22: Models for b_{ij}^{Δ}

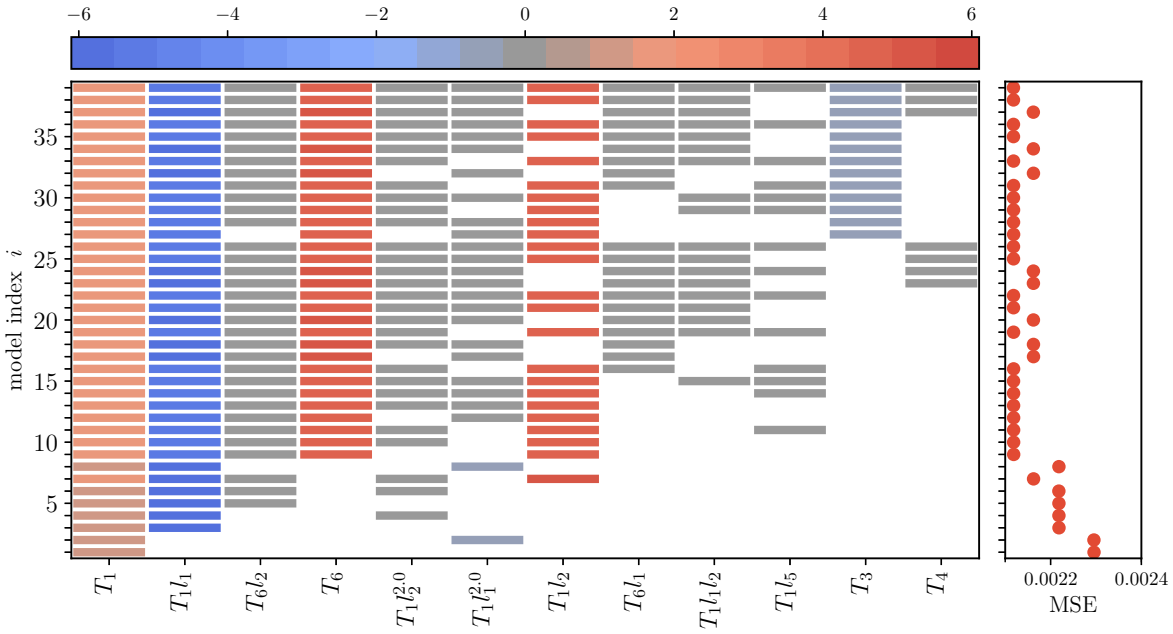
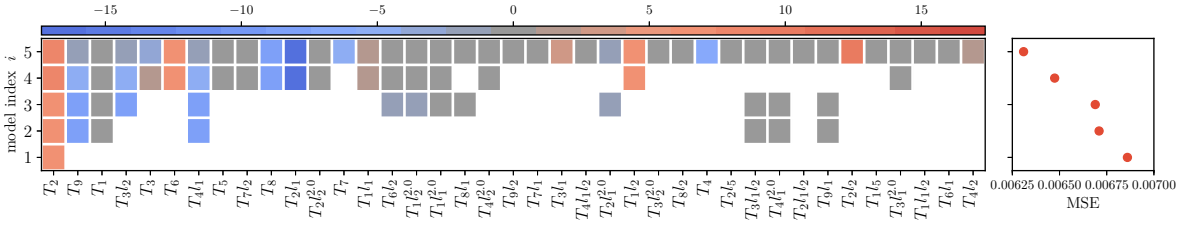


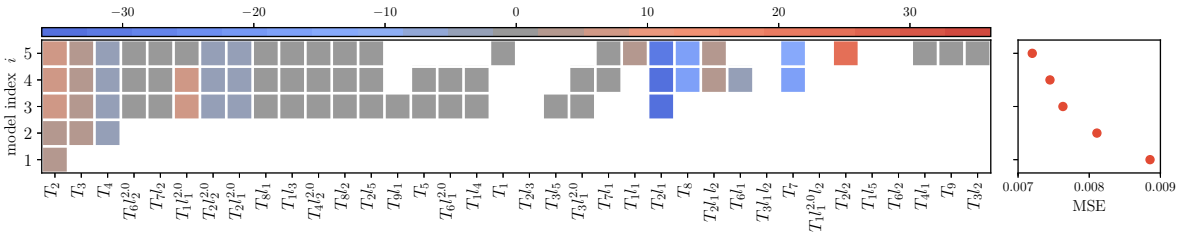
Figure 6.23: Models for R

The discovered models for both b_{ij}^{Δ} and b_{ij}^{R} for the converging-diverging channel and the curved backward-facing step can be found in Appendix D. It can be seen that 72 models have been discovered for b_{ij}^{Δ} for the converging-diverging flow case. In addition, 37 different models have been discovered for b_{ij}^{R}. For the curved backward-facing step flow case, 107 models have been discovered for b_{ij}^{Δ} and 19 models for b_{ij}^{R}. These discovered models for b_{ij}^{R} are significantly less than those of the periodic hill and converging-diverging channel flow cases.

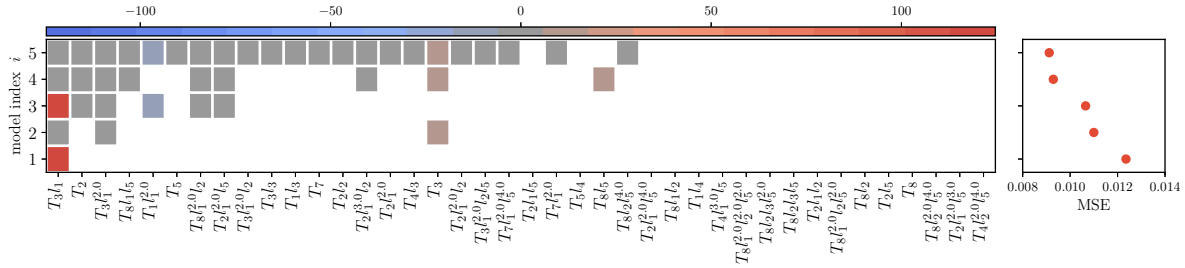
The selected models for b_{ij}^{Δ} for the various flow cases are visualized in Fig. 6.24. These models will be used for cross-validation, which is the next step.



(a) Periodic hill.



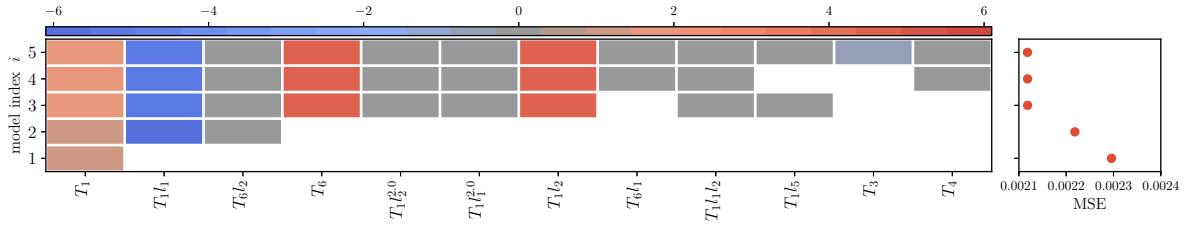
(b) Converging-diverging channel.



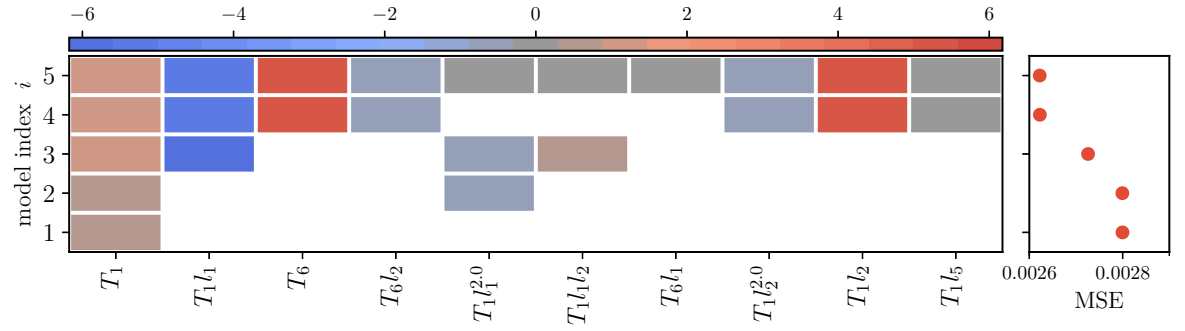
(c) Curved backward-facing step.

Figure 6.24: Model-structure and mean-squared error of selected models for b_{ij}^{Δ} for the various flow cases. The matrix indicates the active (coloured) candidate functions (x-axis) for each model M_i with index i (y-axis). Value of the coefficients is indicated by the colour.

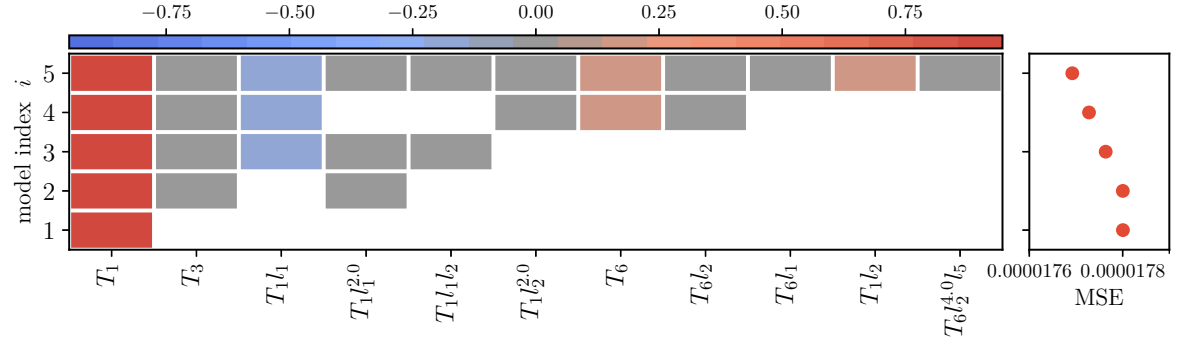
A visualization of the selected models for b_{ij}^{R} for the three flow cases is given in Fig. 6.25.



(a) Periodic hill.



(b) Converging-diverging channel.

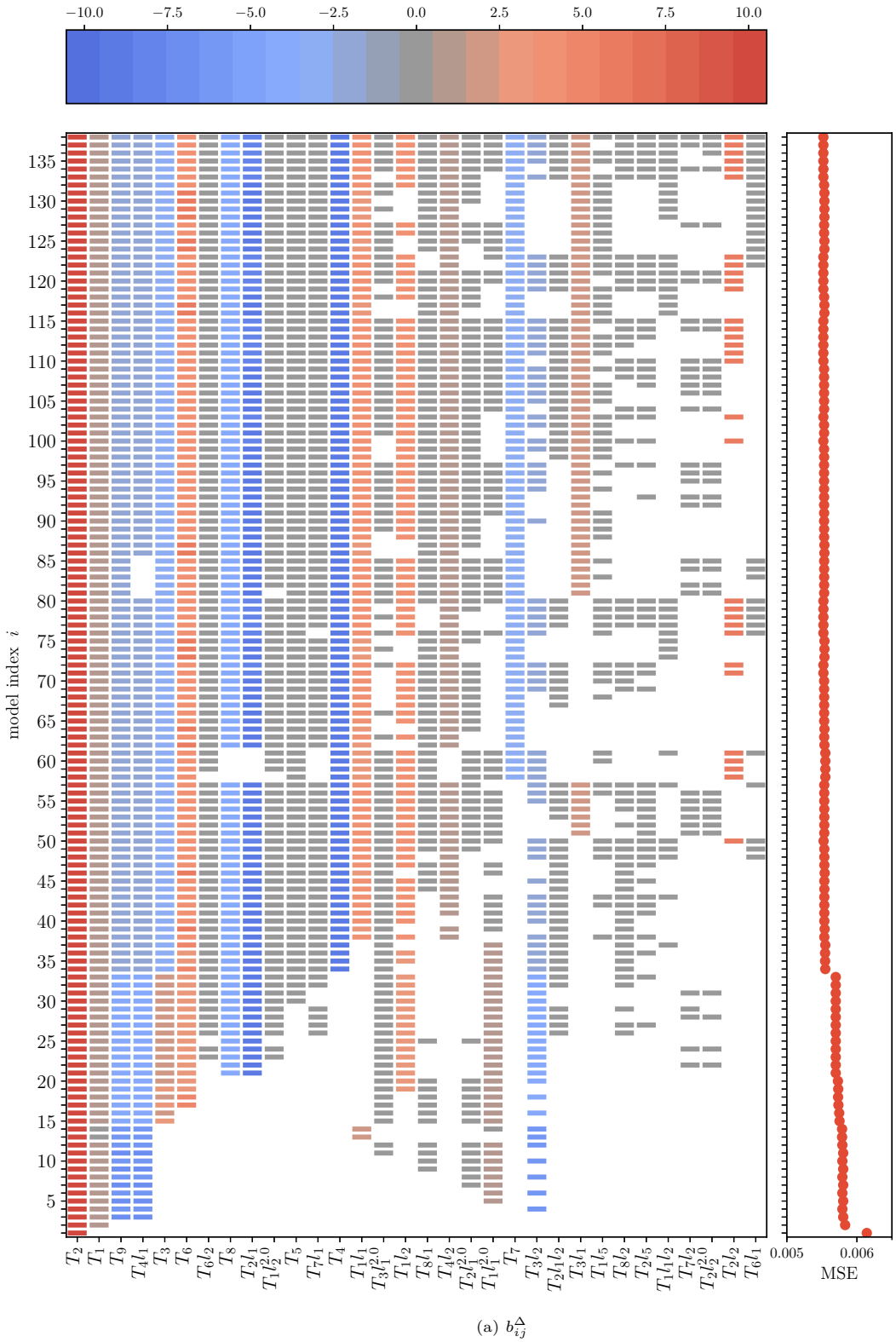


(c) Curved backward-facing step.

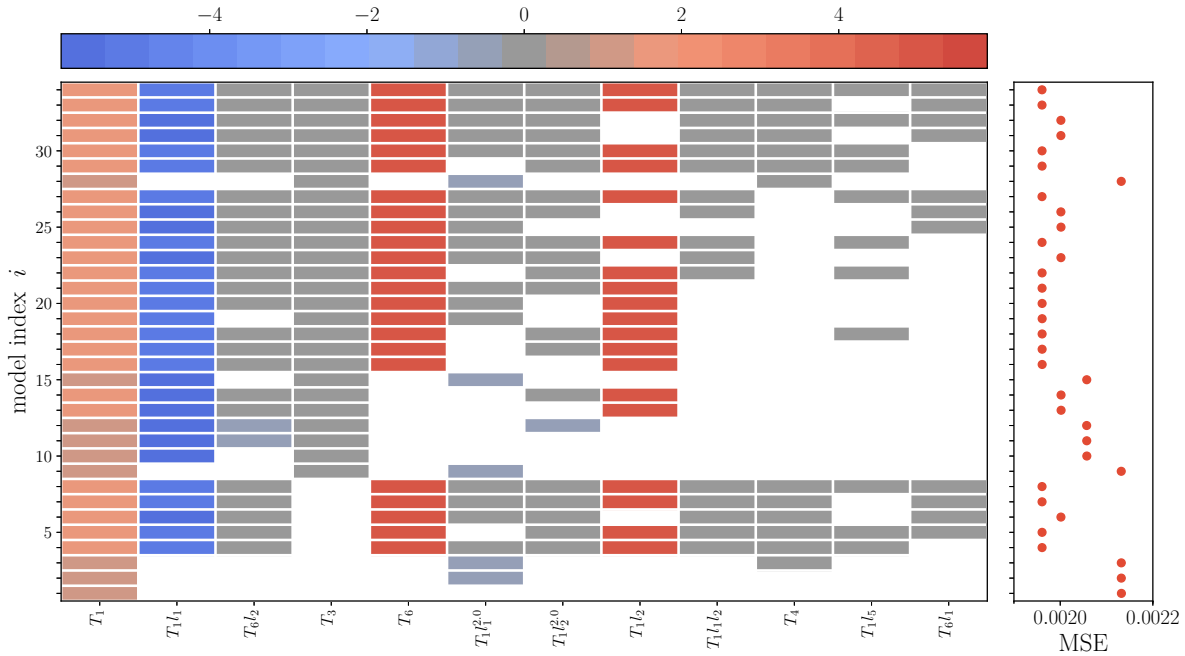
Figure 6.25: Model-structure and mean-squared error of identified models for $\hat{\boldsymbol{R}}$ for the various flow cases. The matrix indicates the active (coloured) candidate functions (x-axis) for each model M_i with index i (y-axis). Value of the coefficients is indicated by the colour.

6.3.2. Framework 2 - Elliptic Blending Lag Parameter Model

The SpaRTA approach applied to the periodic hill flow case led to the models shown in Figs. 6.26a and 6.26b for b_{ij}^{Δ} and b_{ij}^{R}, respectively. As can be seen, 138 distinct models were identified for b_{ij}^{Δ} and 34 distinct models for b_{ij}^{R}.



(a) b_{ij}^{Δ}

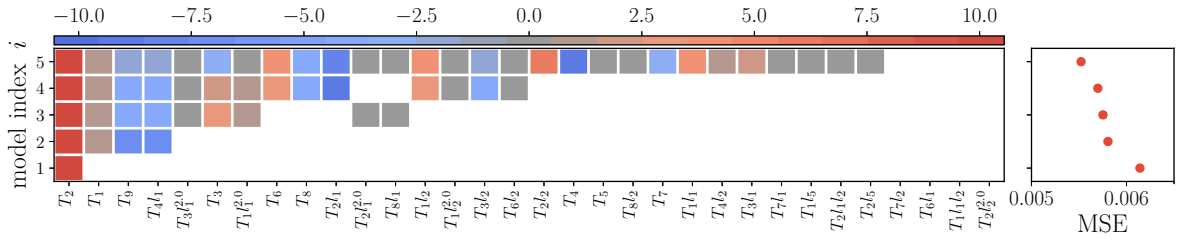


(b) b_{ij}^R

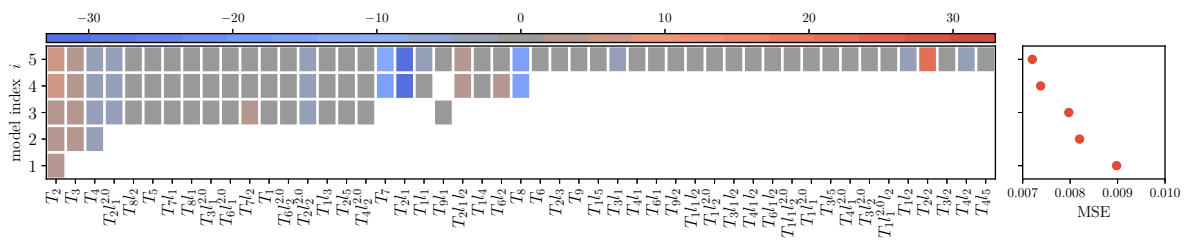
Figure 6.26: Model-structure and mean-squared error of identified models based on training data for the periodic hill flow case. The matrix indicates the active (coloured) candidate functions (x-axis) for each model M_i with index i (y-axis). Value of the coefficients is indicated by the colour.

In a similar fashion, the models that were obtained by applying the SpaRTA approach to the converging diverging channel flow case are shown in Fig. D.5. As can be seen, 72 models for b_{ij}^{Δ} and 27 models for b_{ij}^R have been discovered. The discovered models for b_{ij}^{Δ} and b_{ij}^R for the curved backward-facing step are shown in Figs. D.7 and D.8, respectively. In total, 133 models have been discovered for b_{ij}^{Δ} and 33 different models for b_{ij}^R.

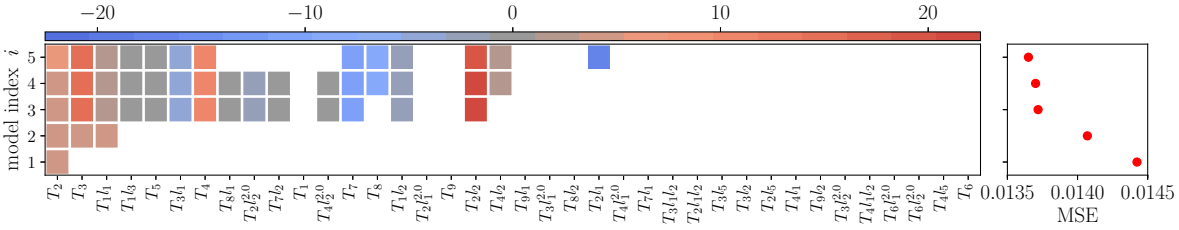
The selected models for b_{ij}^{Δ} that have been trained on the periodic hill, converging diverging channel and curved backward-facing step training cases are shown in Fig. 6.27.



(a) Periodic hill



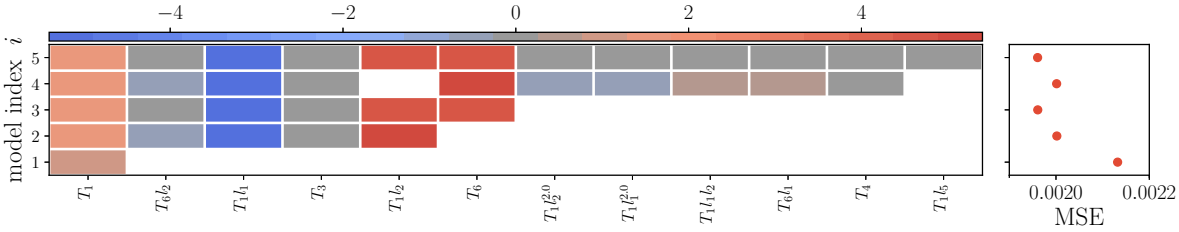
(b) Converging diverging channel



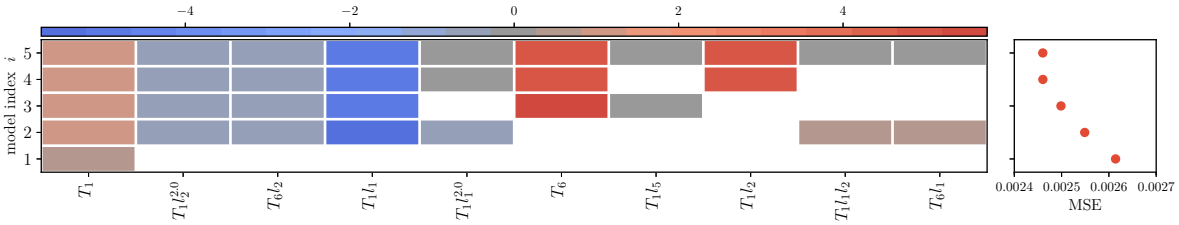
(c) Curved backward-facing step

Figure 6.27: Selected models for b_{ij}^{Δ}.

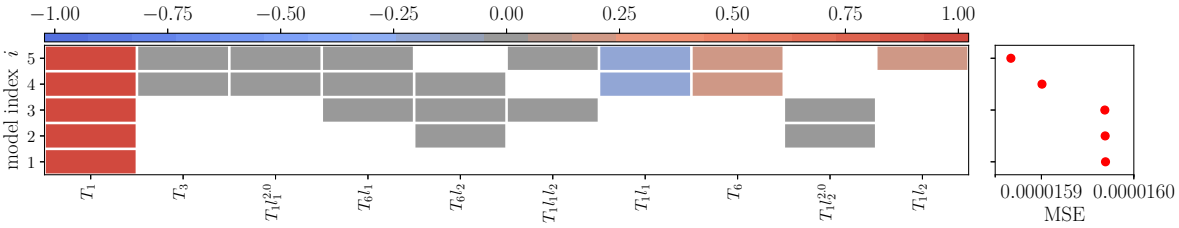
The selected models for b_{ij}^{R} are presented in Fig. 6.28.



(a) Periodic hill



(b) Converging diverging channel



(c) Curved backward-facing step

Figure 6.28: Selected models for b_{ij}^{R}.

The performance of the selected models of both frameworks in terms of predictive capability and generalizability will be tested and ranked using cross-validation, which will be elaborated upon in the next section.

6.4. Cross-Validation

In order to test the performance of the selected models on unseen data, cross-validation is used [82]. This way, it is possible to select the best predictive models. Combinations of models for b_{ij}^{Δ} and b_{ij}^{R} have been tested, as well as individual models for b_{ij}^{Δ} and b_{ij}^{R}, since the corrections are regressed individually. Thus, a model with only a correction for b_{ij}^{Δ} and no correction for b_{ij}^{R} can be implemented in a CFD solver, and vice-versa. For each flow case, five models for b_{ij}^{Δ} and five for b_{ij}^{R} are used, leading to a total of 35 distinct models per training case. Thus, in total, 105 distinct models have been tested and a total of 315 simulations have been conducted.

6.4.1. Framework 1 - Standard k-ω Model

For all individual and combination of models, Fig. 6.29 shows the mean squared error on the velocity field $\epsilon(U)$ of each model, normalized by the mean squared error on the velocity field $\epsilon(U_0)$ of the baseline $k-\omega$ model.

