Data-driven Augmentation of Turbulence Models for Complex Fluid Flows

by

Omid Bidar

Dissertation submitted to the University of Sheffield
in partial fulfilment of the requirements for the
degree of Doctor of Philosophy
June 2024

Supervisors:

Dr. Sean R Anderson
Department of Automatic Control and Systems Engineering

Professor Ning Qin
Department of Mechanical Engineering

ORCID iD: 0009-0000-9889-7351

Abstract

Turbulent flows are broadly investigated using physical experiments and/or computational fluid dynamics (CFD) where the accuracy, level of fidelity, and associated costs play competing roles. For practical CFD analyses such as design and analysis of aircraft, there is a heavy reliance on the Reynolds-averaged Navier–Stokes (RANS) simulations due to their relative simplicity and low computational cost. This comes at the cost of limited accuracy for complex flows. A number of data-driven techniques have emerged that augment RANS-based turbulence models using high-fidelity data. The novel contributions of this thesis in data-driven augmentation of RANS-based turbulence models, categorised under two themes, are: 1) the data in adjoint-based data assimilation, and 2) developing and employing neural network-augmented turbulence models for automated aerodynamic design optimisation.

Firstly, we develop a novel sensor placement approach for generating sparse experimental data for data assimilation. It leverages systematic eigenspace perturbations of the Reynolds-stress tensor to identify regions of flow where RANS turbulence modelling assumptions are most erroneous and concentrate measurements in these areas using a novel greedy algorithm. Results show that the error reduction using our placement strategy is close in accuracy to the instances where two to three orders of magnitude more data points than the placed sensors. We further investigate fusing data for multiple quantities (e.g. velocity and surface pressure measurements) to perform data assimilation, and investigate the relative

importance of various physical quantities on the flow reconstruction.

Secondly, we employ the field inversion and machine learning technique to augment an existing turbulence model for aerodynamic shape optimisation. As a proof-of-concept we apply the framework to a separated periodic hill flow. Our investigations highlight the role of the turbulence model on the optimal shapes achievable, with the neural network-augmented turbulence model achieving approximately 4-times more drag reduction compared to the baseline model.

Dedication

To my parents

for their dreams, sacrifices, and unwavering faith.

Acknowledgements

I am grateful to my supervisors, Dr Sean Anderson and Professor Ning Qin, for agreeing to supervise me in this work. Their guidance and mentorship has been instrumental to the contents of this thesis. I particularly appreciate their accommodating supervision style.

I would like to thank Dr Ping He, from Iowa State University, for his collaboration and support developing the software in this work.

I would like to thank my thesis examiners, Professor Krzysztof Fidkowski from the University of Michigan and Professor Shuisheng He from the University of Sheffield, for their insightful suggestions to improve this thesis.

I am thankful to the University of Sheffield, and Engineering and Physical Sciences Research Council (EPSRC) for funding this work.

I am thankful to my family: Rahjo, Parima, Pouya, Atay and Abay. Working from home throughout this project would not have been possible without your unconditional love, support, and encouragement. Your presence is a constant source of inspiration. Finally, I would like to thank my beloved Husnia. I am very excited about the journey ahead of us.

Contents

List of figures

List of tables

CHAPTER 1

Introduction

1.1 Motivations

1.1.1 The need for predicting turbulent flows

Turbulent fluid flow phenomena are present across natural and engineered systems. The understanding and predictions of turbulent flows are crucial across many scientific and engineering fields, such as environmental sciences; aerospace, automotive and nuclear engineering; and beyond. Broadly, turbulent flows are investigated using real-world experiments, such as wind tunnel measurements and numerical simulations. Computational fluid dynamics (CFD) analyses have become an indispensable tool across many industries by allowing physical insights into flow physics and enabling design optimisation of engineering applications while reducing the expense, complexity, and technical limitations of a purely experimental design approach[144]. For instance, Fig. 1.1 shows an overview of the use of CFD in the aircraft design process at Airbus.

1.1.2 High-fidelity approaches and their limitations

The Navier-Stokes equations, based on conservation of mass, momentum and energy, describe turbulent fluid flows. For Newtonian, incompressible flows the

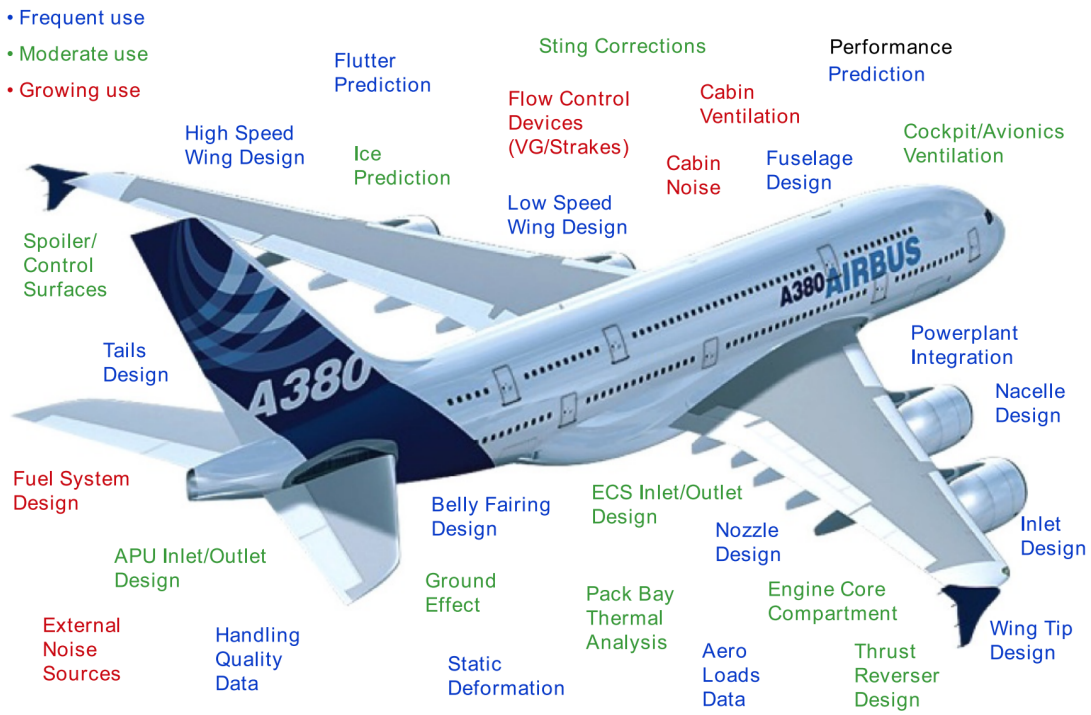


Figure 1.1: Graphical representation of how "CFD moved from an exploratory tool to a full flight physics production" in Airbus aircraft design cycle, image for CFD application in 2010, and quote from the company's head of aerodynamic research and technology, [2].

N-S equations are:

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{1.1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{1.2}$$

where u_i is the velocity, x_i is the spatial coordinates, p is pressure, $Re = \rho l_{\text{ref}} U_{\text{ref}}/\mu$ is the Reynolds number based on density, reference length and velocity, and dynamic viscosity respectively.

The main approaches to the use of N-S equations is illustrated in Figure 1.2. Highest-fidelity can be achieved through direct numerical simulation (DNS) which *fully resolves* the N-S equations in space and time [121]. The computational re-

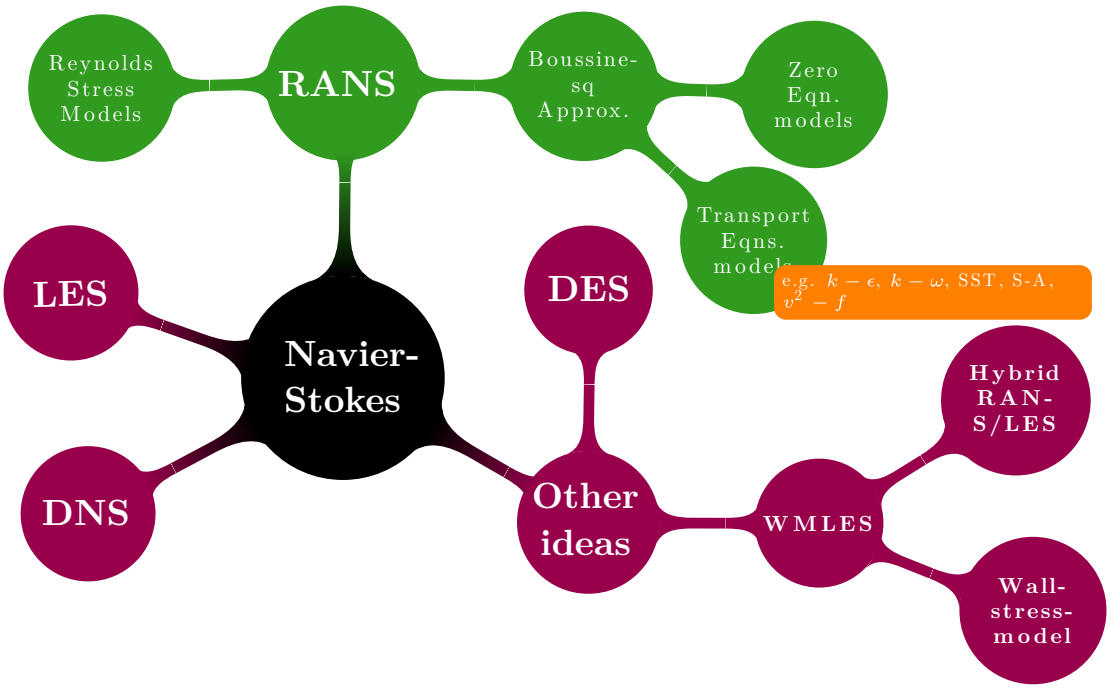


Figure 1.2: Classification of the main turbulence simulation and modelling approaches, diagram inspired from [87]. Abbreviations: DNS, direct numerical simulations; LES, large eddy simulations; DES, detached eddy simulations; WMLES, wall-modelled LES; RANS, Reynolds-averaged Navier-Stokes; SST, shear stress transport; S-A, Spalart-Allmaras.

quirements for DNS can be estimated using Kolmogorov's hypothesis, which states that at sufficiently high Reynolds numbers, small-scale turbulent motion is statistically isotropic, and can be uniquely determined by the viscosity η and dissipation rate ϵ. This leads to the definition of the Kolmogorov length, velocity, and time scales defined as: $\eta \equiv (\nu^3/\epsilon)^{1/4}$, $u_\eta \equiv (\epsilon\nu)^{1/4}$ and $\tau_\eta \equiv (\nu/\epsilon)^{1/2}$, respectively. Thus the ratio of the smallest to largest Kolmogorov scales can be determined as follows: $\eta/l_0 \sim \mathrm{Re}^{-3/4}$, $u_\eta/u_0 \sim \mathrm{Re}^{-1/4}$, and $\tau_\eta/\tau_0 \sim \mathrm{Re}^{-1/2}$ [121]. To fully resolve the N-S equations, the spatial and temporal resolution should scale: $(\Delta x)^3 \sim (l_0/\eta)^3$ (in three dimensions) and $\Delta t \sim \tau_0/\tau_\eta$. Therefore, the total computational cost can be estimated as $(\mathrm{Re}^{3/4})^3 \cdot \mathrm{Re}^{1/2} \sim \mathcal{O}(\mathrm{Re}^{2.75})$. As an example of the computational costs involved, in a 2022 DNS study of a transonic aerofoil at $Re = 3 \times 10^5$ and

freestream Mach number $M_\infty = 0.7$, the meshes employed had 536 - 1800 million cells, with a time step $\Delta t = 1.5 \times 10^{-4}$ (baseline mesh) and $\Delta t = 1 \times 10^{-4}$ (fine mesh), and the flow statistics were accumulated over a time $40c/U_\infty$, where c is the chord length, and U_∞ is the freestream velocity [124]. Given that for complex aerodynamic flows the Reynolds number is $\mathcal{O}(10^6)$ and higher (e.g. flow over a wing), DNS will require huge computational resources.

An alternative is large-eddy simulations (LES) [101] which involves resolving the large-scale turbulence effects while a filtering process is applied to scales below a threshold. The filtering process gives rise to the Smagorinsky stress tensor term, in the momentum equation (Eqn. 1.2), which leads to the need for sub-grid scale models to close the equation—unless the implicit LES approach is used [58]. The computational cost of LES, based on the grid requirements, is estimated to scale as $N \sim Re^{0.4}$ away from the wall, while scaling as $N \sim Re^{1.8-2}$ near the wall [7, 60, 79]. Despite reducing the computational costs compared to the DNS approach, LES unfortunately still remains expensive for wall-bounded high Reynolds number flows as the energetic scales – even if small – dominate the dynamics in the near-wall region [146].

Other ideas that are computationally cheaper include wall-modelled LES (WMLES). In WMLES the energetic motions in the outer boundary layer are resolved while the near-wall eddies are modelled. In WMLES the number of grid points (N) required is estimated to be proportional to Re_{L_x}, where L_x is the flat plate length in the streamwise direction, compared to $N \sim Re_{L_x}^{13/7}$ for wall-resolved LES [30]. Reviews in Refs. [79, 20] further divide WMLES into two categories: hybrid RANS-LES and wall-stress-modelled LES approaches. In the former RANS equations are used near walls and interfacing to LES away from walls, while in the latter LES governing equations are solved throughout the domain, with the wall boundary conditions augmented to account for the effect of the unresolved inner

layer of the boundary layer [20]. While WMLES is computationally cheaper than DNS and LES, it can still be orders of magnitude more expensive than RANS [79]. For many-query studies, e.g. iterative aerodynamic design optimisation where multiple flow solutions are required, WMLES can still be very expensive.

1.1.3 Reynolds-averaged Navier-Stokes and associated inaccuracies

Turbulence models based on Reynolds-averaged Navier-Stokes (RANS) equations [4] remain the workhorse of most CFD analyses due to their relative simplicity, ease of implementation, and considerably lower computational cost. The RANS equations are so-called because the instantaneous velocity and pressure (u_i and p) in the N-S equations (Eqn. 1.1, and 1.2) are decomposed into a mean (U_i and P) and fluctuating (u'_i and p') component through statistical ensemble averaging [128], leading to:

$$\frac{\partial U_i}{\partial x_i} = 0, \tag{1.3}$$

$$\frac{\partial U_i}{\partial t} + \frac{\partial (U_i U_j)}{dx_j} = -\frac{\partial P}{dx_i} + \frac{1}{Re}\frac{\partial^2 U_i}{\partial x_j \partial x_j} - \frac{\partial \overline{u'_i u'_j}}{\partial x_j}. \tag{1.4}$$

The rationale behind the averaging process is that for most of the engineering applications where turbulent flow behaviour simulations are required, the quantities of interest mainly depend on mean flow (e.g. force coefficients for aerodynamic design optimisation)*, thus the instantaneous fields are not resolved [170]. However, the averaging process results in the Reynolds stress tensor, $-\rho\overline{u'_i u'_j}$ in Eqn. 1.4. As the Reynolds stress tensor is a function of the velocity fluctuations—which are not resolved—it must be modelled/"closed" in terms of the averaged quantities.

*Counterexamples however exist where unsteady values are of interest, e.g. when assessing aircraft ride quality.

The prevalent RANS closures use the Boussinesq hypothesis by assuming a linear relation between the eddy viscosity and the Reynolds stresses, and then using surrogate variable(s) to model the eddy viscosity[81]. Widely used models of this type are the two-equation $k-\epsilon$ [82], $k-\omega$ [162], and the $k-\omega$ shear stress transport [105], and the one-equation Spalart-Allmaras (S-A) [143] models and their variants, which have been formulated based on a combination of intuition, empiricism, and theoretical constraints [41].

Despite their prevalent use, RANS-based simulations have well-documented deficiencies for complex turbulent flows, such as flow separation, strong unsteadiness, and strong flow curvatures [148]. For instance, Fig. 1.3 illustrates the RANS-based prediction variability of lift and pitching momemnt coefficient for the NASA common research model (CRM) at a Mach number of 0.85 and a Reynolds number of 20 million. The data is from the seventh AIAA Drag Prediction Workshop in 2022.

For instance, RANS models are prone to inaccuracies in predicting flows over wings at high angles of attack as the flows tend to separate. This is illustrated in the comparative RANS predictions of the lift coefficient for the NASA high-lift common research model (HL-CRM) in Fig. 1.3 from the 2017 American Institute of Aeronautics and Astronautics (AIAA) High Lift Prediction Workshop[133]. Although most RANS predictions at lower angles of attack, α, have reasonable agreement with wind-tunnel measurements, the prediction capabilities at higher angles show high variability due the complex flows not being captured accurately by the RANS models.

Figure 1.4 illustrates the rise in the use of popular RANS models against wind tunnel tests in the design cycle of commercial aircraft. It is important to note that the linear eddy-viscosity models developed by the 1990s are still in use. While a plethora of elegant ideas have spurred in the turbulence research community such

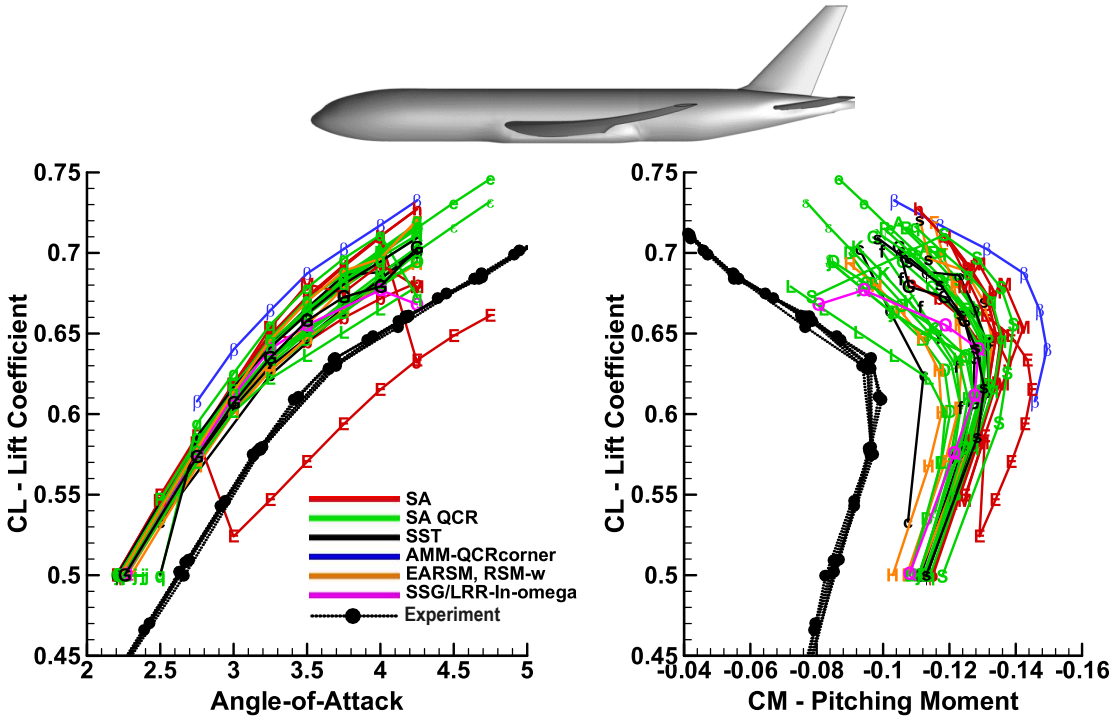


Figure 1.3: Comparison of the wind tunnel lift and pitching moment coefficients against various RANS-based turbulence models for the NASA Common Research Modle (CRM). Data from the seventh AIAA Drag Prediction Workshop in 2022. Letters on the lines designate various participants. For details refer to Ref. [151].

as non-linear eddy viscosity models [52], elliptic relaxation [59], and hybrid RANS-LES [116] to address the deficiencies, their complexity and lack of generalisation have limited their applications, leading to the conclusion that traditional RANS modelling approaches may have reached a plateau [141].

1.1.4 Data-driven RANS-based turbulence modelling

Data-driven approaches such as machine learning have gained significant traction across many disciplines, including the field of fluid dynamics, extensively reviewed in Refs. [38, 22, 24, 37, 3, 155, 92, 117, 134]. Turbulence closure modelling is one such avenue which will be reviewed in detail in the subsequent chapter.

Though traditional RANS models already make use of data by, for example,

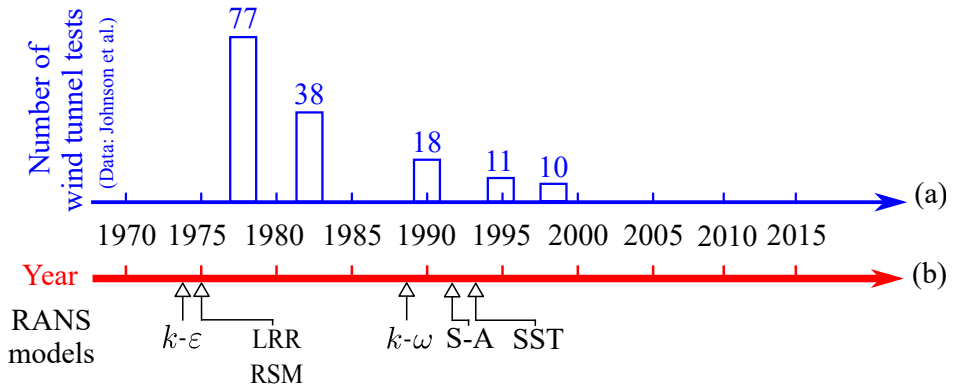


Figure 1.4: Timeline of popular RANS-based turbulence models. (a) The number of wind tunnel tests for wing design during Boeing's aircraft development program, data reported up to the year 2000, from Johnson *et al.* [72], (b) the introduction of popular RANS-based turbulence models: $k-\epsilon$ [82], Launder, Reece and Rodi's Reynolds Stress Model (RSM) [80], Wilcox $k-\omega$ [164], Spalart-Allmaras [143] and $k-\omega$ shear stress transport model [104]. Figure based on Xiao & Cinnella [170].

using experimental or DNS results from limited canonical cases to tune model coefficients, the rise in data-driven turbulence modelling over the last decade or so, is a more comprehensive effort to leverage data assimilation and machine learning tools and the existence of richer high-fidelity datasets—from experiments and scale resolving simulations—for improved turbulence modelling.

In this work we are interested in the *augmentation* of existing turbulence models using the field inversion and machine learning framework (FIML) [119, 139]. The field inversion step—an adjoint-based data assimilation approach—is a highly-dimensional optimisation problem where the transport equation for an existing model is perturbed by a scalar spatial field which is then adjusted iteratively such that the error between the RANS prediction for a given quantity of interest compared to the high-fidelity data is reduced. Machine learning techniques, such as neural networks, are then used to generalise the augmentations for flows beyond the training dataset. It is a relatively well-established framework, with many variants, which will be reviewed in detail in chapter 2.3.1. Key advantages

of the FIML framework include its ability to work with sparse training datasets, and model-consistency since the governing equations are part of the training—this will be further discussed in chapter 2.

1.2 Contributions of the present work

The contributions of this thesis can be categorised into two themes: the role of sparse, experimentally measureable, data in adjoint-based data assimilation; and the development and use of a neural network-augmented turbulence model for aerodynamic shape optimisation.

Adjoint-based data assimilation has been shown to achieve considerable improvements using sparse, experimentally measureable, data. The types of data used in literature have included volume/field data (e.g. velocity profiles [119]), surface data (e.g. skin friction [39], pressure coefficient [12]), and integral data (e.g. lift coefficient [139])—listed in terms of lowest to highest data sparsity. Unexplored aspects to be investigated in this work are outlined below.

Sparse sensor placement using eigenspace perturbations

Most applications of data assimilation for turbulence model augmentation thus far have been primarily dictated by the availability of existing data, usually measured for benchmarking and validation. This thesis presents a novel *a priori* sensor placement strategy to measure the experimental data for data assimilation. The strategy relies on eigenspace perturbations of the Reynolds stress tensors to identify regions of uncertainty in the flow domain due to the structural uncertainties in the turbulence model and targets the sensors in these regions using a greedy placement algorithm.

Research on this topic led to the following publications which are presented in **chapter 4** of the thesis:

- Omid Bidar, Sean Anderson, and Ning Qin. "*Sensor placement for data assimilation of turbulence models using eigenspace perturbations*". In: Physics of Fluids 36.1 (Jan. 2024). DOI: 10.1063/5.0182080. Author contributions: Omid Bidar: Conceptualisation; Formal analysis; Investigation; Methodology; Software; Writing – original draft; Writing – review & editing. Sean Anderson: Conceptualisation; Formal analysis; Funding acquisition; Methodology; Project administration; Supervision; Writing – review & editing. Ning Qin: Conceptualisation; Formal analysis; Methodology; Supervision; Writing – review & editing.

- Omid Bidar, Sean Anderson, and Ning Qin. "*A Priori Sensor Placement Strategy for Turbulent Mean Flow Reconstruction Using Parametric Model Perturbations*". In: AIAA SCITECH 2024 Forum. American Institute of Aeronautics and Astronautics, Jan. 2024. DOI: 10.2514/6.2024-1580. Author contributions: Omid Bidar: Conceptualisation; Formal analysis; Investigation; Methodology; Software; Writing - original draft, review & editing. Sean Anderson: Project administration; Supervision; Writing - review & editing. Ning Qin: Supervision; Writing - review & editing.

Multi-sensor data for data assimilation

To date, data assimilation using adjoint-based methods have been limited to using only a single source of data, e.g. surface pressure measurements, or velocity profiles. This is a limitation because multiple sources can be used simultaneously to reconstruct the flow and in theory improve the accuracy of the model and ensure that it fits all available data correctly. This work extends the adjoint-based field inversion framework, to make use of multiple data sources for enhanced data assimilation using multi-objective optimisation.

Research on this topic led to the following publication, and is presented in

chapter 5 of the thesis:

- Omid Bidar, Ping He, Sean Anderson, and Ning Qin. "*Turbulent mean flow reconstruction based on sparse multi-sensor data and adjoint-based field inversion*". In: AIAA Aviation 2022 Forum. American Institute of Aeronautics and Astronautics, June 2022. DOI: 10.2514/6.2022-3900. Author contributions: Omid Bidar: Conceptualisation; Formal analysis; Investigation; Methodology; Software; Writing - original draft, review & editing. Ping He: Conceptualisation; Methodology; Software; Writing - review & editing. Sean Anderson: Conceptualisation; Methodology; Project administration; Supervision; Writing - original draft; Writing - review & editing. Ning Qin: Conceptualisation; Methodology; Supervision; Writing - review & editing.

Relative importance of various physical quantities

It has been observed that the impact of data assimilation in terms of error reduction in all physical quantities of interest is sensitive to the type of data used. However, there has been no quantitative study of the relative importance of physical quantities on data assimilation outcomes. In this work we compare the accuracy of turbulent flow reconstruction using field inversion when using experimentally measurable data such as surface pressure, skin friction, and velocity profiles.

The results from this work was presented at the UK Fluids Conference, and is presented in **chapter 3** of the thesis:

- Omid Bidar, "*Relative Importance of Physical Quantities for Data-driven RANS-based Turbulence Modelling*". Presented at the UK Fluids Conference, Sheffield, UK. 2022. DOI: 10.13140/RG.2.2.16569.44641.

Aerodynamic shape optimisation with data-driven turbulence modelling

Although the debate about universality of data-driven turbulence model is an important one to be had, in the context of aerodynamic flows, there is still scope for customised models tailored to specific types of applications—aerodynamic shape optimisation is one such case. It has become an important tool in the aircraft design process [140]. The iterative process involves: starting with a given parameterised shape, manipulate the surfaces to optimise a given characteristic (e.g. minimum drag) while satisfying various constraints: such as geometric constraints (e.g. twist, span, thickness, etc.), aerodynamic constraints (e.g. the lift generated), and/or constraints on flow conditions.

Flow predictions during the iterative design optimisation is typically limited to RANS-based simulations due to the computational costs of higher-fidelity approaches. However, when performing shape optimisation for mission critical applications in complex operating conditions, one of the limitations faced is the inaccuracies due to turbulence modelling, for example, as discussed earlier in the case of the NASA high-lift common research model at high angles of attack (Fig. 1.3). Therefore, this thesis explores the use of a neural network-augmented turbulence model for aerodynamic shape optimisation of a proof-of-concept, highly complex separated flow.

Research on this subject led to the following publications, and are presented in **chapter 6 & 7**:

- Omid Bidar, Ping He, Sean Anderson, and Ning Qin. "*Aerodynamic Shape Optimisation Using a Machine Learning-Augmented Turbulence Model*". In: AIAA SCITECH 2024 Forum. American Institute of Aeronautics and Astronautics, Jan. 2024. DOI: 10.2514/6.2024-1231. Author contributions: Omid Bidar: Conceptualisation; Formal analysis; Investigation; Methodol-

ogy; Software; Writing - original draft, review & editing. Ping He: Conceptualisation; Methodology; Software; Writing - review & editing. Sean Anderson: Conceptualisation; Project administration; Supervision; Writing - review & editing. Ning Qin: Conceptualisation; Supervision; Writing - review & editing.

- Omid Bidar, Sean Anderson, and Ning Qin. "*A Hybrid RANS-LES Dataset for Data-driven Turbulent Mean Flow Reconstruction*". Presented at the Cambridge Unsteady Flow Symposium 2024. Author contributions: Omid Bidar: Conceptualisation; Formal analysis; Investigation; Methodology; Software; Writing - original draft, review & editing. Sean Anderson: Project administration; Supervision; Writing - review & editing. Ning Qin: Supervision; Writing - review & editing.

Open-source implementation

As previously stated, data assimilation via field inversion involves the solution of a high dimensional optimisation problem, where the number of design variables/degrees of freedom is equivalent to the size of the mesh. To make the optimisation task computationally tractable, a gradient-based strategy is adopted which require the adjoint approach [75] for efficient gradient evaluations. The intrusive nature of this approach involving the CFD flow solver, the adjoint solver, and the optimiser makes its implementation a high entry barrier for most researchers [37]. In this work we integrate existing open-source codes to enable a flexible implementation of the FIML framework available to all researchers.

This led to the following publication, and is presented in **chapter 3** of the thesis:

- Omid Bidar, Ping He, Sean Anderson, and Ning Qin. "*An open-source adjoint-based field inversion tool for data-driven RANS modelling*". In:

AIAA Aviation 2022 Forum. American Institute of Aeronautics and Astronautics, June 2022. DOI: 10.2514/6.2022-4125. Author contributions: Omid Bidar: Conceptualisation; Formal analysis; Investigation; Methodology; Software; Writing - original draft, review & editing. Ping He: Conceptualisation; Methodology; Software; Writing - review & editing. Sean Anderson: Conceptualisation; Project administration; Supervision; Writing - review & editing. Ning Qin: Conceptualisation; Supervision; Writing - review & editing.

1.3 Thesis structure

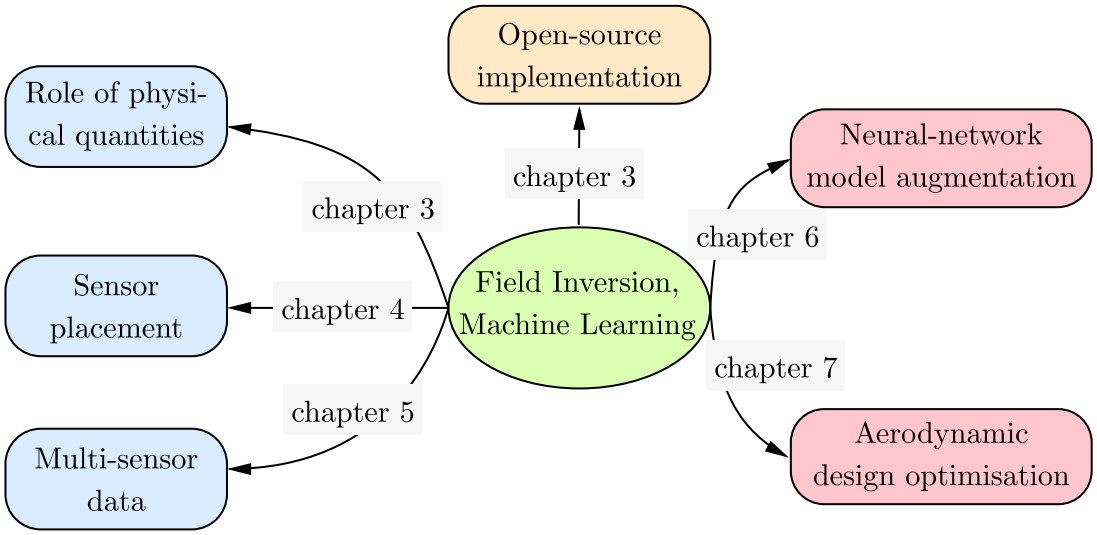


Figure 1.5: Summary of thesis structure.

CHAPTER 2 presents a literature survey of data-driven RANS-based turbulence modelling, with a particular focus on field inversion and machine learning.

CHAPTER 3 formulates the adjoint-based data assimilation problem, briefly outlines our open-source implementation, and investigates data assimilation on a range of separated flows. The ten data assimilation scenarios consider

different turbulence models, various data sources (e.g. experimental, scale-resolving simulations) and data types (e.g. surface data such as lift coefficient and skin friction, or field data such as velocity provides). It also investigates the role of different physical quantities on the data assimilation.

CHAPTER 4 formulates and tests a novel sensor placement framework for data assimilation using the eigenspace perturbations approach.

CHAPTER 5 investigates the use of multi-sensor data for adjoint-based data assimilation.

CHAPTER 6 develops a neural-network augmented turbulence model for a proof-of-concept case, and performs *a priori* and *a posteriori* tests of the augmented model on different separated flows.

CHAPTER 7 investigates the application of the neural network-augmented model, developed in chapter 6, for performing aerodynamic shape optimisation, comparing the augmented model against baseline Wilcox $k-\omega$ and shear stress transport models. In addition hybrid RANS-LES predictions using the improved delayed detached eddy simulation are performed on the RANS-based optimal shapes to verify the flow predictions.

CHAPTER 8 concludes the thesis, and suggests avenues for future work.

CHAPTER 2

Data-driven turbulence modelling: a literature review

This chapter provides a high-level overview of the state-of-the-art on data-driven approaches in the context of Reynolds-averaged Navier-Stokes (RANS) simulations.

Most of the development in data-driven turbulence modelling has occurred over the last decade. Beginning with quantifying uncertainties in existing models, researchers have then attempted to reduce the turbulence modelling errors, broadly categorised as [38]: a) *parametric* errors: arising from the closure coefficients, which are tuned based on a limited set of canonical flows, b) *functional* errors: arising from the mathematical formulations of the turbulence model variables, e.g. turbulent kinetic energy, k, and turbulent dissipation rate, ϵ, and c) *structural* errors: arising from the simplifying modelling assumptions, such as the Boussinesq approximation, which assumes that Reynolds stresses are linearly proportional to the mean strain rate. The three stated sources of errors relate specifically to the turbulence closure modelling, but there also exist mesh-related and numerical errors due to the finite dimensional discretisations of continuous systems in any CFD simulation (reviewed in Ref. [48]) which is beyond the remit of this thesis.

2.1 Uncertainty quantification and parametric error reduction

Some representative ideas, based on data-driven uncertainty quantification, are briefly outlined in this section—a comprehensive review has been conducted by Xiao and Cinnella [170].

Deterministic closure model constants, empirically tuned on experimental data for fundamental flows are one of the sources of uncertainties. Oliver *et al.* [118] and Cheung *et al.* [29] have used a Bayesian framework, by treating the closure coefficients of a number of eddy viscosity models as random variables, and used DNS data to calibrate the models by producing posterior probability density functions for the model constants. These, and similar studies, have shown that: a) there are no universal closure parameter values, b) parameters must be adjusted continuously when changing the dataset used for calibration, and c) closure coefficients calibrated on one dataset of flows are not necessarily valid for prediction of a different set of flows [170], thus highlighting the limitations of a parametric error reduction approach. Additionally, a fully Bayesian analysis based on sampling techniques can become computationally intractable for complex, high dimensional flows.

As opposed to parametric errors, Dow *et al.* [36] quantified model-form uncertainties in channel flows by leveraging DNS data to modify the eddy viscosity field that minimised the difference between the velocity field predicted by $k-\omega$ turbulence model and the DNS velocity field.

On the other hand, Ling *et al.* [88] utilises machine learning algorithms to classify regions of high RANS uncertainties related to functional and structural errors. They employ high-fidelity DNS and LES data for canonical flows to classify linear eddy-viscosity model breakdown in the entire flow domain due to the following modelling assumptions: isotropic eddy viscosity, linear Reynolds stresses

and strain rate relationship in the Boussinesq hypothesis, and non-negative eddy viscosity. Although, promising results were achieved in terms of the models capability to classify uncertain regions and the capability to generalise to unseen, albeit similar, flows, one of the key disadvantages of this method is the extent of high-fidelity data required—nearly impossible to acquire for practical aerospace applications.

Clearly uncertainty error quantification is important, but the turbulence modelling community have come up with a plethora of ideas to use data for reducing these errors, which will be reviewed in the following sections.

2.2 *A priori* learning

Most of the recent work on data-driven turbulence modelling has been concerned with enhancing the predictive capability of traditional models by directly addressing the sources of functional and/or structural errors. Among these, a considerable number attempt to train machine learning models directly on inputs and features taken from high-fidelity data for the Reynolds stress tensor—an approach known as *a priori* or CFD-free training [37].

Many *a priori* learning approaches start with a representation of the Reynolds stress tensor that addresses the inadequacies of the existing turbulence model. The Reynolds stress tensor, $\boldsymbol{\tau}$, based on eigen-decomposition can be expressed as [121],

$$\boldsymbol{\tau} = -\overline{u_i' u_j'} = 2k\left(\frac{1}{3}\boldsymbol{I} + \boldsymbol{b}\right) = 2k\left(\frac{1}{3}\boldsymbol{I} + \boldsymbol{V}\Lambda\boldsymbol{V}^T\right), \tag{2.1}$$

where $k = \frac{1}{2}\overline{u_i' u_i'}$ is the turbulent kinetic energy, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{b}$ is the normalised anisotropy or deviatoric part of the Reynolds stresses which can be expressed as a function of its eigenvalues Λ and eigenvectors $\boldsymbol{V}$. The eigen-decomposition is essential, because the turbulent kinetic energy, and eigenvalues and eigenvectors of the Reynolds stress anisotropy, represent the amplitude, shape

and orientation of the Reynolds stress tensor, and thus allow for a framework of addressing the discrepancies in the Reynolds stress tensor while ensuring physical realisability [42].

In one of the earlier attempts, Tracey *et al.* [152] used kernel regression and DNS data to build a machine learning model of the error in the turbulence anisotropy, and successfully showed *a priori* improvements to the baseline $k-\omega$ SST model for a limited set of flows. Specifically, for the machine learning model input they used: the eigenvalue anisotropy, Λ; the production-to-dissipation rate of the turbulent kinetic energy k; and a marker function which indicates the flow regions with high deviations from parallel shear flow—known to be poorly modelled by existing eddy-viscosity models. As the output, they selected the eigenvalues predicted by the DNS dataset. However, they used a very limited set of flow features for machine learning training, and did not consider the physical requirement of Galilean invariance*.

Wang *et al.* [159] used the Reynolds stress tensor representation in Eqn. 2.1, and built a random forest model of the Reynolds stress anisotropy based on DNS data for not only the eigenvalues, but also the turbulent kinetic energy and the eigenvectors. Additionally, as the machine learning inputs, they used a broader range of mean flow features (ten in total) which better characterised the flow, while also ensuring these were Galilean invariant. They achieved significant improvements in anisotropy predictions over the baseline models, however, these improvements did not necessarily translate to better predictions of propagated quantities of interest, such as the mean velocity field. Wu *et al.* [166], from the same group, assigned this issue to the limited set of flow features used for training the random forest regressor. To alleviate this, they constructed 47 invariant bases

*This is the property that requires that the physics of the fluid flow should be independent of coordinates frame of the observer. If this is not the case then identical flows will have different predictions if the axes are defined in different directions by observers moving at different velocities.

from the following non-dimensionalised mean flow features: the mean strain and rotation rates, the pressure gradient, and the gradient of the turbulent kinetic energy. With these, they demonstrated improved results for not only the Reynolds stress tensor, but also the associated propagated fields such as the mean velocity.

Ling *et al.* [91] argued that *a priori* training using traditional machine learning algorithms such as random forest will have limited success, due to the above difficulties in enforcing invariance properties, and proposed a novel neural network architecture, called tensor basis neural network (TBNN), shown in Figure 2.1. This framework embeded Galilean invariance, by exploiting the proof by Pope [120], who showed that a general eddy viscosity model of the Reynolds stress anisotorpy, $\boldsymbol{a}$ in Eqn. 2.1, can be expressed as linear combination of 10 isotropic basis tensors, which are functions of the mean strain and rotation rate tensors (which are functions of the gradient of mean velocity), and automatically satisfy Galilean invariance:

$$\boldsymbol{\tau} = 2k\left(\frac{1}{3}\boldsymbol{I} + \boldsymbol{a}\right) = 2k\left(\frac{1}{3}\boldsymbol{I} + \sum_{n=1}^{10} g^{(n)}\left(\lambda_1, ..., \lambda_5\right)\boldsymbol{T}^{(n)}\right), \tag{2.2}$$

where the goal of the tensor basis neural network is to determine the values of the scalar coefficients fields $g^{(n)}$, given the tensor invariants λ_i with $i \in [1, 5]$ are known functions of the strain and rotation rate tensors, as the model input, see Figure 2.1.

The TBNN was shown to make substantial improvement over existing linear and non-linear eddy viscosity models, and data-driven models with off-the-shelf neural network algorithms, both in terms of the predictions of the Reynolds stresses and other quantities of interest, such as the velocity fields. The framework was also shown to have good generalisability capabilities for improved predictions of cases not used in training. Other successful applications have been in turbulence scalar flux [107] and turbulent heat flux modelling [108] for jet in crossflow

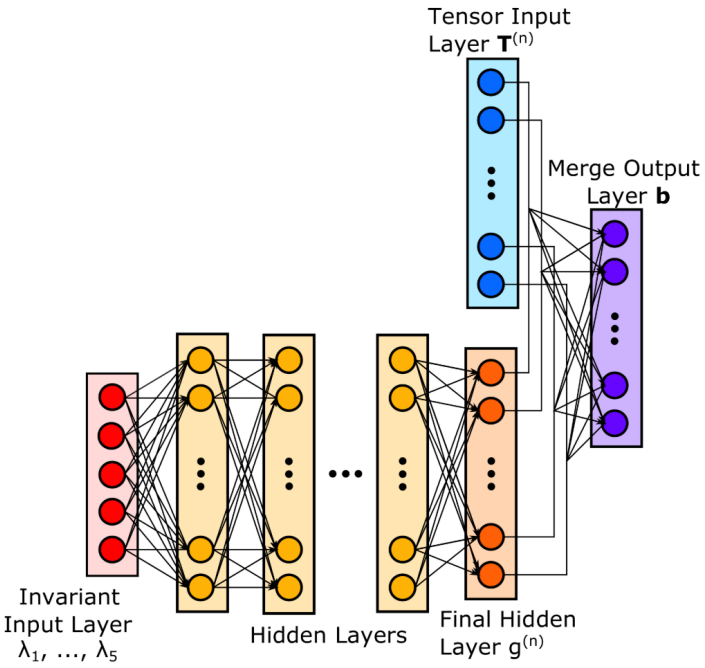


Figure 2.1: The tensor basis neural network architecture proposed by [91], with two input layers: the invariant input layers, for the five known tensor invariants λ_i, followed by hidden layers, where the final hidden layer has ten elements representing the scalar coefficient $g^{(n)}$ for $n = 1, ..., 10$, and the tensor input layer, representing the ten isotropic basis tensors, $\boldsymbol{T}$ in Eqn. 2.2. The output layer sums the results from the final hidden layer and the tensor input layer, to give the Reynolds stress anisotropy.

and film cooling flows, respectively. Kaandorp *et al.* [73] have successfully implemented a random forest regression version of this novel approach.

An aspect not entirely addressed in the frameworks reviewed so far, is that of interpretability—the ability of expressing the machine learning-based model closure in an explicit analytical form [22].

Weatheritt *et al.* [161] have attempted to use a regression technique based on evolutionary algorithms, called gene expression programming, to build explicit nonlinear models for the Reynolds stress anisotropy, of the form represented in Eqn. 2.2. They only considered the first four isotropic basis tensors, $\boldsymbol{T}$, and two tensor variants λ. One of the advantages of such a framework is that the explicit

form allows analysis of the relative importance of the different terms in the closure. The results show good improvements in the prediction of the Reynolds stresses and other quantities of interest, although there are still slight discrepancies in components of the Reynolds stresses tensor as some of the invariant-basis tensors in Eqn. 2.2 were ignored. This also adversely impacts the generalisability of the model for problems it was not trained on.

On the other hand, Schmelzer *et al.* [135] have proposed using linear symbolic regression to infer Reynolds stress anisotropy closures based on a library of candidate functions. They use three of the invariant bases in Eqn. 2.2, and two of the tensor invariants, λ_i, to model the Reynolds stress anisotropy. Over-fitting to data is avoided through sparsity-promoting regression: the chosen model is constrained such that the error reduction is balanced against complexity of the candidate model-form expression. In a similar vein, Beetham *et al.* [11] have used linear sparse regression to form explicit, non-linear, Galilean invariant turbulence closures for multi-phase flows.

Even though, *a priori* data-driven techniques have demonstrated promising results, there are some considerable drawbacks:

- the amount of high-fidelity data required for model training, which for aerospace applications are very challenging, if not impossible, to acquire;
- model consistency: while directly training on high-fidelity data may ensure consistency with the quantities of the data, it does not necessarily guarantee consistency in the predictive RANS setting, where these models operate after training [149]. These issues have been ascribed to the difference between features used for training, and those used for prediction, e.g. the DNS quantities for length and time scales do not have a one-to-one correspondence to the RANS equivalents; among other possible causes [37];

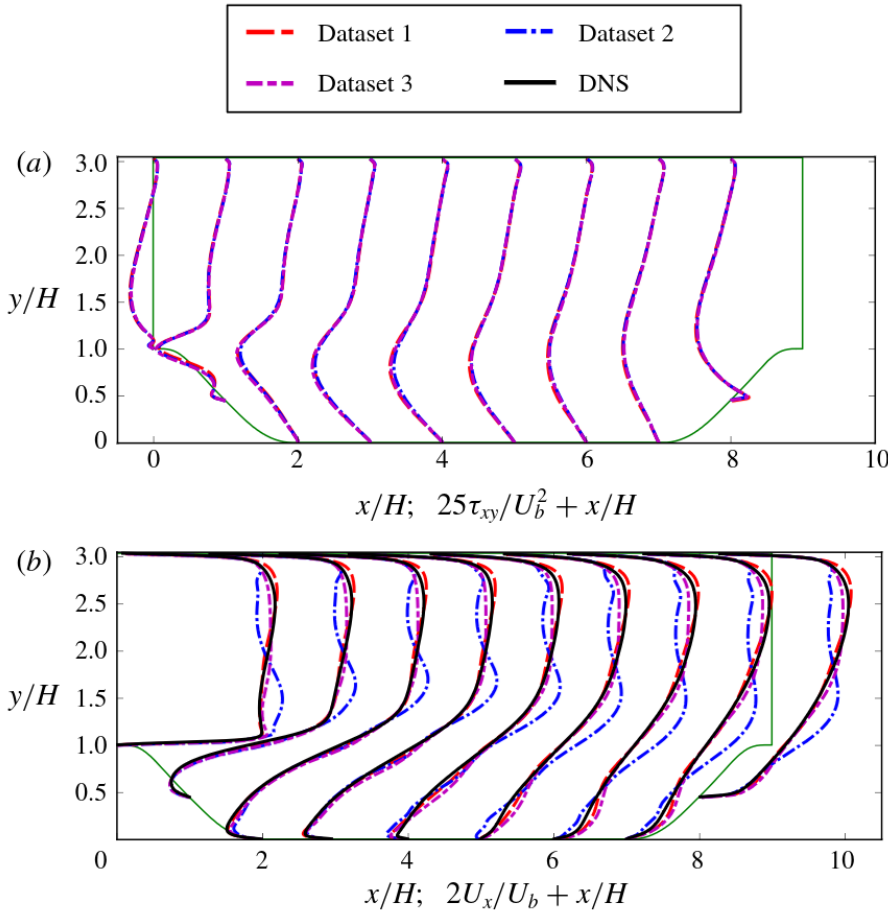


Figure 2.2: Ill-conditioning of RANS equations by explicit propagation of Reynolds stress tensor from high fidelity data. Figure from Wu *et al.* [167]. Dataset 1 is based on LES predictions of the Reynolds stress tensor using the LESOCC solver [23], Dataset 2 and 3 are DNS predictions using the Incompact3d solver [9] with mesh resolutions $512 \times 257 \times 128$ and $768 \times 385 \times 128$, respectively. (a) Reynolds shear stress profiles from all three datasets, (b) streamwise velocity profiles by solving the RANS equations where the Reynolds stress tensor is replaced by the high-fidelity data, instead of using a RANS-based closure model.