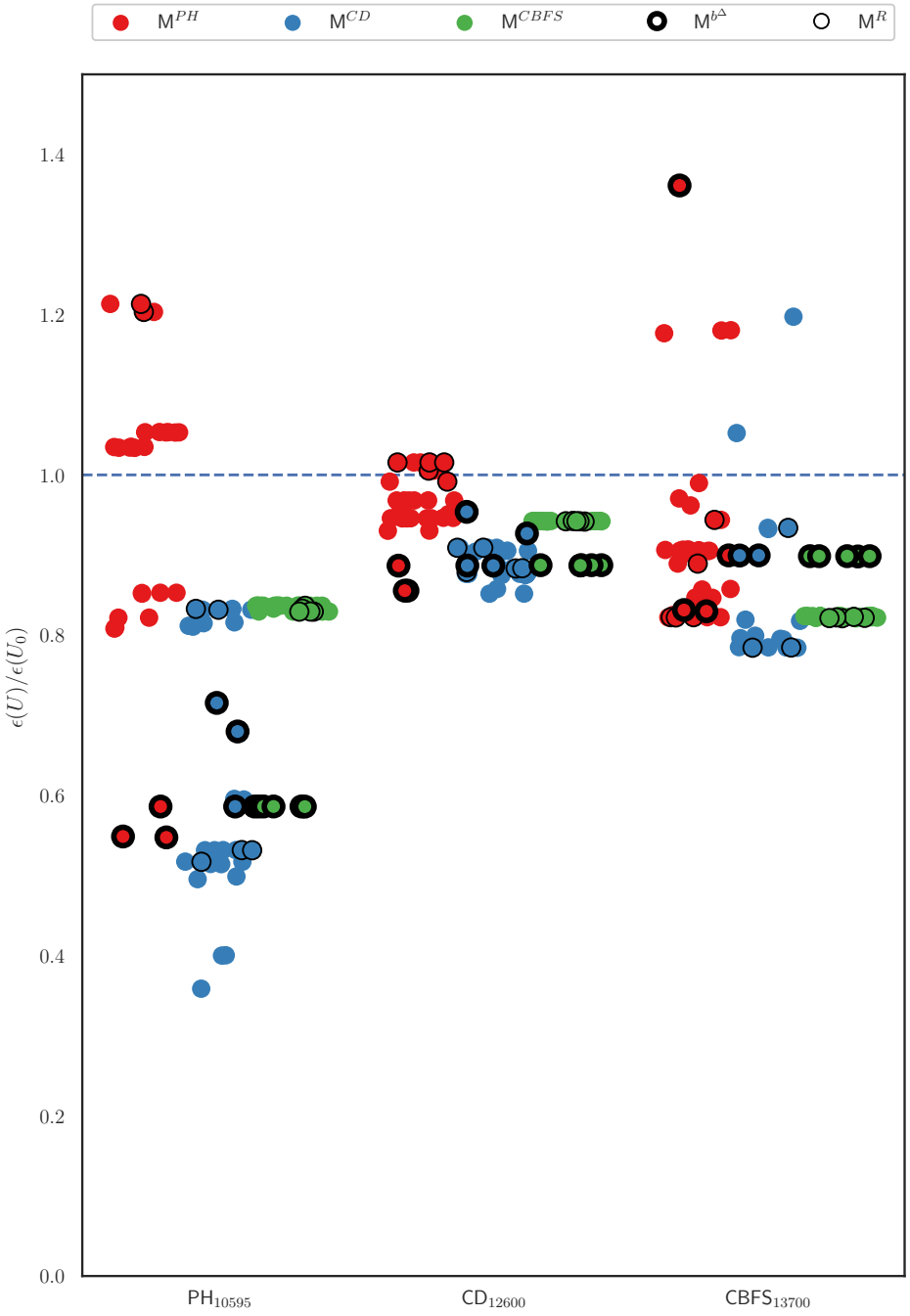


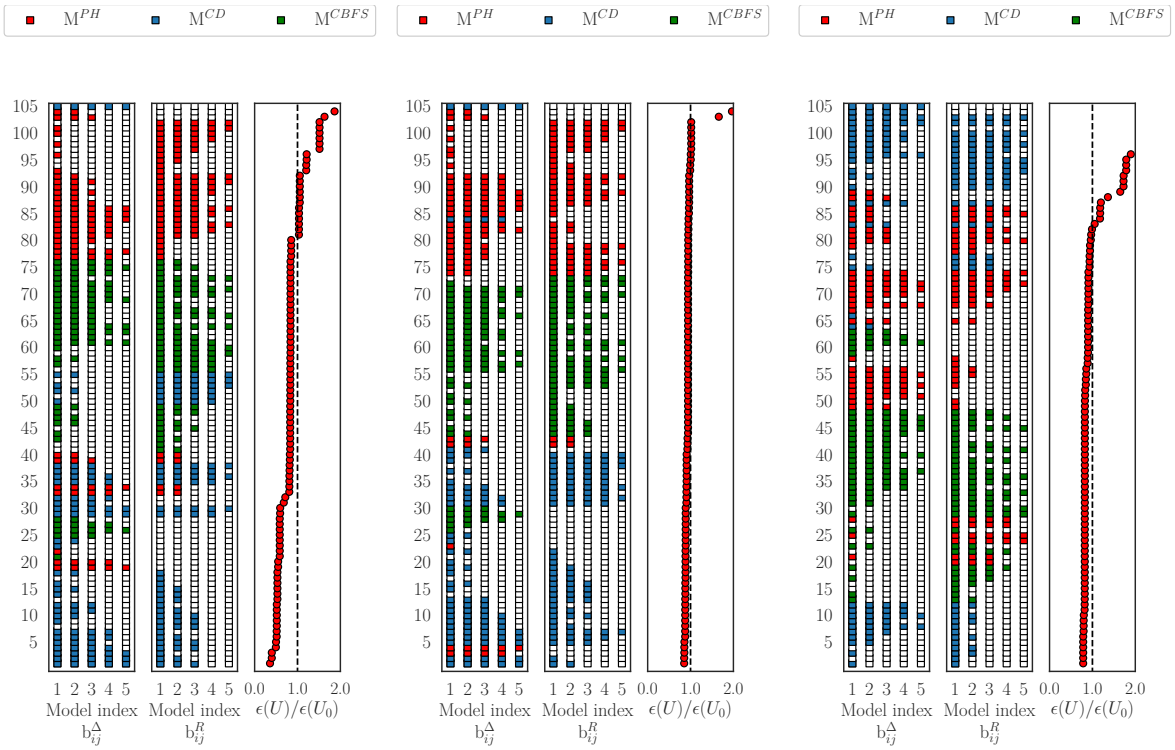
Figure 6.29: Mean squared error of each model on the velocity field, normalized by the mean squared error of the baseline $k-\omega$ model. Colors indicate on which flow case the models have been trained. The labels on the x-axis indicate the test cases. Individual models for b_{ij}^{Δ} and b_{ij}^{R} are indicated by the thickness of the markers' edge, as can be seen from the legend.

Out of the three different test cases, the most significant improvements are observed for the periodic hill flow case. An interesting thing to note is that some of the models that have a different train and test case actually performed better than models that had already seen the data, i.e., trained and tested

on the same flow case. For example, some of the models trained on the converging-diverging channel show a significant improvement for the periodic hill test case in comparison with the models that were actually trained on the periodic hill case.

As can be seen, the best performing models consists of combinations of corrections, as well as individual corrections. A clear exception to this is the periodic hill test case, as individual models for b_{ij}^{Δ}, which are trained on the periodic hill and curved backward-facing step flow cases, show a significant improvement over combinations of models for b_{ij}^{Δ} and b_{ij}^{R}. Although there are some models that do not improve or only show minimal improvement to the baseline prediction, it can be stated that overall the predictions are improved.

Figures 6.30a to 6.30c presents an ordered list of the models' performance for each test case, in which the vertical position indicates the accuracy, i.e., the best performing model starts at index 1 and the worst performing model has the index 105.



(a) Periodic hill. (b) Converging-diverging channel. (c) Curved backward-facing step.

Figure 6.30: Left and middle matrices indicate the index of the selected model for b_{ij}^{Δ} and b_{ij}^{R}, respectively. On the right, the mean squared error of the velocity field normalized by the mean squared error of the $k-\omega$ model is shown.

The colours of the matrices indicate on which flow case the models have been trained. If all squares are empty, no correction has been used. The amount of squares indicate which of selected models of has been used, in which one colored square indicates the simplest model and five colored squares indicate the most complex model. Overall, the models trained on the converging-diverging channel show the best performance in terms of accuracy. For the periodic hill and curved backward-facing step test cases, three groups can be identified, namely a (small) group that does not show an improvement compared to the baseline, a group that is close or similar to the baseline and a group that clearly shows an improvement to the baseline. However, for the converging-diverging channel test case, the majority of the models show a result that is close or similar to the baseline, with only a few models that show an improvement.

Based on this cross-validation, the following models are selected:

$$
\begin{aligned}
M_{b^\Delta}^{(1)} =& \left(0.04774 + 2.60258 I_1 + 0.19508 I_1^2 + 0.2869 I_3\right) T_{ij}^{(1)} \\
&+ \left(7.25465 - 31.60593 I_1 + 1.7335 I_1 I_2 - 2.17851 I_1^2 + 25.41991 I_2 - 2.14463 I_2^2 + 0.86626 I_5\right) T_{ij}^{(2)} \\
&+ \left(2.47908 + 1.32854 I_2\right) T_{ij}^{(3)} \\
&+ \left(-2.51541 + 1.40518 I_1 + 0.28088 I_2^2\right) T_{ij}^{(4)} \\
&+ - 0.14138 I_2^2 T_{ij}^{(6)} \\
&+ \left(-12.70996 - 0.86675 I_1 + 1.07231 I_2\right) T_{ij}^{(7)} \\
&+ \left(-15.7871 - 1.08849 I_1 + 0.86577 I_2\right) T_{ij}^{(8)} \\
&+ 1.32336 T_{ij}^{(9)} , \\
M_R^{(1)} =& \left(0.65403 - 0.38525 I_1^2\right) T_{ij}^{(1)}
\end{aligned}
\tag{6.1}
$$

$$
\begin{aligned}
M_{b^\Delta}^{(2)} =& \left(0.44424 + 1.14252 I_1 + 0.08237 I_1^2 + 5.98153 I_2 - 0.04833 I_2^2\right) T_{ij}^{(1)} \\
&+ \left(7.64381 - 17.35012 I_1 - 0.58208 I_2^2\right) T_{ij}^{(2)} \\
&+ \left(1.67877 + 0.02441 I_1^2 - 5.04813 I_2\right) T_{ij}^{(3)} \\
&+ \left(-5.04803 I_1 - 0.17578 I_2^2\right) T_{ij}^{(4)} \\
&+ - 0.00739 T_{ij}^{(5)} \\
&+ \left(6.02141 - 0.0493 I_2\right) T_{ij}^{(6)} \\
&+ 0.29104 I_2 T_{ij}^{(7)} \\
&+ - 8.67492 T_{ij}^{(8)} \\
&+ - 5.04816 T_{ij}^{(9)} , \\
M_R^{(2)} =& 0
\end{aligned}
\tag{6.2}
$$

$$
\begin{aligned}
M_{b^\Delta}^{(3)} =& 4.21876 T_{ij}^{(2)} , \\
M_R^{(3)} =& 0.65327 T_{ij}^{(1)}
\end{aligned}
\tag{6.3}
$$

The data on which each of the selected models is trained is indicated by 'TRAIN' and their respective ranks per test case are shown in Table 6.1.

Table 6.1: Specification of ranking and training data for selected models from framework 1.

	PH_{10595}		CD_{12600}		$CBFS_{13700}$	
Model	Index i	Error	Index j	Error	Index k	Error
$M^{(1)}$	2	0.40004	TRAIN	0.851741	10	0.799251
$M^{(2)}$	TRAIN	0.548669	3	0.85551	52	0.831338
$M^{(3)}$	16	0.531676	TRAIN	0.883539	1	0.784259

6.4.2. Framework 2 - Elliptic Blending Lag Parameter Model

The results of the SpaRTA approach applied to the lag parameter model are visualized in Fig. 6.31, in which the mean squared error on the velocity field for each model is shown for the various test cases.

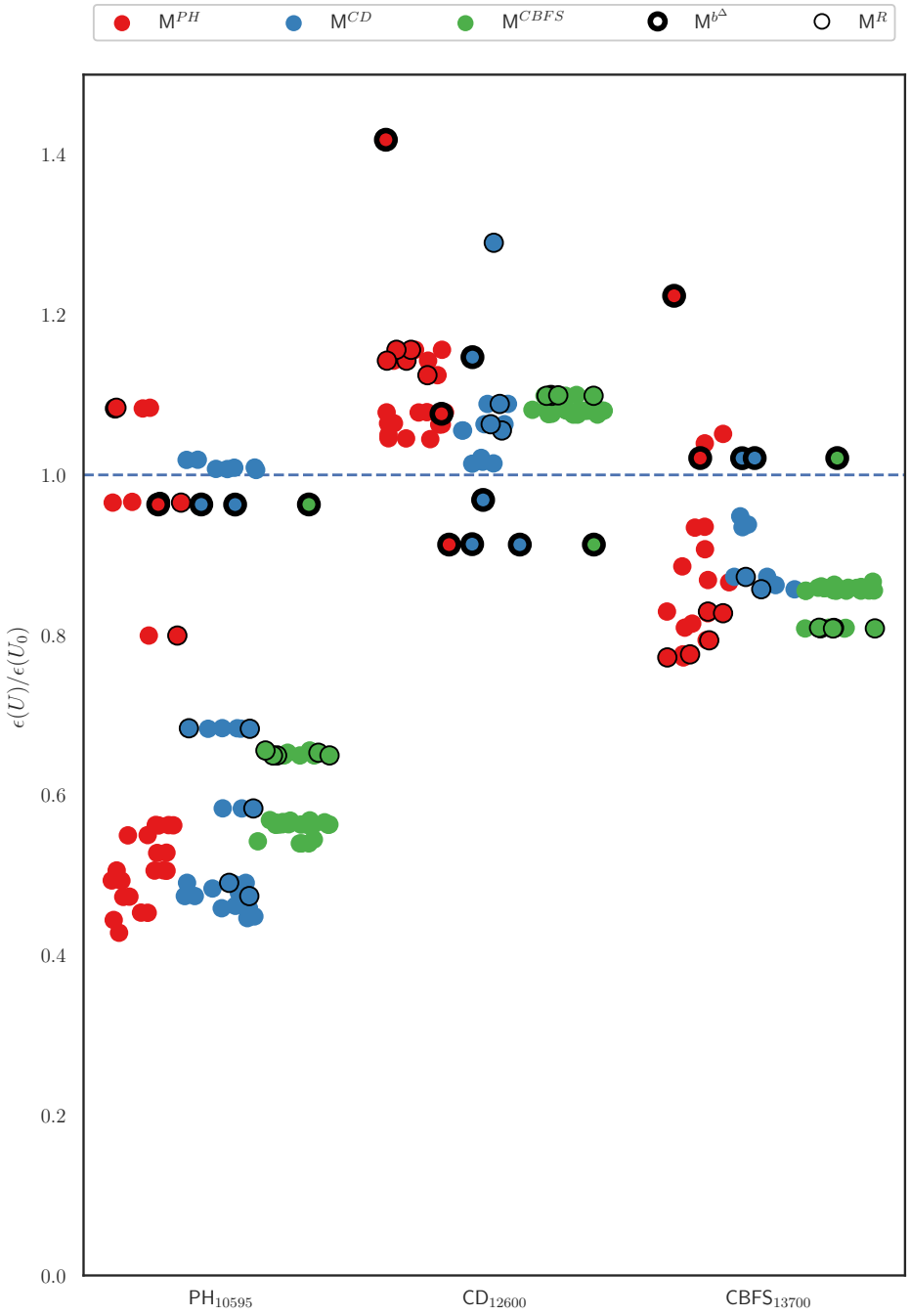


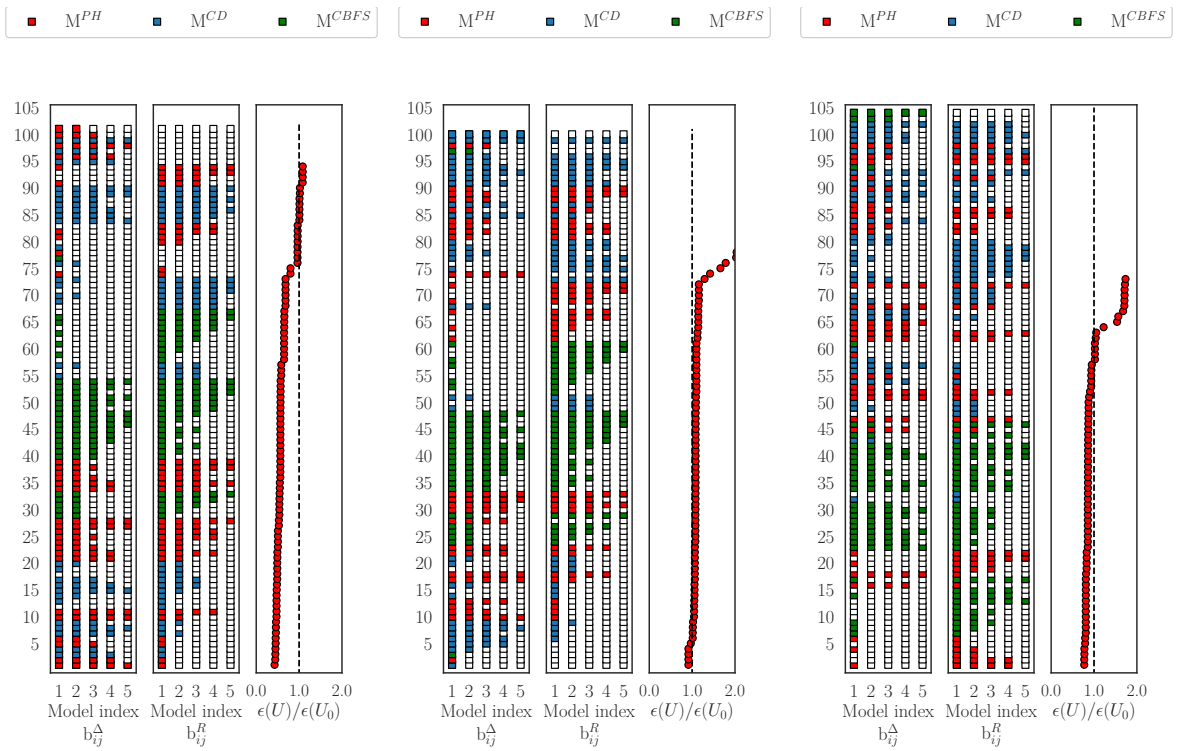
Figure 6.31: Mean squared error of each model on the velocity field, normalized by the mean squared error of the baseline $k-\omega$ model. Colors indicate on which flow case the models have been trained. The labels on the x-axis indicate the test cases. Individual models for b_{ij}^{Δ} and b_{ij}^{R} are indicated by the thickness of the markers' edge, as can be seen from the legend.

Most of the models tested on the periodic hill and curved backward-facing step test cases show significant improvements. The majority of the models tested on the converging-diverging channel has a slightly lower or equal performance to the baseline $k-\omega$ model. This is in contrast to what one has seen from framework 1, in which almost all of the models were slightly below the line of $\epsilon(U)/\epsilon(U_0) = 1.0$, but neither showed a significant improvement as is the case for the periodic hill and the curved backward-facing step.

The majority of the models based on combination of corrections that are tested on the periodic hill flow case show a significant improvement, which is in contrast with the result of framework 1.

Individual corrections for b_{ij}^{Δ} show minor improvements over the baseline $k-\omega$ model. However, no improvements are shown for the curved backward-facing step when individual corrections for b_{ij}^{Δ} are used. Furthermore, these individuals models for b_{ij}^{Δ} are the only models that showed an improvement over the baseline for the converging-diverging channel. Individual models for b_{ij}^{R} and combination of corrections had a significant impact on the performance for the periodic hill and curved backward-facing step flow cases, as these were among the best performing models. For the converging-diverging channel, almost no improvements were observed, as only some of the individuals individual models for b_{ij}^{Δ} showed an improvement over the baseline.

The performance of each model per test case is ranked based on the lowest mean squared error on the velocity field in Fig. 6.32.



(a) Periodic hill. (b) Converging-diverging channel. (c) Curved backward-facing step.

Figure 6.32: Left and middle matrices indicate the index of the selected model for b_{ij}^{Δ} and b_{ij}^{R}, respectively. On the right, the mean squared error of the velocity field normalized by the mean squared error of the $k-\omega$ model is shown.

The most significant improvements can be observed for the periodic hill test case. For the converging-diverging channel, most of the models show a performance that is similar to that of the baseline $k-\omega$ model. Only a few models, all individual corrections for b_{ij}^{Δ}, show a slightly increases accuracy. For the curved backward-facing step, three categories of models can be observed. Roughly half of the models have an increased accuracy, while the remaining models have a similar or worse performance in terms of accuracy compared to the baseline $k-\omega$ model. A clear difference with the results from framework 1 is the amount of models that have a performance in the range of $\epsilon(U)/\epsilon(U_0) \leq 2.0$, which is a lower amount for framework 2. This can be attributed to an increase in numerical instability, most probable a result stemming from the addition of an extra transport equation.

Based on this cross-validation, the following models are selected:

$$\begin{aligned} M_{b^{\Delta}}^{(4)} &= 4.55351 T_{ij}^{(2)} \,, \\ M_{R}^{(4)} &= 0.71476 T_{ij}^{(1)} \end{aligned} \tag{6.4}$$

$$
\begin{aligned}
M_{b^\Delta}^{(5)} =& \left(0.62733 + 0.676 I_1^2 + 3.44735 I_2 + 0.24795 I_2^2\right) T_{ij}^{(1)} \\
&+ \left(10.36939 - 8.69553 I_1\right) T_{ij}^{(2)} \\
&+ \left(2.49994 + 0.0059 I_1^2 - 4.00825 I_2\right) T_{ij}^{(3)} \\
&- 4.0099 I_1 T_{ij}^{(4)} \\
&+ \left(3.47917 + 0.24724 I_2\right) T_{ij}^{(6)} \\
&- 4.34745 T_{ij}^{(8)} \\
&- 4.00827 T_{ij}^{(9)} , \\
M_R^{(5)} =& 1.17598 T_{ij}^{(1)}
\end{aligned}
\tag{6.5}
$$

$$
\begin{aligned}
M_{b^\Delta}^{(6)} =& \left(0.55506 + 3.53055 I_1 + 0.35542 I_1^2 + 3.91245 I_2 + 0.20333 I_2^2 + -0.0874 I_5\right) T_{ij}^{(1)} \\
&+ \left(10.54569 - 8.31959 I_1 + 0.29447 I_1 I_2 + -0.39661 I_1^2 + 5.8563 I_2 + 0.14723 I_5\right) T_{ij}^{(2)} \\
&+ \left(-3.06951 + 1.57336 I_1 + 0.07269 I_1^2 - 1.72248 I_2\right) T_{ij}^{(3)} \\
&+ \left(-8.60453 - 1.72392 I_1 + 1.24944 I_2\right) T_{ij}^{(4)} \\
&+ 0.03653 T_{ij}^{(5)} \\
&+ \left(3.93653 + 0.20289 I_2\right) T_{ij}^{(6)} \\
&+ \left(-2.92815 - 0.14723 I_1\right) T_{ij}^{(7)} \\
&+ \left(-4.1595 - 0.1983 I_1 + 0.14722 I_2\right) T_{ij}^{(8)} \\
&- 1.72244 T_{ij}^{(9)} , \\
M_R^{(6)} =& 1.17598 T_{ij}^{(1)}
\end{aligned}
\tag{6.6}
$$

The rank of the selected models and their respective training data are indicated in Table 6.2.

Table 6.2: Specification of ranking and training data for selected models from framework 2.

	PH_{10595}		CD_{12600}		$CBFS_{13700}$	
Model	Index i	Error	Index j	Error	Index k	Error
$M^{(4)}$	13	0.473898	TRAIN	1.05551	32	0.8571
$M^{(5)}$	TRAIN	0.444028	13	1.05021	16	0.809185
$M^{(6)}$	TRAIN	0.428085	10	1.04469	18	0.814275

6.4.3. Discussion

A significant portion of the models developed by both frameworks are able to capture the model-form error of different flow cases, which validates the combination of the k-corrective-frozen-RANS approach and SpaRTA method again as an effective way of enhancing RANS models using full-field high-fidelity data. Models developed by the second framework suffer more from numerical instability, which can be seen from their relatively large corresponding mean-squared error of the velocity field that is normalized by the mean-squared error of the conventional $k-\omega$ model. No significant improvements are observed in terms of predictive capability and generalizability when comparing the two frameworks. The cross-validation results of framework 1 for the converging-diverging channel even show an improvement over the second framework. Another observation is that the results by both frameworks are (to a certain extent) grouped, which means that the errors on the velocity field are similar, which is especially true for the predictions by the individuals models M_{b^Δ} and M_R. Generally speaking, neither the individual or combined corrections showed a clear advantage over one another for the various test cases. All corrections showed the ability to improve the predictions over the baseline prediction. In order to have a more in-depth comparison of the two frameworks, the three best overall models of each framework are compared in Section 6.5, in which the velocity- and turbulence kinetic energy profiles of these selected models are shown in order to assess their predictive capability and generalizability.

6.5. Assessment of Predictive Capability and Generalizability

This section assesses the predictive capability and generalizability of the selected models of framework 1 and framework 2. These models are tested on the periodic hill, converging-diverging channel and curved backward-facing step.

6.5.1. Periodic Hill

The velocity profiles predicted by the selected models are shown in Fig. 6.33. The models $M^{(2)}$, $M^{(5)}$ and $M^{(6)}$ have been trained on the periodic hill flow case.

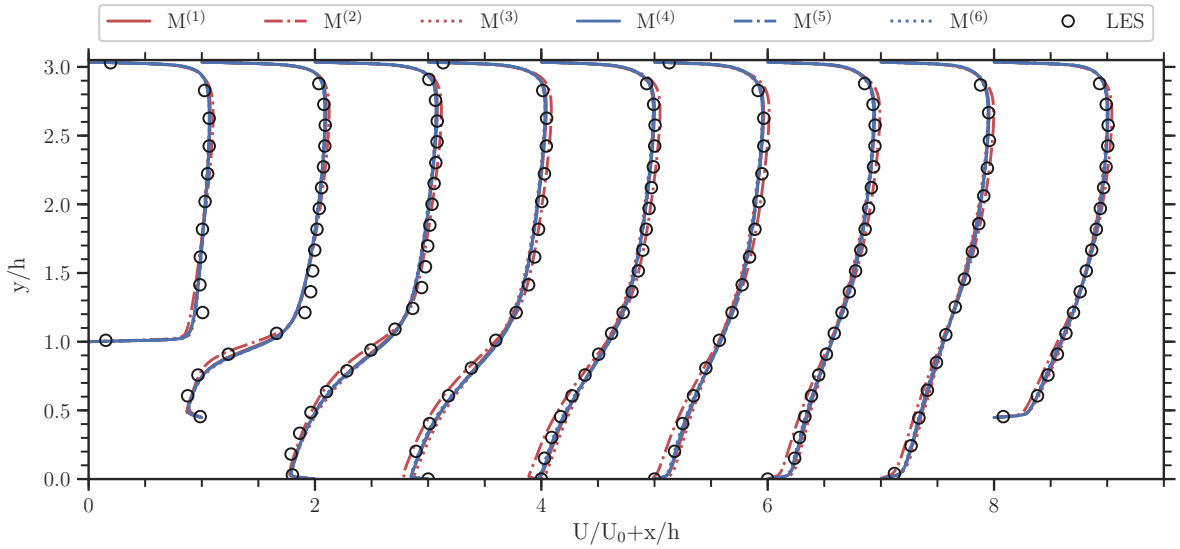


Figure 6.33: Streamwise velocity profiles.

In general, the predicted velocity profiles are in close agreement with the high-fidelity data by Breuer et al. [6]. The largest discrepancy is observed for $M^{(2)}$ at streamwise locations where $y/h \leq 1.0$, as the velocity profiles are slightly underpredicted. A possible explanation could be that $M^{(2)}$ has no correction $M_R^{(2)}$, which in turn leads to a lower k.

The turbulence kinetic energy profiles are visualized in Fig. 6.34.

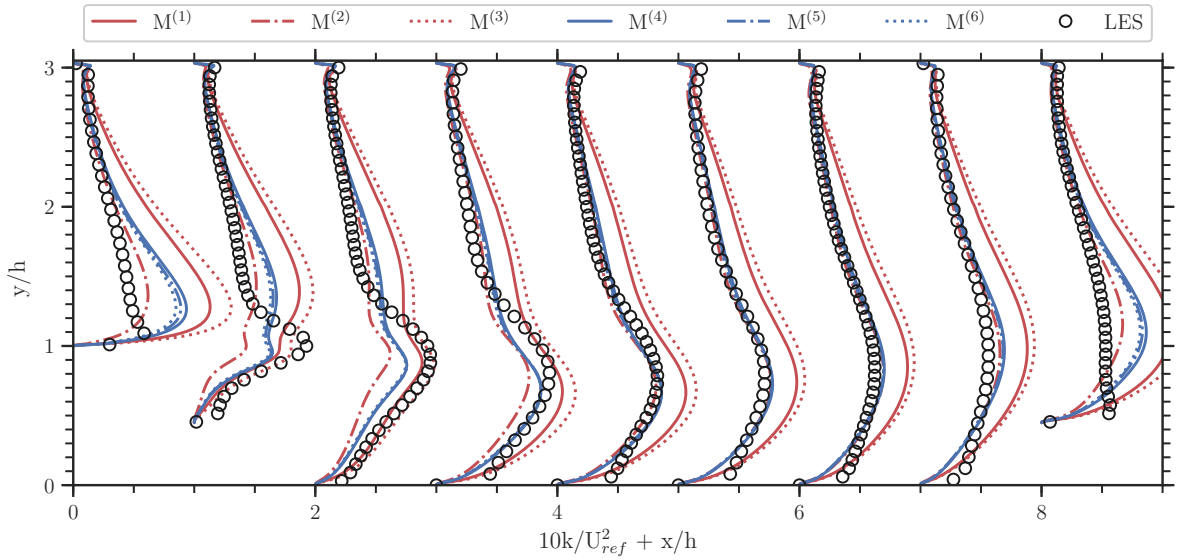


Figure 6.34: Streamwise turbulence kinetic energy profiles.

The differences between the selected models are more significant, as well as the discrepancies with the high-fidelity data. The two models from framework 1 ($M^{(1)}$ and $M^{(3)}$) that are not trained on the periodic hill data show the largest discrepancy out of all models. These two models also seem to systematically overpredict k, which could be explained by the fact that a correction for M_R is used, while $M^{(2)}$ has no correction $M_R^{(2)}$. Although model $M^{(4)}$ from framework 2 has not seen this training data, it shows a significant improvement over the models from framework 1 and is in close agreement with the two models that are actually trained on this data.

6.5.2. Converging-Diverging Channel

In order to assess the predictive capability of the models, the converging-diverging channel has been used as a second test case. The predicted streamwise velocity profiles by the various models are shown in Fig. 6.35.

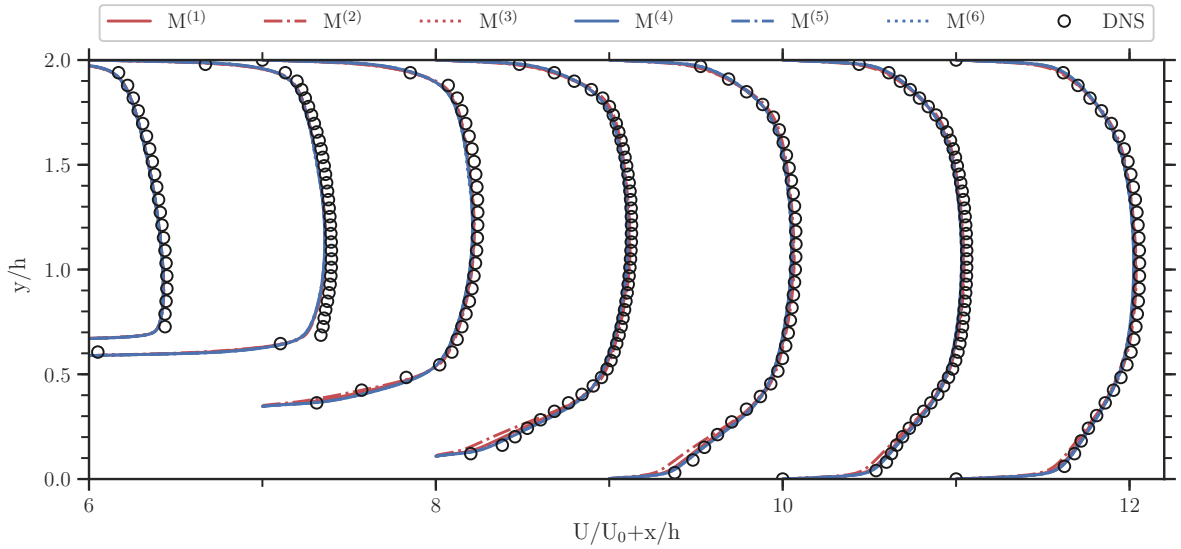


Figure 6.35: Streamwise velocity profiles.

Models $M^{(1)}$, $M^{(3)}$ and $M^{(4)}$ have been trained on high-fidelity data from the converging-diverging channel flow case. As can be seen, all models are in close agreement with the DNS data, except for model $M^{(2)}$. Being the only model from framework 1 that has not been trained on data from the converging-diverging channel, it shows minor discrepancies around $y/h \leq 0.3$ for the streamwise profiles located at $x/h = 8$, $x/h = 9$ and $x/h = 10$. None of these discrepancies have been observed for the models from framework 2.

The turbulence kinetic energy profiles are shown in Fig. 6.36.

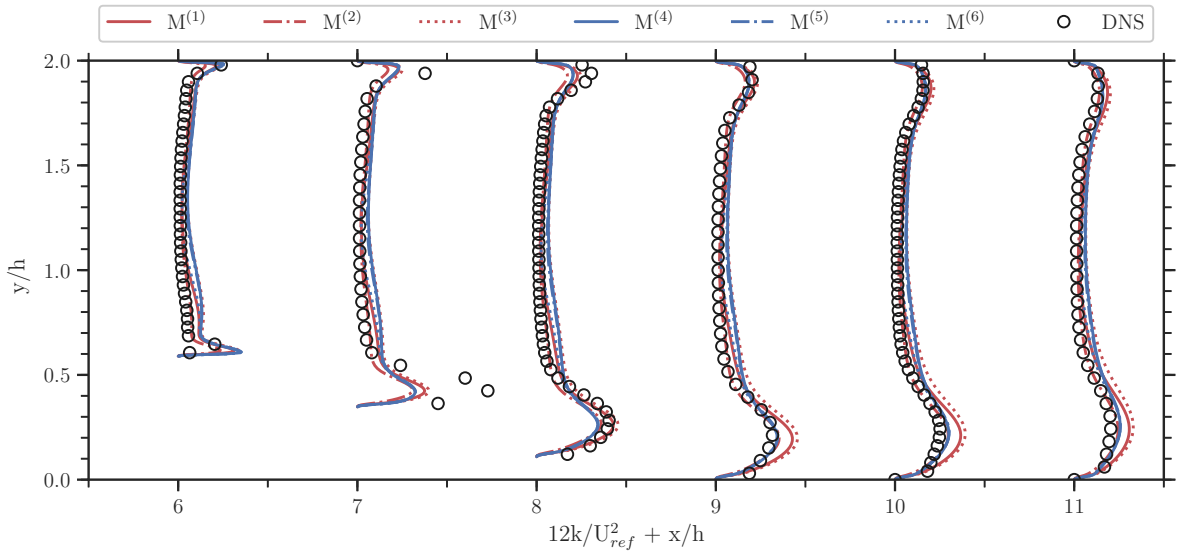


Figure 6.36: Streamwise turbulence kinetic energy profiles.

Again, the largest discrepancies are observed for the turbulence kinetic energy profiles. It can be observed that most of the models overpredict the turbulence kinetic energy. However, certain models show a larger discrepancy than others. Model $M^{(2)}$ most accurately predicts the turbulence kinetic energy out of the models from framework 1, which is remarkable since it is the only model out of the three that has not been trained on the converging-diverging channel. A possible explanation could be that for $M^{(1)}$ and $M^{(3)}$ corrections have been used, while this was not the case for $M^{(2)}$. The models obtained from framework 2 show a more significant overprediction for $y/h = 0.7 - 1.5$, except for model $M^{(6)}$. However, for streamwise profiles $x/h = 9$, $x/h = 10$ and $x/h = 11$, the models of framework 2 are more accurate.

6.5.3. Curved Backward-Facing Step

The third and final test case is the curved backward-facing step. None of the models from framework 1 and 2 have been trained on high-fidelity data from the current flow case. The velocity profiles are shown in Fig. 6.8.

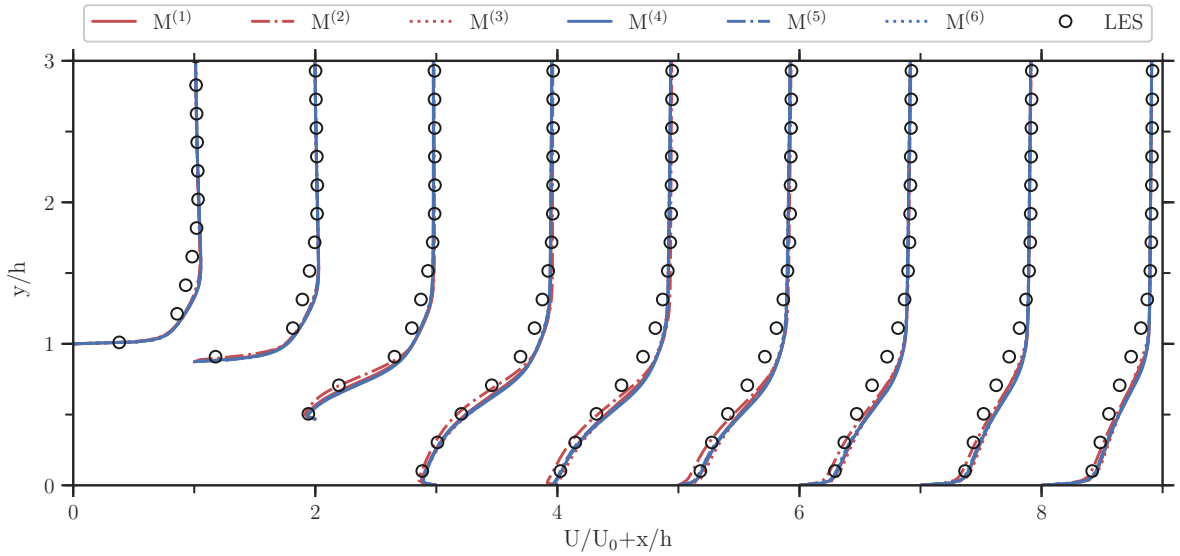


Figure 6.37: Streamwise velocity profiles.

All models systematically overpredict the velocity profiles around $y/h \leq 1.5$ Again, model $M^{(2)}$ shows a slightly different prediction compared to the other models, as it predicts lower velocity profiles than the other models. A possible explanation could be that this results from not having a correction $M_R^{(2)}$.

The turbulence kinetic energy profiles are shown in Fig. 6.38.

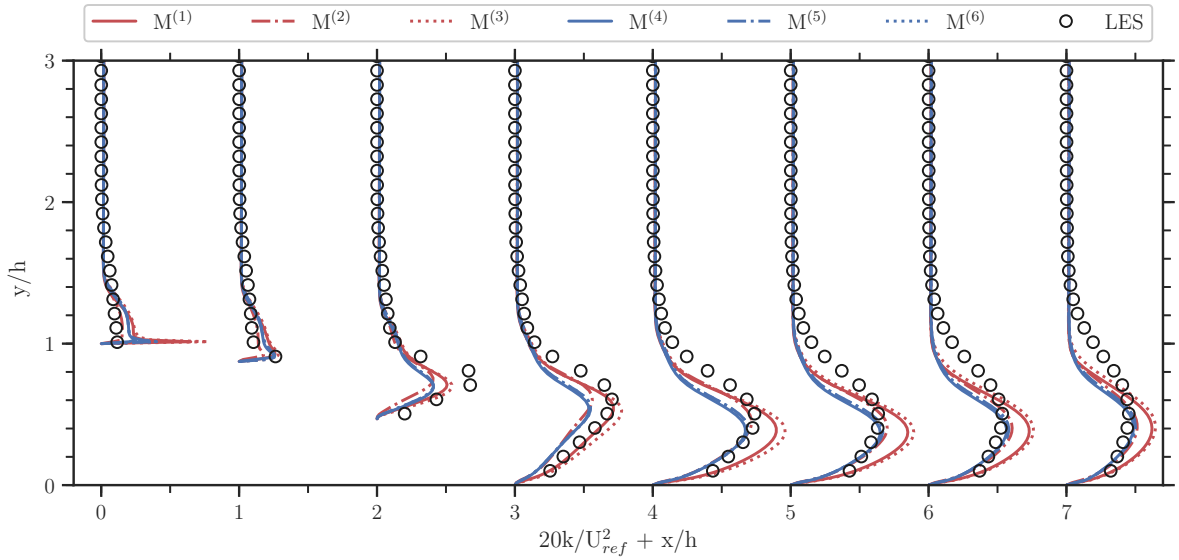


Figure 6.38: Streamwise turbulence kinetic energy profiles.

The models obtained by using framework 1 overpredict the turbulence kinetic energy from the streamwise profile at $x/h = 4$ onwards, especially models $M^{(1)}$ and $M^{(3)}$. The opposite can be observed for the models obtained using framework 2, which underpredict or are close to the high-fidelity data of the turbulence kinetic energy profiles. Among the models obtained from the second framework, no significant differences are observed.

6.5.4. Discussion

A more detailed view on the predictions by the best overall selected models confirms the observations and conclusions drawn in Section 6.4. The velocity profiles predicted by the various selected models are truly similar, with an exception for $M^{(2)}$, which showed a minor difference with the other five models. This confirms the ability of the models to generalize their predictions to two-dimensional flow cases that differ in complexity. It also shows that there is no significant improvement observed between the two frameworks with respect to predictive capability or generalizability, and thus, the additional transport equation in the data-driven lag parameter model does not result in an improvement in performance.

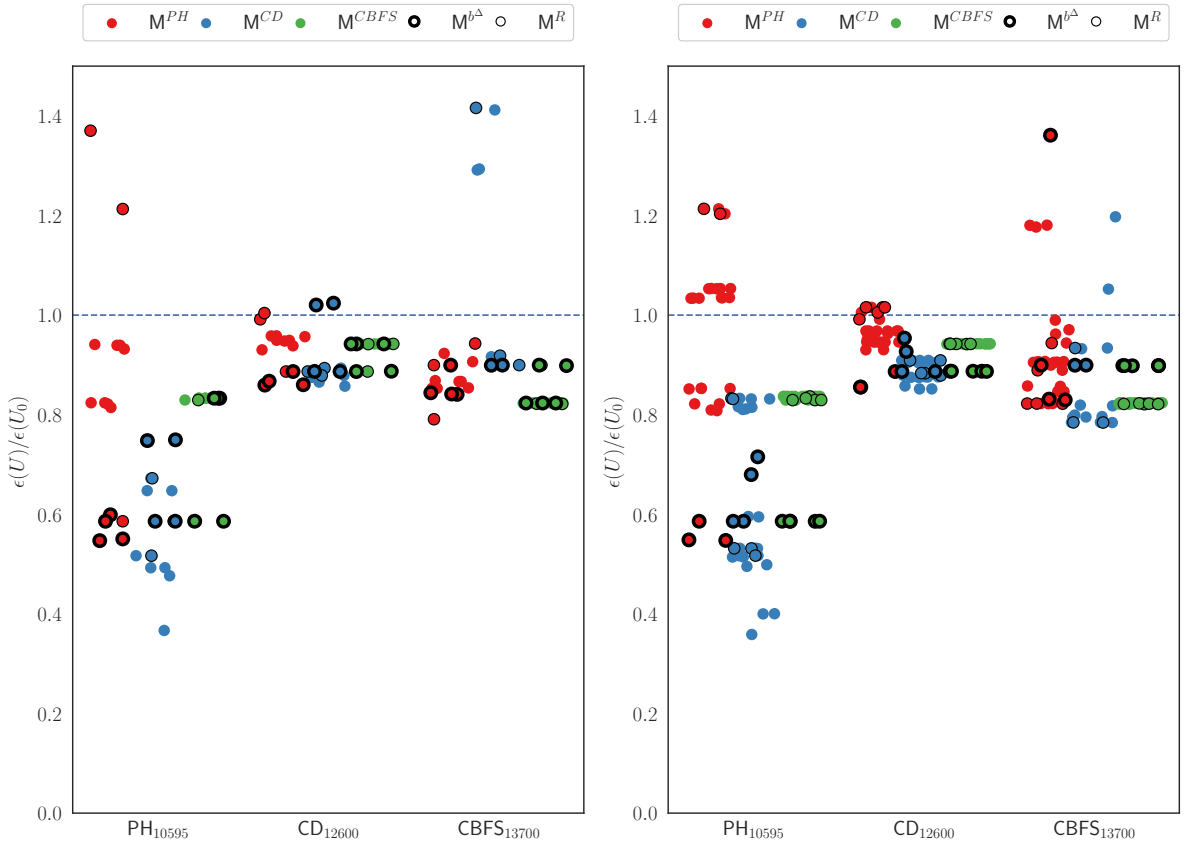
6.6. Reduced versus Full Integrity Basis

In the work of Schmelzer et al. [68], a reduced integrity basis is used to construct a library of candidate functions. This reduced integrity basis consists of the tensors $T_{ij}^{(1)}$, $T_{ij}^{(2)}$, $T_{ij}^{(3)}$ and the invariants I_1 and I_2. Comparing the results in this study with the results in [68] would not make a fair comparison to assess the performance of a full integrity versus that of reduced integrity basis, since a different turbulence model is used and minor differences might be present in the setup. Instead, framework 1, which is an augmented version of the conventional $k-\omega$ model, has been used to compute the additive terms b_{ij}^{Δ} and R. Then, SpaRTA is utilized to explore the functional forms of the additive terms using the reduced integrity basis, i.e. in the form of

$$b_{ij}\left(S_{ij}, \Omega_{ij}\right) = \sum_{n=1}^{3} T_{ij}^{(n)} \alpha_n \left(I_1, I_2\right) , \tag{6.7}$$

$$R = 2kb_{ij}^{R}\frac{\partial \overline{u_i}}{\partial x_j}\,. \tag{6.8}$$

Following the approach in this study, five models for b_{ij}^{Δ} have been handpicked for each flow case, such that a representative subset is selected. For b_{ij}^{R}, three models have been handpicked per flow case to form a representative subset. Please refer to Appendix E for a visualization of the full set of discovered models and for the handpicked models that are used for cross-validation purposes. The results of the cross-validation are shown in Fig. 6.39 and are compared to the cross-validation results of framework 1, which made use of the full integrity basis.



(a) Reduced integrity basis

(b) Full integrity basis

Figure 6.39: Mean squared error of each model on the velocity field, normalized by the mean squared error of the baseline $k-\omega$ model. Colors indicate on which flow case the models have been trained. The labels on the x-axis indicate the test cases. Individual models for b_{ij}^{Δ} and b_{ij}^{R} are indicated by the thickness of the markers' edge, as can be seen from the legend.

As can be seen from Fig. 6.39, the result of the cross-validation using the reduced integrity basis is similar to that of the full integrity basis. For each test case, similar improvements over the baseline $k-\omega$ model are observed when comparing the reduced integrity basis with the full integrity basis. Furthermore, the aforementioned grouping of the results, which refers to the errors on the velocity field being similar, is also observed for the reduced integrity basis. It is concluded that, in its current form, a full integrity basis shows no advantage over a reduced integrity basis.

7

Conclusions and Recommendations

A novel framework has been introduced that utilizes the k-corrective-frozen-RANS approach and the Sparse Regression of Turbulent Stress Anisotropy (SpaRTA) method to infer algebraic stress models from high-fidelity data. These algebraic stress models have been formed using the full integrity basis and are added as corrections to existing turbulence models, such that new data-driven models are formed. This concept has been applied to the standard $k-\omega$ model by Wilcox [55] and the elliptic blending lag parameter model by Biswas et al. [21] such that the impact of the additional transport equation for the lag parameter can be assessed. Both frameworks have been tested on the periodic hill, converging-diverging channel and curved backward-facing step flow cases.

Conclusions regarding the findings in this work are given in Section 7.1. In Section 7.2, recommendations for future work are provided.

7.1. Conclusions

The findings in this study will be concluded by answering the main research questions.

RQ1 **What is the ability of the data-driven elliptic blending lag parameter model in terms of generalizing its predictions to other flow cases?**

The elliptic blending lag parameter model was implemented in OpenFOAM and the results were verified against the results from Biswas et al. [21]. The k-corrective-frozen-RANS approach has been used to identify the corrective terms b_{ij}^{Δ} and R, which are corrections to the stress-strain relation and the transport equations, using high-fidelity data. In order to assess the impact of the lag parameter, two frameworks have been utilized. In the first framework, the standard two-equation $k-\omega$ has been used as a baseline for the data-driven approach. In the second framework, the elliptic blending lag parameter model has been used, which has a transport equation for the lag parameter in addition to the conventional $k-\omega$ transport equations, to test the data-driven approach. It was validated that the computed corrective terms capture the model-form error for both frameworks and that the high-fidelity mean flow data was reproduced. SpaRTA has been used to infer algebraic stress models for these corrective terms. Both frameworks produced models using high-fidelity data of the periodic hill, converging-diverging channel and the curved backward-facing step, of which a representative subset in terms of complexity and accuracy was handpicked. Cross-validation was done using CFD in order to test the performance of the selected models on unseen data and to rank the models. In general, neither the individual or combined corrections showed a clear advantage over one another for the various test cases, but they all showed the ability to improve the predictions over the baseline prediction. A significant portion of these data-driven models showed an increase in accuracy over the baseline two-equation $k-\omega$ model, even on unseen data. Some models even performed superior compared to models that actually had the same training and test data. The final three models of each framework were selected based on a low mean-squared error on the various test cases. All the models were able to generalize their predictions to two-dimensional flow cases with similar Reynolds number that had different complexity, as significant improvements over the baseline two-equation $k-\omega$ model

were observed. Features from the integrity basis that were observed in all the selected models from different training data were $T_{ij}^{(2)}$ for M_{b^Δ} and $T_{ij}^{(1)}$ for M_R. Utilizing an additional transport equation for the lag parameter in this data-driven approach did not result in any significant improvements in terms of predictive capability or generalizability, as both data-driven approaches showed a similar performance, although the data-driven $k-\omega$ models were more numerically stable. It was found that corrective terms formulated using a reduced integrity basis yields data-driven models that have a similar predictive capability compared to models that used the full integrity basis to construct the corrective terms.