- ill-conditioning: Wu *et al.* [167] have shown that RANS models explicitly trained on high-fidelity Reynolds stress tensor data, can have major robustness and ill-conditioning issues in a predictive setting, especially when deviating from the training conditions, e.g. using data-driven model at higher Reynolds numbers than the training cases. They showed that when the

Reynolds stress tensor from DNS is explicitly used as a source term in the RANS equations, the predicted mean velocity fields can have significant discrepancies. This is because the mean velocity fields can be hugely sensitive to small errors in the Reynolds stresses. For example, Fig. 2.2 compares the streamwise velocity profiles in the periodic hill case (introduced in chapter 3.5.3) by solving RANS equations with explicit Reynolds stress tensor from three different high-fidelity datasets. It is clear that while the Reynolds shear stress is the same with all three datasets, there are differences in the streamwise velocity profiles, especially with Dataset 2.

2.3 Model consistent learning

To deal with model consistency issues, a number of frameworks have been proposed where the data-driven model training involves the imperfect RANS environment by explicitly involving the baseline turbulence model. This is done by solving a number of inverse problems where the goal is to minimise the discrepancy between the baseline RANS models and higher fidelity data, a so-called *field inversion* approach (sometimes also broadly termed data assimilation).

2.3.1 Field inversion and machine learning

In general, the field inversion problem is posed as follows: given some high-fidelity data Y, with the transport equation of the augmented model represented by $\mathcal{R}_a$, infer the spatiotemporal discrepancy field (also called the corrective, or augmentation field) $\boldsymbol{\delta}(\boldsymbol{x}, t)$ by solving the following optimisation problem [37]:

$$\min_{\boldsymbol{\delta}} \quad \mathcal{L}\left[Y, Y_m\left(\boldsymbol{\delta}\right)\right], \quad \text{s.t.} \quad \mathcal{R}_a\left[\boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{\delta}\right] = 0, \tag{2.3}$$

where Y_m are the model outputs corresponding to the quantity from data; and the transport equations represented by $\mathcal{R}_a$ are functions of the RANS state variables, $\boldsymbol{Q}$, such as velocity and pressure; secondary model variables, $\boldsymbol{T}$ such as eddy viscosity and turbulent quantities. While the notation $\boldsymbol{\delta}(\boldsymbol{x}, t)$ is used for a generalised field inversion formulation, we will be using β for the spatial field perturbing the production term of the turbulence transport equation(s) in the following chapters.

Field inversion on its own has limited capability, as the corrective field $\boldsymbol{\delta}(\boldsymbol{x}, t)$ is specific to the case for which the optimisation problem is solved. In other words, it cannot be used for prediction of a different case — which is the ultimate goal. Therefore, field inversion must be performed for a representative number of cases to build a set of augmentation fields $\boldsymbol{\delta}^i = [\boldsymbol{\delta}^{(1)}, ..., \boldsymbol{\delta}^{(n)}]$, and then machine learning tools can be used to map the fields in terms of a set of local flow features $\boldsymbol{\eta}^i = [\boldsymbol{\eta}^{(1)}, ..., \boldsymbol{\eta}^{(n)}]$ which are functions of the baseline model. A supervised learning process can be expressed as follows:

$$\min_{w} \quad \mathcal{L}[\boldsymbol{\delta}^i(\boldsymbol{x}, t), \boldsymbol{\delta}(\boldsymbol{\eta}^i; w)], \tag{2.4}$$

where w are the machine learning model parameters (weights and biases), and $\boldsymbol{\delta}_m$ is the machine learning model of the corrective fields that can be injected in the baseline model $\mathcal{R}_a$ for improved predictions.

Aside from model consistency, one of the advantages of these approaches is that, in principle, even sparse measurement data—which is a realistic scenario for aerospace experimental datasets—can be used to augment existing models.

One of the earliest model consistent approaches, is the field inversion and machine learning (FIML) framework, shown in Figure 2.3, originally proposed by Duraisamy *et al.* [40] and further explored by Parish *et al.* [119]. As a proof-of-concept they augmented the popular $k-\omega$ model to improve predictions for planar

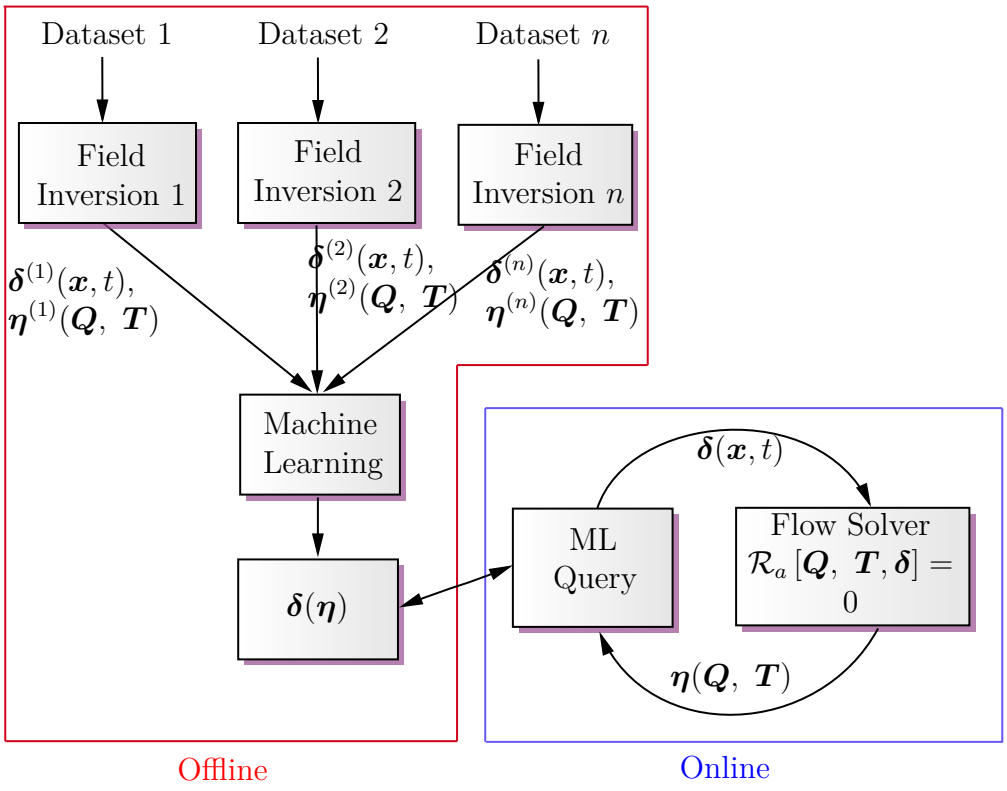


Figure 2.3: The processes in the field inversion and machine learning framework, based on the diagram from Ref. [137].

channel flows. In the field inversion step, they modified the transport equation for the turbulent kinetic energy, k, by multiplying the production term with a scalar field $\boldsymbol{\beta}(\boldsymbol{x})$ (the discrepancy field, generally represented by $\boldsymbol{\delta}$ in the discussion above). Note that this field is a scalar value defined across the entire flow domain in the CFD mesh. The optimisation problem was posed as a Bayesian inversion, where the goal was to reduce the error between the velocity profiles from DNS data, and the baseline model prediction.

The results were promising as the FIML-augmented model provided better predictions compared to the baseline case when simulating channel flows at different Reynolds numbers compared to the training. Singh *et al.* [138] extended the field inversion step to more complex cases such as separated flow over aerofoils

where substantial improvements were achieved with sparse data such as pressure coefficient measurements on the aerofoil surface. In a follow-up publication they also used neural networks to successfully model the field inversion corrective fields using just lift coefficients for training: data-driven model trained on two horizontal-axis wind turbine aerofoils at different incident angles and Reynolds numbers was able to improve predictions of separated flow on an unseen aerofoil with different operating conditions [139].

The results above also highlighted some challenges. Firstly, a fully Bayesian inversion requires the use of sampling algorithms which quickly become computationally intractable because the number of design variables for optimisation is the same as the number of mesh cells, which for aerospace flows can easily be of $\mathcal{O}(10^5)$ and higher. They bypassed this problem by approximating the posterior distribution of $\boldsymbol{\beta}(\boldsymbol{x})$ by its maximum a posterior (MAP) estimate [6], or formulating it as a regularised deterministic optimisation problem [139]. Secondly, a fully Bayesian inversion also requires high quality statistics related to the data to build the full covariance matrices, which is not always available. Finally, the highly dimensional optimisation problem is computationally expensive, since a model consistent framework by definition involves the governing equation in the objective function (Eqn. 2.3), so for gradient-based iterative optimisation methods the derivatives of model state variables and secondary variables must be computed. This problem can be resolved by using the adjoint method. However, the software implementation is an entry barrier to most researchers [37].

Singh *et al.* [39] also explored a different formulation of field inversion, by introducing the discrepancy field in the Reynolds stress anisotropy through the eigenvalues, as in some of the *a priori* formulations described previously. But unlike the *a priori* formulation where the DNS/LES anisotropy fields were used for training, they trained the FIML model on DNS mean velocity data alone.

Application to turbulent channel flow yielded interesting results: although the FIML-model resulted in improvements in the mean velocity field and Reynolds shear stress predictions, it failed to improve the Reynolds stress anisotropy.

Belligoli *et al.* [12] have also sought to address the Reynolds stress anisotropy using field inversion, by perturbing the Reynolds stress tensor, through its eigenvalues and eigenvectors. They ensure physical realisability by guaranteeing the Reynolds stresses are bounded by the barycentric map: a triangular domain proposed by Banerjee *et al.* [8] for visualising physically realisable turbulence states. While shown to make some improvements over the original formulations, the authors do not directly compare the predicted Reynolds stresses with high-fidelity data. Thus it is unclear if this method can improve anisotropy predictions. The technique is also computationally expensive as the number of distinct optimisation variables increases from one in the original formulation (i.e. $\boldsymbol{\beta}(\boldsymbol{x})$), to six (three to perturb the eigenvalues and three for eigenvector perturbations). Yet still, Volpiani *et al.* [156] have used a different formulation of filed inversion by introducing an additive vectorial source term as the corrective field in the momentum equation of the RANS equations, Eqn 1.4. While demonstrated to improve the baseline Spalart-Allmaras model for the periodic hill flow, the results are not compared to original formulation, nor are the turbulent quantities compared between the baseline model and their field inversion results – still leaving the effectiveness of the new formulation over the original approach an open question.

Holland *et al.* [65] proposed the so-called "direct" FIML approach. They argue that while field inversion step is model consistent, the learnability of the discrepancies from field inversion cannot be guaranteed. Therefore, they coupled the neural network learning of the discrepancy field as part of the field inversion process to promote a self-consistent framework. Directly optimising the neural network parameters while perturbing the turbulence model means that the gradi-

ent computations become more expensive. This could also make the optimisation process difficult since the CFD model is now a direct part of the training, whereas in the original FIML framework the neural network is trained on static data. However, advantages include: a self-consistent framework where no offline training is required, and the correction is optimally regularised for the chosen neural network structure, since the limitations of the network structure is considered during the inversion.

The FIML approaches discussed thus far have relied on neural network-based augmentation of turbulence models, which is considered a black-box approach. Jäckel adapted the FIML approach, such that a close-form correction to the Spalart-Allmaras turbulence model can be achieved using Gaussian radial basis functions [68]. It was shown that the corrective field ($\boldsymbol{\beta}$) from field inversion can be described in an algebraic form with only seven parameters and improve the baseline model for a number of separated aerofoil flows. In a similar vein, Wu *et al.* [165] replaced the neural network-based learning algorithm with symbolic regression to learn analytical relationship between the corrective field and the local flow features using the $k-\omega$ SST model. The interpretable augmented $k-\omega$ SST model showed promising generalisability capacities [165].

Other notable variants of the original FIML approach include its application in transitional modelling using the ensemble-based field inversion approach and random forests [172, 173], and using the adjoint-based field inversion and neural networks [44]. Fidkowski also utilised the FIML approach for output-based error estimation and mesh adaptation in the context of unsteady turbulent flows [46]. Specifically, FIML is used to forego the need for unsteady adjoint solutions (computationally expensive for flows of industrial interest, and prone to being unstable for chaotic systems), and instead provide an adjoint of the averaged solution combined with averaged unsteady flow residual. Recently, Fang *et al.* [43] developed

a framework for field inversion for unsteady flow simulations, by implementing a time-accurate segregated adjoint method.

The FIML approach has received considerable attention due to the advantages it offers: model consistency, addressing functional errors in baseline models, and ability to improve turbulence model predictions with relatively sparse datasets, such as those measured in experiments. In literature, particular attention has been paid to various formulations of the FIML approach, as discussed above, however, the following open questions remain:

- What are effective sensor configurations to experimentally collect sparse data for use in data assimilation? Is it possible to systematically design the placement of sensors without the need to undertake the experiments in the first place? Can we utilise existing insights on the underlying assumptions leading to the uncertainties and inaccuracies in existing RANS-based turbulence models to place sensors?
- Which physical quantities, if any, are most effective for data assimilation?
- Can adjoint-based data assimilation be enhanced using disparate data, i.e. multiple physical quantities in different parts of the flow domain with varying dimensions and quality?
- Can the FIML approach be extended to develop customised turbulence model augmentation for aerodynamic shape optimisation?

2.3.2 Other model consistent approaches

Other model-consistent approaches besides FIML, include the CFD-driven version of the gene-expression programming-based turbulence model augmentation by Zhao *et al.* [176]. They propose modifying the baseline $k-\omega$ SST model by introducing an additional term to the Reynolds stress tensor ($\boldsymbol{\tau}$, Eqn. 2.1),

which will be learnt using an evolutionary algorithm. Similar to the FIML approach, a cost function is constructed based on a given quantity of interest from the high-fidelity data and the RANS-based predictions. Saïdi *et al.* [13] have also proposed a similar framework using deterministic symbolic regression and the modifications of the production terms of the baseline $k-\omega$ SST turbulence model. The advantages of these approaches include interpretability, and easier implementation in existing RANS codes. While promising results have been achieved for two-dimensional turbulent flows, further investigations for more complex three-dimensional flows are required.

While the current review has been mainly concerned with data-driven turbulence closure modelling in the context of Reynolds-averaged Navier Stokes equations which is the remit of the present work, there has been sustained interest on the use of data-driven methods in fluid dynamics more broadly. Examples include discovering governing equations [130, 126] for simple flows; using physics-informed neural networks to directly simulate incompressible Navier-Stokes equations for laminar and low Reynolds number turbulent flows [71]; flow control through deep learning reviewed by Rebault *et al.* [125]; development of reduced-order models of complex flows review by Taira *et al.* [150]; mesh adaptation [47]; data-driven closure models for large-eddy simulations reviewed by Beck *et al.* [10]; and aerodynamic shape optimisation reviewed by Li *et al.* [85].

2.4 Summary

This literature review has highlighted a number of research gaps in data assimilation for turbulence modelling. The first gap is the lack of a standard, open-source,

computational software tool for adjoint-based field inversion. This lack of software tool is important because it is a barrier to developing research methods in this domain. Therefore, chapter 3 in this thesis develops an open-source computational tool for adjoint based data assimilation and demonstrates the use of the tool on some standard field inversion problems. This software tool is then used throughout the thesis in all subsequent chapters. The second gap is the lack of attention paid to sensor placement in adjoint-based field inversion. This is important because data-driven methods are highly dependent on the data collected through experiments. This challenge is addressed in chapter 4, where a sensor placement strategy is developed based on the technique of eigenspace perturbations - a more specialised literature review of this method is given in the relevant chapter. The third research gap highlighted is the lack of attention given to the use of multiple, different types of sensor data in field inversion - current work has focused only on a single sensor type. This gap is addressed in chapter 5, where a multi-objective optimisation approach is developed for field inversion, which can incorporate multiple different types of sensor data. The fourth and final gap addressed here, is the lack of methods where field inversion is used to perform a useful task such as in aerodynamic shape optimisation. Therefore, chapter 6 and 7 contribute a new method of shape optimisation that uses the field inversion methods developed through the thesis.

CHAPTER 3

Adjoint-based data assimilation

3.1 Introduction

This chapter addresses the problem of developing an open-source, computational software tool for adjoint-based data assimilation. This is an important gap in the literature because the lack of an open software tool is a barrier to research being done in this domain. The novel contribution of this chapter is therefore an open-source computational software tool, which has been published [18] and incorporated into the computational package DAFoam*, an extension to OpenFOAM for adjoint-based optimisation. The novel addition from this thesis extends the capability of DAFoam to performing adjoint-based field inversion.

In this chapter, we:

- formulate the DA task as a deterministic inverse problem which is ultimately a gradient-based optimisation problem,
- introduce the discrete adjoint approach for efficient gradient computations during the optimisation,
- summarise the implementation of the DA approach in an open-source framework,

*https://github.com/mdolab/dafoam

- and demonstrate the capabilities of the data assimilation implementation for complex turbulent flows for three commonly used turbulence models (Spalart-Allmaras, Wilcox $k-\omega$, and $k-\omega$ shear stress transport (SST)), and three different types of data (i.e. integral data in the form of lift coefficient, surface data in the form of skin friction and surface pressure, and volume/field data in the form velocity profiles and fields).

By data assimilation in this work, we refer to the use of high-fidelity data[†] to reconstruct turbulent mean flows. This process is also termed field inversion in literature. Broadly, two approaches to field inversion has been pursued in parallel: ensemble-based, and adjoint-based methods. Xiao and colleagues, amongst others, have used the ensemble-based Kalman filter (EnKF) approach to model the discrepancy in baseline turbulence models through eigenvalue perturbations [169]. The advantages of this method include: relatively easy code development, and the ability to compute confidence bounds for the quantity/quantities of interest, albeit, complicated by the need to map the control parameters from a high to low-dimensional space to reduce computational costs.

The alternative adjoint-based field inversion method originally proposed by Duraisamy and co-workers relies on solving a gradient-based optimisation problem, where the derivatives of the cost function is computed using the adjoint approach. A number of different variants of this method has been studied: perturbations of the Reynolds stress anisotropy eigenvalues [39]; perturbations of the eigenvalues as well as the eigenvectors of the anisotropy tensor [12]; and modification of the turbulent model transport equation through a spatial scalar field defined over the entire flow domain [119, 139, 66]. In this work, we use the latter since it has proven to be simple and computationally cheap relative to the other variants. The adjoint-method is, in principle, capable of recovering finer scales of

[†]Data from experimental measurements and/or turbulence resolving simulations such as LES and DNS.

turbulent mean flow compared to the ensemble-based method. However, a fully Bayesian formulation with this method is computationally expensive [119], and most researchers have opted for deterministic formulations.

Another limitation of the adjoint-based field inversion is the time-consuming and laborious software development process due to the intrusive nature of adjoint-based optimisation, which is a high barrier to entry for many researchers. We tackle this problem by introducing an open-source tool which interested practitioners can use to apply and investigate the adjoint-based field inversion method.

3.2 Field inversion: mathematical formulation

The incompressible, steady Reynolds-averaged Navier-Stokes equations, Eqn. 1.2, are re-stated:

$$\frac{\partial U_i}{\partial x_i} = 0, \tag{3.1}$$

$$\frac{\partial U_i}{\partial t} + \frac{\partial (U_i U_j)}{dx_j} = -\frac{\partial P}{dx_i} + \frac{1}{Re}\frac{\partial^2 U_i}{\partial x_j \partial x_j} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j}, \tag{3.2}$$

where $\tau_{ij} = -\overline{u_i' u_j'}$ is the Reynolds stress tensor. Employing the Boussinesq assumption, a linear eddy viscosity model for the stress tensor is derived [81]:

$$\tau_{ij} = 2\nu_t S_{ij} - \frac{2}{3} k \delta_{ij}, \tag{3.3}$$

$$\text{with} \quad S_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \tag{3.4}$$

where ν_t is the turbulent eddy viscosity, S_{ij} is strain rate, k is the turbulent kinetic energy and δ_{ij} is the Kronecker delta function. The turbulence model can be introduced through the turbulent kinematic viscosity, ν_t. In this work we have explored data assimilation with three of the most commonly used turbulence models:

1. Spalart-Allmaras [143] (equations in Appendix A.1):

$$\nu_t = \tilde{\nu} f_{v1}, \tag{3.5}$$

where $\tilde{\nu}$ is a surrogate eddy viscosity, f_{v1} is an intermediate (Eqn. A.2). In the standard S-A model the term involving the turbulent kinetic energy, k, in Eqn. 3.3 is ignored since it is not readily available.

2. Wilcox $k-\omega$ [163] (equations in Appendix A.2):

$$\nu_t = \frac{k}{\omega}, \tag{3.6}$$

where ω is specific dissipation rate.

3. $k-\omega$ shear stress transport (SST) [105, 106] (equations in Appendix A.3):

$$\nu_t = a_1 \frac{k}{\max\left(a_1\omega, SF_2\right)}, \tag{3.7}$$

where a_1 is a model constant, S is the strain rate (Eqn. 3.4), and F_2 is an intermediate function (Eqn. A.9).

The transport equation(s) for the turbulent variables, $\boldsymbol{T}$, (e.g. $\tilde{\nu}$ in the S-A model, or k in the two equation models), are of the following general form:

$$\frac{DT}{Dt} = \mathcal{P}(\boldsymbol{Q}, \boldsymbol{T}) + \boldsymbol{T}(\boldsymbol{Q}, \boldsymbol{T}) - \mathcal{D}(\boldsymbol{Q}, \boldsymbol{T}), \tag{3.8}$$

where $\boldsymbol{Q}$ represents the conversed flow variables such as velocity and pressure, the source terms $\mathcal{P}$, $\mathcal{T}$, and $\mathcal{D}$ represent the turbulence production, transport, and dissipation.

The adjoint-based field inversion approach aims to recover the *functional* discrepancy in the existing turbulence model. This is achieved by perturbing the

production term of the transport equation, Eqn. 3.8 by a spatial scalar field, $\boldsymbol{\beta}(\boldsymbol{x}) \in \mathbb{R}^{N_{\text{cells}}}$,

$$\frac{DT}{Dt} = \boldsymbol{\beta}(\boldsymbol{x}) \cdot \mathcal{P}(\boldsymbol{Q}, \boldsymbol{T}) + \mathcal{T}(\boldsymbol{Q}, \boldsymbol{T}) - \mathcal{D}(\boldsymbol{Q}, \boldsymbol{T}), \tag{3.9}$$

where $\boldsymbol{\beta}(\boldsymbol{x}) = 1$ recovers the baseline model. For most linear eddy viscosity models, by modifying the production term, the corrective field directly influences the Reynolds stress term, e.g. $\mathcal{P}_k = \tau_{ij}\partial U_i/\partial x_j$ in the Wilcox $k-\omega$ model (Appendix A.2), and most studies have chosen to follow this approach (e.g. Refs. [119, 139, 12, 45]). In principle, it is also possible to perturb the other terms (e.g. Ref. [165] modifies the destruction term of the ω equation in the SST model), or perturb the RANS momentum equation by adding a source term [156]. Recently, Cato *et al.* [26] compared six different types of corrections to eddy viscosity models. Their results illustrated that while all the approaches led to improvements to the baseline model, adding a source term to the momentum equation was the most effective.[‡]

For the Spalart-Allmaras model the perturbation is straight-forward as it involves a single transport equation for turbulence modelling. For models with two or more transport equations, e.g. Wilcox $k-\omega$, or SST, previous studies have opted to perturb the turbulent kinetic energy (k) (e.g. [137]), or the turbulent dissipation (ω) (e.g. [12, 64, 165]). We follow the latter approach as we have found that the equation for the turbulent kinetic energy is very sensitive to the $\boldsymbol{\beta}$ perturbations and to numerical instabilities and diverging flow predictions, while the dissipation rate ω is relatively more robust to the perturbations. Not modifying the transport equation for the turbulent kinetic energy may restrict the impact of field inversion for data assimilation. Future work will investigate simultaneously perturbing both k and ω.

[‡]Majority of the work in this thesis were completed prior to this publication.

After the perturbation, the following optimisation problem is solved,

$$\min_{\boldsymbol{\beta}^{\mathrm{FI}}} \quad \mathcal{L} = \|q_i^{\mathrm{RANS}}\left(\boldsymbol{\beta}^{\mathrm{FI}}\right) - q_i^{\mathrm{data}}\|_2^2 + \lambda\|\boldsymbol{\beta}^{\mathrm{FI}} - 1\|_2^2, \tag{3.10}$$

$$\text{subject to} \quad \boldsymbol{R}(\boldsymbol{Q}, \boldsymbol{T}) = \mathbf{0}, \tag{3.11}$$

where Eqn. 3.10 is the loss/objective function, and Eqn. 3.11 represents the RANS equations. $q_i^{\mathrm{data}} \in \mathcal{D}$ represents high-fidelity physical quantities of size N_s, q_i^{RANS} represents the equivalent quantity predicted by the turbulence model, and $\|.\|_2$ is the L_2 norm. The first term in the equation is normalised by the baseline L_2 norm such that its value is unity at the start of the optimisation. The second term in Eqn. 3.10 tuned by the relaxation parameter λ is to regularise the problem by avoiding huge deviations from the baseline turbulence model, avoiding an ill-posed optimisation problem and reducing the risk of over-fitting.

As previously mentioned in Chapter 3.1, the highly dimensional nature of the optimisation (i.e. number of design parameter is equivalent to the number of mesh cells) means gradient-free approaches will be computationally expensive even for very simple flows. Thus, a gradient-based approach with efficient gradient evaluations is required.

3.3 Discrete adjoint approach

In terms of the adjoint implementation there are two approaches: continuous and discrete adjoint. He *et al.* [61] have applied the field inversion framework to the Spalart-Allmaras model using a continuous adjoint implementation. In this framework, the adjoint equations for the governing equations are derived in the continuous form, and then discretised for numerical solutions. They achieved promising results on a number of cases that included a three-dimensional wall-mounted cube. The continuous adjoint method has the advantages of low com-

putational cost due to lower memory requirement and is simpler to implement in existing CFD codes. However, derivations of the adjoint equations are complex, and has to be repeated for every new turbulence models, boundary conditions, and objective function formulations (depending on what quantity is used from data) [75]. To avoid these, we use the discrete-adjoint method.

In the discrete approach the adjoint equations are derived for the discretised governing equation from the outset. The advantages are: ability to achieve more accurate gradient information since these are consistent with the discretised objective function evaluations; and the ability to use algorithmic differentiation (AD) which does not require an updated adjoint equation derivation for every new model, boundary condition, or objective function formulation. However, using AD to compute partial derivatives (to be introduced in section 3.3.1) can lead to large memory requirements. AD relies on using the chain rule to differentiate the sequence of elementary arithmetic operations that constitute a computer code, by producing differentiated version of the code [115]. There are two derivative computation modes for AD: forward, and reverse-mode. The latter is computationally much more efficient when one is interested in the sensitivity of a small number of output quantities with respect to a large number of input parameters [53], which is the case for data assimilation. The memory requirement in reverse-mode AD arises from the need to store intermediate variables for use in reverse accumulation of the derivatives [75]. However, the memory requirements can be overcome if the source-transformation implementation of AD is utilised [113].

3.3.1 Discrete adjoint equations

The derivations here are based on the theory presented in [75]. The goal is to compute the derivative $\mathrm{d}\mathcal{L}/\mathrm{d}\boldsymbol{\beta}$ efficiently, where the scalar $\mathcal{L}$ is the objective function, and $\boldsymbol{\beta} \in \mathbb{R}^{n_\beta}$ is the vector of design variables. As mentioned earlier it is

assumed that the governing equations are available in a discretised form, and the discrete residual equations $\boldsymbol{R}(\boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{\beta}) = \boldsymbol{0}$, where $\boldsymbol{w} = [\boldsymbol{Q}, \boldsymbol{T}]$ is the vector of all state variables, is satisfied.

Using the chain rule the derivative $\mathrm{d}\mathcal{L}/\mathrm{d}\boldsymbol{\beta}$ can be expressed as,

$$\underbrace{\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\beta}}}_{1 \times n_{\boldsymbol{\beta}}} = \underbrace{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}}}_{1 \times n_{\boldsymbol{\beta}}} + \underbrace{\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}}_{1 \times n_w} \underbrace{\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}\boldsymbol{\beta}}}_{n_w \times n_{\boldsymbol{\beta}}}, \tag{3.12}$$

where the computational cost of computing the partial derivatives $\partial\mathcal{L}/\partial\boldsymbol{\beta}$ and $\partial\mathcal{L}/\partial\boldsymbol{w}$ is relatively cheap as these only involve explicit computations. However, the total derivative matrix $\mathrm{d}\boldsymbol{w}/\mathrm{d}\boldsymbol{\beta}$ must be implicitly determined by the residual equations $\boldsymbol{R}(\boldsymbol{\beta}, \boldsymbol{w}) = 0$, and is thus computationally expensive. To compute the total derivative $\mathrm{d}\boldsymbol{w}/\mathrm{d}\boldsymbol{\beta}$, the chain rule can be applied for $\boldsymbol{R}$, noting that $\mathrm{d}\boldsymbol{R}/\mathrm{d}\boldsymbol{\beta}$ must equal zero in order for the governing equations to satisfy $\boldsymbol{R}(\boldsymbol{\beta}, \boldsymbol{w}) = 0$:

$$\frac{\mathrm{d}\boldsymbol{R}}{\mathrm{d}\boldsymbol{\beta}} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\beta}} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{w}} \frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}\boldsymbol{\beta}} = 0, \tag{3.13}$$

which can be expressed as the following linear system

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}\boldsymbol{\beta}} = -\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{w}}^{-1} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\beta}}. \tag{3.14}$$

The expression for $\mathrm{d}\boldsymbol{w}/\mathrm{d}\boldsymbol{\beta}$ is now substituted in Eqn. 3.13 to get the following:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\beta}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} - \underbrace{\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{w}}^{-1}}_{\boldsymbol{\psi}^T} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\beta}}, \tag{3.15}$$

where $\boldsymbol{\psi}$ is the adjoint vector. Transposing the state Jacobian matrix $\partial\boldsymbol{R}/\partial\boldsymbol{w}$ and solving with $[\partial\mathcal{L}/\partial\boldsymbol{w}]^T$ as the right-hand side of Eqn. 3.15 yield the adjoint equation,

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{w}}\boldsymbol{\psi} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}^{T}. \tag{3.16}$$

Having solved Eqn. 3.16, the total derivative $\mathrm{d}\mathcal{L}/\mathrm{d}\boldsymbol{\beta}$ can now be computed by substituting the adjoint vector $\boldsymbol{\psi}$ into Eqn. 3.15 as follows:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\beta}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} - \boldsymbol{\psi}^{T}\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\beta}}. \tag{3.17}$$

The efficiency of the adjoint approach can now be illustrated using the fact that for each computation of the objective function, the adjoint equation is only solved once since the design variable is not explicitly present in Eqn. 3.16. In other words, the computational cost is proportional to the number of objective functions, e.g. one for our purposes, and is independent of the number of design variables, e.g. the number of mesh cells, $\mathcal{O}(10^5)$ for our purposes.

The partial derivative $\partial \boldsymbol{R}/\partial \boldsymbol{\beta}$ in Eqn. 3.17 is solved using reverse-mode algorithmic differentiation (AD) utilising the CoDiPack (Code Differentiation Package) suite [97] which relies on the operation-overloading approach to performing AD [115].

3.4 Implementation

A high-level flow chart for the field inversion process is shown in Fig. 3.1. The process involves:

1. solution of the governing equations, including the baseline turbulence model using OpenFOAM,

2. using these results to compute the objective function (Eqn. 3.10),

3. computing the derivative of the objective function with respect to the design variable, ($\boldsymbol{\beta}$), using DAFoam,

4. using an optimiser to update the $\boldsymbol{\beta}$ field such that the least-squares difference between model predictions and data is minimised, using the optimiser suite pyOptSparse,

5. and repeating steps 1-4 until a user-specified optimisation convergence criterion has been met.

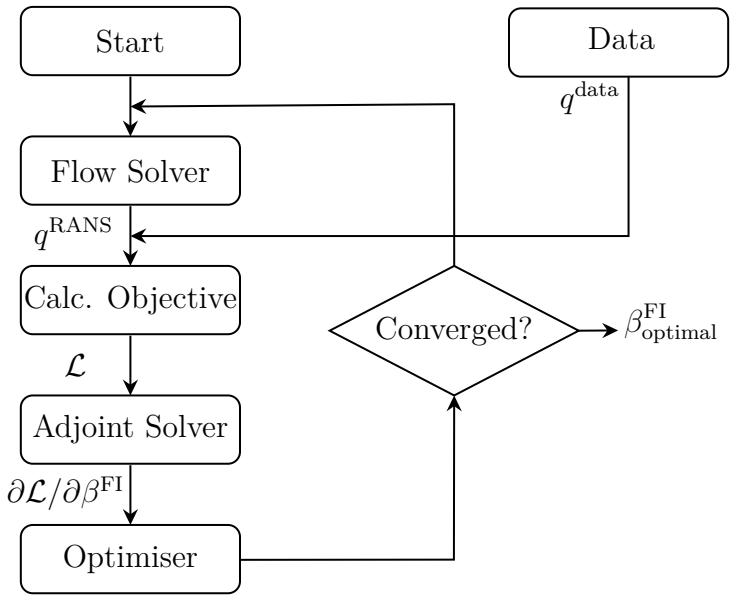


Figure 3.1: Flow diagram for the iterative adjoint-based method for field inversion.

Flow solver: OpenFOAM

Open field operation and manipulation (OpenFOAM) is popular, open-source CFD package, based on the finite volume method and written in C++, with an active developer and user base [69]. In this work, we have employed two of its main solvers: `simpleFoam`, used for the solution of the steady Navier-Stokes equations for incompressible fluids, and `rhoSimpleFoam` which is a compressible steady-state solver. Both solvers, use the semi-implicit method for pressure-linked equations (SIMPLE) algorithm to solve the coupled continuity and momentum equations.

Many of the popular turbulence models are available on OpenFOAM, and the modified Spalart–Allmaras and $k-\omega$ turbulence models for field inversion, are based on the original OpenFOAM implementation. The field inversion models require minimal code modification, and extensions to other turbulence models do not require deep C++ programming skills.

Adjoint solver: DAFoam

DAFoam is another open-source package, that allows effective adjoint solutions. It has been specifically tailored for OpenFOAM and has been successfully applied for multi-disciplinary design optimisation [62].

Large-scale optimiser: pyOptSparse

Once the total derivative $\partial \mathcal{J}/\partial \boldsymbol{\beta}$ is computed, we use the pyOptSparse package to find the optimal $\boldsymbol{\beta}$ field that minimises the objective function. pyOptSparse is a object-oriented Python interface of various optimisers which can be used for formulating and solving constrained nonlinear optimisation [168]. It allows efficient handling of large-scale optimisations through the use of sparse matrices in the code. We have successfully employed two of the large-scale optimisation packages available in pyOptSparse: SNOPT (based on sequential quadratic programming [54]), and IPOPT (based on a primal-dual interior point method [158]). All the field inversion results presented in the thesis are based on the IPOPT optimiser.

Python user interface

A high-level Python layer is used to set and run the field inversion simulation process outlined in Fig. 3.1. Specifically, the following parameters are set in a Python script: primal flow solver (e.g. `simpleFoam`); flow solver boundary conditions and residuals convergence tolerance; the field inversion objective function

specialisation and the relevant parameters (so far, the following have been implemented: full fields, velocity profiles, surface pressure and skin friction, and aerodynamic force coefficients) and the regularisation constant, λ in Eqn. 5.1; adjoint solver parameters such as state normalisation constants, and the equation solution options; and finally the optimiser and its parameters such as $\boldsymbol{\beta}$ field constraints (upper and lower bounds), convergence tolerance, maximum number of iterations, etc.

Enabling field inversion with DAFoam

Having been designed for multidisciplinary design optimisation, DAFoam had many preexisting capabilities to enable field inversion–the process shown in flow diagram in Fig. 3.1. Here we summarise the additional capabilities in DAFoam to enable field inversion:

- Flow solver: we added $\boldsymbol{\beta}$ modified versions of the Spalart-Allmaras, Wilcox $k-\omega$ and $k-\omega$ SST models.

- Objective functions: we added a new class needed to compute the loss functions based on the different reference high-fidelity data. This included adding capability to read the reference data from disk, and compute the equivalent RANS predictions for the loss function calculations.

- Adjoint solver: algorithmic differentiation capability was introduced to compute the partial derivatives in the adjoint equations. Previously, finite-differencing was used to compute these derivatives.

- Optimiser: originally, DAFoam mainly relied on the SLSQP and SNOPT optimisers from pyOptSparse. SLSQP is freely available, but only works with relatively small number of design variables. SNOPT requires a paid license. Since, field inversion involves a very large number of design variables,

we enabled the free open-source large-scale optimiser IPOPT in DAFoam as an alternative optimiser to SNOPT.

- To enable neural-network augmentation of turbulence models (to be detailed in Chapter 6) we also enabled the calculations of various neural network inputs/features, and the coupling with TensorFlow.

- All the additions have been incorporated on the DAFoam GitHub repository: https://github.com/mdolab/dafoam.

3.5 Results

In this section we demonstrate data assimilation using the adjoint-based field inversion approach on a number of cases, summarised in Table 5.1. The selected cases involve both internal and external flows, with the Reynolds number ranging $5.6 \times 10^3 - 2 \times 10^6$. All the cases involves various levels of flow separation, which is inaccurately predicted by commonly employed turbulence models.

These cases were chosen based on data availability, and to demonstrate likely field inversion scenarios based on types of data used for flow reconstruction. Generally, three types of data sources can be considered: *integral data* (e.g. lift or drag coefficients), *surface data* (e.g. surface pressure, or skin friction), and *volume data* (e.g. velocity fields/profiles at certain locations). We consider all three scenarios in the following sections.

3.5.1 Aerofoil flows at high angles of attack

Existing turbulence models are known to perform poorly in predicting the flow over aerofoils at high incidence angles, where the flow generally separates. We utilise field inversion to improve the prediction of separated flow over the NREL

Table 3.1: Summary of case setups, where C_l, C_p, C_f, and U_x are the lift coefficient, pressure coefficient, skin friction coefficient, and the streamwise velocity, respectively. Dimension column refers to the size of the data used.

Case	Geometry	Data	Data size	Data source	Turbulence model	Re
1a	S809 Aerofoil	C_l	1	Integral	Spalart–Allmaras	2×10^6
1b		C_p	32	Surface		
2	NACA 4412	U_x	41,968	Volume	Spalart-Allmaras	1.52×10^6
3a	Conv.-Div. Channel	C_f	564	Surface	Spalart–Allmaras	12,600
3b		C_p	564	Surface		
3c		U_x	1,400	Volume		
3d		U_x	98,000	Volume		
4a	Periodic Hill	U_x	447	Volume	Spalart–Allmaras	5,600
4b					Wilcox $k-\omega$	
4c					$k-\omega$ SST	

S809 horizontal-axis wind turbine section, also investigated in [139, 12], and the NACA 4412 aerofoil.

S809 aerofoil: field inversion using C_l and C_p data

Experimental studies of the S809 aerofoil by Somers [142] found that at high angles-of-attack ($\alpha \gtrapprox 10^\circ$) the flow separates near mid-chord. We take the flow at $\alpha = 14.24^\circ$ as a test case, where the available experimental data include surface pressure C_p, and lift coefficient C_l, at the following flow conditions: Reynolds number based on chord length, $Re_c = 2 \times 10^6$, and freestream Mach number, $M_\infty = 0.2$.

Since the Mach number is relatively low, the flow can be assumed to be incompressible. However, to demonstrate the capabilities of the developed tool, we employ both an incompressible (`simpleFOAM`) and a compressible solver (`rhoSimpleFOAM`). The latter case solves an energy equation along with the Navier-Stokes

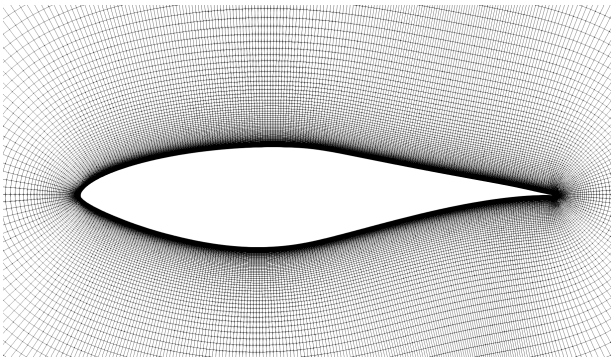


Figure 3.2: Close-up of the mesh for the S809 aerofoil, with around 8.9×10^5 cells.

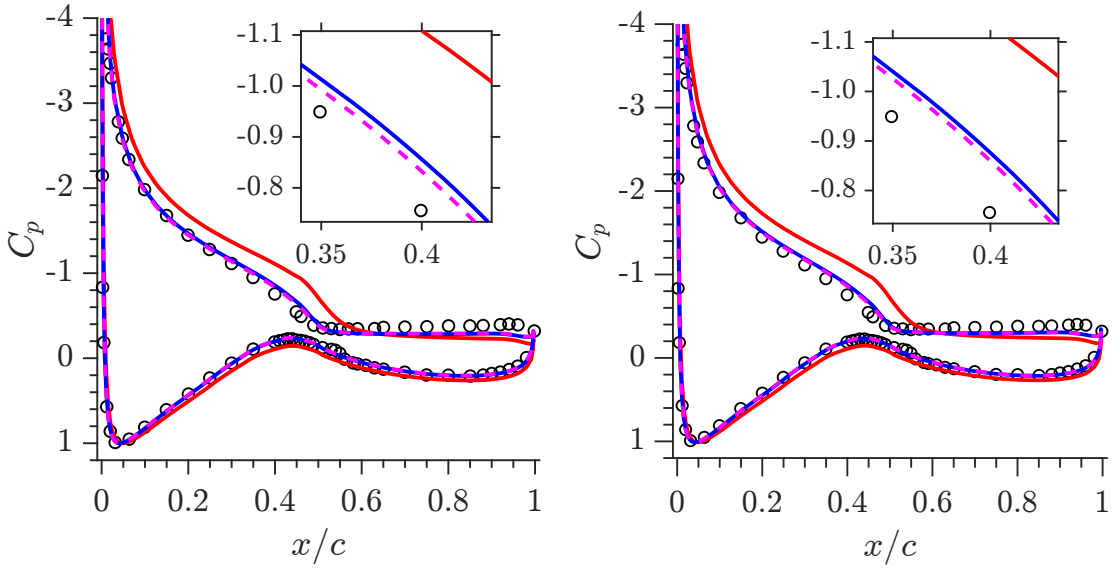
equations. The flow is assumed to be two-dimensional and steady, and the Spalart–Allmaras model is used as the baseline turbulence model. The simulations use an structured O-grid mesh, Fig. 3.2, with an average non-dimensional wall distance, $y^+ < 1$ on the aerofoil. The relatively dense mesh is used to reduce mesh-related inaccuracies.

As summarised in Table 5.1, two types of data are considered for field inversion: lift coefficient C_l and surface pressure coefficient C_p. The C_p data is extracted from [142], and only the values on the suction-side are used for field inversion—this is the region most prone to inaccurate predictions by the baseline model. The regularisation constant λ in Eqn. 5.1 is set to 10^{-4} following Ref. [139].

Table 3.2: Comparison of lift-coefficient prediction. Experimental $C_l = 1.083$.

Scenario	Incompressible	Error	Compressible	Error
Baseline Spalart–Allmaras	1.310	20.9%	1.346	24.3%
Field inversion, C_l data	1.107	2.2%	1.145	5.7%
Field inversion, C_p data	1.093	0.9%	1.133	4.6%

All field inversion scenarios result in significant error reduction in the C_l and C_p predictions, as shown in Table 3.2 and Fig. 3.3, respectively. Compared to only lift-coefficient value, the use of surface pressure data results in slightly better



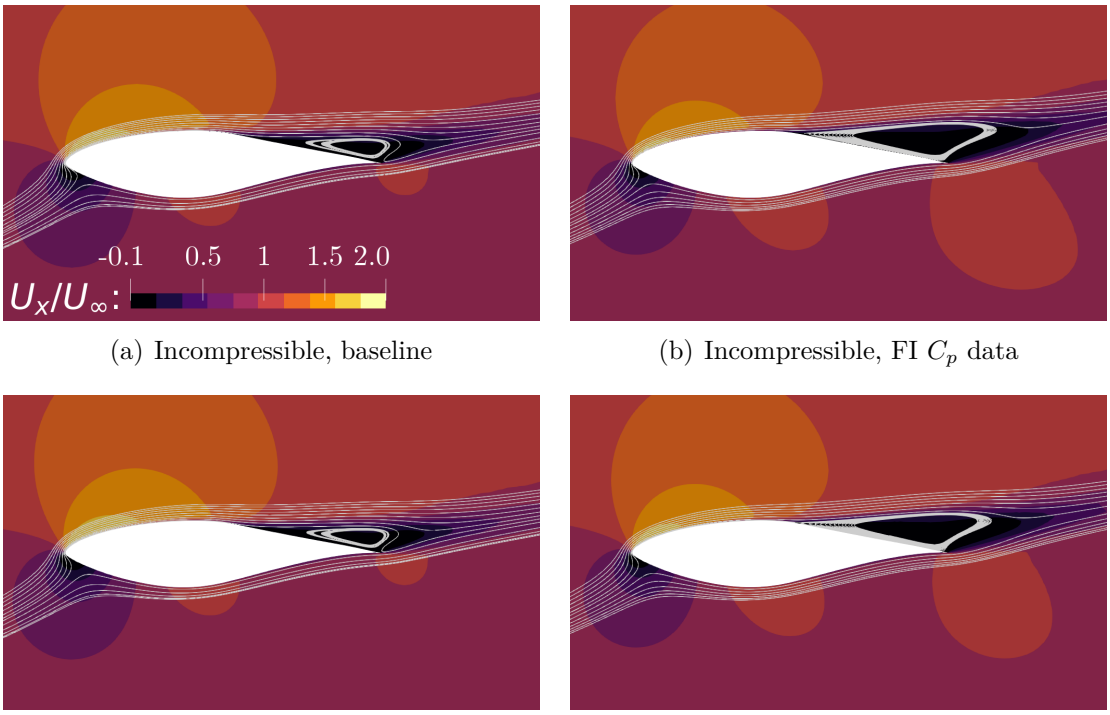
(a) Incompressible solver

(b) Compressible solver

Figure 3.3: Comparison of the pressure distribution for S809 aerofoil. Legend: Experiment (○), Spalart–Allmaras (—), field inversion, C_l data (—), and field inversion, C_p data (- - -). Panels in each diagram show close-ups of C_p prediction on the suction side at $0.35 \lesssim x/c \lesssim 0.45$. Marginally better performance is achieved when inverting with C_p data.

improvement of the baseline results as shown in C_p predictions in the close-up panels in Fig. 3.3. This is expected due to the significant difference between the sizes of two datasets used for field inversion, as outline in Table 5.1. Both the compressible and incompressible solvers produce similar predictions for the surface pressure distribution, lift coefficient, and velocity fields.

The baseline model over-predicts the lift generated, while under-predicting the pressure on the suction side. The baseline model also under-predicts the flow separation location and the size of the separation bubble, as shown in the velocity field contours in Fig. 3.4. The corrective fields, $\boldsymbol{\beta}$, shown in Fig. 3.5, account for the errors in the baseline model by reducing the turbulent production (i.e. regions with $\boldsymbol{\beta} < 1$), and hence predicting an earlier separation, and a larger separation bubble.



(a) Incompressible, baseline

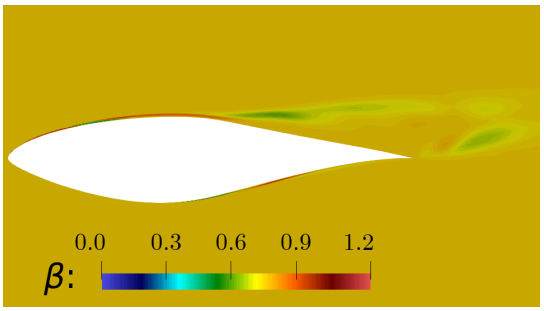
(b) Incompressible, FI C_p data

(c) Compressible, baseline

(d) Compressible, FI C_l data

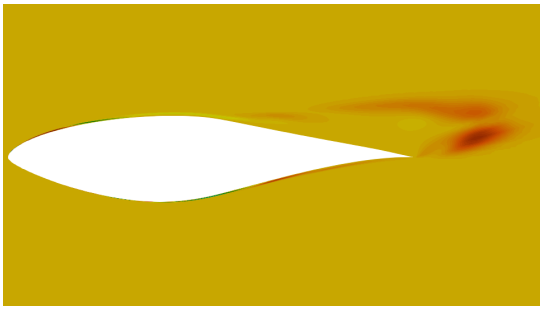
Figure 3.4: Comparison of the streamwise velocity field along with streamlines for the different S809 simulations. The velocity fields for incompressible FI using C_l data and compressible FI using C_p data are very similar to two field inversion velocity predictions shown here, thus removed for brevity.

Most significant changes made by the $\boldsymbol{\beta}$ field for the different scenarios are in the boundary layer close to the aerofoil. Additionally, it is interesting to note that the relatively different $\boldsymbol{\beta}$ field distributions shown in Fig. 3.5 lead to similar distribution of the surrogate turbulence variable $\tilde{\nu}$ and the velocity field, as shown in Fig. 3.6 and Fig. 3.4, respectively. A similar observation was made by He *et al.* [61] who argue that this might be due to the eddy viscosity hypothesis, which assumes that the Reynolds stress tensor can be modelled using a scalar in the form of eddy viscosity. The corrective field changes the entire balance of the turbulence model transport equation, which may also explain the multi-optimal nature of the optimisation results. Additionally, these differences could also be explained as the tendency of the optimiser towards driving the solutions to local minima.