RQ2 **What are the strengths and weaknesses of applying the k-corrective-frozen-RANS approach to the lag parameter model?**

For the two-dimensional flow cases considered in this study, it was demonstrated for both frameworks that the k-corrective-frozen-RANS approach has the ability to capture the model-form error and reproduce the high-fidelity mean flow quantities. When the corrective terms were modelled using the integrity basis, it was shown that a large portion of the resulting models was able to infer the quantities of interest and improve the prediction of the mean velocity field over the baseline prediction by the conventional $k-\omega$ model. Accurately predicting the turbulence kinetic energy profiles proved to be more challenging, as these showed discrepancies with the high-fidelity data. The data-driven lag parameter model has the ability to generalize its predictions to two-dimensional flow cases that have a similar Reynolds number and different complexity. The k-corrective-frozen-RANS approach does not depend on an optimization procedure and is a very cost-efficient method. It is, however, also limited to full-field data. Applying the k-corrective-frozen-RANS approach and SpaRTA method to the lag parameter model resulted in an increase in numerical instability, compared to the data-driven $k-\omega$ model, which is likely to be a result of the additional transport equation. The computational cost associated with the k-corrective-frozen-RANS approach is not significant, as the computation of the corrective terms b_{ij}^{Δ} and R is performed within the order of a couple of minutes on a normal personal computer. Propagating these terms through a CFD solver results in a computational time that is in the order of tens of minutes, similar to RANS simulations. Furthermore, the computational costs associated with the machine learning phase are not significant, as SpaRTA utilizes sparse regression to identify the models, which is computationally inexpensive.

7.2. Recommendations

Machine Learning

Currently, two-dimensional flow cases with similar Reynolds number have been used for the train and test phase of the models. It would be interesting to see how a model, that is trained on three-dimensional data, performs on a two-dimensional test case. Two- and three-dimensional flows have different characteristics, and thus, different features. It is an interesting way to see if the model is able to recognize the differences and possibly improve the generalizability. Another important and natural step is to gather more full-field high-fidelity data in order to train and test the machine learning algorithm on a larger variety of flow cases. A nice follow up study would be to investigate the performance on the periodic hills of parametrized geometries by Xiao et al. [51]. Preferably also flow cases that have a higher Reynolds number, as the current study involves test cases that have similar Reynolds numbers. Therefore, a nice extension of this work could be to investigate the predictions of higher Reynolds number flows to further research the ability of the model to generalize.

In this study, the models were trained separately on each of the data sets, after which cross-validation was used to test the performance of the models. As a next step, the data sets from the periodic hill, converging-diverging channel and curved backward-facing step could be combined to create one large data set for training purposes. This would require more available full field high-fidelity data such that the performance of the trained models can be tested. By combining the data sets, new patterns in the data might be discovered that, when using the data set separately, might have been missed. As a result, the ability to infer quantities of interest and generalizability of the models could possibly be improved.

Validation has shown that the k-corrective-frozen-RANS approach has the ability to capture the model-form error. However, in this research, utilizing an additional transport equation for the lag parameter didn't result in an improvement over the data-driven $k-\omega$ approach, as both frameworks

showed a similar performance. An improvement could be the usage of more complex function approximators for the corrective terms, which would be a nice extension for future work. More complex function approximators such as neural networks or random forests capture more details of the flow [68], which in combination with an additional transport equation could enhance the ability to generalize and infer quantities of interest.

Another very interesting approach would be the use of Bayesian inversion in combination with adjoint theory to enhance the turbulence model. High-fidelity data such as DNS or LES will be used to infer the corrective terms. This approach has already been demonstrated on a simple problem in Appendix B, namely a one-dimensional heat conduction problem. However, due to time constraints and the implementational complexity of adjoints it was decided to continue in a different way. Extending Bayesian inversion together with this adjoint approach to the elliptic blending $k-\omega$ lag parameter model would provide useful insight into how this approach influences the generalizability of the algorithm, as the eddy viscosity is altered in a more physical manner. This indirect modification of the eddy viscosity could pave the way for innovative developments in the prediction of eddy viscosity in future research.

Generalizations of k-Corrective-Frozen-RANS Approach

In its current form, the corrective term R is added as an additive term to the production term P in the additional transport equation of the lag parameter. Some research has already been done into a separate corrective term that accounts for the deficit between the lag parameter computed from high-fidelity data and that from a RANS simulation. It was decided to not continue down this path, since the distributions are significantly different and the assumptions taken to derive the transport equation of the lag parameter are too large to allow a comparison. A truly interesting topic for future research would be to further investigate the generalizations of the k-corrective-frozen-RANS approach. There are so many possibilities left with regard to the placement of a single (or multiple) corrective term(s). First of all, the location of the placement of a corrective term could be an interesting topic for future work. Secondly, the type of corrective term to use in an additional transport equation should be further investigated. In this study, all the corrections were constructed using the integrity basis. It would also be interesting to see how different features influence the performance of the algorithm. For example, a corrective term that is trained on different mean flow features could be a topic of future research.

A

Field Inversion and Adjoint Methodology

In this study, field inversion is used to enhance the model by using high-fidelity data such as DNS, LES or experimental data. In practice, these observations always contain some noise. Thus, the data is assumed to consist of the output by the forward model and noise, i.e.:

$$\boldsymbol{d} = h(\boldsymbol{\beta}) + \varepsilon \tag{A.1}$$

where $\boldsymbol{\beta}$ denotes a spatially varying corrective term and ε indicates the noise. In a forward problem, the goal is to find the data given a certain model $\boldsymbol{h}$, which is referred to as the forward model. The forward model that is used in this study is the lag model augmented with the corrective term. The goal of field inversion in this research is to obtain a corrective term $\boldsymbol{\beta}_{\text{true}}$ such that the solution $\boldsymbol{h}(\boldsymbol{\beta}_{\text{true}})$ best approximates the high-fidelity data $\boldsymbol{d}_{\text{true}}$.

A possible approach for a discrete linear inverse problem is to assume that the data errors are independent and Gaussian. Then, it can be shown that the maximum likelihood principle solution is the solution to a least-squares problem, which is found by minimizing the 2-norm of the residual between the output by the forward model and the data, i.e.:

$$\boldsymbol{\beta}_{MLE} = \arg\min_{\boldsymbol{\beta}} \|h(\boldsymbol{\beta}) - \boldsymbol{d}\|_2^2 \,. \tag{A.2}$$

However, there are challenges that come along with this approach. In many cases, the least-squares problem is not well-conditioned [83], which leads to a large set of solutions that can significantly vary from each other. In such cases, regularization may be required in order to obtain a stable solution. For nonlinear problems, several local optima might exists, and thus, it can be difficult to find the global minimum.

In this study, the Bayesian approach is used, in which the solution has the form of a probability distribution over the corrective fields, referred to as the posterior distribution. This is a fundamental difference with respect to the classical approach, in which a single corrective field is the final solution. It is possible to derive an expression for the posterior distribution using conditional probability and Bayes' formula. Conditional probability is given by

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \tag{A.3}$$

which can be rewritten to obtain Bayes' formula:

$$\rho(\boldsymbol{\beta} \mid \boldsymbol{d}) = \frac{\rho(\boldsymbol{d} \mid \boldsymbol{\beta})\,\rho(\boldsymbol{\beta})}{\rho(\boldsymbol{d})} \tag{A.4}$$

where $\rho(\boldsymbol{d} \mid \boldsymbol{\beta})$ is the likelihood, $\rho(\boldsymbol{\beta})$ is the prior distribution and $\rho(\boldsymbol{d})$ is used as normalization. In the case that the observations $\boldsymbol{d}$, the corrective field $\boldsymbol{\beta}$ and the likelihood $\rho(\boldsymbol{d} \mid \boldsymbol{\beta})$ are normally distributed, it can be shown that the process of finding the distribution of the corrective field reduces to finding the maximum a posteriori (MAP) solution [83], i.e.

$$\boldsymbol{\beta}_{\text{MAP}} = \arg\max_{\beta} \rho(\boldsymbol{\beta} \mid \boldsymbol{d}) \,. \tag{A.5}$$

The observation noise is assumed to be unbiased Gaussian noise, i.e.:

$$\varepsilon \sim \mathcal{N}(0, C_m) \tag{A.6}$$

where C_m is the observational covariance matrix. Then, because the noise has a Gaussian distribution, the likelihood has a Gaussian distribution as well. The resulting expression for the likelihood is given by

$$\rho(\boldsymbol{d}|\boldsymbol{\beta}) = \rho_\varepsilon(\boldsymbol{d} - h(\boldsymbol{\beta})) \sim \mathcal{N}(0, C_m) \ . \tag{A.7}$$

In addition to the aforementioned assumptions, the corrective field $\boldsymbol{\beta}$ is assumed to have a Gaussian prior:

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_{prior}, C_\beta) \tag{A.8}$$

where $\boldsymbol{\beta}_{prior}$ is the prior mean and C_β is the prior covariance matrix. In the work of [22], Parish and Duraisamy suggested to use a Gaussian prior and determine the prior variance based on the following procedure:

1. Assume prior variance σ_β^2.
2. Sample the prior distribution of β with assumed covariance matrix C_β.
3. Propagate samples through forward model.
4. Check whether observed solution falls within the $\pm 2\sigma$ limits of the distribution predicted by the forward model.
5. If necessary, adjust σ_β^2 to ensure observed solution falls within the aforementioned limits.

The posterior is then proportional to

$$\rho(\boldsymbol{\beta}|\boldsymbol{d}) \propto \exp\left[-\frac{1}{2}(\boldsymbol{d} - h(\boldsymbol{\beta}))^T C_m^{-1}(\boldsymbol{d} - h(\boldsymbol{\beta})) - \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_{prior})^T C_\beta^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_{prior})\right] . \tag{A.9}$$

However, the optimization problem involves an exponential function and can be rewritten in a simpler form as

$$\boldsymbol{\beta}_{\text{MAP}} = \arg\min_\beta \underbrace{\frac{1}{2}(\boldsymbol{d} - h(\boldsymbol{\beta}))^T C_m^{-1}(\boldsymbol{d} - h(\boldsymbol{\beta})) + \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{prior}})^T C_\beta^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{prior}})}_{\text{J}} \tag{A.10}$$

where J is the cost function. However, this optimization is difficult, as β is high-dimensional. Therefore, gradient-based optimization is used. A naive approach would be to evaluate the chain rule directly as

$$\frac{dJ}{d\beta_i} = \frac{\partial J}{\partial \beta_i} + \frac{\partial J}{\partial Q_k}\frac{dQ_k}{d\beta_i} \tag{A.11}$$

and differentiate the residuals of the primal equations $R(Q, \beta)$, which gives a linear system of equations to solve for $\frac{dQ_k}{d\beta_i}$:

$$\frac{dR_m}{d\beta_i} = \frac{\partial R_m}{\partial \beta_i} + \frac{\partial R_m}{\partial Q_k}\frac{dQ_k}{d\beta_i} = 0 \tag{A.12}$$

This leads to the requirement of solving N linear systems to obtain $\frac{dQ_k}{d\beta_i}$, which is the drawback of the so-called direct approach. A more clever way to obtain $\frac{dJ}{d\beta_i}$ is by introducing the Lagrangian:

$$L(\beta, \Psi) = J(Q, \beta) + \Psi^T R(Q, \beta) \tag{A.13}$$

where Ψ is the adjoint vector. Note that $L \equiv J$, since $R(Q, \beta) = 0$, and thus

$$\begin{aligned}\frac{dJ}{d\beta_i} = \frac{dL}{d\beta_i} &= \frac{\partial J}{\partial \beta_i} + \frac{\partial J}{\partial Q_k}\frac{dQ_k}{d\beta_i} + \Psi^T\left(\frac{\partial R_m}{\partial \beta_i} + \frac{\partial R_m}{\partial Q_k}\frac{dQ_k}{d\beta_i}\right) \\ &= \frac{\partial J}{\partial \beta_i} + \Psi^T\frac{\partial R_m}{\partial \beta_i} + \left(\frac{\partial J}{\partial Q_k} + \Psi^T\frac{\partial R_m}{\partial Q_k}\right)\frac{dQ_k}{d\beta_i}\end{aligned} \tag{A.14}$$

The freedom of choice of Ψ can be used to eliminate $\frac{dQ_k}{d\beta_i}$. In order to do so, Ψ has to satisfy the adjoint equation:

$$\Psi^T \frac{\partial R_m}{\partial Q_k} = -\frac{\partial J}{\partial Q_k} \tag{A.15}$$

One linear system has to be solved in order to find the solution to the adjoint equation. Equation (A.14) then reduces to the form of (A.16), where the sensitivity of the primal variable Q with respect to the corrective term β is removed from the equation:

$$\frac{dJ}{d\beta_i} = \frac{\partial J}{\partial \beta_i} + \Psi^T \frac{\partial R_m}{\partial \beta_i} \tag{A.16}$$

Generally, the observational covariance matrix C_m depends upon the statistics of the data that is observed and can be constructed in various ways [22]. The most simple model is defined as

$$C_m = \sigma_{obs}^2 \boldsymbol{I} \tag{A.17}$$

where σ_{obs} is a scalar that only represents the mean standard deviation, and thus, all covariances are neglected. A more elaborate model is defined as

$$C_m = \boldsymbol{\sigma}_{obs}^2 \boldsymbol{I} \tag{A.18}$$

where $\boldsymbol{\sigma}_{obs}$ is a vector that represents the standard deviation at each observation point. The last model is the most elaborate and makes the assumption that multiple realizations of the same dataset D are available. Then, the exact covariance matrix is given by

$$C_m = \mathbb{E}\left[\left(D_i - \overline{D_i}\right)\left(D_j - \overline{D_j}\right)\right] \tag{A.19}$$

in which $\overline{D_i} = \mathbb{E}\left[D_i\right]$.

In order to determine the posterior covariance, the computation of the Hessian is required. The adjoint-adjoint approach is used to perform this computation, which is given by

$$H_{ij} = \frac{\partial^2 J}{\partial \beta_i \partial \beta_j} + \psi_m \frac{\partial^2 R_m}{\partial \beta_i \partial \beta_j} + \mu_{i,m} \frac{\partial R_m}{\partial \beta_j} + \nu_{i,m} \frac{\partial^2 J}{\partial Q_n \partial \beta_j} + \nu_{i,n} \psi_m \frac{\partial^2 R_m}{\partial Q_n \partial \beta_j} \quad m, n \,\epsilon\, [1, M] \tag{A.20}$$

where

$$\nu_{i,n} \frac{\partial R_m}{\partial Q_n} = -\frac{\partial R_m}{\partial \beta_i} \quad m, n \,\epsilon\, [1, M] \quad i \,\epsilon\, [1, N] \tag{A.21}$$

$$\mu_{i,m} \frac{\partial R_m}{\partial Q_k} = -\frac{\partial^2 F}{\partial \beta_i \partial u_k} - \psi_m \frac{\partial^2 R_m}{\partial \beta_i \partial Q_k} - \nu_{i,n} \frac{\partial^2 J}{\partial Q_n \partial Q_k} - \nu_{i,n} \psi_m \frac{\partial^2 R_m}{\partial Q_n \partial Q_k}\,. \tag{A.22}$$

The posterior covariance can be determined by evaluating the inverse of the Hessian of the cost function J at the MAP point [22]

$$\boldsymbol{C}_{\beta_{map}} = \mathbf{H}^{-1}\Big|_{\beta_{map}} = \left(\frac{d^2 J}{d\beta_i d\beta_j}\right)^{-1}\Big|_{\beta_{map}}. \tag{A.23}$$

The Newton-CG method, which is a gradient-based optimization algorithm, is given by

$$\frac{\beta_i^{k+1} - \beta_i^k}{\alpha^k} = -\left[\frac{d^2 J^k}{d\beta_i d\beta_j}\right]^{-1} \frac{\partial J^k}{\partial \beta_i} \tag{A.24}$$

in which the positive scalar α^k denotes the step length. This method uses the conjugate gradient (CG) method to compute the search direction [84].

B

Field Inversion - 1D Heat Conduction

B.1. Inversion

The field inversion phase and the machine learning phase is first illustrated on a simple test case, the one-dimensional heat equation from Parish and Duraisamy [22]. This one-dimensional heat equation has a radiative and a convective term and is given by

$$\frac{d^2T}{dz^2} = \varepsilon(T)\left(T_\infty^4 - T^4\right) + h\left(T_\infty - T\right), \quad z\,\epsilon\,[0\ldots1] \tag{B.1}$$

where ε is the emissivity of the material, the convection coefficient has a value of $h = 0.5$, T_∞ is the temperature of the surroundings and z is the spatial coordinate. Equation (B.1) is referred to as the 'true' model, in which the emissivity is given by

$$\varepsilon\left(T\right) = \left[1 + 5\sin\left(\frac{3\pi T}{200}\right) + \exp\left(0.02T\right) + \mathcal{N}\left(0, 0.1^2\right)\right] \cdot 10^{-4}\,. \tag{B.2}$$

This process is imperfectly modelled as

$$\frac{d^2T}{dz^2} = \varepsilon_0\left(T_\infty^4(z) - T^4\right) \tag{B.3}$$

which is referred to as the base model. The constant emissivity is given as $\varepsilon_0 = 5 \cdot 10^{-4}$. The model outputs for the base and true solution for surroundings temperatures $T_\infty = [10, 20, 30, 40, 50]$ are shown in Fig. B.1.

The inversion phase is initiated by placing a spatial multiplier to ε_0, which yields

$$\frac{d^2T}{dz^2} = \beta(z)\varepsilon_0\left(T_\infty^4(z) - T^4\right)\,. \tag{B.4}$$

Equation (B.4) is referred to as the augmented model. The goal of this phase is to optimize $\beta(z)$ such that the augmented model closely resembles the true model. The true form of the corrective term can be derived from the expressions for the true model and the augmented model and is given by

$$\beta_{\text{true}} = \frac{1}{\varepsilon_0}\left[1 + 5\sin\left(\frac{3\pi T}{200}\right) + \exp\left(0.02T\right) + \mathcal{N}\left(0, 0.1^2\right)\right] \cdot 10^{-4} + \frac{h}{\varepsilon_0}\frac{T_\infty - T}{T_\infty^4 - T^4}\,. \tag{B.5}$$

The cost function is given by

$$J = \frac{1}{2}\left(\boldsymbol{d} - h\left(\boldsymbol{\beta}\right)\right)^T C_m^{-1}\left(\boldsymbol{d} - h\left(\boldsymbol{\beta}\right)\right) + \frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{prior}}\right)^T C_\beta^{-1}\left(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{prior}}\right)\,. \tag{B.6}$$

The data $\mathbf{d}$ consists of 100 realizations of the true model. Three different models are used for the observational covariance matrix, which are described in Appendix A. The prior has been chosen to be $\boldsymbol{\beta}_{\text{prior}} = 1$, such that the augmented model reduces to the base model. The prior of the corrective term

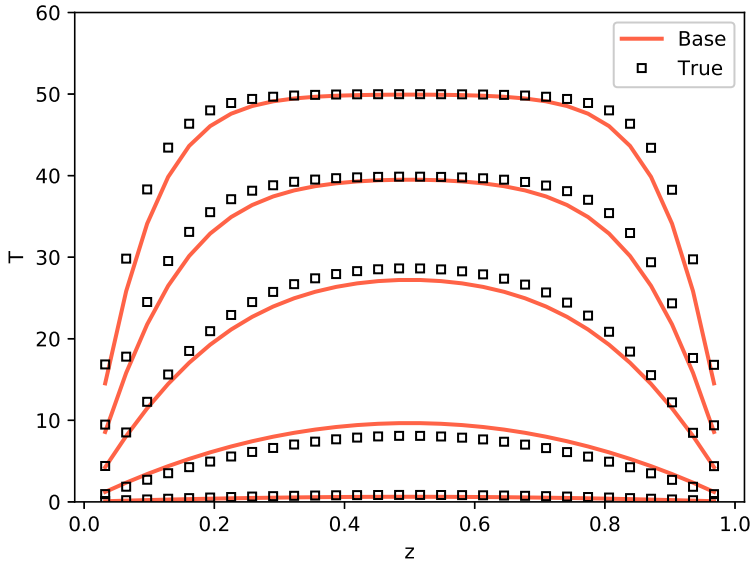


Figure B.1: Model output for the base and true solution for several values of T_∞.

is sampled using a selected prior variance, such that the observed temperature is within 2σ limits of the prior PDF of temperature [22]. The goal is to find the maximum a posteriori (MAP) solution, i.e.

$$\boldsymbol{\beta}_{\text{MAP}} = \arg\max_{\beta} \rho(\boldsymbol{\beta} \mid \mathbf{d}) \,. \tag{B.7}$$

However, as this optimization problem is high dimensional, gradient-based methods are required to efficiently minimize the cost function J.

B.2. Discrete adjoint

The discrete adjoint approach, which is discussed in Appendix A, is used to calculate the derivatives that are required for the gradient-based methods. These derivatives are given by (B.10) to (B.19). The equations that govern the one-dimensional heat equation are solved using a second order central difference scheme on an equidistant grid consisting of 31 nodes. The optimization routine uses the Newton-CG method, which is described in Appendix A. The discretization of the true process is, which is done using an iterative scheme, is given by

$$T_i = \frac{1}{2}\left[T_{i+1} + T_{i-1} - (\Delta z)^2 \left(\varepsilon(T_i)\left(T_i^4 - T_\infty^4\right) + h\left(T_i - T_\infty\right)\right)\right]. \tag{B.8}$$

In a similar fashion the augmented model, including the spatial corrective term $\beta(z)$, is discretized as

$$T_i = \frac{1}{2}\left[T_{i+1} + T_{i-1} - (\Delta z)^2 \beta(z)\varepsilon_0\left(T_i^4 - T_\infty^4\right)\right]. \tag{B.9}$$

The first order partial derivatives of the cost function and the governing equations are given by

$$\frac{\partial J}{\partial \beta_i} = C^{-1}_{\beta,ij}\left(\beta_j - \beta_{\text{prior},j}\right) \tag{B.10}$$

$$\frac{\partial J}{\partial T_l} = -H_{il} C^{-1}_{m,ij}\left(d_j - H_{jk} T_k\right) \tag{B.11}$$

$$\frac{\partial R_j}{\partial \beta_i} = \frac{1}{2}(\Delta z)^2 \varepsilon_0\left(T_j^4 - T_{\infty,j}^4\right)\delta_{ji} \tag{B.12}$$

$$\frac{\partial R_j}{\partial T_i} = \left(1 + 2(\Delta z)^2 \beta_j \varepsilon_0 T_j^3\right)\delta_{ji} - \frac{1}{2}\delta_{j,i-1} - \frac{1}{2}\delta_{j,i+1} \,. \tag{B.13}$$

Consequently, the expressions for the second order partial derivatives are given by

$$\frac{\partial^2 J}{\partial \beta_i \partial \beta_k} = C^{-1}_{\beta,ij}\delta_{jk} \tag{B.14}$$

$$\frac{\partial^2 J}{\partial \beta_i \partial T_k} = 0 \tag{B.15}$$

$$\frac{\partial^2 J}{\partial T_l \partial T_n} = H_{il} C^{-1}_{m,ij} H_{jn} \tag{B.16}$$

$$\frac{\partial^2 R_j}{\partial \beta_i \partial \beta_k} = 0 \tag{B.17}$$

$$\frac{\partial^2 R_j}{\partial \beta_i \partial T_k} = 2\left(\Delta z\right)^2 \varepsilon_0 T_j^3 \delta_{ji}\delta_{jk} \tag{B.18}$$

$$\frac{\partial^2 R_j}{\partial T_i \partial T_k} = 6\left(\Delta z\right)^2 \beta_j \varepsilon_0 T_j^2 \delta_{ji}\delta_{jk}\,. \tag{B.19}$$

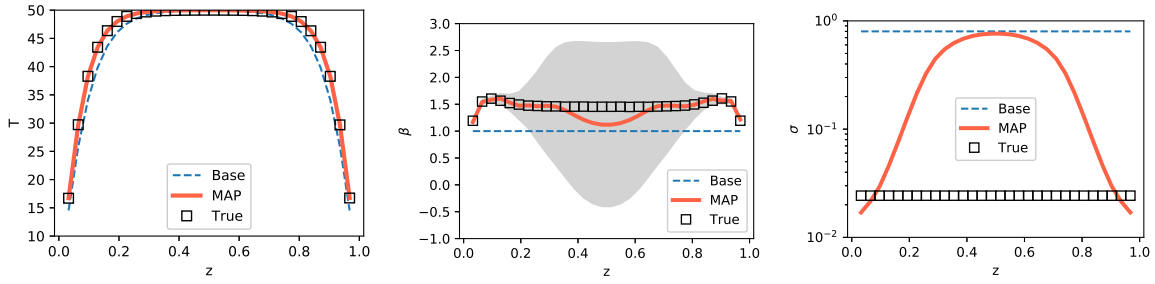
B.3. Results

The results of the inversion phase are shown in Fig. B.2 for the various expressions of the observational covariance matrix C_m, which are discussed in Appendix A. These figures show the temperature T, the corrective term β and the standard deviation σ for the base, MAP and true solution where the surroundings temperature $T_\infty = 50$. It can be seen that the MAP solutions for the temperature closely resembles the true solution for all three observational covariance matrices. The MAP solutions for the corrective term only resembles the true solution when the full covariance matrix is used. For the other two covariance matrices, the MAP solution for β is especially erroneous near the center of the domain. The MAP solution for the standard deviation is overestimated compared to the true solution when the diagonal covariance matrices are used. Only the full covariance matrix yields a MAP solution for σ that closely resembles the true solution. Parish and Duraisamy [22] used several cases for the inversion, which are listed in Table B.1.