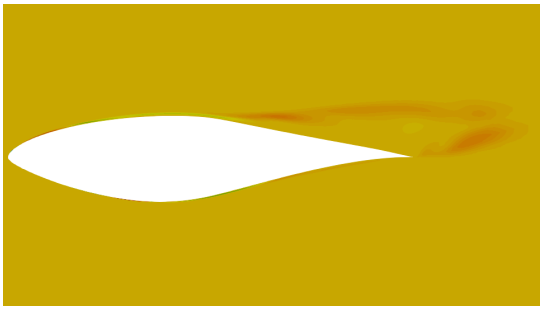


(a) Incompressible, C_l data

(b) Incompressible, C_p data

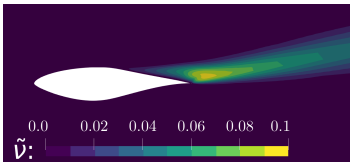


(c) Compressible, C_l data

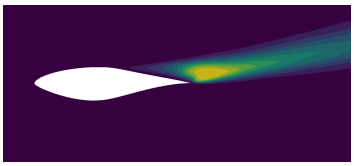


(d) Compressible, C_p data

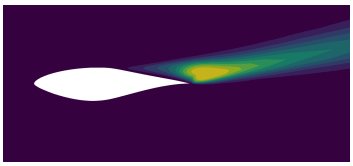
Figure 3.5: Comparison of the corrective field, $\boldsymbol{\beta}$, for the different S809 cases.



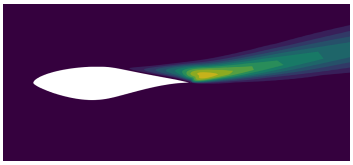
(a) Incomp., baseline



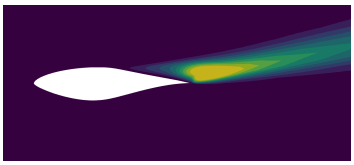
(b) Incomp., FI C_l data



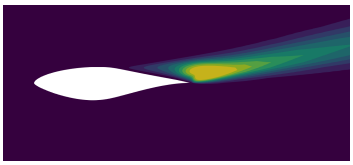
(c) Incomp., FI C_p data



(d) Compressible, baseline



(e) Compressible, FI C_l data



(f) Compressible, FI C_p data

Figure 3.6: Comparison of the surrogate turbulence variable $\tilde{\nu}$ in the S-A model before and after modification by the corrective scalar field $\boldsymbol{\beta}$, shown in Fig. 3.5.

NACA 4412: field inversion using velocity wake measurements

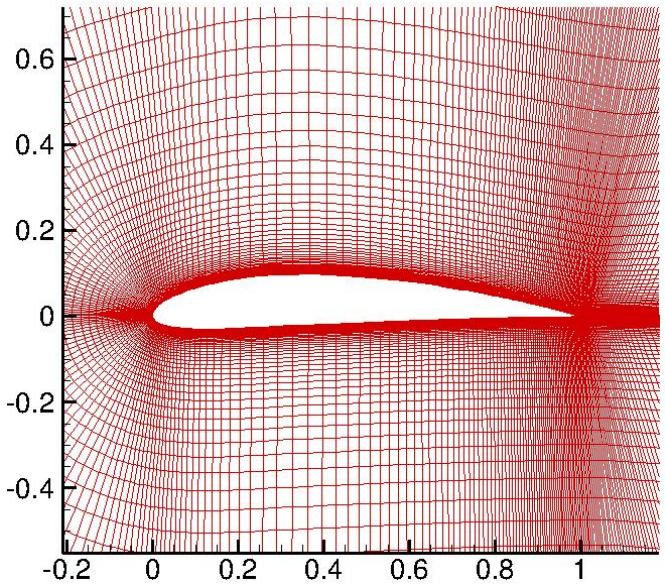
The goal of this test case is to demonstrate data assimilation using velocity data (vs. surface pressure and lift coefficient data used for the S809 case in the previ-

ous section). Flying-hot-wire streamwise velocity measurements from Coles and Wadcock [32] near the trailing edge (Fig. 3.7(b)) is used as high-fidelity data. The angle of attack is set to $\alpha = 13.87°$, with the Reynolds number based on the chord length $Re_c = 1.52 \times 10^6$, and the freestream Mach number $M_\infty = 0.09$.

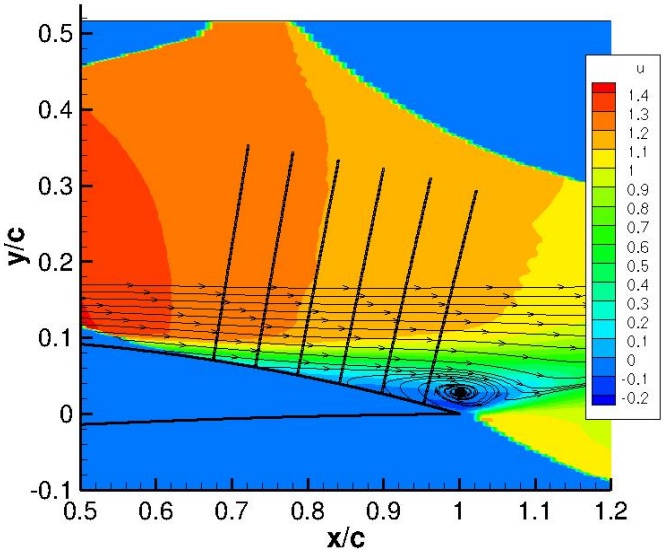
Like all the other cases in this work, we use a relatively fine mesh to minimise the numerical/discretisation errors. DAFoam also requires relatively fine meshes, to avoid using wall functions, for better performance. It is assumed that the modelling errors are the dominant source of uncertainty in the selected cases, as most of these have also been used in other studies where RANS-based models are shown to perform poorly. The NACA 4412 mesh from the NASA Turbulence Modelling Resource used over 230×10^3 cells, with an average $y^+ < 0.5$ on the aerofoil.

The streamwise velocity profiles, locations illustrated in Fig. 3.7(b), are compared in Fig. 3.8. The baseline Spalart-Allmaras model is unable to accurately capture the small separation bubble near the trailing edge as it under-predicts the effects of adverse pressure gradients in the boundary layer. Field inversion is able to match the experimental measurements well in the boundary layer close to the aerofoil surface, however, slight discrepancies remain at the edge of the boundary layer. The corrective field, $\boldsymbol{\beta}$, is shown in Fig. 3.9(a).

The corrections are achieved by reducing the turbulent production in the boundary layer on the suction side, which results in lower surrogate eddy viscosity production in the wake of the aerofoil, as illustrated in the $\tilde{\nu}$ predictions before and after data assimilation in Figs. 3.9(b-c), respectively. The reduced turbulence production means there is less turbulence mixing to delay flow separation.



(a) NACA 4412 mesh



(b) Trailing edge velocity measurements

Figure 3.7: The mesh used for the NACA 4412 case from the NASA Turbulence Modelling Resource [131], along with the normalised streamwise velocity measurements near the trailing edge by Coles and Wadcock [32]. The six perpendicular black lines mark the profiles where velocity predictions will be compared.

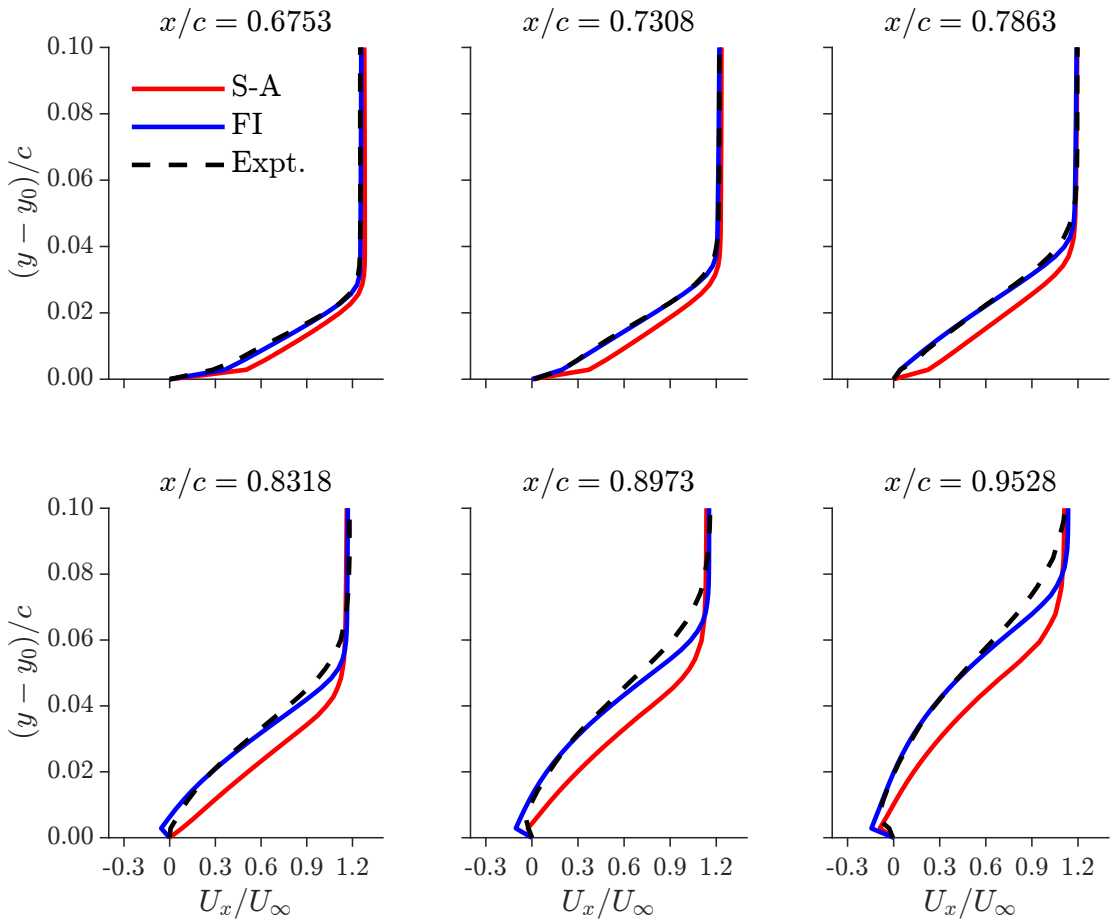
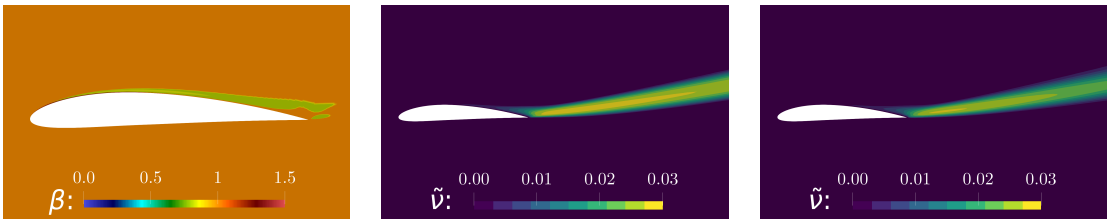


Figure 3.8: Comparison of the velocity profiles for the NACA 4412 case.



(a) Corrective field $\boldsymbol{\beta}$ (b) Baseline $\tilde{\nu}$ (c) FI $\tilde{\nu}$

Figure 3.9: The corrective field, the baseline and modified surrogate viscosity for the NACA 4412 aerofoil.

3.5.2 On the relative importance of different physical quantities

It has been variously observed that the impact of field inversion in terms of error reduction in all physical quantities of interest are highly sensitive to the type of

data used. However, there has been no quantitative study of the relative importance of physical quantities for data-driven turbulence modelling. In this section we will explore this topic.

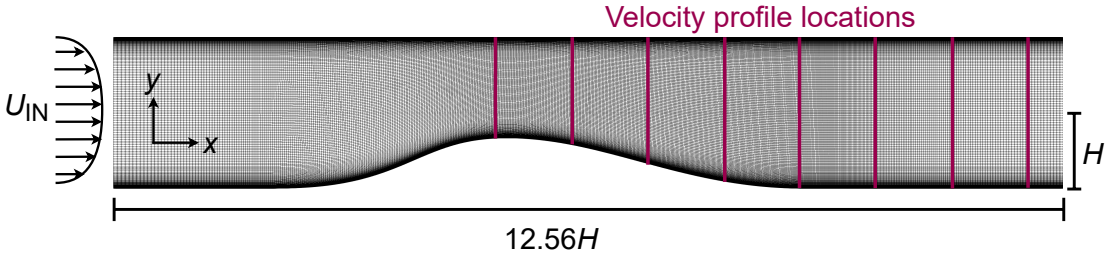


Figure 3.10: Mesh for the converging-diverging channel, with 9.87×10^5 cells, with the locations of the velocity profiles used for data assimilation.

We will use limited data from a DNS dataset to emulate experimental scenarios (e.g. pressure taps, hot-wire anemometers, or PIV measurements), and perform field inversion. We will then compare the data-driven turbulence model predictions against the rich DNS data in terms of physical quantities of interest (e.g. surface pressure, skin friction, velocity field). This will allow us to use error metrics to analyse the relative importance of different quantities, and will be the first step towards creating a list of ingredients for the experimental fluid dynamicist, and bridge the gap between experimental and computational approaches.

The next case is the flow over a smooth converging-diverging channel, Fig. 3.10. The flow involves adverse pressure gradients and a small separation bubble on the curved region of the lower wall which cannot be predicted accurately by the S-A model. Of the cases investigated in this work, the converging-diverging channel has the richest dataset available based on direct numerical simulation (DNS) results of Laval *et al.* [83]. We use the two-dimensional, steady, incompressible Navier–Stokes equations and the Spalart–Allmaras model to simulate the flow. The Reynolds number based on the channel half-height and maximum inlet velocity is 12,600. The structured mesh used for the simulations are from

[103], with an average $y^+ < 0.2$. Using the same approach as McConkey *et al.* [103] the inlet boundary conditions are generated by simulating a fully-developed boundary layer using the same turbulence model, and Reynolds number. The regularisation constant λ is set to 10^{-6}—a small value to reflect high-confidence in the available low-noise data.

For data assimilation we use the following quantities: streamwise velocity profiles at the locations outlined in Fig. 3.10, the skin friction on the lower wall, and the surface pressure on the lower wall. The effectiveness of the flow reconstruction will be quantitatively assessed as a function of the root-mean-square error reduction defined as follows:

$$\Delta\epsilon(q) = 1 - \frac{\epsilon_{\text{field inversion}}}{\epsilon_{\text{baseline}}}, \tag{3.18}$$

with q representing the mean physical quantities, which are the streamwise and wall normal velocities, the skin friction, and the surface pressure. The root-mean-square error ϵ is defined as,

$$\epsilon = \sqrt{\frac{\sum_{i=1}^{N} \left(q_i^{\text{DNS}} - q_i^{\text{RANS}}\right)^2}{N}}. \tag{3.19}$$

The data assimilation error reduction using different quantities are outlined in Table 3.3 and summarised as follows:

- While using a single type of data results in improvements across all quantities, the relative error reduction in each quantity is clearly impacted by the type of data used.

- Usually, maximum error reduction is observed for the quantity used for data assimilation. Even using a large amount of data, i.e. the entire streamwise velocity field does not lead to maximum error reduction in the skin friction

Table 3.3: Comparison of the root-mean-square error reduction (Eqn. 3.18) with different physical quantities used for data assimilation. n represents the size of data used for field inversion.

Data	$\Delta\epsilon(\boldsymbol{U})$	$\Delta\epsilon(C_f)$	$\Delta\epsilon(C_p)$	avg. $\Delta\epsilon$	n	avg. $\Delta\epsilon/n$
C_f	0.50	0.81	0.46	0.59	564	1.0×10^{-3}
C_p	0.25	0.49	0.83	0.52	564	9.2×10^{-4}
U_x profiles	0.47	0.46	0.64	0.52	1,400	3.7×10^{-4}
U_x field	0.79	0.58	0.48	0.62	98,700	6.4×10^{-6}

and surface pressures relative to the scenarios where the surface quantities are used for data assimilation.

- The average error reductions (column five in Table 3.3) illustrate that the velocity profiles and surface pressure lead to a similar mean flow reconstruction. More interestingly, data assimilation using the skin friction on the lower wall is 7% more effective than the other two scenarios (i.e. using surface pressure and velocity profiles). Using much more data in the form of the entire velocity field data leads to a 3% additional error reduction only (compared to the case with C_f data).

- Admittedly, the average error reduction (avg. $\Delta\epsilon$ in Table 3.3) does not account for the sizes of the different quantities, i.e. $N = 98,700$ in the root-mean-square error calculations for the velocity components, while it is $N = 564$ for the surface quantities. As a crude way of accounting for this, we normalise the average error reduction by the number of data points used for data assimilation, n. Results in the last column clearly shows that of the various data assimilation scenarios considered, using skin friction data leads to the most effective flow reconstruction. (Note: a larger value for avg. $\Delta\epsilon/n$ signifies a better data assimilation as a function of the amount

of data used.)

Figs. 3.11-3.12 compare the skin friction and surface pressure predictions for the different field inversion scenarios, respectively. The baseline model over-predicts the extent of the separated shear layer, hence a much delayed reattachment location, as recorded in Table 3.4. This is significantly improved by all field inversion scenarios. The skin friction is more sensitive to the quantity used for data assimilation, for instance, some discrepancies still exist in the reconstructed predictions for the field inversion scenarios when the entire U_x field (Fig. 3.11(a)), the surface pressure (Fig. 3.11(c)), and the velocity profiles (Fig. 3.11(d)), is used. On the other hand, the surface pressure is less sensitive as seen in the plots in Fig. 3.12.

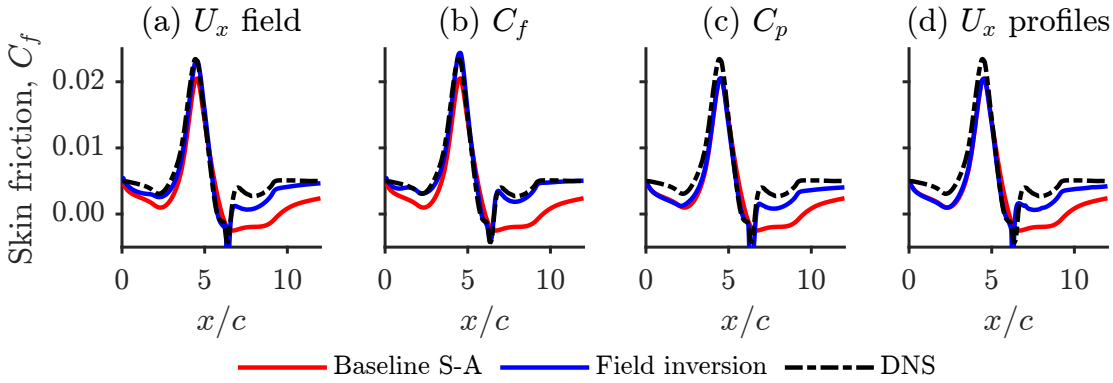


Figure 3.11: Skin friction comparison with different physical quantities used for data assimilation.

The streamwise velocity profiles, shown in Fig. 3.13 illustrate that field inversion scenarios with the entire streamwise velocity field (Fig. 3.13 (a)) and the skin friction (Fig. 3.13 (b)) are more effective than the wall pressure and velocity profiles. While the former is expected, the latter is of particular interest as skin friction measurements (e.g. using hot-wire anemometer) require much less experimental resources compared to detailed velocity field measurements (e.g. using particle image velocimetry techniques). While all scenarios lead to significantly

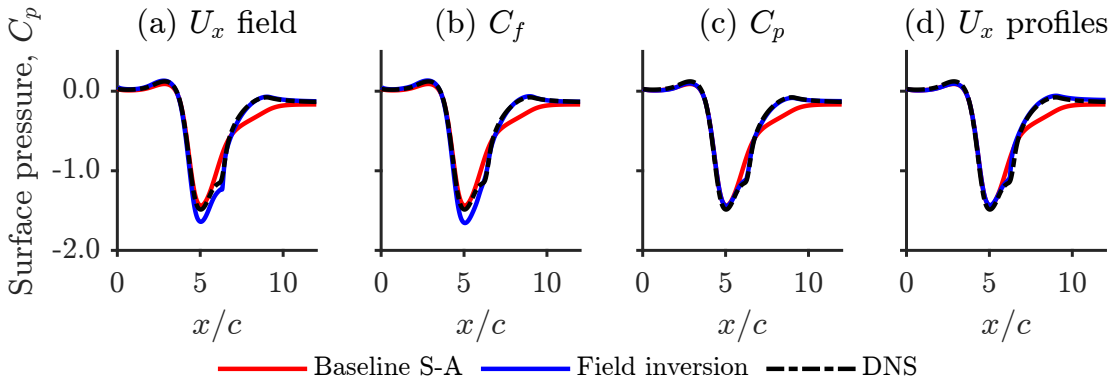


Figure 3.12: Surface pressure comparison with different physical quantities used for data assimilation.

Table 3.4: Flow re-attachment location comparison with different physical quantities used for data assimilation

Scenario	Re-attachment location, x/c	Error
DNS	6.60	-
Baseline S-A	9.54	30.82%
FI, U_x field	6.62	0.30%
FI, C_f	6.58	0.30%
FI, C_p	6.67	1.05%
FI, U_x profiles	6.44	2.48%

improved velocity predictions, some errors remain in the field inversion scenarios with C_p (Fig. 3.13 (c)) and velocity profiles data (Fig. 3.13(d)), especially at $x/H = [5.5, 6.5]$. This is also reflected in the corrective fields shown in Fig. 3.14, where for the stated field inversion scenarios, minimal corrections are made near the lower wall apex at $x/H \approx 5$ (Fig. 3.14 (c-d)). This may be explained by the optimiser concentrating the corrections in the near-wall regions with the largest discrepancy between the DNS and baseline S-A results, i.e. $7.5 \lesssim x/H \lesssim 11.5$.

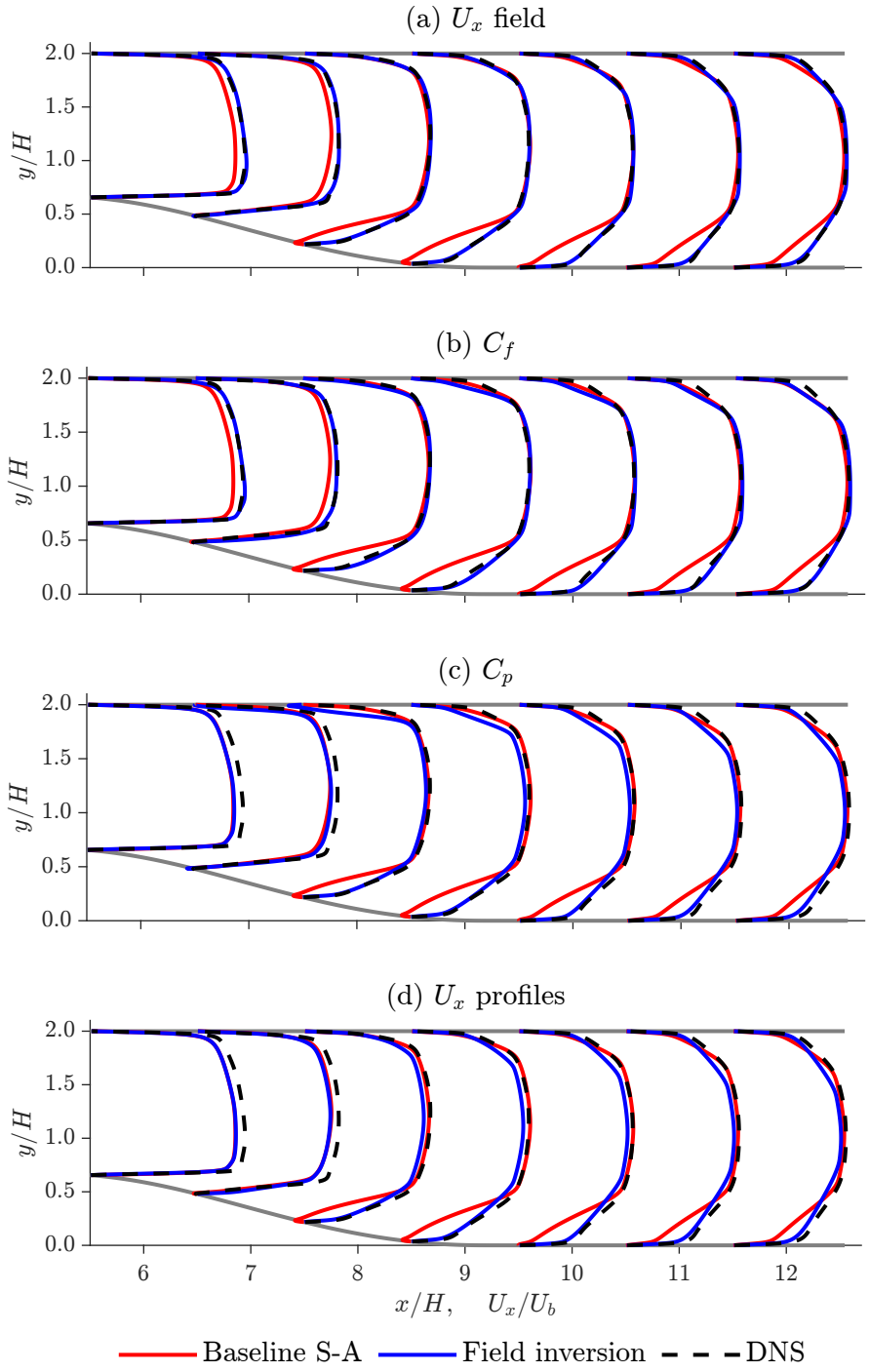


Figure 3.13: Velocity profiles comparison with different physical quantities used for data assimilation.