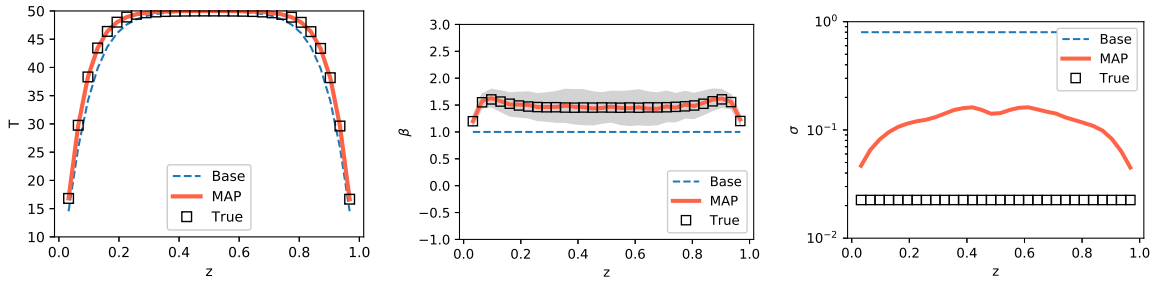
Table B.1: Various cases used for the inversion.

Case	T_∞	σ_{prior}	β_{prior}
1	5	20	1
2	10	2	1
3	15	1	1
4	20	1	1
5	25	0.5	1
6	30	1	1
7	35	1	1
8	40	1	1
9	45	1	1
10	50	0.8	1

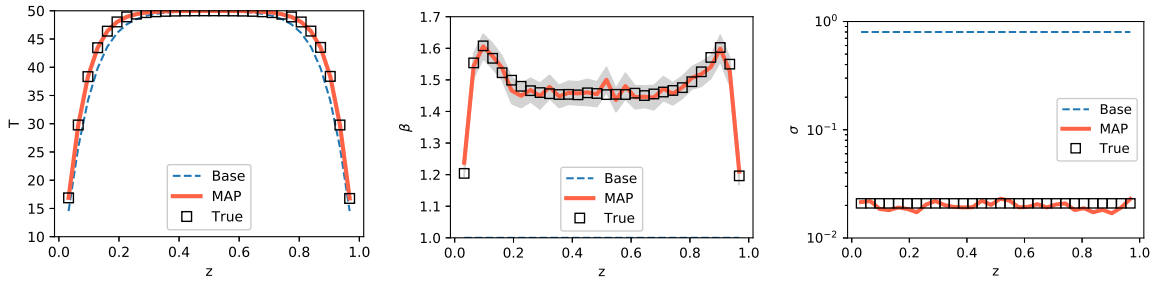
The results of the inversion for the ten cases in Table B.1 are shown in Fig. B.3 for all previously discussed types of observational covariance matrices. The radiative and convective term, β_r and β_c respectively, are extracted from β_{map} and are plotted against the true solution. The radiative term is shown along with uncertainty bounds. It can be seen that both the radiative and convective terms have been inferred correctly for all the different covariance matrices. The uncertainty bounds decrease as the amount of information contained in the covariance matrices increases. In both cases with the diagonal observational covariance matrices, the posterior variance does not accurately represent the true solution, especially at either high or low temperatures. However, the full covariance matrix yields truly accurate results for the posterior variance compared to the true solution for the whole domain.



(a) Diagonal covariance matrix with $C_m = \sigma^2 \boldsymbol{I}$; $\sigma = 0.02$.

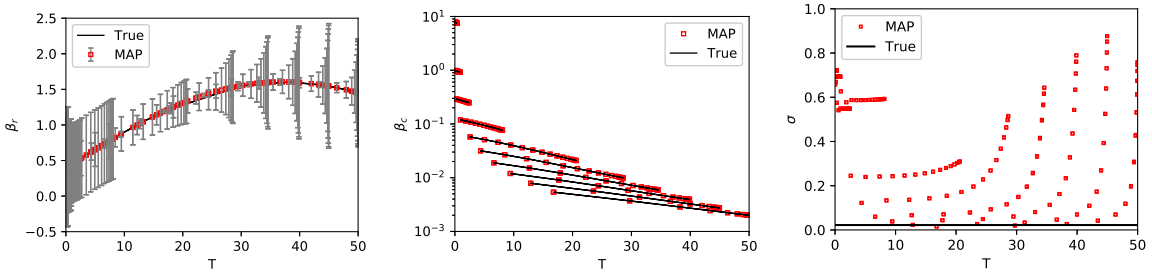


(b) Diagonal covariance matrix with $C_m = \boldsymbol{\sigma}^2 \boldsymbol{I}$.

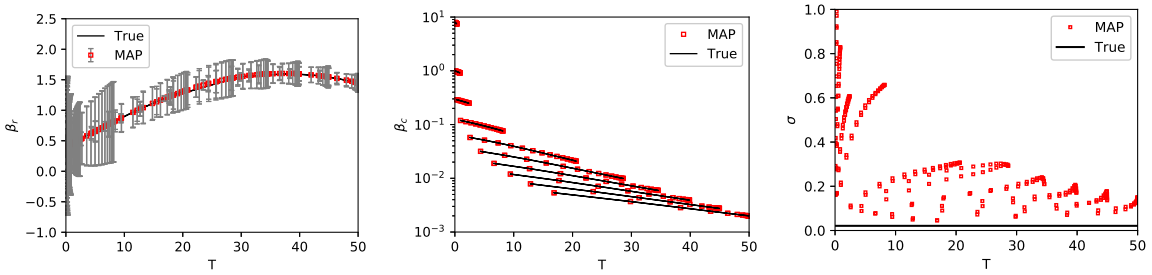


(c) Full covariance matrix C_m.

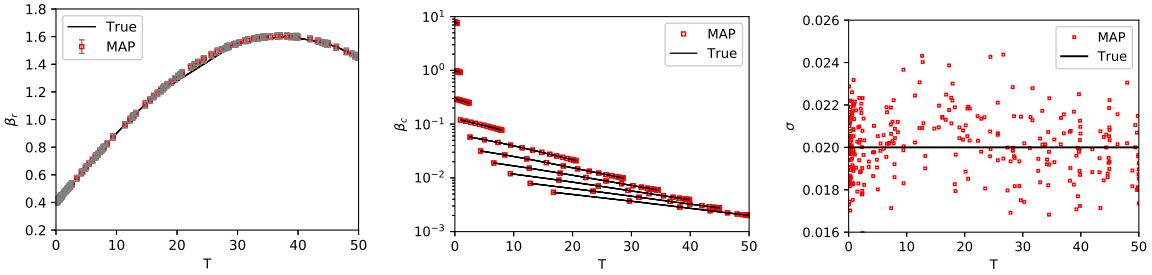
Figure B.2: Comparison of temperature, corrective term and standard deviation distributions for the base, MAP and true solution.



(a) Diagonal covariance matrix with $C_m = \sigma^2 \boldsymbol{I}$; $\sigma = 0.02$.



(b) Diagonal covariance matrix with $C_m = \boldsymbol{\sigma}^2 \boldsymbol{I}$.



(c) Full covariance matrix C_m.

Figure B.3: Inferred radiative terms, convective terms and posterior variances for the ten cases from Table B.1

C

Selected Models

C.1. Framework 1 - Standard k-ω Model

C.1.1. Periodic Hill

Models for b_{ij}^{Δ}:

$$
\begin{aligned}
M_{b_{ij}^{\Delta}} =& + 6.95898 T_{ij}^{(2)} \\
M_{b_{ij}^{\Delta}} =& + 0.25167 T_{ij}^{(1)} + 6.92681 T_{ij}^{(2)} + -8.68381 T_{ij}^{(9)} + -8.6851 I_1 T_{ij}^{(4)} - 0.33265 I_1 T_{ij}^{(9)} \\
& - 0.33265 I_1 I_2 T_{ij}^{(3)} + -0.33273 I_1 I_1 T_{ij}^{(4)} \\
M_{b_{ij}^{\Delta}} =& + 0.25271 T_{ij}^{(1)} + 6.92837 T_{ij}^{(2)} + -8.20327 T_{ij}^{(9)} + -8.20456 I_1 T_{ij}^{(4)} + -0.54308 I_1 T_{ij}^{(8)} \\
& - 0.31481 I_1 T_{ij}^{(9)} + -0.3148 I_1 I_2 T_{ij}^{(3)} + 0.00224 I_1 I_1 T_{ij}^{(1)} + -1.08618 I_1 I_1 T_{ij}^{(2)} \\
& - 0.31488 I_1 I_1 T_{ij}^{(4)} + -8.20318 I_2 T_{ij}^{(3)} + -1.00354 I_2 T_{ij}^{(6)} + -1.00242 I_2 I_2 T_{ij}^{(1)} \\
M_{b_{ij}^{\Delta}} =& + 0.44424 T_{ij}^{(1)} + 7.64381 T_{ij}^{(2)} + 1.67877 T_{ij}^{(3)} + -0.00739 T_{ij}^{(5)} + 6.02141 T_{ij}^{(6)} - 8.67492 T_{ij}^{(8)} \\
& - 5.04816 T_{ij}^{(9)} + 1.14252 I_1 T_{ij}^{(1)} + -17.35012 I_1 T_{ij}^{(2)} + -5.04803 I_1 T_{ij}^{(4)} + 0.08237 I_1 I_1 T_{ij}^{(1)} \\
& + 0.02441 I_1 I_1 T_{ij}^{(3)} + 5.98153 I_2 T_{ij}^{(1)} + -5.04813 I_2 T_{ij}^{(3)} + -0.0493 I_2 T_{ij}^{(6)} + 0.29104 I_2 T_{ij}^{(7)} \\
& - 0.04833 I_2 I_2 T_{ij}^{(1)} + -0.58208 I_2 I_2 T_{ij}^{(2)} + -0.17578 I_2 I_2 T_{ij}^{(4)} \\
M_{b_{ij}^{\Delta}} =& + 0.44328 T_{ij}^{(1)} + 7.966 T_{ij}^{(2)} + -2.52868 T_{ij}^{(3)} + -6.44656 T_{ij}^{(4)} + -0.01231 T_{ij}^{(5)} \\
& + 5.93728 T_{ij}^{(6)} + -4.89558 T_{ij}^{(7)} + -8.41777 T_{ij}^{(8)} + -2.41489 T_{ij}^{(9)} + 1.07082 I_1 T_{ij}^{(1)} \\
& - 16.83575 I_1 T_{ij}^{(2)} + 2.50246 I_1 T_{ij}^{(3)} + -2.41444 I_1 T_{ij}^{(4)} + 0.10516 I_1 T_{ij}^{(6)} + -0.2951 I_1 T_{ij}^{(7)} \\
& - 0.46934 I_1 T_{ij}^{(8)} + -0.12645 I_1 T_{ij}^{(9)} + 0.10438 I_1 I_2 T_{ij}^{(1)} + 0.59019 I_1 I_2 T_{ij}^{(2)} \\
& - 0.12645 I_1 I_2 T_{ij}^{(3)} + 0.07378 I_1 I_2 T_{ij}^{(4)} + 0.07813 I_1 I_1 T_{ij}^{(1)} + -0.93869 I_1 I_1 T_{ij}^{(2)} \\
& + 0.13216 I_1 I_1 T_{ij}^{(3)} - 0.12644 I_1 I_1 T_{ij}^{(4)} + 5.9027 I_2 T_{ij}^{(1)} + 9.79117 I_2 T_{ij}^{(2)} - 2.41492 I_2 T_{ij}^{(3)} \\
& + 1.08829 I_2 T_{ij}^{(4)} + -0.04082 I_2 T_{ij}^{(6)} + 0.25623 I_2 T_{ij}^{(7)} + 0.29509 I_2 T_{ij}^{(8)} + 0.0738 I_2 T_{ij}^{(9)} \\
& - 0.04004 I_2 I_2 T_{ij}^{(1)} + -0.51247 I_2 I_2 T_{ij}^{(2)} + 0.0738 I_2 I_2 T_{ij}^{(3)} + -0.03352 I_2 I_2 T_{ij}^{(4)} \\
& + 0.05219 I_5 T_{ij}^{(1)} + 0.29509 I_5 T_{ij}^{(2)}
\end{aligned}
$$

Models for b_{ij}^{R}:

$$
\begin{aligned}
M_{b_{ij}^{R}} &= +1.19914T_{ij}^{(1)} \\
M_{b_{ij}^{R}} &= +1.40585T_{ij}^{(1)} + -6.09832I_1T_{ij}^{(1)} + -0.2814I_2T_{ij}^{(6)} \\
M_{b_{ij}^{R}} &= +1.71729T_{ij}^{(1)} + 4.86737T_{ij}^{(6)} + -5.48I_1T_{ij}^{(1)} + 0.23333I_1I_2T_{ij}^{(1)} + -0.26264I_1I_1T_{ij}^{(1)} \\
&\quad + 4.86849I_2T_{ij}^{(1)} + -0.22973I_2T_{ij}^{(6)} + -0.22977I_2I_2T_{ij}^{(1)} + 0.11664I_5T_{ij}^{(1)} \\
M_{b_{ij}^{R}} &= +1.71744T_{ij}^{(1)} + 0.09945T_{ij}^{(4)} + 4.86649T_{ij}^{(6)} + -5.47924I_1T_{ij}^{(1)} + 0.23323I_1T_{ij}^{(6)} \\
&\quad + 0.23328I_1I_2T_{ij}^{(1)} + -0.26259I_1I_1T_{ij}^{(1)} + 4.86762I_2T_{ij}^{(1)} + -0.22967I_2T_{ij}^{(6)} \\
&\quad - 0.22972I_2I_2T_{ij}^{(1)} \\
M_{b_{ij}^{R}} &= +1.71749T_{ij}^{(1)} + -0.29509T_{ij}^{(3)} + 0.09891T_{ij}^{(4)} + 4.86487T_{ij}^{(6)} + -5.47767I_1T_{ij}^{(1)} \\
&\quad + 0.23312I_1T_{ij}^{(6)} + 0.23317I_1I_2T_{ij}^{(1)} - 0.26249I_1I_1T_{ij}^{(1)} + 4.86599I_2T_{ij}^{(1)} - 0.22956I_2T_{ij}^{(6)} \\
&\quad - 0.22961I_2I_2T_{ij}^{(1)} + 0.11656I_5T_{ij}^{(1)}
\end{aligned}
$$

C.1.2. Converging-Diverging Channel

Models for b_{ij}^{Δ}:

$$
\begin{aligned}
M_{b_{ij}^{\Delta}} &= +4.21876T_{ij}^{(2)} \\
M_{b_{ij}^{\Delta}} &= +4.22029T_{ij}^{(2)} + 2.51177T_{ij}^{(3)} + -2.30284T_{ij}^{(4)} \\
M_{b_{ij}^{\Delta}} &= +5.8025T_{ij}^{(2)} + 2.50522T_{ij}^{(3)} + -2.31291T_{ij}^{(4)} + -0.37375T_{ij}^{(5)} + -36.16018I_1T_{ij}^{(2)} \\
&\quad - 1.24737I_1T_{ij}^{(8)} + 0.11536I_1T_{ij}^{(9)} + 6.52162I_1I_1T_{ij}^{(1)} - 2.49637I_1I_1T_{ij}^{(2)} \\
&\quad - 0.14347I_1I_1T_{ij}^{(3)} + -0.20554I_1I_1T_{ij}^{(6)} + 1.69973I_2T_{ij}^{(7)} + 1.25783I_2T_{ij}^{(8)} \\
&\quad - 3.39946I_2I_2T_{ij}^{(2)} + 0.28318I_2I_2T_{ij}^{(4)} + -0.32012I_2I_2T_{ij}^{(6)} + 0.38367I_3T_{ij}^{(1)} \\
&\quad - 0.05155I_4T_{ij}^{(1)} + 1.25844I_5T_{ij}^{(2)} + 0.05789I_5T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= +6.44398T_{ij}^{(2)} + 2.50604T_{ij}^{(3)} + -2.31386T_{ij}^{(4)} + -0.32665T_{ij}^{(5)} + -17.75277T_{ij}^{(7)} \\
&\quad - 16.97685T_{ij}^{(8)} - 33.98599I_1T_{ij}^{(2)} - 4.28498I_1T_{ij}^{(6)} - 1.12079I_1T_{ij}^{(7)} - 1.17122I_1T_{ij}^{(8)} \\
&\quad + 2.24157I_1I_2T_{ij}^{(2)} + 6.29929I_1I_1T_{ij}^{(1)} + -2.34408I_1I_1T_{ij}^{(2)} + -0.1436I_1I_1T_{ij}^{(3)} \\
&\quad - 0.19591I_1I_1T_{ij}^{(6)} + 1.49811I_2T_{ij}^{(7)} + 1.11961I_2T_{ij}^{(8)} + -2.99622I_2I_2T_{ij}^{(2)} \\
&\quad + 0.28282I_2I_2T_{ij}^{(4)} - 0.31019I_2I_2T_{ij}^{(6)} + 0.3819I_3T_{ij}^{(1)} + -0.04959I_4T_{ij}^{(1)} + 1.1202I_5T_{ij}^{(2)} \\
M_{b_{ij}^{\Delta}} &= +0.04774T_{ij}^{(1)} + 7.25465T_{ij}^{(2)} + 2.47908T_{ij}^{(3)} + -2.51541T_{ij}^{(4)} + -12.70996T_{ij}^{(7)} \\
&\quad - 15.7871T_{ij}^{(8)} + 1.32336T_{ij}^{(9)} + 2.60258I_1T_{ij}^{(1)} + -31.60593I_1T_{ij}^{(2)} + 1.40518I_1T_{ij}^{(4)} \\
&\quad - 0.86675I_1T_{ij}^{(7)} - 1.08849I_1T_{ij}^{(8)} + 1.7335I_1I_2T_{ij}^{(2)} + 0.19508I_1I_1T_{ij}^{(1)} - 2.17851I_1I_1T_{ij}^{(2)} \\
&\quad + 25.41991I_2T_{ij}^{(2)} + 1.32854I_2T_{ij}^{(3)} + 1.07231I_2T_{ij}^{(7)} + 0.86577I_2T_{ij}^{(8)} - 2.14463I_2I_2T_{ij}^{(2)} \\
&\quad + 0.28088I_2I_2T_{ij}^{(4)} + -0.14138I_2I_2T_{ij}^{(6)} + 0.2869I_3T_{ij}^{(1)} + 0.86626I_5T_{ij}^{(2)}
\end{aligned}
$$

Models for b_{ij}^R:

$$
\begin{aligned}
M_{b_{ij}^R} &= +0.65327T_{ij}^{(1)} \\
M_{b_{ij}^R} &= +0.65403T_{ij}^{(1)} + -0.38525I_1I_1T_{ij}^{(1)} \\
M_{b_{ij}^R} &= +0.92018T_{ij}^{(1)} + -6.17505I_1T_{ij}^{(1)} + 0.35712I_1I_2T_{ij}^{(1)} + -0.3466I_1I_1T_{ij}^{(1)} \\
M_{b_{ij}^R} &= +1.34476T_{ij}^{(1)} + 5.33914T_{ij}^{(6)} + -5.19059I_1T_{ij}^{(1)} + 5.33913I_2T_{ij}^{(1)} + -0.29873I_2T_{ij}^{(6)} \\
&\quad - 0.29873I_2I_2T_{ij}^{(1)} + 0.14516I_5T_{ij}^{(1)} \\
M_{b_{ij}^R} &= +1.3459T_{ij}^{(1)} + 5.33469T_{ij}^{(6)} + -5.1864I_1T_{ij}^{(1)} + 0.29004I_1T_{ij}^{(6)} + 0.29004I_1I_2T_{ij}^{(1)} \\
&\quad - 0.28236I_1I_1T_{ij}^{(1)} + 5.33469I_2T_{ij}^{(1)} - 0.29842I_2T_{ij}^{(6)} - 0.29842I_2I_2T_{ij}^{(1)} + 0.14502I_5T_{ij}^{(1)}
\end{aligned}
$$

C.1.3. Curved Backward-Facing Step

Models for b_{ij}^{Δ}:

$$
\begin{aligned}
M_{b_{ij}^{\Delta}} &= +124.67831I_1T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= +10.82959T_{ij}^{(3)} + 3.98802I_1T_{ij}^{(3)} + 0.25672I_1I_1T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= +2.05925T_{ij}^{(2)} - 1.39788T_{ij}^{(5)} + 124.5461I_1T_{ij}^{(3)} + 0.04705I_1I_5T_{ij}^{(8)} - 9.79427I_1I_1T_{ij}^{(1)} \\
&\quad - 1.67015I_1I_1T_{ij}^{(2)} + 5.89337I_1I_1T_{ij}^{(3)} + -0.12909I_1I_1I_2T_{ij}^{(3)} + 0.24327I_1I_1l54.0T_{ij}^{(7)} \\
&\quad + 0.09899I_2T_{ij}^{(2)} - 5.65612I_2l54.0T_{ij}^{(8)} + 1.99256I_3T_{ij}^{(1)} + 0.05137I_3T_{ij}^{(3)} + 3.26606I_3T_{ij}^{(4)} \\
M_{b_{ij}^{\Delta}} &= +1.79842T_{ij}^{(2)} + 10.83082T_{ij}^{(3)} + 3.99853I_1T_{ij}^{(3)} + 0.49862I_1I_5T_{ij}^{(8)} + 0.25744I_1I_1T_{ij}^{(3)} \\
&\quad + 1.03903I_1I_1I_2T_{ij}^{(8)} + 1.08606I_1I_1I_5T_{ij}^{(2)} + 2.26625l13.0I_2T_{ij}^{(2)} + 11.60962I_5T_{ij}^{(8)} \\
M_{b_{ij}^{\Delta}} &= +2.06944T_{ij}^{(2)} + 10.83628T_{ij}^{(3)} + -1.36459T_{ij}^{(5)} + 0.00669T_{ij}^{(7)} + 3.98346I_1T_{ij}^{(3)} \\
&\quad + 0.04359I_1I_5T_{ij}^{(8)} + -10.06837I_1I_1T_{ij}^{(1)} + -1.66905I_1I_1T_{ij}^{(2)} + 0.25611I_1I_1T_{ij}^{(3)} \\
&\quad - 1.11934I_1I_1T_{ij}^{(7)} + 2.23884I_1I_1I_2T_{ij}^{(2)} + 0.07814I_1I_1I_2T_{ij}^{(3)} + 0.08271I_1I_1I_2T_{ij}^{(8)} \\
&\quad + 0.0369I_1I_1I_2I_5T_{ij}^{(3)} + 0.07691I_1I_1I_5T_{ij}^{(2)} - 0.10205I_1I_1l54.0T_{ij}^{(7)} + 0.14224l13.0I_2T_{ij}^{(2)} \\
&\quad + 0.10681I_2T_{ij}^{(2)} + 1.73698I_2l54.0T_{ij}^{(8)} + 0.21912I_2I_2l54.0T_{ij}^{(8)} + -1.89351I_3T_{ij}^{(1)} \\
&\quad + 0.03354I_3T_{ij}^{(3)} + 2.80618I_3T_{ij}^{(4)}
\end{aligned}
$$

Models for b_{ij}^R:

$$
\begin{aligned}
M_{b_{ij}^R} &= +0.9289T_{ij}^{(1)} \\
M_{b_{ij}^R} &= +0.92892T_{ij}^{(1)} + -0.00073T_{ij}^{(3)} + -0.01203I_1I_1T_{ij}^{(1)} \\
M_{b_{ij}^R} &= +0.93499T_{ij}^{(1)} + -0.00072T_{ij}^{(3)} - 0.15838I_1T_{ij}^{(1)} + 0.01197I_1I_2T_{ij}^{(1)} - 0.01198I_1I_1T_{ij}^{(1)} \\
M_{b_{ij}^R} &= +0.94096T_{ij}^{(1)} + -0.00071T_{ij}^{(3)} + 0.15694T_{ij}^{(6)} + -0.15759I_1T_{ij}^{(1)} + -0.01194I_2T_{ij}^{(6)} \\
&\quad - 0.01194I_2I_2T_{ij}^{(1)} \\
M_{b_{ij}^R} &= +0.94693T_{ij}^{(1)} + -0.00071T_{ij}^{(3)} + 0.15613T_{ij}^{(6)} + -0.1568I_1T_{ij}^{(1)} + 0.01187I_1T_{ij}^{(6)} \\
&\quad + 0.01187I_1I_2T_{ij}^{(1)} + -0.01189I_1I_1T_{ij}^{(1)} + 0.1561I_2T_{ij}^{(1)} + -0.01189I_2T_{ij}^{(6)} \\
&\quad - 0.01189I_2I_2T_{ij}^{(1)}
\end{aligned}
$$