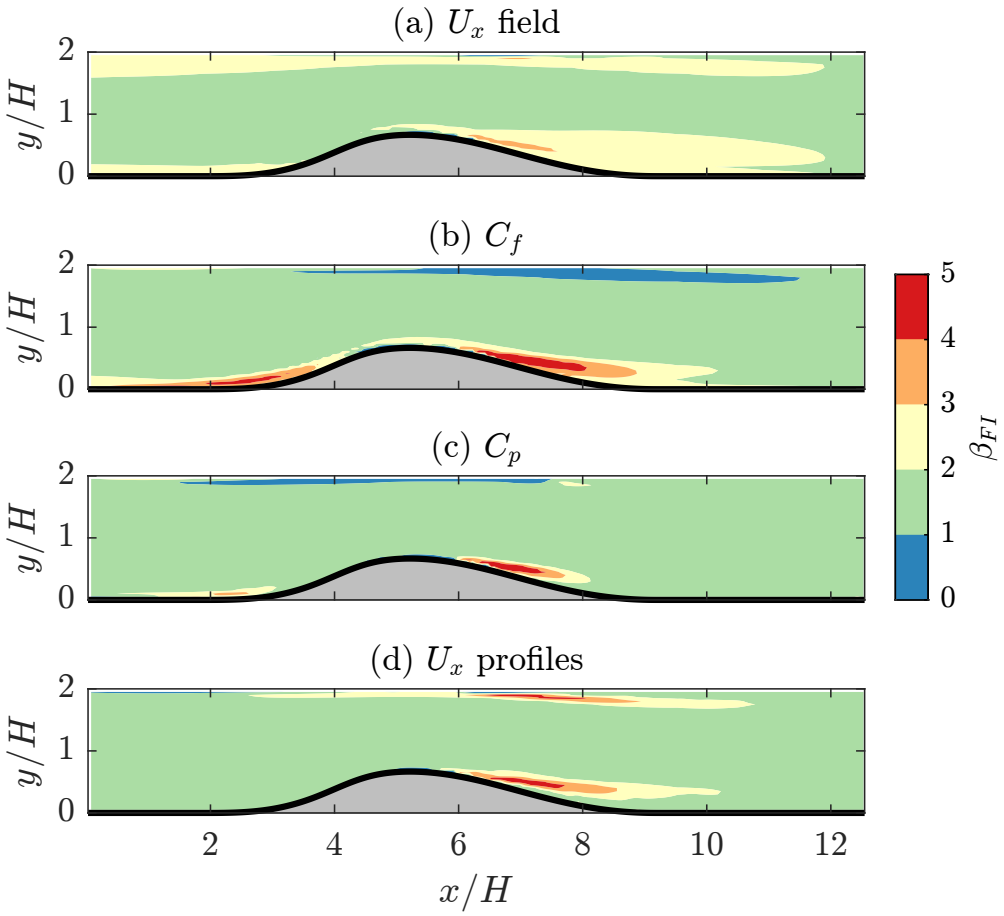


Figure 3.14: Comparison of the corrective field $\boldsymbol{\beta}$ for the various field inversion scenarios.

3.5.3 Data assimilation with different turbulence models

The last case, is the periodic hill geometry, which has become a prototypical case for testing turbulence models. The periodic hill case, shown in Fig. 3.15, is a simple geometry comprised of curved surfaces joined by a flat plate. As the flow moves over the first hill, it separates with a large recirculating bubble surrounded by an unsteady shear layer. It then reattaches, and undergoes strong acceleration at the subsequent hill. The flow at the top wall is observed to remain attached with high pressure gradients. These complex features are poorly predicted by RANS turbulence models [84]. Additionally, this case has well defined boundary

conditions, and relatively affordable computational cost which has made it an oft-used benchmark case for high-order simulations (LES e.g. Refs. [55, 50], and DNS e.g. Refs. [76, 171]), evaluation of RANS turbulence models, and data-driven frameworks. Extensive experimental data are also available: [127, 23].

Most linear eddy-viscosity based RANS models are known to perform poorly in predicting the flow with large separation after the initial hill. Data assimilation is performed using the Spalart-Allmaras, the Wilcox $k-\omega$, and the $k-\omega$ shear stress transport (SST) models.

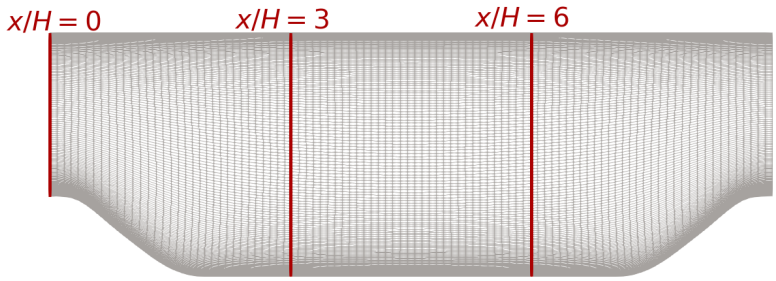


Figure 3.15: The mesh used for the periodic hills case supplied with the dataset, with $\sim 1.4 \times 10^4$ cells.

We use the DNS dataset by Xiao *et al.* [171]. In particular, we use the streamwise velocity profiles at $(x/H = 0, 3, 6)$ for field inversion. The Reynolds number is set to 5,600 following the data, and the two-dimensional, incompressible, Navier-Stokes equations are solved. A source/forcing term is added to the x-momentum equation to maintain a constant bulk velocity to achieve the LES Reynolds number, defined as:

$$Re_b = \frac{u_b H}{\nu}, \tag{3.20}$$

$$u_b = \frac{1}{2.035H} \int_H^{3.035H} U_x\,(y)\,dy, \tag{3.21}$$

where u_b is the bulk velocity, H is the hills height, ν is the kinematic viscosity, and U_x is the streamwise velocity component. No-slip boundary conditions are

applied at the walls, and cyclic boundary conditions are set at the inlet and outlet. The structured mesh has an average $y^+ < 1$ at the walls.

Table 3.5: Comparison of the root-mean-square error for the periodic hill case with three different turbulence models.

Model	Baseline $\epsilon(\boldsymbol{U})$	Field inversion $\epsilon(\boldsymbol{U})$	$\Delta\epsilon(\boldsymbol{U})$
Spalart-Allmaras	1.619×10^{-3}	5.904×10^{-4}	63.5%
Wilcox $k-\omega$	1.258×10^{-3}	6.656×10^{-4}	47.1%
$k-\omega$ SST	1.857×10^{-3}	5.487×10^{-4}	70.4%

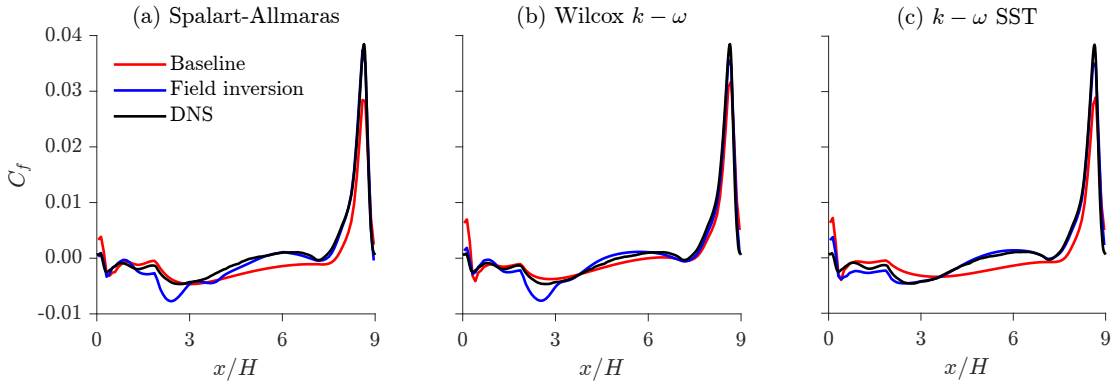


Figure 3.16: Comparison of the baseline and reconstructed surface friction distribution on the periodic hill lower wall with the three different turbulence models considered. The reference DNS surface friction data is from [76] *et al.* (for the geometry and flow conditions) as it is not available from the dataset used for data assimilation.

Table 3.5 summarises the baseline and reconstructed errors in the velocity predictions for the three turbulence models considered. The Wilcox $k-\omega$ model is the most accurate in terms of velocity predictions, while the $k-\omega$ SST is the least accurate. The most effective field inversion scenario is for the SST model, with an error reduction of 70.4%. This is also reflected in the streamwise velocity profiles, and the surface friction distribution on the lower wall, in Figs. 3.17, and 3.16, respectively. The velocity profiles and skin friction predictions show that

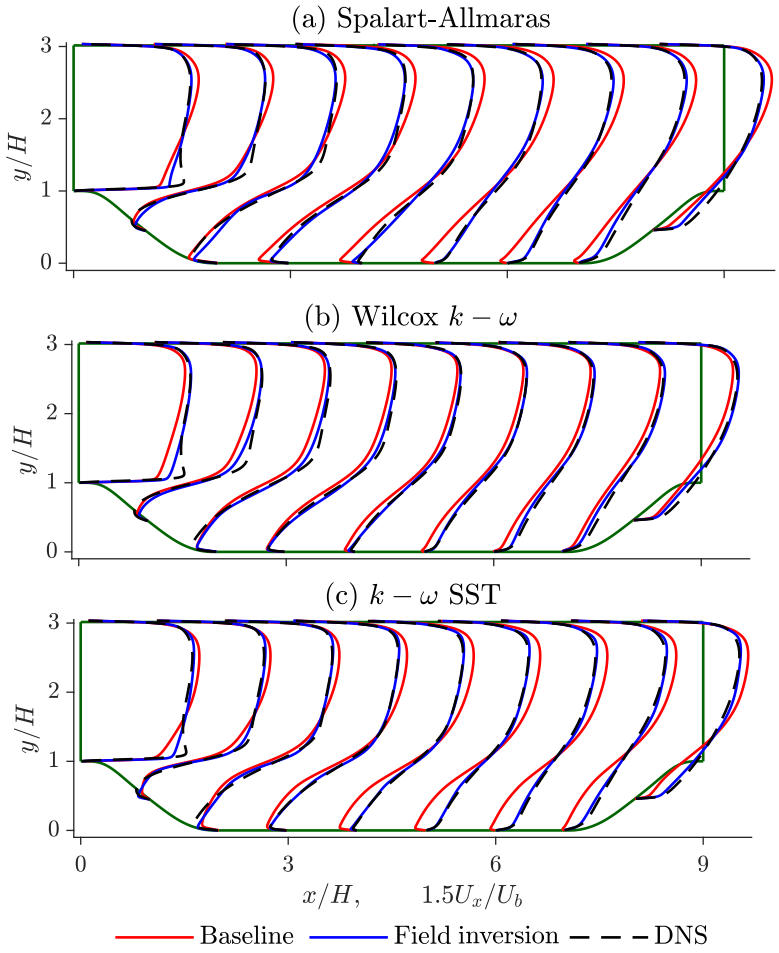


Figure 3.17: Periodic hill streamwise velocity profiles comparison with different turbulence models.

all the baseline turbulence models over-predict the size of the separation bubble, leading to flow re-attachment at a later point compared to the reference DNS data. This is due to the under-predicted turbulent viscosity, especially in the shear layer, as illustrated in the baseline and field inversion ν_t in Fig. 3.18. For all the three turbulence models, field inversion magnifies the eddy viscosity in the separated shear layer, and throughout the centre of the periodic hill domain. These are achieved through a complex, highly non-linear magnification and dampening of the turbulent variable ($\tilde{\nu}$ for the S-A model, and ω for the two-equation models)

in the domain, as illustrated in the corrective $\boldsymbol{\beta}$ fields in Fig. 3.19. However, there are differences between the baseline and field inversion turbulent viscosity contours for all three models, which could be attributed to the wholly different model structures. The beta field range is between 0 to 10 for the S-A model where the production term in the surrogate viscosity is adjusted by upto a factor of 10, whereas the range for the Wilcox $k-\omega$ and the SST model is 0 to 3. We found that the latter two models are more sensitive to the corrective field, and higher values of $\boldsymbol{\beta}$ could cause stability issues, where the flow would not converge.

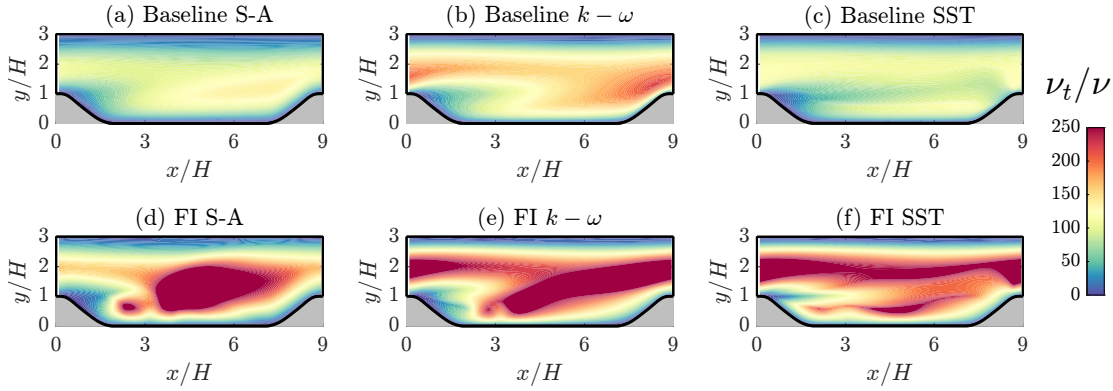
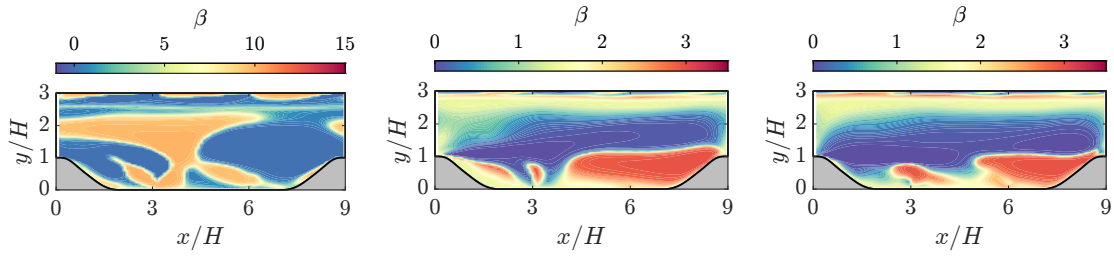


Figure 3.18: Comparison of the baseline and field inversion turbulent eddy viscosity for the periodic hill case.



(a) Spalart-Allmaras (b) Wilcox $k-\omega$ (c) $k-\omega$ SST

Figure 3.19: The corrective fields $\boldsymbol{\beta}$ for the three different turbulence models.

3.6 Summary

This chapter introduced and implemented the adjoint-based field inversion approach for turbulent mean flow reconstruction using data from high-fidelity simulations and experiments. Field inversion involved introducing a spatially varying field in the transport equation of the turbulence model, and optimising this field such that the error between the data and model predictions is minimised. This highly dimensional inverse problem is solved with gradient-based optimisation, driven by efficient derivative computations of the cost function using the discrete-adjoint method. The open-source implementation required intrusive code implementation involving the CFD flow solver, the adjoint solver, and the optimiser. In this work, open-source packages OpenFOAM, DAFoam and pyOptSparse were integrated to achieve this. Ten field inversion scenarios (summarised in Table 5.1) were presented for wall-bounded and external aerodynamic turbulent flows at Reynolds number in the range $10^4 - 10^6$. All the flows involved some level of flow separation which were poorly predicted by the baseline turbulence model(s). Data assimilation using limited, experimentally measureable, data was demonstrated to be highly effective in reconstructing the flows to match the high-fidelity reference data.

CHAPTER 4

Sparse sensor placement using eigenspace perturbations

4.1 Introduction

This chapter addresses the problem of sensor placement in field inversion methods. Obtaining high quality data is important in data driven methods and currently there is a lack of research performed on where to place sensors for field inversion problems. Existing methods tend to rely on expert knowledge and experience for predicting the most useful locations of sensors. This chapter provides the novel contribution of a sensor placement method to guide users, where sensors locations are chosen in regions of high uncertainty in the physics model. This uncertainty is obtained through eigenspace perturbations of the physics model - it is computationally efficient, simple to use (with few hyper-parameters) and is a principled, objective way for a user to choose sensor locations. This work has been published in [16, 15].

Sensor placement in fluid flows has been studied in the context of model reduction, for example using proper orthogonal decomposition (POD) [110, 31, 174], resolvent analysis [70], and deep learning [122]. It has also been explored for flow control, for example Refs. [77, 5]. Manohar *et al.* [99] introduced a sparse sensor placement strategy using singular value decomposition and QR pivoting. Callaham *et al.* [25] proposed using sparse representation techniques to reconstruct

flow fields given sparse measurements. Lu *et al.* [93] reconstructed flow around a surface-mounted prism using a combination of POD and a data-driven estimator using sparse velocity and scalar measurements. Recently, Karnik *et al.* [74] introduced a constrained sensor placement strategy using a greedy algorithm for flow reconstruction in nuclear digital twins. High-dimensional fields were reconstructed based on sparse sensor data using reduced order models.

In the context of RANS-based turbulent flow reconstruction, sensor placement has been investigated both in the context of ensemble- and adjoint-based data assimilation. Deng *et al.* [35] proposed a deep neural network (DNN)-based strategy to obtain the spatial sensitivity of the velocity field with respect to perturbations to RANS model constants. The *a priori* sensitivity analysis requires running ensembles of flow simulations with modified RANS model constants (100 CFD calculations in Ref. [35]) and using a feature importance layer in a deep neural network to map the velocity fields to the respective modified constants. The feature importance layer is effectively weights in the entire CFD mesh, and after DNN training the sensors are placed at the locations with the highest weights. The EnKF-based data assimilation by calibrating model coefficients may not be sufficient to reconstruct turbulent flows, as RANS-based discrepancy is thought to be mainly due to the structural forms of the turbulence model equations (i.e. uncertainties due to the Boussinesq hypothesis), rather than model parameters [170]. In addition, it is not clear how the bounds for model constants should be set for generating the ensemble data for DNN training.

Mons *et al.* [111] investigated sensor placement in the context of variational data assimilation for an unsteady *laminar* flow past a rotationally oscillating cylinder. The proposed framework involves the maximisation of the sensitivity of observations with respect to changes in initial and boundary conditions using a first-order adjoint approach. It is unclear how this method can be extended to

turbulent flows, and if the initial and boundary conditions are effective metrics to account for discrepancies due to parametric, functional, and structural uncertainties in turbulence models. Recently, Mons *et al.* [112] also proposed a more comprehensive linear and non-linear sensor placement strategies for laminar mean-flow variational data assimilation. The data assimilation is based on inferring a forcing term that corresponds to the divergence of the Reynolds stress tensor in the RANS momentum equation (as in Ref. [49]). The optimisation problem was formulated in two contexts: linear and non-linear. In the linear approach, the sensor placement strategy involves identifying the forcings that result in the most important variations in the mean velocity field, and placing sensors at the dominant locations to allow for accurate reconstruction. In order for this approach to be most effective, the analysis needs to be performed close to the true states of the flow—which is a limitation for *a priori* analyses. To overcome this issue a non-linear approach based on the second-order adjoint method was proposed that involves the minimisation of the condition number of the Hessian of the assimilated flow. While this approach was found to be considerably more effective compared to the linear approach, the Hessian evaluation is computationally expensive—especially when applied to high Reynolds number three-dimensional turbulent flows—and difficult to implement.

4.2 Proposed novel approach

In this work we propose an optimisation-based approach to variational sensor placement that involves the following:

1. Initially, we generate a spatial sensitivity map of the flow by employing the eigenspace perturbation approach for epistemic structural uncertainty in a baseline turbulence model [67]. Unlike the examples presented from literature, here the aim is to directly address the main source of discrepancy—the

epistemic structural errors in turbulence models [170]—and use it to tailor sensor placement for mean turbulent flow assimilation.

2. After the uncertainty map is generated, an optimisation problem is solved where the regions of highest uncertainties are targeted. This process is decoupled from the RANS-based solution, unlike the variational sensor placement strategies discussed above. The advantages are a very low computational cost, an easy implementation and algorithm tuning, compared to the adjoint-based analyses in Refs. [111, 112].

3. Once sensor locations have been identified, we then perform the variational data assimilation using the field inversion framework. This approach has the advantage of model-consistency [38], and the aforementioned capacity to perform reasonably with limited data. In addition, unlike the method in Ref. [35], this approach is not limited to parametric adjustment of the baseline turbulence model, and addresses the functional discrepancy.

4.3 Mathematical formulation

We will begin by motivating the approach to sensor placement, with a general overview of our proposed strategy, followed by the details of components involved in the framework, described sequentially. Very broadly, the task of sensor placement can be posed as an optimisation problem as follows:

$$\begin{aligned} &\max_{\mathbf{x}} \quad f[\mathbb{S}(\mathbf{x})], \quad \text{or} \quad \min_{\mathbf{x}} \quad f[\mathbb{S}(\mathbf{x})]^{-1}, \\ &\text{subject to} \qquad \mathcal{C}(\mathbf{x}), \end{aligned} \tag{4.1}$$

where $\mathbf{x}$ represents the desired sensor locations, which we will also denote $\mathcal{S}$ henceforth, $\mathbb{S}$ is an operator for the desired sensor locations, and $\mathcal{C}$ represents any/all the constraints (e.g. number of sensors, regions of interest, minimum

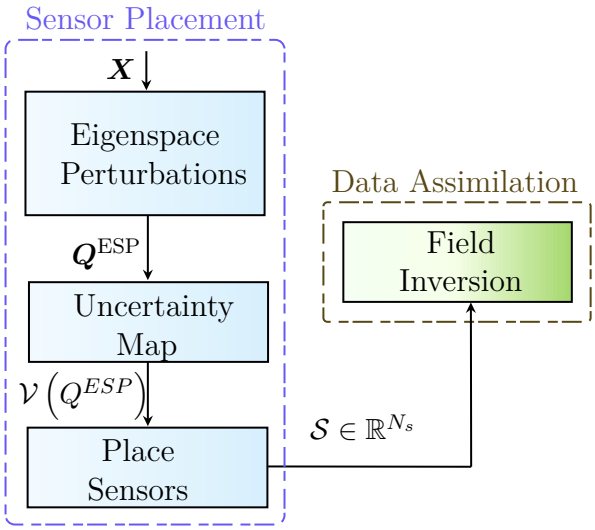


Figure 4.1: Overview of the proposed framework for sensor placement and data assimilation. The sensor placement and data assimilation problems are decoupled. $\boldsymbol{X}$: spatial coordinates of flow domain; $\boldsymbol{Q}$: flow variables such as velocity and pressure; $\mathcal{V}$ variance of any given quantity (q) following the five eigenspace perturbations; and $\mathcal{S}$: coordinates of optimised sensor locations. The relevant sections (§) of the paper are highlighted in each block.

distance between sensors, other physics-based constraints including the governing equations of the system, etc.).

Short of a brute-force exhaustive search, the so-called "generate and test" approach, which will have huge time and cost implications, the goal is to have an *a priori* sensor placement approach, i.e. placing sensors by approximating regions of uncertainty based on physical and/or expert knowledge before undertaking the experiment(s). The function $f[\mathbb{S}(\mathbf{x})]$ can be based on the determinant of the Fisher information matrix (e.g. Ref. [114]), some measure from a deep neural network (e.g. Refs. [35, 122]), or any other appropriate cost function (e.g. Refs. [111, 112, 74]). We stress, the goal is for $f[\mathbb{S}(\boldsymbol{x})]$ in Eqn. 4.1 to effectively capture the uncertainty in the underlying system, and target spatial sensors in the regions of high uncertainty.

Fig. 4.1 illustrates the proposed framework. Starting with a linear-eddy vis-

cosity RANS closure, we perturb the Reynolds stress tensor to their extrimal states to generate an uncertainty map, a surrogate for the operator $\mathbb{S}(\mathbf{x})$ in Eqn. 4.1. Specifically, five CFD simulations are run sequentially: two simulations perturbing the eigenvectors, and three simulations perturbing the eigenvalues—for all cells at once. All five simulations result in as many realisations of the flow prediction, allowing us to generate uncertainty maps–based on the variances–for various quantities of interest, e.g. a variance map of the streamwise velocity, with a value for each mesh cell, etc. Then, for a prescribed number of sensors and for a given quantity of interest, an optimisation problem is solved by a greedy search to ensure sensors are placed in regions of flow field with highest uncertainty. After identifying these locations the turbulent flow is reconstructed using adjoint-based data assimilation.

4.3.1 Eigenspace perturbations

The eigen-decomposition of the Reynolds stress tensor in Eqn. 3.3 results in:

$$\overline{u_i u_j} = 2k\left(\frac{\delta_{ij}}{3} + \nu_{in}\Lambda_{nl}\nu_{lj}\right), \tag{4.2}$$

where ν_{in} is the orthornormal eigenvectors matrix and Λ_{nl} is a diagonal matrix composed of eigenvalues, λ_l. The eigen-decomposition is useful because now the shape and orientation of the Reynolds stress tensor are directly represented by the eigenvalues and eigenvectors, respectively.

The eigenvalues can be represented in the Barycentric map [8], shown in Fig. 4.2. This is an alternative to the anisotropy-invariant maps (AIM) introduced by Lumley and Newman [94]. AIM is constructed as a function of the nonlinear principal invariants (II and III) of the anisotropy tensor $a_{ij} = \overline{u_i u_j}/(2k) - \delta_{ij}/3$, with $II = a_{ij}a_{ji}$, and $III = a_{ij}a_{in}a_{jn}$. The Barycentric map is constructed from the eigenvalues of the Reynolds stress tensor in Eqn. 4.3.1. The Barycentric

map (represented in 2D space with an arbitrary turbulence state defined by the coordinates $\boldsymbol{x} = (x, y)$) is formed such that all realisable turbulent states are encompassed by the equilateral triangle, with the vertices representing the limiting states.

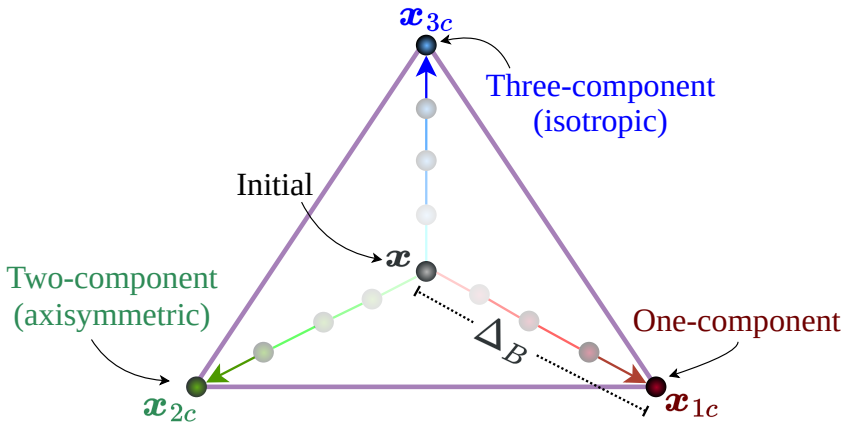


Figure 4.2: Barycentric triangle used to perturb eigenvalues to their three limits. Mapping turbulence states on the Barycentric map is useful as all realisable turbulence states are bounded and presented by the edges and vertices of the equilateral triangle. In eigenspace perturbations an arbitrary Reynolds stress state $\boldsymbol{x}$ is sequentially perturbed from the initial state to the three vertices $\boldsymbol{x}_{1c}$, $\boldsymbol{x}_{2c}$, and $\boldsymbol{x}_{3c}$.

Using spectral theorem for second-order tensors, Banerjee *et al.* [8] show that the Barycentric triangle vertices ($\boldsymbol{x}_{1c}$, $\boldsymbol{x}_{2c}$, $\boldsymbol{x}_{3c}$) in Fig. 4.2 represent one-component, two-component (axisymmetric) and three-component (isotropic) turbulence states, respectively. These vertices correspond to the following eigenvalue vectors:

$$\boldsymbol{\lambda}_{\boldsymbol{x}_{1c}} = \begin{pmatrix} 2/3 \\ -1/3 \\ -1/3 \end{pmatrix}, \quad \boldsymbol{\lambda}_{\boldsymbol{x}_{2c}} = \begin{pmatrix} 1/6 \\ 1/6 \\ -1/3 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\lambda}_{\boldsymbol{x}_{3c}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{4.3}$$

Therefore, a given Reynolds stress tensor in a mesh cell, can be represented on the

Barycentric map as $\boldsymbol{x} = (x, y)$. It is formed as a linear function of the eigenvalues of the Reynolds stress tensor, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)^T$, and the vertices of the triangle, as follows [8]:

$$\boldsymbol{x} = \boldsymbol{x_{1c}}(\lambda_1 - \lambda_2) + \boldsymbol{x_{2c}}(2\lambda_2 - 2\lambda_3) + \boldsymbol{x_{3c}}(3\lambda_3 + 1). \tag{4.4}$$

The mapping in Eqn. 4.4 and the requirement that the sum of eigenvalues must be zero, can be written as an invertible one-to-one mapping, $\boldsymbol{x} = B\boldsymbol{\lambda}$. Eqn. 4.3 maps the eigenvalues to the three limiting vertices, e.g. $\boldsymbol{\lambda_{x_{1c}}} = B^{-1}\boldsymbol{x}_{1c} = (2/3, -1/3, -1/3)^T$.

Uncertainty can now be introduced in the Reynolds stresses by perturbing the eigenvalues to their limiting states, i.e. the three vertices in the Barycentric map. The perturbed states $\boldsymbol{x}^*$ can be defined as [42, 67, 109]:

$$\boldsymbol{x}^* = \boldsymbol{x} + \Delta_B(\boldsymbol{x}^{(t)} - \boldsymbol{x}), \tag{4.5}$$

where $\boldsymbol{x}$ is the arbitrary initial state, $\Delta_B \in [0, 1]$ is the relative distance between $\boldsymbol{x}$, and the target vertex $\boldsymbol{x}^{(t)}$ of the Barycentric triangle. Using the mapping $\boldsymbol{\lambda} = B^{-1}\boldsymbol{x}$, the perturbed eigenvalues, $\boldsymbol{\lambda}^*$ can be calculated as

$$\begin{aligned}
\boldsymbol{\lambda}^* &= B^{-1}\boldsymbol{x}^* && (4.6)\\
&= B^{-1}\left[\boldsymbol{x} + \Delta_B(\boldsymbol{x}^{(t)} - \boldsymbol{x})\right], && (4.7)\\
&= (1 - \Delta_B)\underbrace{B^{-1}\boldsymbol{x}}_{=\boldsymbol{\lambda}} + \Delta_B\underbrace{B^{-1}\boldsymbol{x}^{(t)}}_{=\boldsymbol{\lambda}^{(t)}}, && (4.8)\\
&= (1 - \Delta_B)\boldsymbol{\lambda} + \Delta_B\boldsymbol{\lambda}^{(t)}, && (4.9)
\end{aligned}$$

where $\boldsymbol{\lambda}$ is the vector of eigenvalues calculated from the eigen-decomposed Reynolds stress tensor, and $\boldsymbol{\lambda}^{(t)}$ is the eigenvalues for any target vertex given in Eqn. 4.3.

In practice, eigenvalue perturbations corresponding to three CFD flow solutions perturbing the baseline eigenvalues to their three limiting states, i.e. three vertices of the Barycentric triangle.

The eigenvector perturbations modulate the turbulent kinetic energy production, $\mathcal{P}_k = -\overline{u_i u_j}(\partial U_i/\partial x_j)$, which represents the transfer of kinetic energy from the mean flow to the fluctuating velocity field [67]. This corresponds to varying the alignment of the Reynolds stress ellipsoid. Mathematically, this modulation is achieved by varying the Frobenius inner product $\langle A, R\rangle_F = \mathrm{tr}(AR)$, with A representing the mean velocity gradient, and R the Reynolds stress tensor. As with the eigenvalue perturbations, the extremal states are sought by considering the bounding values of the inner product to consider all permissible dynamics. [67] show that the bounds on the inner product are

$$\langle A, R\rangle_F \in \left[\lambda_1\gamma_3 + \lambda_2\gamma_2 + \lambda_3\gamma_1, \lambda_1\gamma_1 + \lambda_2\gamma_2 + \lambda_3\gamma_3\right], \tag{4.10}$$

with $\lambda_1 \geq \lambda_2 \geq \lambda_3$ representing the eigenvalues of the symmetric components of the mean velocity gradient A, which is also the strain rate tensor. Thus in the coordinate system defined by the eigenvectors of the strain rate tensor, the bounds for the Reynolds stress alignments of the Reynolds stress eigenvectors are:

$$\nu_{\min} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \nu_{\max} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Therefore, two additional CFD flow solutions are needed to perturb the eigenvectors to their limiting states. The eigenspace perturbations implementation of [109] in the SU2 CFD suite is used in this work. The implementation in Ref. [109] has been tested on both two- and three-dimensional flows.

4.3.2 Greedy search placement algorithm

The uncertainty map generated as described in the previous section can now be used as a surrogate for the $f[\mathbb{S}(\mathbf{x})]$ operator in Eqn. 4.1. One approach of formulating the optimisation problem for sensor placement, with a prescribed number of sensors, N_s, can be

$$\max_{\mathbf{x}} \quad \mathcal{J} = \frac{1}{\|\mathcal{J}_0\|} \sum_{i=1}^{N_s} \mathcal{V}^q(\mathbf{x}_i), \quad \text{subject to} \quad d(\mathbf{x}) \geq d_{\min}, \tag{4.11}$$

where $\mathcal{V}^q$ represents the variance-based uncertainty map from eigenspace perturbation for a quantity q, $\mathbf{x} \in \chi$ represents the sensor coordinates, where $\chi \subseteq \boldsymbol{X}$ represents the space of potential sensor locations, and for $N_s > 1$, $d(\mathbf{x})$ represents a minimum distance constraints between any two sensors, prescribed by the user (based, for example, on operational parameters/constraints, etc.). Additionally, the distance constraint is essential for the problem to be well-posed in order to avoid sensor clustering.

One approach to solving the optimisation problem in Eqn. 4.11 can be using global search methods, such as genetic algorithms, with many off-the-shelf implementations to choose from. However, this can be a computationally challenging optimisation problem due to the very large design space involved in turbulent flows, i.e. the computational budget required in high Reynolds number, three-dimensional flows of industrial interest is not anticipated to scale well. Given this, we propose a more computationally efficient solution based on a greedy search, which uses the heuristic of making the locally optimal decision at each choice of sensor location, i.e. the sensor is placed at the next permissible location of maximal uncertainty in each iteration.

The sensor placement procedure, summarised in Fig. 4.3 and Algorithm 1, is as follows:

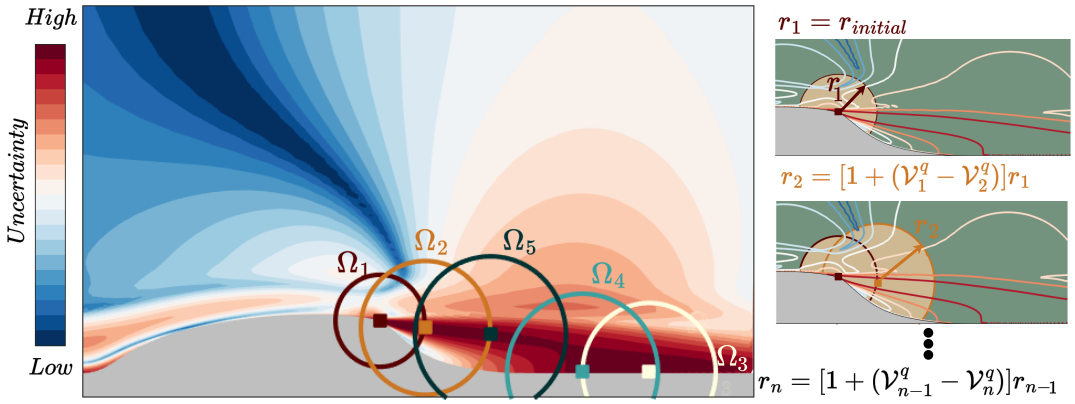


Figure 4.3: Sensor placement visualisation using the proposed greedy algorithm, in Algorithm 1. The contour represents the uncertainty in the streamwise velocity predictions from eigenspace perturbations, $\mathcal{V}(U_x)$. The number of sensors $N_s = 5$ in this example, placed sequentially. The sensors are represented by the square markers, and the circles labelled Ω_n represent the exclusion domain defined by radius r_n, where the radius is calculated using the expression in Eqn. 4.12 for $n > 1$, and the initial radius r_1 is selected iteratively using $\mathcal{M}_1$ in Eqn. 4.13.

1. Given a prescribed number of sensors, N_s, the potential sensor sites $\chi \subseteq \boldsymbol{X}$, uncertainty map for a given quantity, $\mathcal{V}^q$, place the first sensor in the cell with the highest variance, $\mathcal{V}^q$, and store the spatial coordinates, $\mathcal{S}_n$. An initial radius, r_{initial}, must also be prescribed for defining the circular (in two-dimensional cases)—or spherical in three-dimensional flows—exclusion zone, Ω_n with $n = 1$, to avoid subsequent sensors clustering around the sensor placed previously. This is an important hyper-parameter which we will return to shortly.

2. Exclude the cells in the domain Ω_n, for $n \geq 1$, and update χ and $\mathcal{V}^q$ accordingly.

3. Place the next sensor at the cell with highest variance (based on the updated $\mathcal{V}^q$ and χ), and store the coordinates $\mathcal{S}_n$.

4. Select the radius, r_n for $n > 1$, for the exclusion domain Ω_n using the following

$$r_n = \left[1 + \left(\mathcal{V}^q_{n-1} - \mathcal{V}^q_n\right)\right] r_{n-1}. \tag{4.12}$$

Using the relation in Eqn. 4.12 we linearly increase the size of the exclusion domain based on the change in variance between subsequent sensors. In the results to be presented in section 4.4 this linear relationship leads to satisfactory outcomes. While this can also be a nonlinear function, the choice of the form of the function is heuristic, which can be investigated in future works.

5. Repeat steps 2-4 until all sensors are placed.

Algorithm 1 Greedy algorithm for sensor placement

Inputs: N_s = number of sensors; $\chi \subseteq \boldsymbol{X} = x$, y, z Cartesian coordinates of mesh cell centres; $\mathcal{V}^q$ = eigenspace perturbation-based variance of quantity to be measured; r_{initial} = initial radius for exclusion domain.
Output: $\mathcal{S}$ = sensor locations (x, y, z coordinates)

procedure PLACESENSORS(N_s, χ, $\mathcal{V}^q$, r_{initial})
  $r \leftarrow r_{\text{initial}}$ ▷ radius for exclusion domain
  for all $n \in 1, \ldots, N_s$ **do**
    $i = \max(\mathcal{V}^q)$ ▷ i = index of cell with max. variance
    $\mathcal{S}\left[n, :\right] = \chi\left[i, :\right]$ ▷ save x, y, z coordinates of sensor n
    $\hat{\chi}, \hat{\mathcal{V}} = []$ ▷ storage for non-excluded regions
    $M \leftarrow \text{size}(\chi)$
    for all $m \in 1, \ldots, M$ **do**
      $x, y, z \leftarrow \chi\left[m, :\right]$
      if $x, y, z \notin \Omega_n$ **then** ▷ only keep cells outside Ω_n
        $\hat{\chi} \leftarrow \text{concatenate}(x, y, z)$
        $\hat{\mathcal{V}} \leftarrow \text{concatenate}(\mathcal{V}^q\left[m\right])$
    $j \leftarrow \max\left(\hat{\mathcal{V}}\right)$ ▷ j = index of cell with max. variance
    $r \leftarrow \left[1 + \left(\mathcal{V}^q[i] - \hat{\mathcal{V}}[j]\right)\right] * r$ ▷ update radius for Ω_{n+1}
    $\chi \leftarrow \hat{\chi}$
    $\mathcal{V}^q \leftarrow \hat{\mathcal{V}}$
  return $\mathcal{S}$

There are two hyper-parameters in the proposed greedy algorithm. The first one is the initial radius for the circular exclusion zone. This determines how far the sensors are placed. For a given number of sensors, we propose an iterative approach to guide this selection, using the following metric $\mathcal{M}_1$:

$$\mathcal{M}_1 = \frac{\sum_i^{N_\chi} \mathcal{V}_i^q}{\sum_n^{N_s} \sum_m^{\Omega_n} \mathcal{V}^q\left(\mathbf{x}_m\right)}, \tag{4.13}$$

where we iteratively vary the initial radius, illustrated in Fig. 4.4. The numerator is the sum of variances, $\mathcal{V}^q$, over the entire potential sensor sites, χ, and the denominator is the sum of variances over the excluded circular domains with the sensors locations at the centres (refer to illustrative exclusion domains, Ω_n, in Fig. 4.3). The initial radius can be defined as a proportion of some characteristic length of the case under consideration. As it increases, $\mathcal{M}_1$ in Eqn. 4.13 approaches unity. Thus, we can select the initial radius iteratively, using some relative threshold (i.e. with respect to unity). This will be practically demonstrated and assessed numerically in the following section. In addition, this approach of selecting the initial radius parameter is a practical one, given the low computational cost of the entire sensor placement algorithm.

The number of sensors can be considered as the second hyper-parameter. We note that based on the experimental approach, this can be dictated by other constraints (e.g. in the instance where only a discrete given set of sensors are available). Nonetheless, as an approximate guide to how many sensors may be considered, we propose the following metric, $\mathcal{M}_2$:

$$\mathcal{M}_2 = \left(\sum_n^{N_s} \mathcal{V}_n^q\right)^{-1}, \tag{4.14}$$

which is the sum of variances over the sensor locations, $\mathcal{S}$. As the number of sensors increase N_s will level off, as illustrated in Fig. 4.4, with detailed numerical

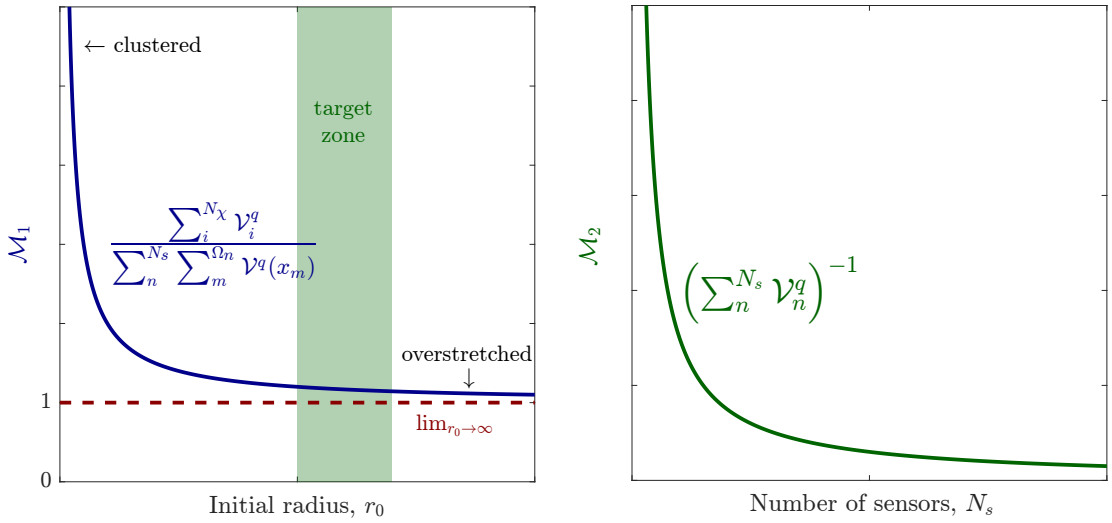


Figure 4.4: Illustrative plot of the metric in Eqn. 4.13 for guiding the selection of the initial radius hyper-parameter. The initial radius for the exclusion domain, required by the greedy sensor placement algorithm, can be iteratively selected, i.e. any point in the "target zone" when $\mathcal{M}_1$ levels off.

analysis in the results section.

4.4 Results

In this section we apply the proposed framework to three fully-turbulent wall-bounded benchmark flows, all involving challenging flow physics (e.g. separation and reattachment) for RANS-based simulations. In all the cases we use the streamwise velocity as the quantity from high-fidelity data, since it is one of the most commonly measured (in terms of volume data, i.e. measured quantities in the flow domain). All flows are simulated as two-dimensional, steady, and incompressible.

4.4.1 NASA wall-mounted hump

The flow over the well-established 2D wall-mounted hump, Fig. 4.5, part of the NASA Turbulence Modelling Resource database [131], is selected as the first test

case. It involves flow separation as a result of adverse pressure gradients over the smooth hump surface. The flow is commonly used as a benchmark case for verifying and validating turbulence models since most linear eddy viscosity models perform poorly in predicting the separation, reattachment and boundary recovery by over-predicting the size of the separation bubble due to under-predicted turbulent shear stress in the separation region. The chord-based Reynolds number is 9.36×10^5 with a Mach number of 0.1.

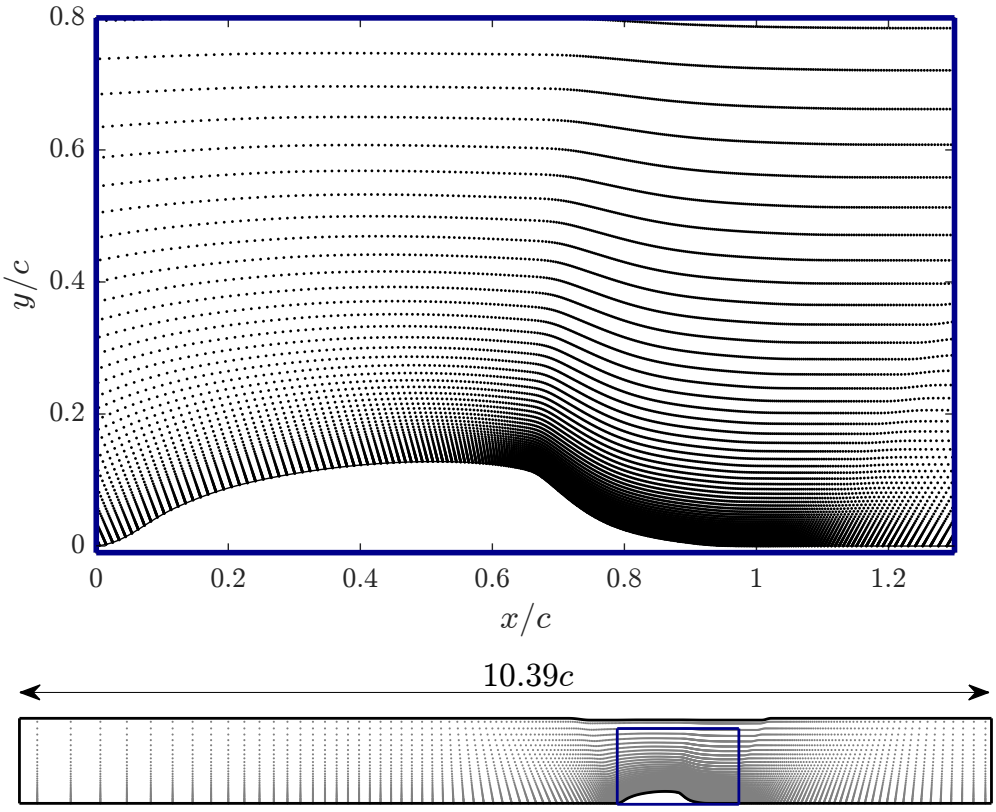


Figure 4.5: 2D NASA smooth hump, with the dots marking cell-centres, $\boldsymbol{X}$, in the mesh used [131]. The blue bounding box represent the region of potential sensor sites, $\chi \subseteq \boldsymbol{X}$.