C.2. Framework 2 - Elliptic Blending Lag Parameter Model

C.2.1. Periodic Hill

Models for b_{ij}^{Δ}:

$$
\begin{aligned}
M_{b_{ij}^{\Delta}} =& 10.12875 \cdot T_2 \\
M_{b_{ij}^{\Delta}} =& 0.52635 T_{ij}^{(1)} + 10.06218 T_{ij}^{(2)} - 6.65905 T_{ij}^{(9)} - 6.6613 I_1 T_{ij}^{(4)} \\
M_{b_{ij}^{\Delta}} =& 0.52639 T_{ij}^{(1)} + 10.06415 T_{ij}^{(2)} + 2.57092 T_{ij}^{(3)} - 4.05816 T_{ij}^{(9)} - 4.06047 I_1 T_{ij}^{(4)} \\
& - 0.22952 I_1 T_{ij}^{(8)} + 0.53856 I_1^2 T_{ij}^{(1)} - 0.45906 I_1^2 T_{ij}^{(2)} + 0.00622 I_1^2 T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} =& 0.62733 T_{ij}^{(1)} + 10.36939 T_{ij}^{(2)} + 2.49994 T_{ij}^{(3)} + 3.47917 T_{ij}^{(6)} - 4.34745 T_{ij}^{(8)} - 4.00827 T_{ij}^{(9)} \\
& - 8.69553 I_1 T_{ij}^{(2)} - 4.0099 I_1 T_{ij}^{(4)} + 0.676 I_1^2 T_{ij}^{(1)} + 0.0059 I_1^2 T_{ij}^{(3)} + 3.44735 I_2 T_{ij}^{(1)} \\
& - 4.00825 I_2 T_{ij}^{(3)} + 0.24724 I_2 T_{ij}^{(6)} + 0.24795 I_2^2 T_{ij}^{(1)} \\
M_{b_{ij}^{\Delta}} =& + 0.55506 T_{ij}^{(1)} + 10.54569 T_{ij}^{(2)} - 3.06951 T_{ij}^{(3)} - 8.60453 T_{ij}^{(4)} + 0.03653 T_{ij}^{(5)} + 3.93653 T_{ij}^{(6)} \\
& - 2.92815 T_{ij}^{(7)} - 4.1595 T_{ij}^{(8)} - 1.72244 T_{ij}^{(9)} + 3.53055 I_1 T_{ij}^{(1)} - 8.31959 I_1 T_{ij}^{(2)} \\
& + 1.57336 I_1 T_{ij}^{(3)} - 1.72392 I_1 T_{ij}^{(4)} - 0.14723 I_1 T_{ij}^{(7)} - 0.1983 I_1 T_{ij}^{(8)} + 0.29447 I_1 I_2 T_{ij}^{(2)} \\
& + 0.35542 I_1^2 T_{ij}^{(1)} - 0.39661 I_1^2 T_{ij}^{(2)} + 0.07269 I_1^2 T_{ij}^{(3)} + 3.91245 I_2 T_{ij}^{(1)} + 5.8563 I_2 T_{ij}^{(2)} \\
& - 1.72248 I_2 T_{ij}^{(3)} + 1.24944 I_2 T_{ij}^{(4)} + 0.20289 I_2 T_{ij}^{(6)} + 0.14722 I_2 T_{ij}^{(8)} + 0.20333 I_2^2 T_{ij}^{(1)} \\
& - 0.0874 I_5 T_{ij}^{(1)} + 0.14723 I_5 T_{ij}^{(2)}
\end{aligned}
$$

Models for b_{ij}^{R}:

$$
\begin{aligned}
M_{b_{ij}^{R}} =& + 1.17598 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} =& + 1.5601 T_{ij}^{(1)} - 0.23889 T_{ij}^{(3)} - 5.40077 I_1 T_{ij}^{(1)} + 5.29245 I_2 T_{ij}^{(1)} - 0.26492 I_2 T_{ij}^{(6)} \\
M_{b_{ij}^{R}} =& + 1.70093 T_{ij}^{(1)} - 0.22426 T_{ij}^{(3)} + 4.77857 T_{ij}^{(6)} - 5.00873 I_1 T_{ij}^{(1)} + 4.77941 I_2 T_{ij}^{(1)} \\
& - 0.23294 I_2 T_{ij}^{(6)} \\
M_{b_{ij}^{R}} =& + 1.56112 T_{ij}^{(1)} - 0.23857 T_{ij}^{(3)} + 0.07992 T_{ij}^{(4)} + 5.28526 T_{ij}^{(6)} - 5.39526 I_1 T_{ij}^{(1)} \\
& + 0.25927 I_1 T_{ij}^{(6)} + 0.2593 I_1 I_2 T_{ij}^{(1)} - 0.26623 I_1^2 T_{ij}^{(1)} - 0.26449 I_2 T_{ij}^{(6)} - 0.26453 I_2^2 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} =& + 1.70176 T_{ij}^{(1)} - 0.22393 T_{ij}^{(3)} + 0.07514 T_{ij}^{(4)} + 4.77307 T_{ij}^{(6)} - 5.00389 I_1 T_{ij}^{(1)} \\
& + 0.22953 I_1 T_{ij}^{(6)} + 0.22956 I_1 I_2 T_{ij}^{(1)} - 0.24174 I_1^2 T_{ij}^{(1)} + 4.77391 I_2 T_{ij}^{(1)} - 0.23257 I_2 T_{ij}^{(6)} \\
& - 0.2326 I_2^2 T_{ij}^{(1)} + 0.11476 I_5 T_{ij}^{(1)}
\end{aligned}
$$

C.2.2. Converging-Diverging Channel

Models for b_{ij}^{Δ}:

$$
\begin{aligned}
M_{b_{ij}^{\Delta}} =& 4.55351 T_{ij}^{(2)} \\
M_{b_{ij}^{\Delta}} =& 4.55382 T_{ij}^{(2)} + 2.57465 T_{ij}^{(3)} - 2.66955 T_{ij}^{(4)} \\
M_{b_{ij}^{\Delta}} =& 0.26671 T_{ij}^{(1)} + 4.5759 T_{ij}^{(2)} + 2.48374 T_{ij}^{(3)} - 2.76012 T_{ij}^{(4)} - 0.06904 T_{ij}^{(5)} - 1.3254 I_1 T_{ij}^{(7)} \\
& - 1.40415 I_1 T_{ij}^{(8)} + 0.11784 I_1 T_{ij}^{(9)} - 2.81038 I_1^2 T_{ij}^{(2)} - 0.14944 I_1^2 T_{ij}^{(3)} - 0.02214 I_1^2 T_{ij}^{(6)} \\
& + 1.62099 I_2 T_{ij}^{(7)} + 1.32401 I_2 T_{ij}^{(8)} - 3.24199 I_2^2 T_{ij}^{(2)} + 0.27434 I_2^2 T_{ij}^{(4)} - 0.14765 I_2^2 T_{ij}^{(6)} \\
& + 0.28756 I_3 T_{ij}^{(1)} + 1.3247 I_5 T_{ij}^{(2)} \\
M_{b_{ij}^{\Delta}} =& 0.31576 T_{ij}^{(1)} + 6.62727 T_{ij}^{(2)} + 2.48105 T_{ij}^{(3)} - 2.76724 T_{ij}^{(4)} + 0.08565 T_{ij}^{(5)} - 15.83194 T_{ij}^{(7)} \\
& - 16.44562 T_{ij}^{(8)} - 1.24742 I_1 T_{ij}^{(1)} - 32.93722 I_1 T_{ij}^{(2)} - 1.01707 I_1 T_{ij}^{(7)} - 1.11562 I_1 T_{ij}^{(8)} \\
& + 2.03414 I_1 I_2 T_{ij}^{(2)} - 2.23335 I_1^2 T_{ij}^{(2)} - 0.14991 I_1^2 T_{ij}^{(3)} - 0.02709 I_1^2 T_{ij}^{(6)} + 1.98223 I_2 T_{ij}^{(6)} \\
& + 1.2791 I_2 T_{ij}^{(7)} + 1.01572 I_2 T_{ij}^{(8)} - 2.5582 I_2^2 T_{ij}^{(2)} + 0.27407 I_2^2 T_{ij}^{(4)} - 0.15057 I_2^2 T_{ij}^{(6)} \\
& + 0.28234 I_3 T_{ij}^{(1)} - 0.05463 I_4 T_{ij}^{(1)} + 1.0164 I_5 T_{ij}^{(2)} \\
M_{b_{ij}^{\Delta}} =& 0.25944 T_{ij}^{(1)} + 7.3372 T_{ij}^{(2)} + 2.58012 T_{ij}^{(3)} - 3.03298 T_{ij}^{(4)} + 0.01566 T_{ij}^{(5)} - 1.30781 T_{ij}^{(6)} \\
& - 11.56889 T_{ij}^{(7)} - 15.27834 T_{ij}^{(8)} + 1.33998 T_{ij}^{(9)} - 1.788 I_1 T_{ij}^{(1)} - 30.6007 I_1 T_{ij}^{(2)} \\
& - 1.85445 I_1 T_{ij}^{(3)} + 1.43566 I_1 T_{ij}^{(4)} + 0.1527 I_1 T_{ij}^{(6)} - 0.79727 I_1 T_{ij}^{(7)} - 1.03581 I_1 T_{ij}^{(8)} \\
& + 0.11001 I_1 T_{ij}^{(9)} + 0.14627 I_1 I_2 T_{ij}^{(1)} + 1.59455 I_1 I_2 T_{ij}^{(2)} + 0.11 I_1 I_2 T_{ij}^{(3)} - 0.15082 I_1 I_2 T_{ij}^{(4)} \\
& + 0.0227 I_1 I_2 T_{ij}^{(6)} + 0.02271 I_1 I_2^2 T_{ij}^{(1)} - 0.14886 I_1^2 T_{ij}^{(1)} - 2.07357 I_1^2 T_{ij}^{(2)} - 0.14161 I_1^2 T_{ij}^{(3)} \\
& + 0.1138 I_1^2 T_{ij}^{(4)} + 0.01206 I_1^2 T_{ij}^{(6)} + 0.01188 I_1^2 I_2 T_{ij}^{(1)} - 1.59973 I_2 T_{ij}^{(1)} + 23.13778 I_2 T_{ij}^{(2)} \\
& + 1.34249 I_2 T_{ij}^{(3)} - 2.53178 I_2 T_{ij}^{(4)} + 0.88954 I_2 T_{ij}^{(6)} + 0.93586 I_2 T_{ij}^{(7)} + 0.79611 I_2 T_{ij}^{(8)} \\
& - 0.14848 I_2 T_{ij}^{(9)} + 0.89334 I_2^2 T_{ij}^{(1)} - 1.87172 I_2^2 T_{ij}^{(2)} - 0.14831 I_2^2 T_{ij}^{(3)} + 0.25731 I_2^2 T_{ij}^{(4)} \\
& - 0.08833 I_2^2 T_{ij}^{(6)} + 0.33182 I_3 T_{ij}^{(1)} - 0.00587 I_3 T_{ij}^{(2)} - 0.05036 I_4 T_{ij}^{(1)} + 0.0751 I_5 T_{ij}^{(1)} \\
& + 0.79669 I_5 T_{ij}^{(2)} + 0.05504 I_5 T_{ij}^{(3)} - 0.07536 I_5 T_{ij}^{(4)}
\end{aligned}
$$

Models for b_{ij}^{R}:

$$
\begin{aligned}
M_{b_{ij}^{R}} =& 0.71476 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} =& 0.95975 T_{ij}^{(1)} - 5.7521 I_1 T_{ij}^{(1)} + 0.33623 I_1 T_{ij}^{(6)} + 0.33623 I_1 I_2 T_{ij}^{(1)} - 0.32471 I_1^2 T_{ij}^{(1)} \\
& - 0.34834 I_2 T_{ij}^{(6)} - 0.34834 I_2^2 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} =& 1.16672 T_{ij}^{(1)} + 5.41532 T_{ij}^{(6)} - 5.20379 I_1 T_{ij}^{(1)} - 0.30963 I_2 T_{ij}^{(6)} - 0.30963 I_2^2 T_{ij}^{(1)} \\
& + 0.14949 I_5 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} =& 1.33829 T_{ij}^{(1)} + 4.93208 T_{ij}^{(6)} - 4.74725 I_1 T_{ij}^{(1)} - 0.25903 I_1^2 T_{ij}^{(1)} + 4.93207 I_2 T_{ij}^{(1)} \\
& - 0.2774 I_2 T_{ij}^{(6)} - 0.2774 I_2^2 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} =& 1.33901 T_{ij}^{(1)} + 4.92864 T_{ij}^{(6)} - 4.74399 I_1 T_{ij}^{(1)} + 0.26773 I_1 T_{ij}^{(6)} + 0.26773 I_1 I_2 T_{ij}^{(1)} \\
& - 0.25881 I_1^2 T_{ij}^{(1)} + 4.92863 I_2 T_{ij}^{(1)} - 0.27717 I_2 T_{ij}^{(6)} - 0.27717 I_2^2 T_{ij}^{(1)} + 0.13386 I_5 T_{ij}^{(1)}
\end{aligned}
$$

C.2.3. Curved Backward-Facing Step

Models for b_{ij}^{Δ}:

$$
\begin{aligned}
M_{b_{ij}^{\Delta}} =& + 3.61771 T_{ij}^{(2)} \\
M_{b_{ij}^{\Delta}} =& + 3.62135 T_{ij}^{(2)} + 3.97027 T_{ij}^{(3)} + 4.56464 I_1 T_{ij}^{(1)} \\
M_{b_{ij}^{\Delta}} =& + 4.89833 T_{ij}^{(2)} + 15.25939 T_{ij}^{(3)} + 11.09354 T_{ij}^{(4)} + -0.35979 T_{ij}^{(5)} + -11.25916 T_{ij}^{(7)} \\
& + 3.06022 I_1 T_{ij}^{(1)} + -4.67446 I_1 T_{ij}^{(3)} + -0.62571 I_1 T_{ij}^{(8)} + -1.15903 I_2 T_{ij}^{(1)} + 22.51906 I_2 T_{ij}^{(2)} \\
& + 0.83324 I_2 T_{ij}^{(7)} + -1.66655 I_2 I_2 T_{ij}^{(2)} + -0.42844 I_2 I_2 T_{ij}^{(4)} + -0.00573 I_3 T_{ij}^{(1)} \\
M_{b_{ij}^{\Delta}} =& + 5.07454 T_{ij}^{(2)} + 15.32806 T_{ij}^{(3)} + 11.24558 T_{ij}^{(4)} + -0.36293 T_{ij}^{(5)} + -10.99617 T_{ij}^{(7)} \\
& - 9.5395 T_{ij}^{(8)} + 3.03859 I_1 T_{ij}^{(1)} + -4.69373 I_1 T_{ij}^{(3)} + -0.60254 I_1 T_{ij}^{(8)} + -1.17477 I_2 T_{ij}^{(1)} \\
& + 21.99308 I_2 T_{ij}^{(2)} + 1.85897 I_2 T_{ij}^{(4)} + 0.81904 I_2 T_{ij}^{(7)} + -1.63816 I_2 I_2 T_{ij}^{(2)} \\
& - 0.42096 I_2 I_2 T_{ij}^{(4)} + -0.0091 I_3 T_{ij}^{(1)} \\
M_{b_{ij}^{\Delta}} =& + 5.66993 T_{ij}^{(2)} + 15.33432 T_{ij}^{(3)} + 11.25222 T_{ij}^{(4)} + -0.38527 T_{ij}^{(5)} + -10.12681 T_{ij}^{(7)} \\
& - 8.40056 T_{ij}^{(8)} + 3.02245 I_1 T_{ij}^{(1)} + -16.74062 I_1 T_{ij}^{(2)} + -4.69201 I_1 T_{ij}^{(3)} + -1.17641 I_2 T_{ij}^{(1)} \\
& + 20.25436 I_2 T_{ij}^{(2)} + 1.85837 I_2 T_{ij}^{(4)} + -0.00923 I_3 T_{ij}^{(1)}
\end{aligned}
$$

Models for b_{ij}^{R}:

$$
\begin{aligned}
M_{b_{ij}^{R}} =& + 1.00552 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} =& + 1.00556 T_{ij}^{(1)} + -0.01205 I_2 T_{ij}^{(6)} + -0.01205 I_2 I_2 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} =& + 1.0056 T_{ij}^{(1)} + 0.01197 I_1 T_{ij}^{(6)} + 0.01196 I_1 I_2 T_{ij}^{(1)} + -0.01205 I_2 T_{ij}^{(6)} + -0.01205 I_2 I_2 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} =& + 1.01701 T_{ij}^{(1)} + 0.00076 T_{ij}^{(3)} + 0.15453 T_{ij}^{(6)} + -0.15265 I_1 T_{ij}^{(1)} + 0.01186 I_1 T_{ij}^{(6)} \\
& - 0.01177 I_1 I_1 T_{ij}^{(1)} + -0.01194 I_2 T_{ij}^{(6)} \\
M_{b_{ij}^{R}} =& + 1.02267 T_{ij}^{(1)} + 0.00076 T_{ij}^{(3)} + 0.15368 T_{ij}^{(6)} + -0.15181 I_1 T_{ij}^{(1)} + 0.01181 I_1 T_{ij}^{(6)} \\
& + 0.01181 I_1 I_2 T_{ij}^{(1)} + -0.01172 I_1 I_1 T_{ij}^{(1)} + 0.15364 I_2 T_{ij}^{(1)}
\end{aligned}
$$

D

Visualization of Discovered Models

This chapter visualizes the discovered models of both frameworks for the converging-diverging channel and the curved backward-facing step. First, Appendix D.1 shows the discovered models of framework 1. Then, the discovered models for framework 2 are presented in Appendix D.2.

D.1. Framework 1 - Standard k-ω Model

Discovered models for b_{ij}^{Δ} for the converging-diverging channel.

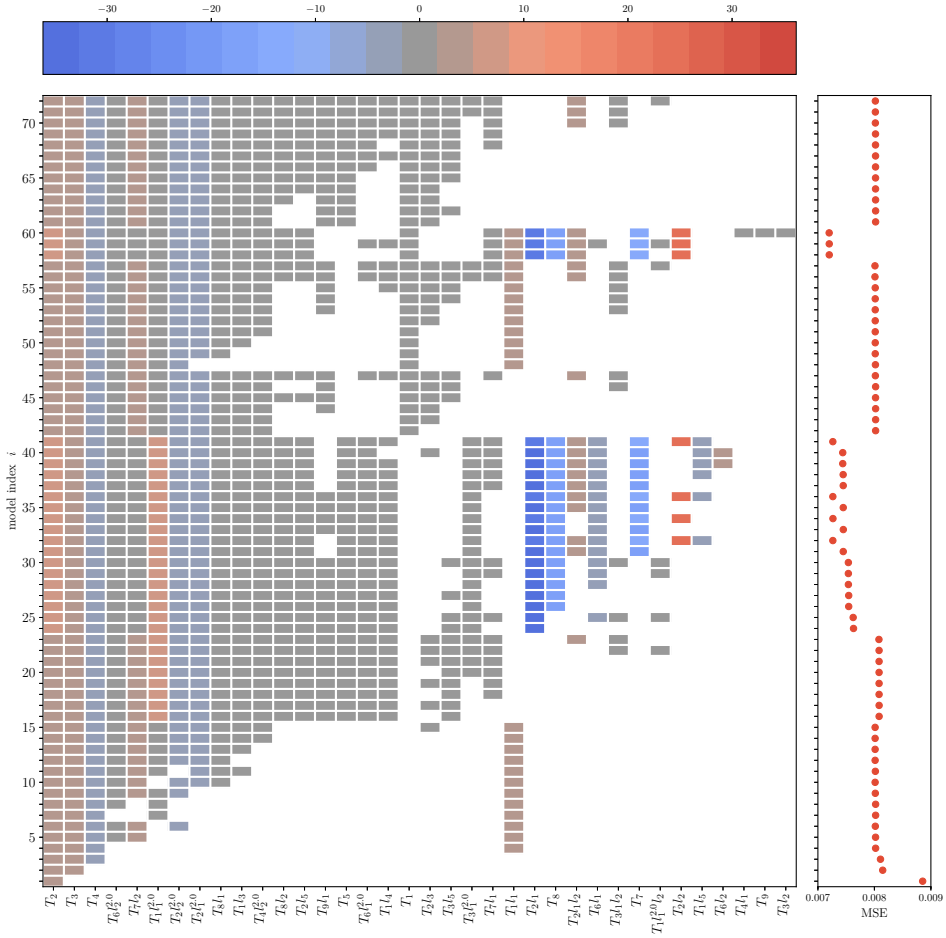


Figure D.1: Visualization of model-structure and mean-squared error of discovered models for b_{ij}^{Δ}.

Discovered models for b_{ij}^R for the converging-diverging channel.

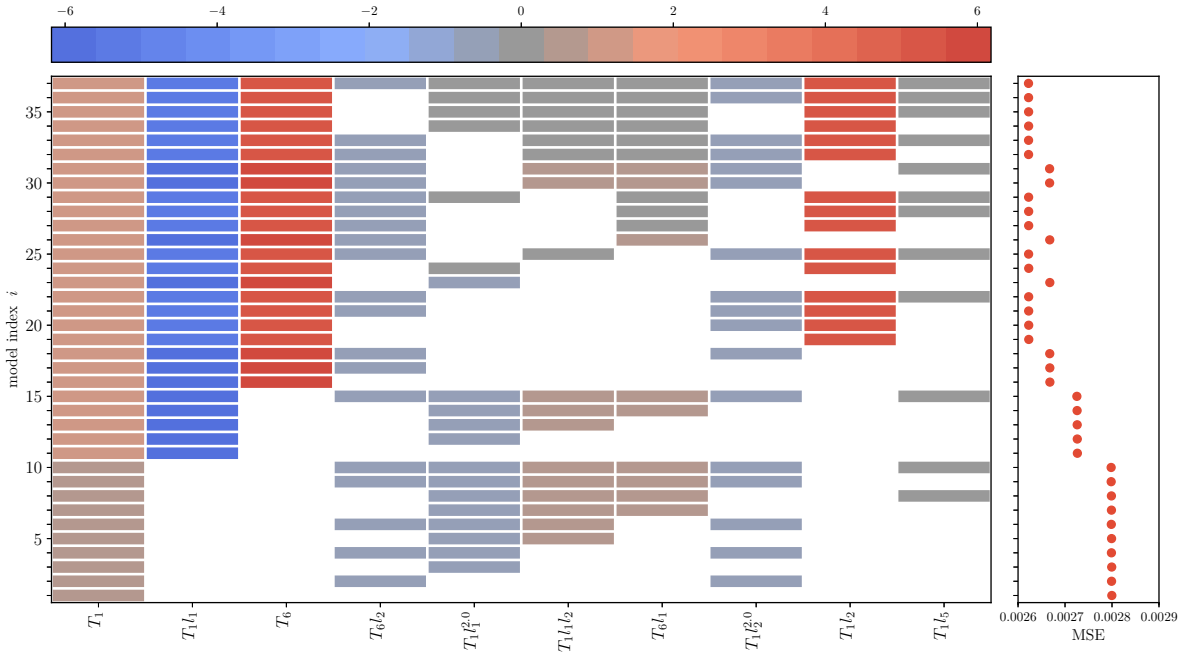


Figure D.2: Visualization of model-structure and mean-squared error of discovered models for b_{ij}^R.

Discovered models for b_{ij}^{Δ} for the curved backward-facing step.

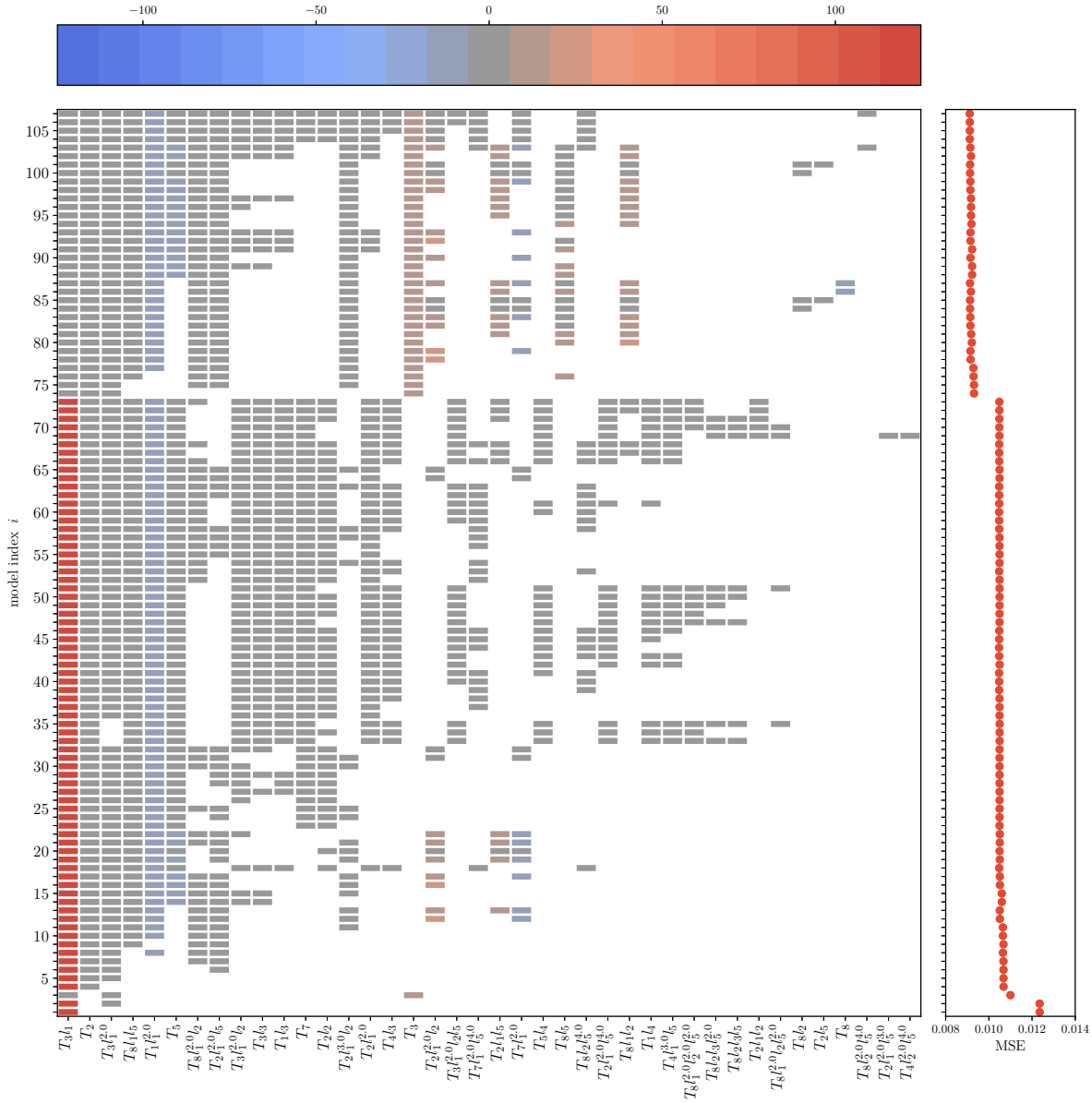


Figure D.3: Visualization of model-structure and mean-squared error of discovered models for b_{ij}^{Δ}.

Discovered models for b_{ij}^R for the curved backward-facing step.

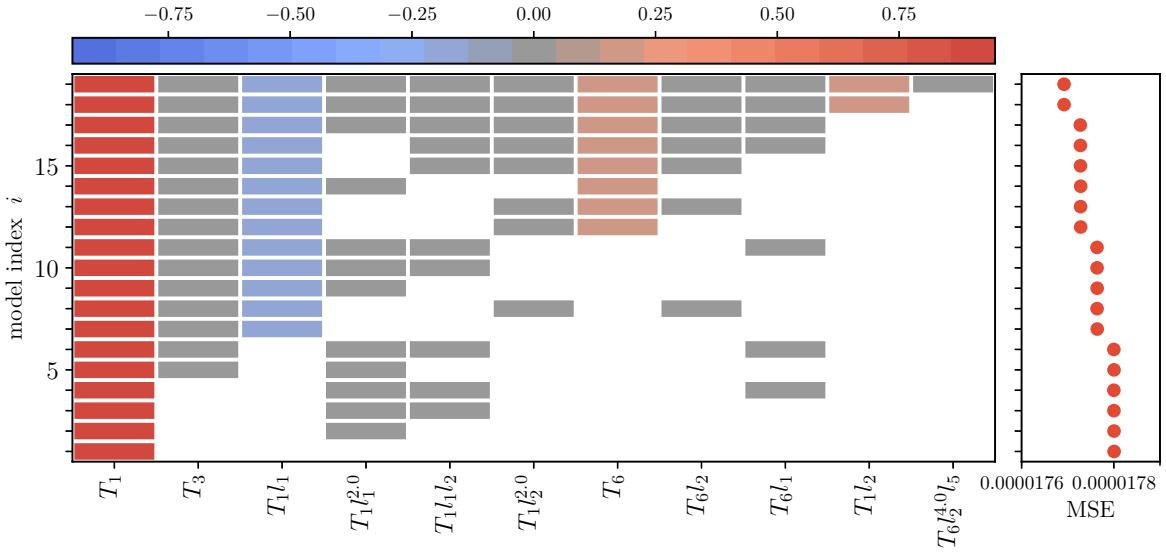


Figure D.4: Visualization of model-structure and mean-squared error of discovered models for b_{ij}^R.

D.2. Framework 2 - Elliptic Blending Lag Parameter Model

Discovered models for b_{ij}^{Δ} for the converging-diverging channel.

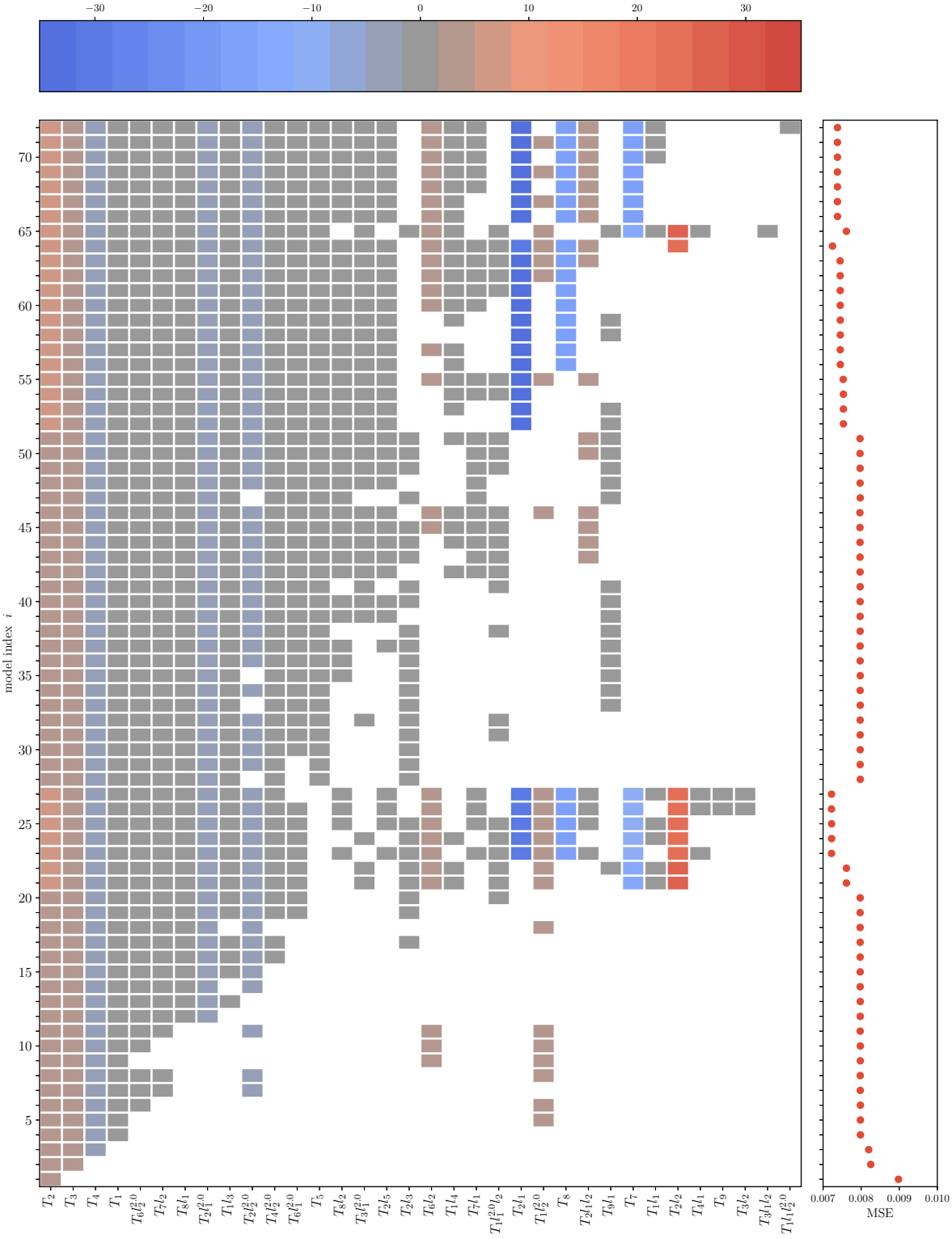


Figure D.5: Visualization of model-structure and mean-squared error of discovered models for b_{ij}^{Δ}.

Discovered models for b_{ij}^R for the converging-diverging channel.

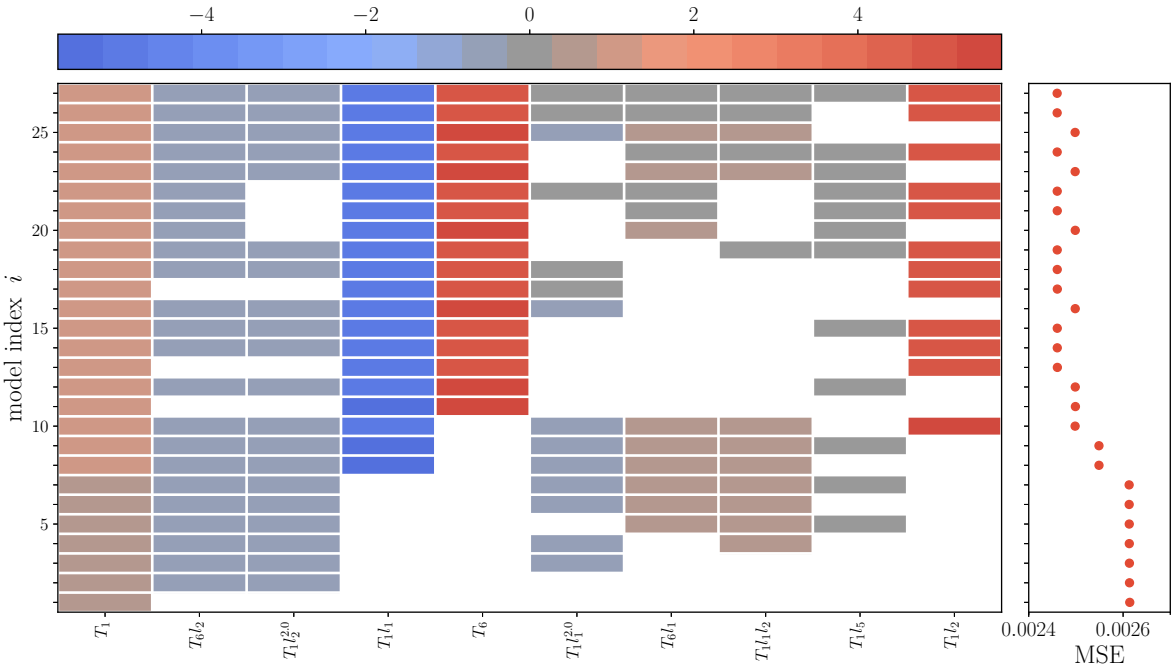


Figure D.6: Visualization of model-structure and mean-squared error of discovered models for b_{ij}^R.

Discovered models for b_{ij}^{Δ} for the curved backward-facing step.

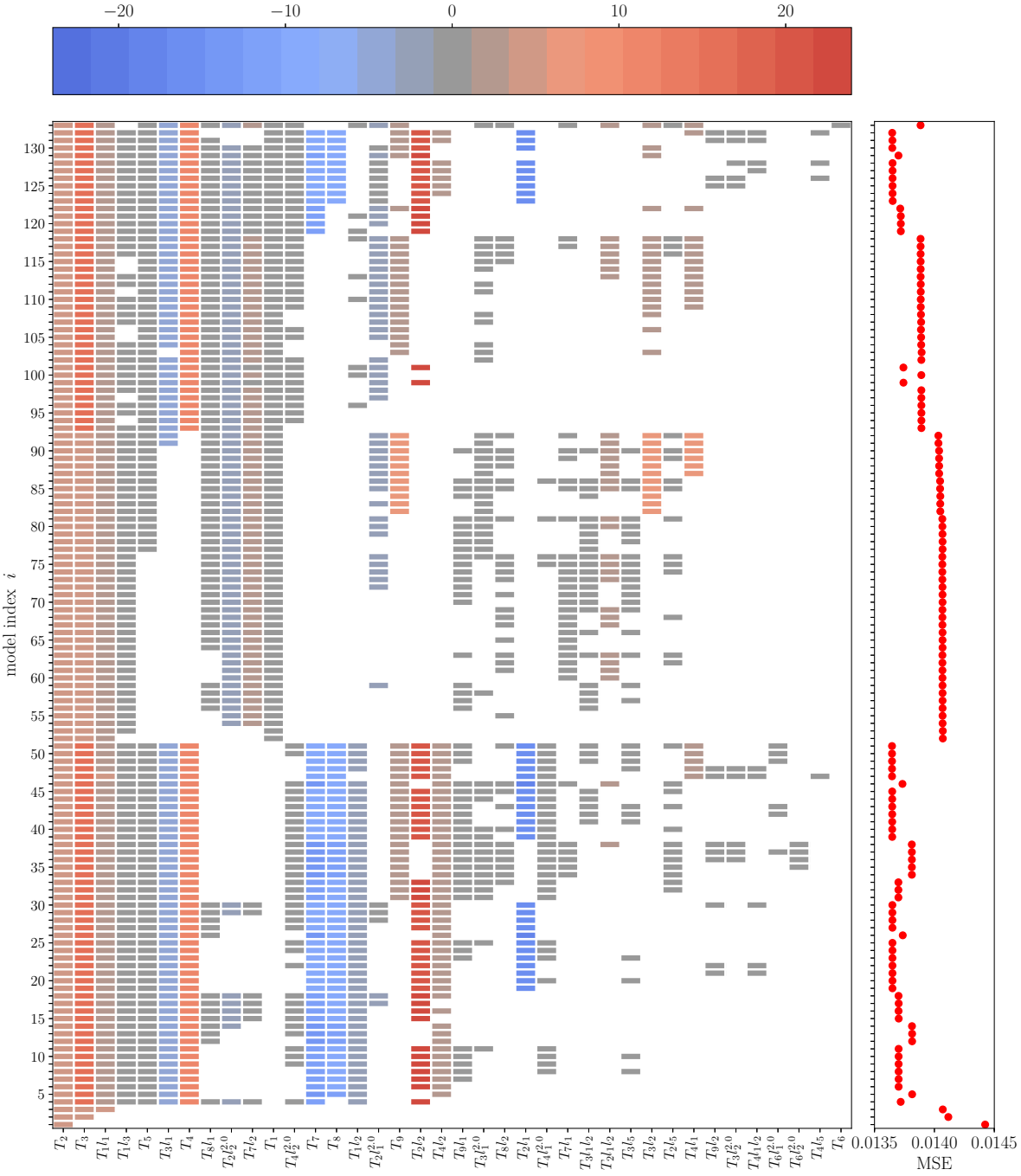


Figure D.7: Visualization of model-structure and mean-squared error of discovered models for b_{ij}^{Δ}.

Discovered models for b_{ij}^R for the curved backward-facing step.

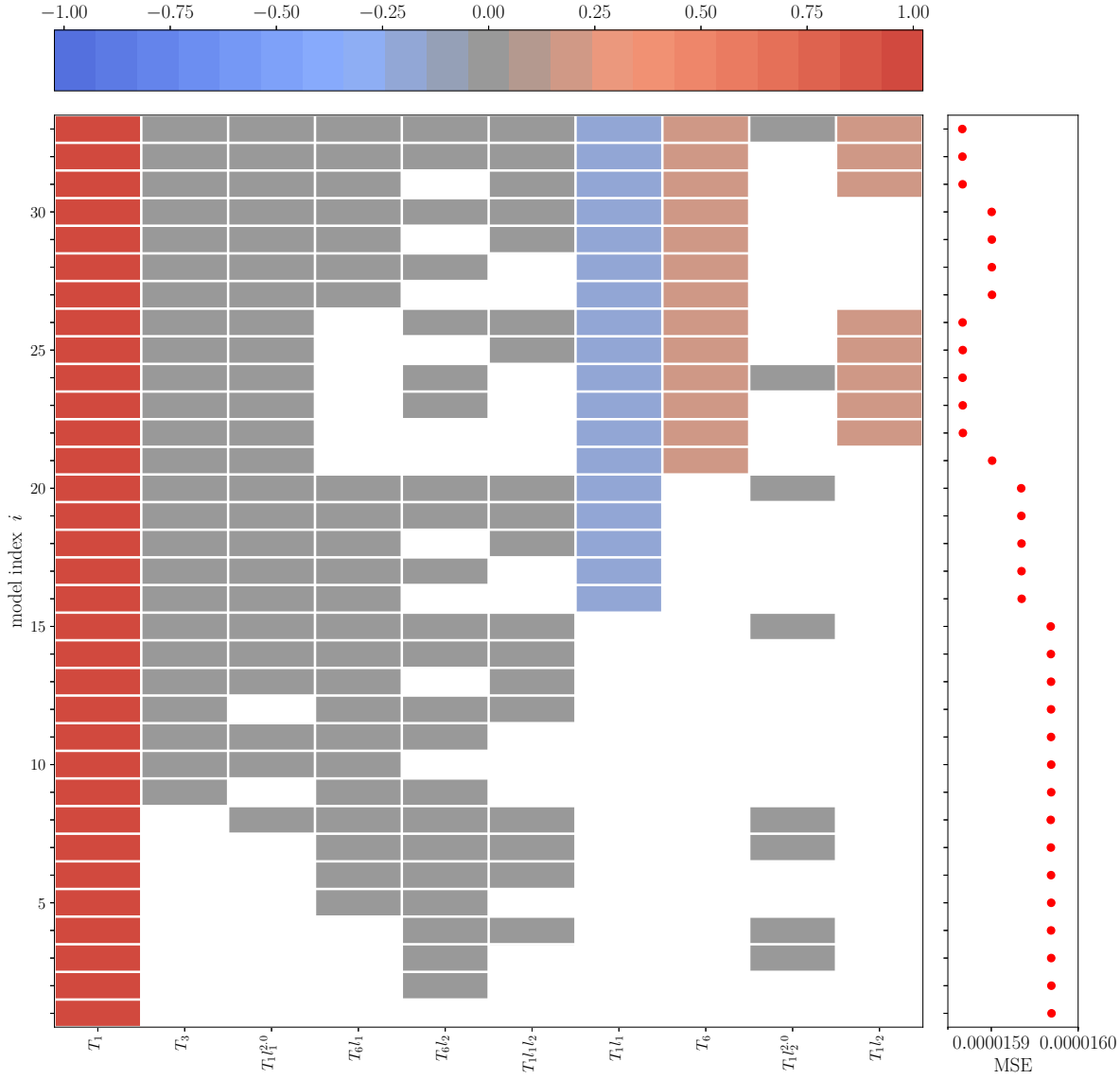


Figure D.8: Visualization of model-structure and mean-squared error of discovered models for b_{ij}^R.

E

Reduced Integrity Basis - Discovery of Models

Visualizations of the discovered models using a reduced integrity basis for the construction of the functionals of the corrective terms are shown in this appendix for the periodic hill, converging-diverging channel and curved backward-facing step.

E.1. Periodic Hill

Discovered models for b_{ij}^{Δ} for the periodic hill.

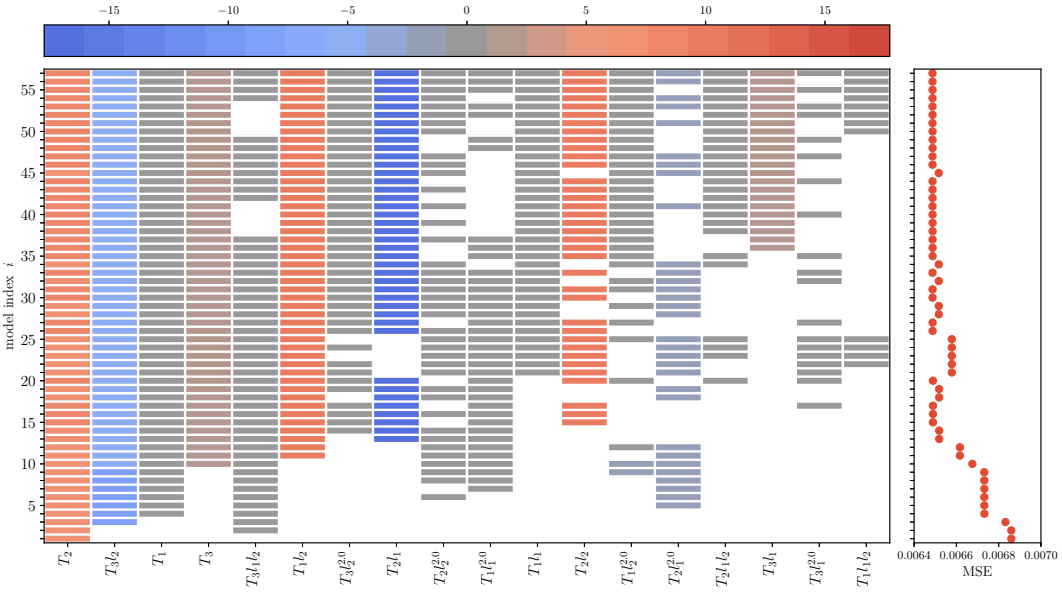


Figure E.1: Models for b_{ij}^{Δ}

Discovered models for b_{ij}^R for the periodic hill.

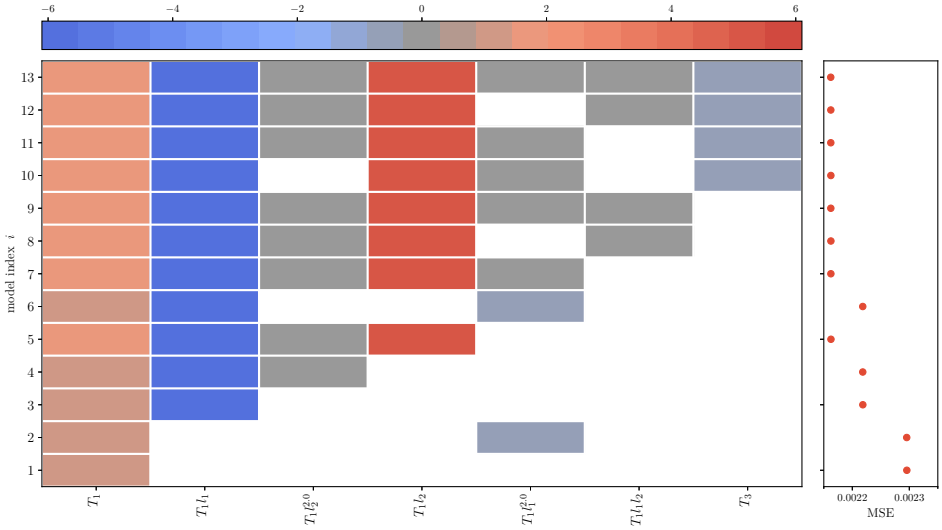


Figure E.2: Models for b_{ij}^R

The handpicked models for b_{ij}^Δ that are used for cross-validation are given by

$$
\begin{aligned}
M_{b_{ij}^\Delta} =& 6.95898 T_{ij}^{(2)} \\
M_{b_{ij}^\Delta} =& +0.25165 T_{ij}^{(1)} + 6.92702 T_{ij}^{(2)} + \left(-0.35278 I_1 I_2 - 9.22585 I_2\right) T_{ij}^{(3)} \\
M_{b_{ij}^\Delta} =& \left(+0.43473 + 0.19091 I_1^2 + 9.70161 I_2\right) T_{ij}^{(1)} + \left(7.51305 - 17.71277 I_1 - 0.59692 I_2^2\right) T_{ij}^{(2)} \\
& + \left(1.89102 - 0.20828 I_1 I_2 + -5.22528 I_2\right) T_{ij}^{(3)} \\
M_{b_{ij}^\Delta} =& \left(0.43281 + 0.18341 I_1^2 + 9.59916 I_2\right) T_{ij}^{(1)} + \left(7.78301 - 17.28072 I_1 + 10.23217 I_2\right) T_{ij}^{(2)} \\
& + \left(1.89125 - 0.20828 I_1 I_2 + -5.22565 I_2 + 0.19459 I_2^2\right) T_{ij}^{(3)} \\
M_{b_{ij}^\Delta} =& \left(0.41372 + 0.73374 I_1 + 0.24649 I_1 I_2 + 0.08832 I_1^2 + 9.72807 I_2 - 0.20618 I_2^2\right) T_{ij}^{(1)} \\
& + \left(7.78426 - 17.28148 I_1 + 0.61479 I_1 I_2 - 0.96481 I_1^2 + 10.21625 I_2 - 0.53598 I_2^2\right) T_{ij}^{(2)} \\
& + \left(1.85382 + 1.05331 I_1 - 0.20802 I_1 I_2 + 0.02296 I_1^2 - 5.2245 I_2 + 0.19449 I_2^2\right) T_{ij}^{(3)}
\end{aligned}
$$

The models for b_{ij}^R are given by

$$
\begin{aligned}
M_{b_{ij}^R} =& +1.19914 T_{ij}^{(1)} \\
M_{b_{ij}^R} =& +\left(1.40553 - 6.09944 I_1\right) T_{ij}^{(1)} \\
M_{b_{ij}^R} =& +\left(1.57506 - 5.76062 I_1 + 0.25543 I_1 I_2 - 0.27876 I_1^2 + 5.28283 I_2 - 0.25327 I_2^2\right) T_{ij}^{(1)} \\
& - 0.32 T_{ij}^{(3)}
\end{aligned}
$$

E.2. Converging-Diverging Channel

Discovered models for b_{ij}^{Δ} for the converging-diverging channel.

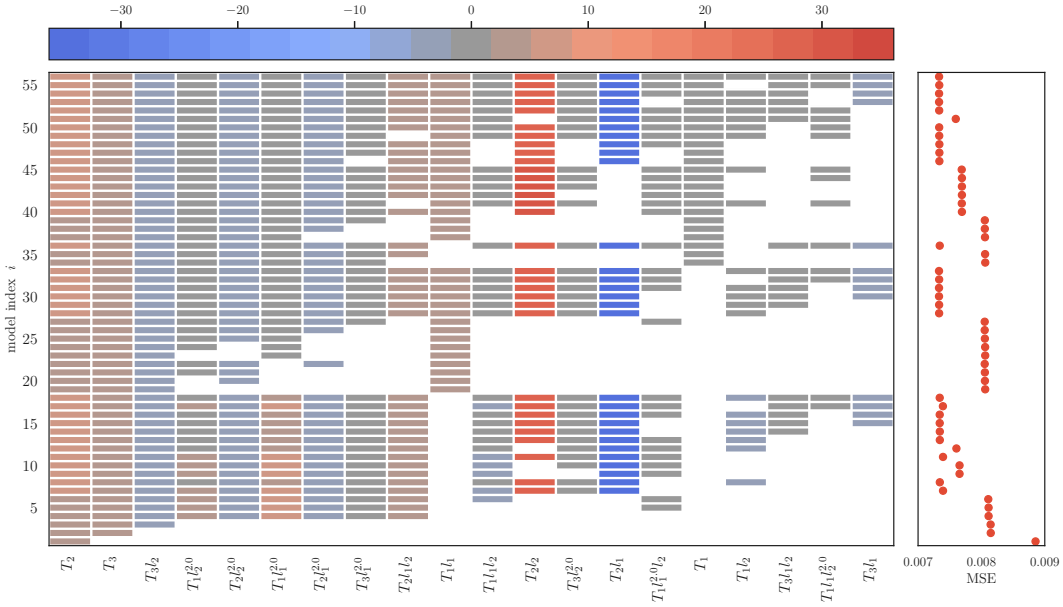


Figure E.3: Models for b_{ij}^{Δ}

Discovered models for b_{ij}^{R} for the converging-diverging channel.

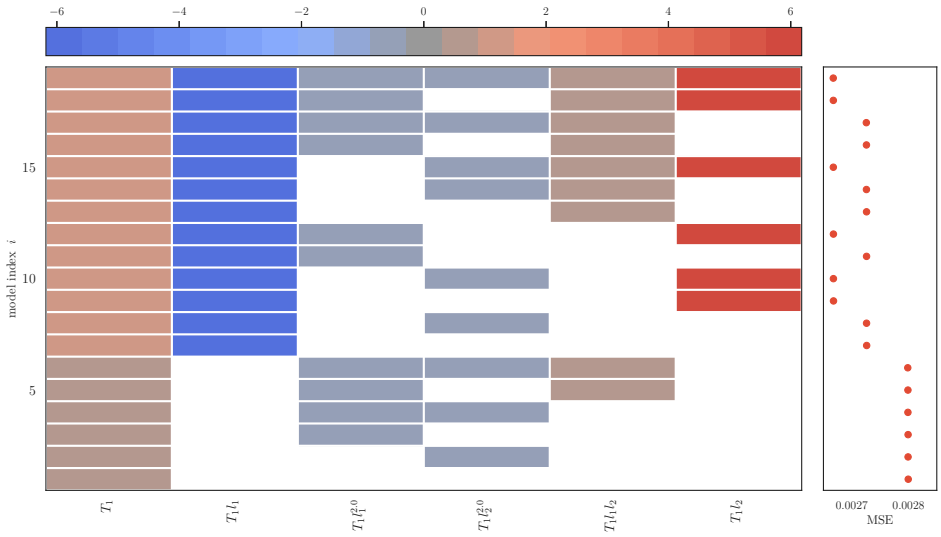


Figure E.4: Models for b_{ij}^{R}

The handpicked models for b_{ij}^{Δ} that are used for cross-validation are given by

$$
\begin{aligned}
M_{b_{ij}^{\Delta}} &= +4.21876 T_{ij}^{(2)} \\
M_{b_{ij}^{\Delta}} &= +4.22119 T_{ij}^{(2)} + 4.21171 T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= \left(-4.05436 I_1 I_2 + 6.02149 I_1^2 - 0.18528 I_1^2 I_2 + 4.8606 I_2^2\right) T_{ij}^{(1)} \\
&\quad + \left(5.8024 - 36.14688 I_1 + 2.5181 I_1 I_2 - 2.49549 I_1^2 - 3.40126 I_2^2\right) T_{ij}^{(2)} \\
&\quad + \left(4.13554 - 0.17882 I_1^2 - 2.39583 I_2\right) T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= \left(-4.08109 I_1 I_2 + 6.04517 I_1^2 - 0.18645 I_1^2 I_2 + 4.92323 I_2^2\right) T_{ij}^{(1)} \\
&\quad + \left(6.81951 - 33.19336 I_1 + 1.88849 I_1 I_2 - 2.29006 I_1^2 + 27.95848 I_2 - 2.34506 I_2^2\right) T_{ij}^{(2)} \\
&\quad + \left(4.14045 - 0.179 I_1^2 - 2.38987 I_2 + 0.07884 I_2^2\right) T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= \left(0.02291 + 2.37067 I_1 + 0.0113 I_1 I_2 + 0.03687 I_1 I_2^2 + 0.18755 I_1^2 + 0.00109 I_1^2 I_2\right. \\
&\quad \left. -1.09365 I_2 + 1.02595 I_2^2\right) T_{ij}^{(1)} \\
&\quad + \left(6.82186 - 33.13731 I_1 + 1.88909 I_1 I_2 - 2.286 I_1^2 + 28.00653 I_2 - 2.35189 I_2^2\right) T_{ij}^{(2)} \\
&\quad + \left(4.12896 - 0.17807 I_1^2 - 2.46437 I_2 + 0.08235 I_2^2\right) T_{ij}^{(3)}
\end{aligned}
$$

The models for b_{ij}^{R} are given by

$$
\begin{aligned}
M_{b_{ij}^{R}} &= +0.65327 T_{ij}^{(1)} \\
M_{b_{ij}^{R}} &= +\left(0.91897 - 6.17912 I_1\right) T_{ij}^{(1)} \\
M_{b_{ij}^{R}} &= +\left(1.15185 - 5.63738 I_1 + 0.32064 I_1 I_2 - 0.31167 I_1^2 + 5.81464 I_2 - 0.33034 I_2^2\right) T_{ij}^{(1)}
\end{aligned}
$$

E.3. Curved Backward-Facing Step

Discovered models for b_{ij}^{Δ} for the curved backward-facing step.

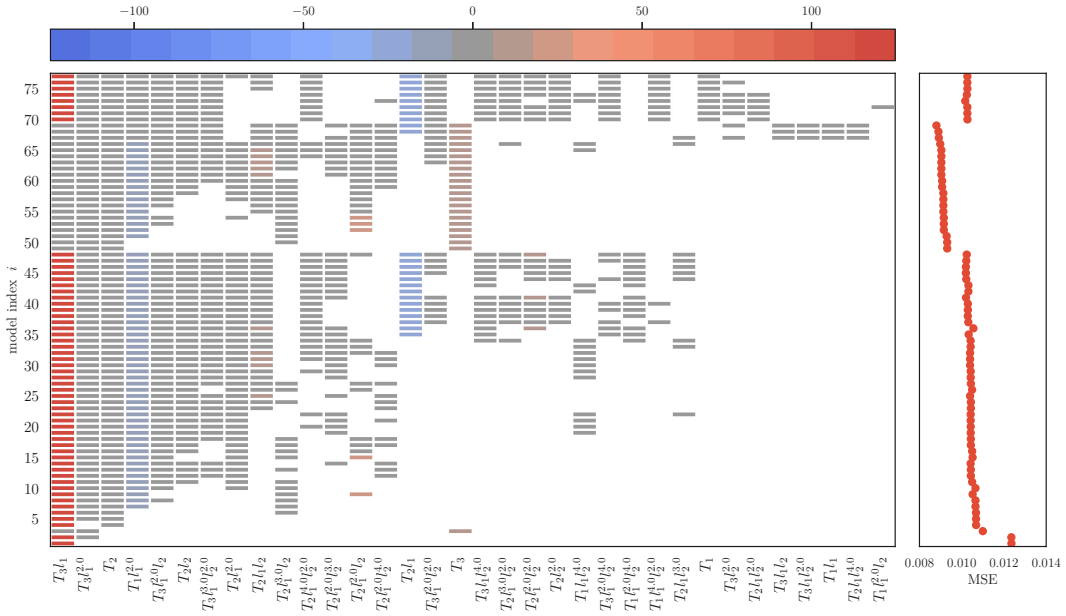


Figure E.5: Models for b_{ij}^{Δ}

Discovered models for b_{ij}^R for the curved backward-facing step.

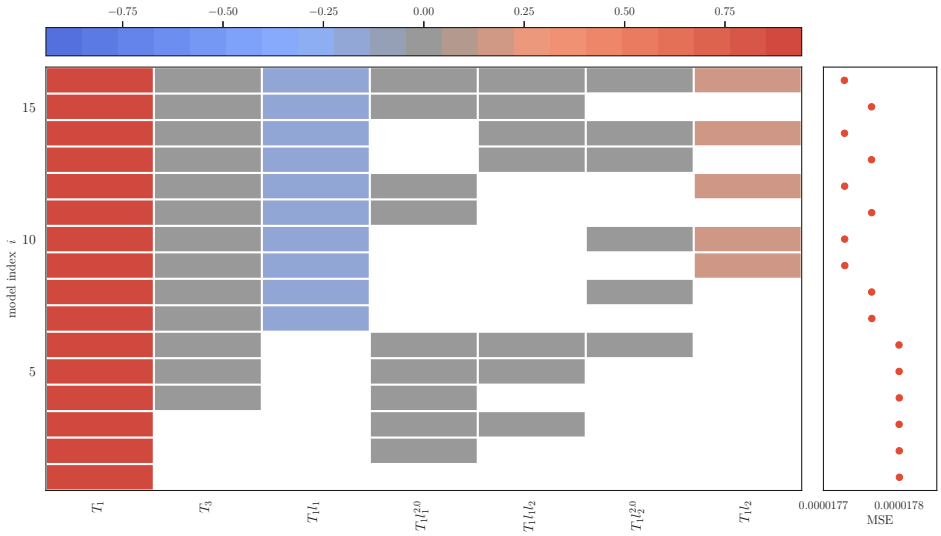


Figure E.6: Models for b_{ij}^R

The handpicked models for b_{ij}^{Δ} that are used for cross-validation are given by

$$
\begin{aligned}
M_{b_{ij}^{\Delta}} &= +\,124.67831 I_1 T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= +\left(10.82959 + 3.98802 I_1 + 0.25672 I_1^2\right) T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= -\,9.83914 I_1^2 T_{ij}^{(1)} \\
&\quad + \left(1.93579 + 21.83418 I_1^2 I_2 + 0.94322 I_1^3 I_2\right) T_{ij}^{(2)} \\
&\quad + \left(124.568 I_1 + 5.89434 I_1^2\right) T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= -\,10.18305 I_1^2 T_{ij}^{(1)} \\
&\quad + \left(1.77897 + 2.44229 I_1^3 I_2\right) T_{ij}^{(2)} \\
&\quad + \left(10.83217 + 3.99599 I_1 + 0.25722 I_1^2\right) T_{ij}^{(3)} \\
M_{b_{ij}^{\Delta}} &= \left(-2.74583 I_1 - 0.58551 I_1^2\right) T_{ij}^{(1)} \\
&\quad + \left(3.11166 - 22.14415 I_1 - 2.7238 I_1 I_2 + 5\exp{-05} I_1 I_2^4 - 0.18466 I_1^2 I_2 - 0.00026 I_1^2 I_2^4\right. \\
&\quad \left. -0.00925 I_1^3 I_2 + 0.28099 I_2\right) T_{ij}^{(2)} \\
&\quad + \left(10.84359 + 3.97393 I_1 + 2.61039 I_1 I_2 + 0.17774 I_1 I_2^2 + 0.25532 I_1^2 + 0.12292 I_1^2 I_2\right. \\
&\quad \left. +0.04917 I_1^2 I_2^2 + 0.00383 I_1^3 I_2^2\right) T_{ij}^{(3)}
\end{aligned}
$$

The models for b_{ij}^R are given by

$$
\begin{aligned}
M_{b_{ij}^R} &= +\,0.9289 T_{ij}^{(1)} \\
M_{b_{ij}^R} &= +\left(0.93496 - 0.15839 I_1\right) T_{ij}^{(1)} \\
&\quad - 0.00072 T_{ij}^{(3)} \\
M_{b_{ij}^R} &= +\left(0.94098 - 0.15759 I_1 + 0.01192 I_1 I_2 - 0.01193 I_1^2 + 0.1569 I_2 - 0.01194 I_2^2\right) T_{ij}^{(1)} \\
&\quad - 0.00071 T_{ij}^{(3)}
\end{aligned}
$$

Bibliography

[1] M. Emory and G. Iaccarino, "Visualizing turbulence anisotropy in the spatial domain with componentality contours," *Cent. Turbul. Res. Annu. Res. Briefs*, pp. 123–138, 2014.

[2] M. Kaandorp, "Machine learning for data-driven RANS turbulence modelling," 2018.

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

[4] C. Jiang, J. Mi, S. Laima, and H. Li, "A novel algebraic stress model with machine-learning-assisted parameterization," *Energies*, vol. 13, no. 1, p. 258, 2020.

[5] M. Schmelzer, R. P. Dwight, and P. Cinnella, "Machine learning of algebraic stress models using deterministic symbolic regression," *arXiv preprint arXiv:1905.07510*, 2019.

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

[7] J.-P. Laval and M. Marquillie, "Direct numerical simulations of converging–diverging channel flow," in *Progress in Wall Turbulence: Understanding and Modeling*, pp. 203–209, Springer, 2011.

[8] H. Le, P. Moin, and J. Kim, "Direct numerical simulation of turbulent flow over a backward-facing step," *Journal of fluid mechanics*, vol. 330, pp. 349–374, 1997.

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

[10] S. Pope, *Turbulent Flows*. Cambridge University Press, 2000.

[11] Z. Zhaoshun, C. Guixiang, and X. Chunxiao, "Modern turbulence and new challenges," *Acta Mechanica Sinica*, vol. 18, no. 4, pp. 309–327, 2002.

[12] P. Moin and K. Mahesh, "Direct Numerical Simulation: a tool in turbulence research," *Annual Review of Fluid Mechanics*, vol. 30, no. 1, pp. 539–578, 1998.

[13] C. Argyropoulos and N. Markatos, "Recent advances on the numerical modelling of turbulent flows," *Applied Mathematical Modelling*, vol. 39, no. 2, pp. 693–732, 2015.

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

[15] K. Hanjalic, "Will RANS survive LES? A view of perspectives," *Journal of fluids engineering*, vol. 127, no. 5, pp. 831–839, 2005.

[16] W. Edeling, G. Iaccarino, and P. Cinnella, "A return to eddy viscosity model for epistemic UQ in RANS closures," *arXiv preprint arXiv:1705.05354*, 2017.

[17] A. J. Revell, T. J. Craft, and D. R. Laurence, "Turbulence modelling of unsteady turbulent flows using the stress strain lag model," *Flow, Turbulence and Combustion*, vol. 86, no. 1, pp. 129–151, 2011.

[18] B. Perot and P. Moin, "A new approach to turbulence modeling," 1996.

[19] S. Lardeau and F. Billard, "Development of an elliptic-blending lag model for industrial applications," in *54th AIAA Aerospace Sciences Meeting*, p. 1600, 2016.

[20] R. Manceau and K. Hanjalić, "Elliptic blending model: A new near-wall Reynolds-stress turbulence closure," *Physics of Fluids*, vol. 14, no. 2, pp. 744–754, 2002.

[21] R. Biswas, P. A. Durbin, and G. Medic, "Development of an elliptic blending lag k-ω model," *International Journal of Heat and Fluid Flow*, vol. 76, pp. 26–39, 2019.

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

[23] A. P. Singh and K. Duraisamy, "Using field inversion to quantify functional errors in turbulence closures," *Physics of Fluids*, vol. 28, no. 4, p. 045110, 2016.

[24] A. P. Singh, S. Medida, and K. Duraisamy, "Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils," *AIAA Journal*, pp. 2215–2227, 2017.

[25] M. Emory, R. Pecnik, and G. Iaccarino, "Modeling structural uncertainties in Reynolds-averaged computations of shock/boundary layer interactions," in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 479, 2011.

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

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

[28] J. Ling, R. Jones, and J. Templeton, "Machine learning strategies for systems with invariance properties," *Journal of Computational Physics*, vol. 318, pp. 22–35, 2016.

[29] J.-X. Wang, J.-L. Wu, and H. Xiao, "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data," *Physical Review Fluids*, vol. 2, no. 3, p. 034603, 2017.

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

[31] J. Weatheritt and R. Sandberg, "The development of algebraic stress models using a novel evolutionary algorithm," *International Journal of Heat and Fluid Flow*, vol. 68, pp. 298–318, 2017.

[32] D. Wilcox, *Turbulence Modeling for CFD*. DCW Industries, Incorporated, 1994.

[33] N. N. Mansour, J. Kim, and P. Moin, "Reynolds-stress and dissipation-rate budgets in a turbulent channel flow," *Journal of Fluid Mechanics*, vol. 194, pp. 15–44, 1988.

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

[35] K.-S. Choi and J. L. Lumley, "The return to isotropy of homogeneous turbulence," *Journal of Fluid Mechanics*, vol. 436, pp. 59–84, 2001.

[36] J. L. Lumley and G. R. Newman, "The return to isotropy of homogeneous turbulence," *Journal of Fluid Mechanics*, vol. 82, no. 1, pp. 161–178, 1977.

[37] U. Schumann, "Realizability of Reynolds-stress turbulence models," *The Physics of Fluids*, vol. 20, no. 5, pp. 721–725, 1977.

[38] P. A. Durbin and B. P. Reif, *Statistical theory and modeling for turbulent flows*. John Wiley & Sons, 2011.

[39] R. Tunstall, S. Lardeau, D. Laurence, and R. Prosser, "An elliptic blending lag model for flows in thermal-hydraulics systems," in *ETMM Conference*, 2016.

[40] R. Biswas and P. A. Durbin, "Assessment of viscosity models that incorporate lag parameter scaling," *International Journal of Heat and Fluid Flow*, vol. 78, p. 108427, 2019.

[41] P. A. Durbin, "Near-wall turbulence closure modeling without damping functions," *Theoretical and Computational Fluid Dynamics*, vol. 3, no. 1, pp. 1–13, 1991.

[42] K. Hanjalić, M. Popovac, and M. Hadžiabdić, "A robust near-wall elliptic-relaxation eddy-viscosity turbulence model for CFD," *International Journal of Heat and Fluid Flow*, vol. 25, no. 6, pp. 1047–1051, 2004.

[43] F.-S. Lien and G. Kalitzin, "Computations of transonic flow with the v2–f turbulence model," *International Journal of Heat and Fluid Flow*, vol. 22, no. 1, pp. 53–61, 2001.

[44] D. Laurence, J. Uribe, and S. Utyuzhnikov, "A robust formulation of the v2-f model," *Flow, Turbulence and Combustion*, vol. 73, no. 3-4, pp. 169–185, 2005.

[45] F. Billard and D. Laurence, "A robust k- ε- v2/k elliptic blending turbulence model applied to near-wall, separated and buoyant flows," *International Journal of Heat and Fluid Flow*, vol. 33, no. 1, pp. 45–58, 2012.

[46] A. Yoshizawa, H. Abe, Y. Matsuo, H. Fujiwara, and Y. Mizobuchi, "A reynolds-averaged turbulence modeling approach using three transport equations for the turbulent viscosity, kinetic energy, and dissipation rate," *Physics of fluids*, vol. 24, no. 7, p. 075109, 2012.

[47] R. Manceau, "Recent progress in the development of the Elliptic Blending Reynolds-stress model," *International Journal of Heat and Fluid Flow*, vol. 51, pp. 195–220, 2015.

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

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

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

[51] H. Xiao, J.-L. Wu, S. Laizet, and L. Duan, "Flows over periodic hills of parameterized geometries: A dataset for data-driven turbulence modeling from direct simulations," *Computers & Fluids*, p. 104431, 2020.

[52] G. A. Gerolymos, E. Sauret, and I. Vallet, "Contribution to single-point closure reynolds-stress modelling of inhomogeneous flow," *Theoretical and Computational Fluid Dynamics*, vol. 17, no. 5-6, pp. 407–431, 2004.

[53] B. Younis, T. Gatski, and C. G. Speziale, "Towards a rational model for the triple velocity correlations of turbulence," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 456, no. 1996, pp. 909–920, 2000.

[54] S. Poroseva and S. M. Murman, "Velocity/pressure-gradient correlations in a FORANS approach to turbulence modeling," in *44th AIAA Fluid Dynamics Conference*, p. 2207, 2014.

[55] D. C. Wilcox *et al.*, *Turbulence modeling for CFD*, vol. 2. DCW industries La Canada, CA, 1998.

[56] A. P. Singh, *A framework to improve turbulence models using full-field inversion and machine learning*. PhD thesis, 2018.

[57] J. R. Holland, J. D. Baeder, and K. Duraisamy, "Towards integrated field inversion and machine learning with embedded neural networks for RANS modeling," in *AIAA Scitech 2019 Forum*, p. 1884, 2019.

[58] M. Olsen and T. Coakley, "The lag model, a turbulence model for non equilibrium flows," in *15th AIAA Computational Fluid Dynamics Conference*, p. 2564, 2001.

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

[60] T. Craft, B. Launder, and K. Suga, "Development and application of a cubic eddy-viscosity model of turbulence," *International Journal of Heat and Fluid Flow*, vol. 17, no. 2, pp. 108–115, 1996.

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

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

[63] R. L. Thompson, L. E. B. Sampaio, F. A. de Bragança Alves, L. Thais, and G. Mompean, "A methodology to evaluate statistical errors in DNS data of plane channel flows," *Computers & Fluids*, vol. 130, pp. 1–7, 2016.

[64] J. Wu, H. Xiao, R. Sun, and Q. Wang, "RANS equations with Reynolds stress closure can be ill-conditioned," *arXiv preprint arXiv:1803.05581*, 2018.

[65] J. Weatheritt and R. D. Sandberg, "Hybrid Reynolds-averaged/large-eddy simulation methodology from symbolic regression: formulation and application," *AIAA Journal*, pp. 3734–3746, 2017.

[66] J. Weatheritt and R. D. Sandberg, "Improved junction body flow modeling through data-driven symbolic regression," *Journal of Ship Research*, 2019.

[67] M. Schmelzer, R. Dwight, and P. Cinnella, "Data-driven deterministic symbolic regression of non-linear stress-strain relation for RANS turbulence modelling," in *2018 Fluid Dynamics Conference*, p. 2900, 2018.

[68] M. Schmelzer, R. P. Dwight, and P. Cinnella, "Discovery of algebraic Reynolds-stress models using sparse symbolic regression," *Flow, Turbulence and Combustion*, pp. 1–25, 2019.

[69] Y. Zhang, R. P. Dwight, M. Schmelzer, J. F. Gomez, S. Hickel, and Z.-h. Han, "Customized data-driven rans closures for bi-fidelity les-rans optimization," *arXiv preprint arXiv:2004.03003*, 2020.

[70] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[71] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[72] A. Beck, D. Flad, and C.-D. Munz, "Deep neural networks for data-driven LES closure models," *Journal of Computational Physics*, vol. 398, p. 108910, 2019.

[73] H.-I. Suk, S.-W. Lee, D. Shen, A. D. N. Initiative, *et al.*, "Deep ensemble learning of sparse regression models for brain disease diagnosis," *Medical image analysis*, vol. 37, pp. 101–113, 2017.

[74] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[75] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[76] R. Biswas. personal communication.

[77] T. McConaghy, *Genetic programming theory and practice IX*. Genetic and Evolutionary Computation, New York, NY: Springer, 2011.

[78] D. R. Chapman, "Computational aerodynamics development and outlook," *AIAA journal*, vol. 17, no. 12, pp. 1293–1313, 1979.

[79] J. N. Kutz, "Deep learning in fluid dynamics," *Journal of Fluid Mechanics*, vol. 814, pp. 1–4, 2017.

[80] C. Mellen, J. Fröhlich, and W. Rodi, "Large eddy simulation of the flow over periodic hills," 2000.

[81] D. M. Driver and H. L. Seegmiller, "Features of a reattaching turbulent shear layer in divergent channel flow," *AIAA journal*, vol. 23, no. 2, pp. 163–171, 1985.

[82] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[83] R. C. Aster, C. H. Thurber, and B. Borchers, "Parameter estimation and inverse problems," 2005.

[84] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.