We use wall-resolved large eddy simulation (LES) data by Uzun *et al.* [154] as a surrogate for experimental data. These results have been validated against experimental data, and are preferred over limited publicly available experimental

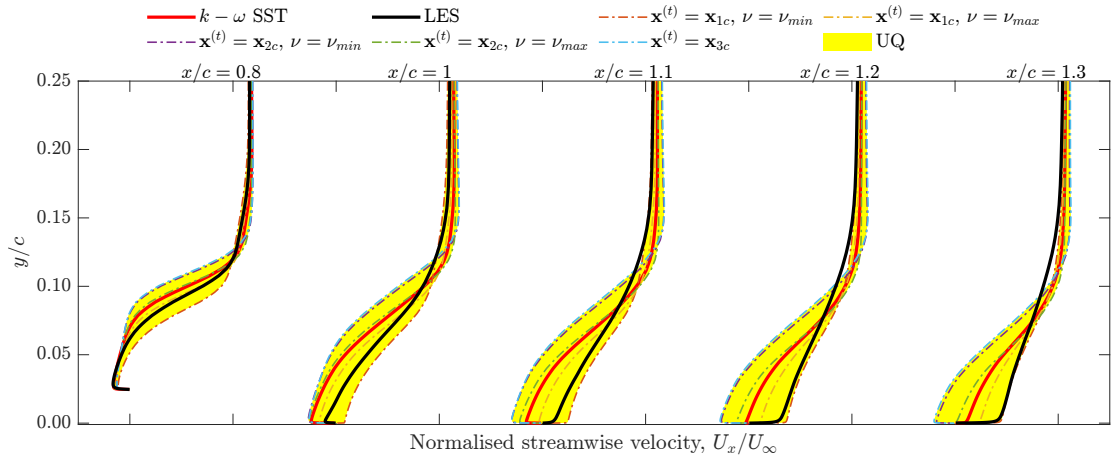


Figure 4.6: Streamwise velocity profiles with uncertainty bounds based on eigenspace perturbations. The dashed-dotted lines represent the five ESP scenarios. The UQ bounds are the minimum and maximum values from the 5 eigenspace perturbations.

results as it allows for benchmarking data assimilation with sparse experimental data against the scenario when extensive data is available, such as those generated in particle image velocimetry (PIV) experiments.

The baseline $k-\omega$ SST, LES, and eigenspace perturbation streamwise velocity profiles are shown in Fig. 4.6. The SST predictions are most erroneous near the wall, in the separated shear layer, due to a very high flow reversal aft of the hump apex, resulting in an over-predicted circulation zone. The eigenspace perturbation results show a high degree of variability compared to the SST predictions in the near-wall region. Increasing the turbulent production mechanism (e.g. $\boldsymbol{x}^{(t)} = \boldsymbol{x}_{1c}$, $\nu = \nu_{\text{min}}$) reduce the flow separation. thus shifting the results closer to the reference data, while damping the production (e.g. $\boldsymbol{x}^{(t)} = \boldsymbol{x}_{3c}$) lead to higher separation. The ESP results essentially subsume the LES references data, especially near the wall, in the separated shear layer, and becomes negligible as we approach the freestream (as expected).

We can now map the uncertainty for any given physical quantity, $q \subseteq \boldsymbol{Q}$ (in the

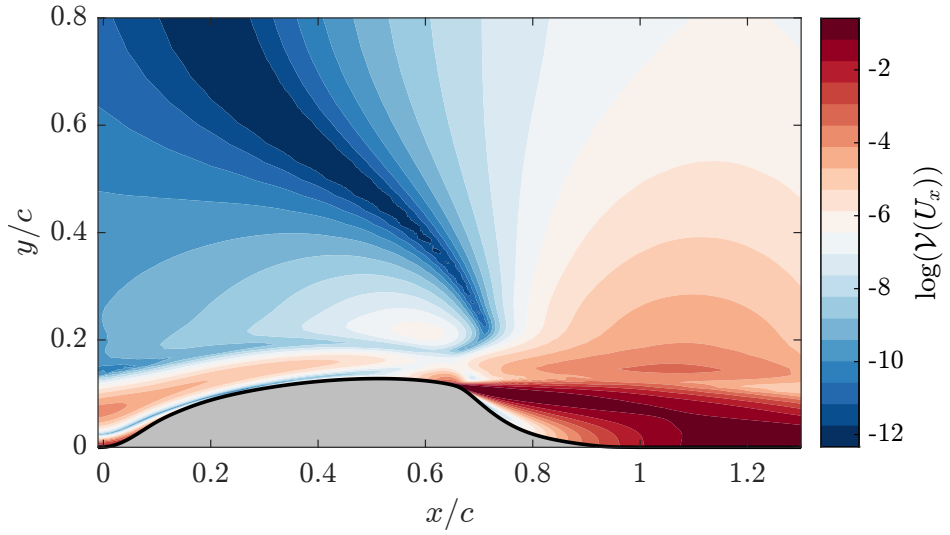


Figure 4.7: The uncertainty map in streamwise velocity for the NASA wall-mounted hump case.

case of vector quantities, we can look at individual scalar components, as at the present work, or some normalised sum of different components) as a function of the variance based on eigenspace perturbation results. As the reference data will be the streamwise velocity, U_x, Fig. 4.7 shows the uncertainty map, normalised by the maximum variance. We use a logarithmic scale to better illustrate the degree of uncertainty in the different regions of the flow domain. The high regions of uncertainty are in the near the hump, aft of hump apex, increasing to the highest uncertainty in the separated shear layer—overall, qualitatively, bearing engineering scrutiny.

Sensors are now placed using the uncertainty map, and our proposed heuristic greedy algorithm. In order to benchmark the framework, we will present results for the following scenarios, shown in Fig. 4.8: a) uniformly spaced, manually placed sensors, ending up with $N_s = 33$ over the entire potential sensor sites, $\chi \subseteq \boldsymbol{X}$, fixed for the subsequent two scenarios, b) randomly placed sensors, c) sensor placed using our algorithm with $N_s = 33$, d) using our algorithm but reducing the number of sensors by over one order of magnitude, i.e. $N_s = 3$, and

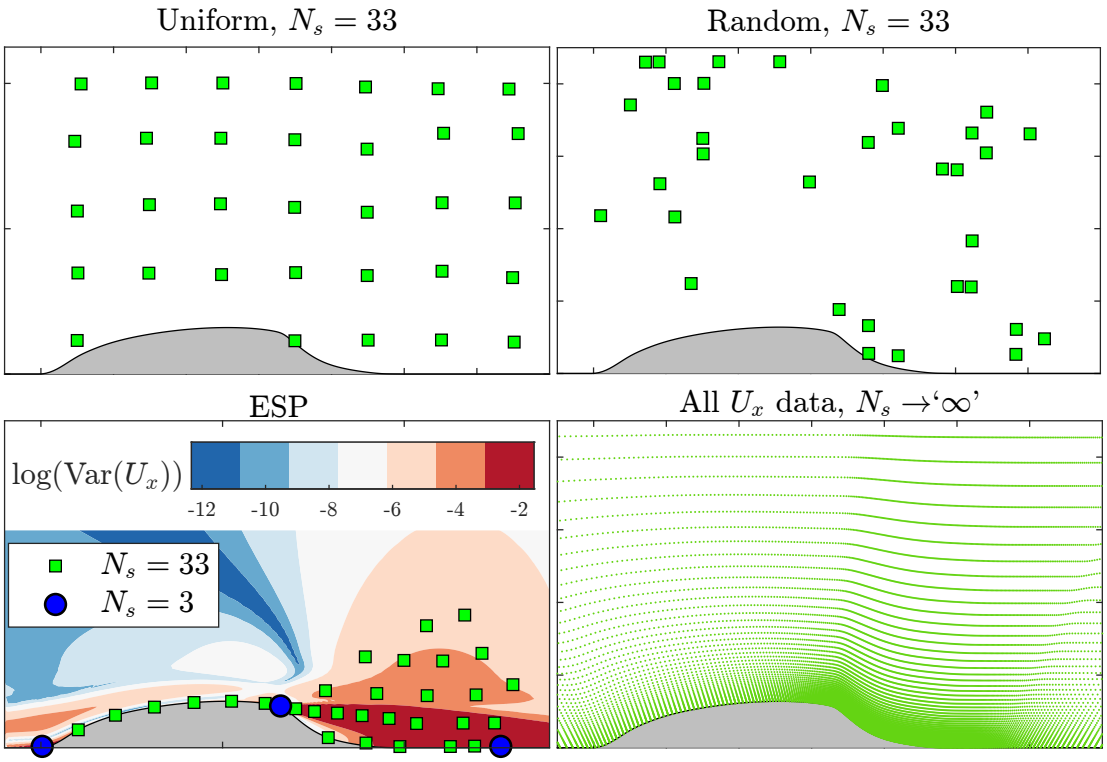


Figure 4.8: Various sensor placement scenarios used for data assimilation. ESP refers to the proposed placement algorithm, and the contour lines represent the uncertainty map in Fig. 4.7.

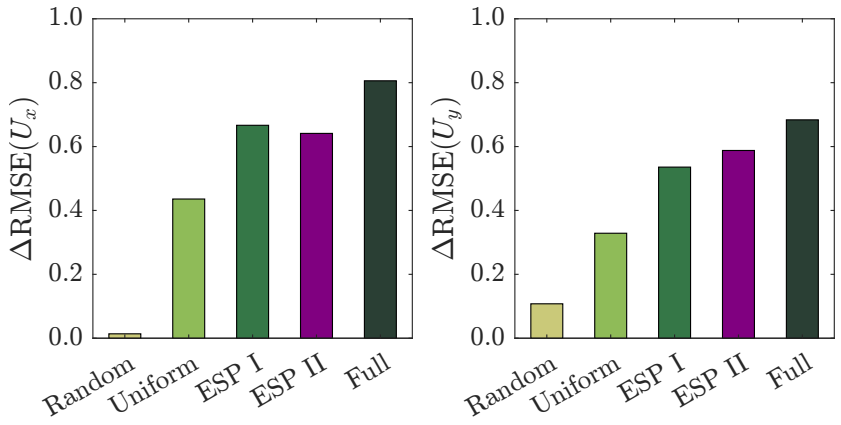


Figure 4.9: Comparing the root-mean-square error change in velocity predictions (left: streamwise component, U_x; right: wall-normal component, U_y) for various sensor placement scenarios shown in Fig. 4.8. ESP I and II refer to the cases with $N_s = 33$, and $N_s = 3$, respectively, where sensors are placed using our proposed algorithm.

e) using the streamwise velocity over the entire potential sensor sites, $\chi \subseteq \boldsymbol{X}$. Note: only one random placement realisation is considered. A more thorough analysis to illustrate the sensitivity of data assimilation to sensor locations should consider multiple realisations, which is an oversight in our analysis. Nevertheless, the uniform placement vs. the other scenarios, clearly illustrate the sensitivity of DA to sensor locations.

We compare the turbulent mean flow reconstruction results for the stated scenarios in Fig. 4.9, in terms of the change in root-mean-square error in the baseline SST predictions of the velocity components. As anticipated, the random sensor placement case is the least effective (negligible improvement in the streamwise velocity despite the use of U_x data in field inversion), highlighting the need for informed sensor placement. The uniform sensor placement case reduces the average error in the velocity components by 38%. For the same number of sensors as the previous two discussed scenarios (i.e. $N_s = 33$), with our proposed sensor placement algorithm, the average errors reduce by 60% (labelled "ESP I" in Fig. 4.9), compared to a 74% reduction when using all the streamwise velocity data (labelled "Full" in Fig. 4.9). The case with all the data shows that using very large datasets does not necessarily results in proportionally similar data assimilation error reduction. Notably, the results for this flow show that with our approach the same error reduction is achievable even if the number of sensors is reduced by over an order of magnitude, i.e. $N_s = 3$, (labelled "ESP II" in Fig. 4.9), with an average error reduction of 61%.

Next, we investigate the two hyper-parameters for the proposed sensor placement algorithm as discussed in Section 4.3.2: initial radius for the circular exclusion domain, and the number of sensors.

Fig. 4.10 compares the initial radius against the metric $\mathcal{M}_1$ in Eqn. 4.13 (proposed to guide selecting an appropriate r_{initial} *a priori*) and the subsequent

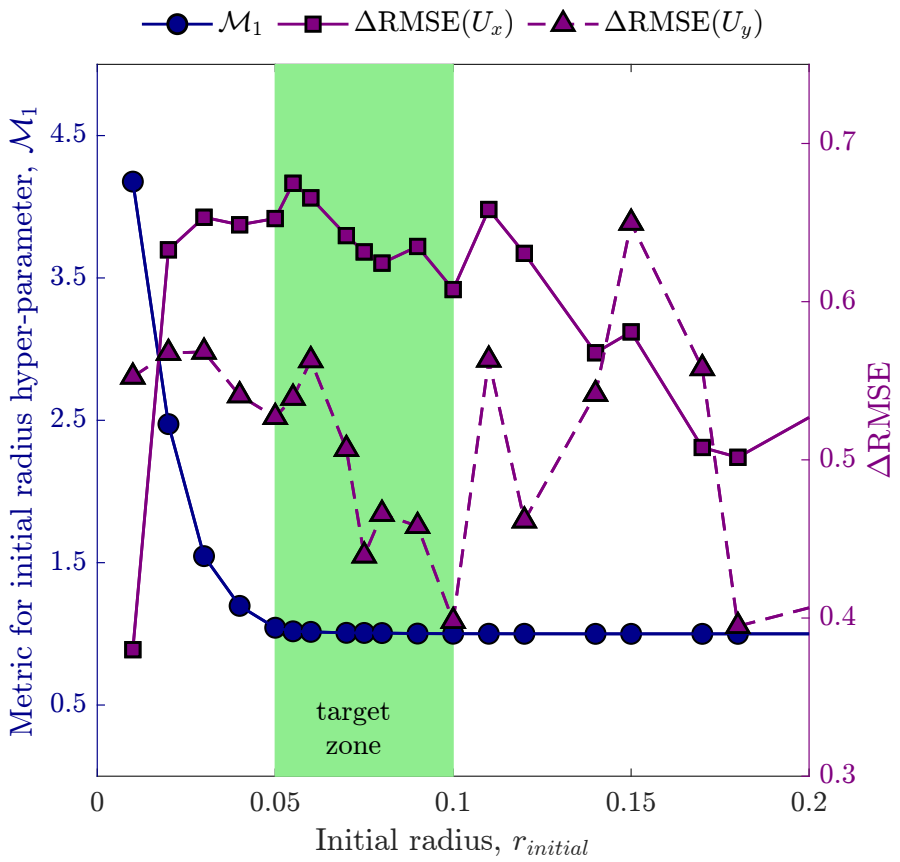


Figure 4.10: The tuning metric $\mathcal{M}_1$ (Eqn. 4.13) on the left axis, and the effect of initial radius on flow reconstruction error reduction on the right axis. Number of sensors is set to ten.

error reduction through data assimilation, with $0.01 \leq r_{\text{initial}}/c \leq 0.20$, where c is chord length for the hump (Fig. 4.5). The number of sensors is fixed to $N_s = 10$. The metric, $\mathcal{M}_1$ sharply decreases in the range $0.01 \leq r_{\text{initial}}/c \leq 0.05$, reaching $\leq 5\%$ of the asymptotic value of $\mathcal{M}_1$, i.e. $\lim_{r_{\text{initial}}\to\infty} \mathcal{M}_1(r_{\text{initial}}) \leq 1.05$. The average percentage error reduction in the velocity is: $44\% \leq \Delta\text{RMSE}(\boldsymbol{U}) \leq 61\%$. (For reference, with only a third of the sensors, the lower bound is still an improvement on the uniformly placed sensors case, Fig. 4.9.) Illustrative placement scenarios are shown in Fig. 4.12. Near the lower initial radius bound the sensors are clearly too clustered, while near the upper bound, the sensors are

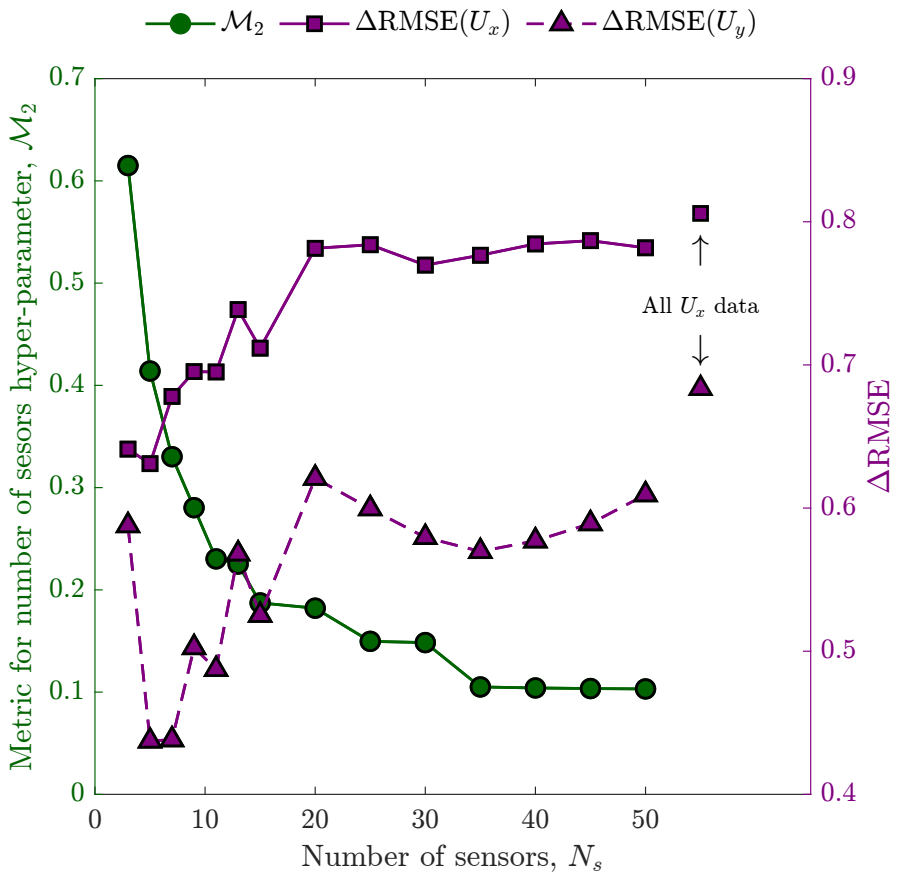


Figure 4.11: The tuning metric $\mathcal{M}_2$ (Eqn. 4.14) on the left axis, and the effect of number of sensors on flow reconstruction error reduction on the right axis.

overstretched (some placed in regions of low uncertainty, while many regions of higher uncertainty are not considered during the placement). The configuration in and close to the "target zone" in Fig. 4.10, $0.05 \leq r_{\text{initial}}/c \leq 0.1$, seem to be good potential candidates. This is broadly confirmed by the similar error reduction in the streamwise component of velocity. Admittedly, the prescribed metric $\mathcal{M}_1$ does not guarantee the optimal solution, and requires the exercises of an element of engineering judgement.

Investigations of the second hyper-parameter, the number of sensors, are summarised in Fig. 4.11 and 4.13. The analysis is with reference to the second metric, $\mathcal{M}_2$ introduced in Section 4.3.2. For all the number of sensors consid-

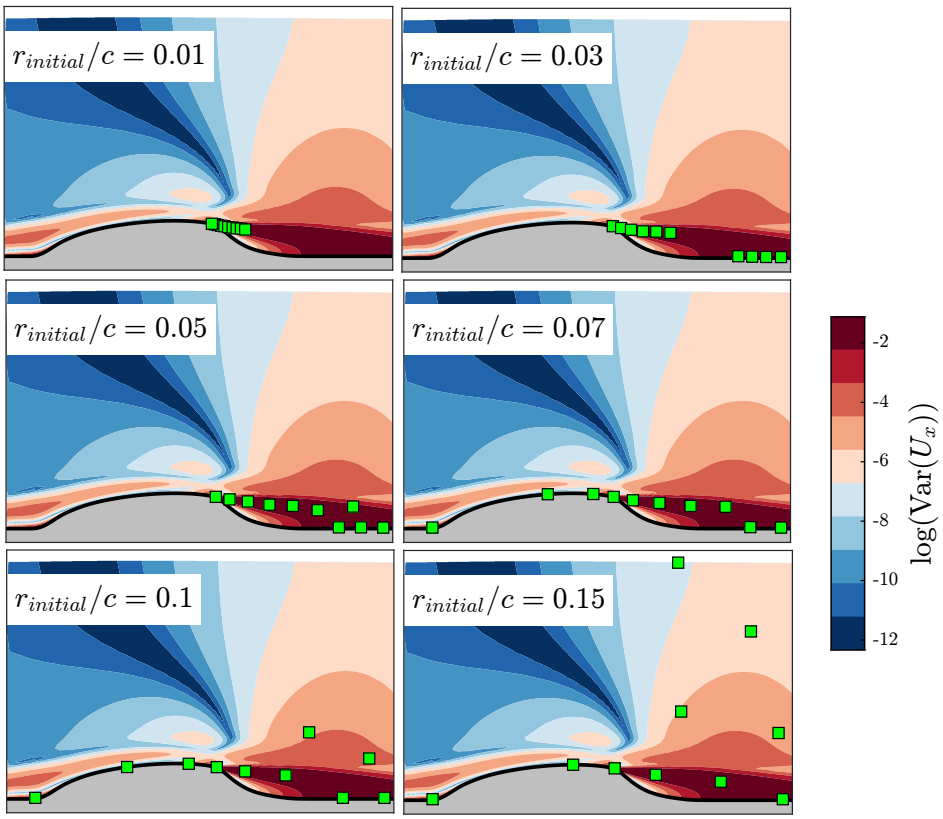


Figure 4.12: Illustrative sensor configurations, for investigating the effect of the initial radius on data assimilation. The number of sensors is fixed to $N_s = 10$.

ered, the initial radius is chosen iteratively using the previously described approach: the final r_{initial} is chosen before $\mathcal{M}_1$ reaches an asymptotic value of 1.02 (see illustrative placement configurations in Fig. 4.13). The value of the metric $\mathcal{M}_2$ initially reduces sharply up to 10-15 sensors, then gradually levelling from $N \geq 35$, in Fig. 4.11. The average percentage error reduction in the velocity is: $53\% \leq \Delta\text{RMSE}(\boldsymbol{U}) \leq 71\%$. The error reduction in the streamwise velocity (the quantity used as reference data for field inversion, and the dominant component in terms of contribution to velocity magnitude) levels after around 20 sensors, and is reasonably close to the scenario when all the data is used. On the other hand, the reduction in the wall-normal velocity component, is less clear as a function of number of sensors. Fig. 4.11 demonstrates that the metric introduced, $\mathcal{M}_2$, can

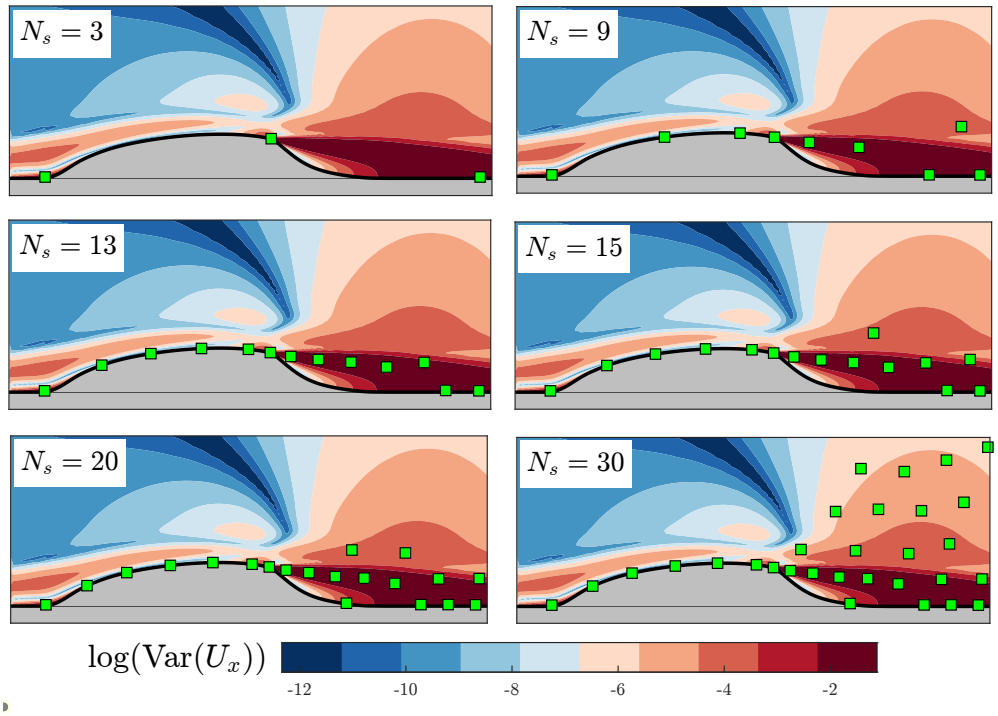


Figure 4.13: Illustrative sensor configurations, for investigating the effect of number of sensors on data assimilation.

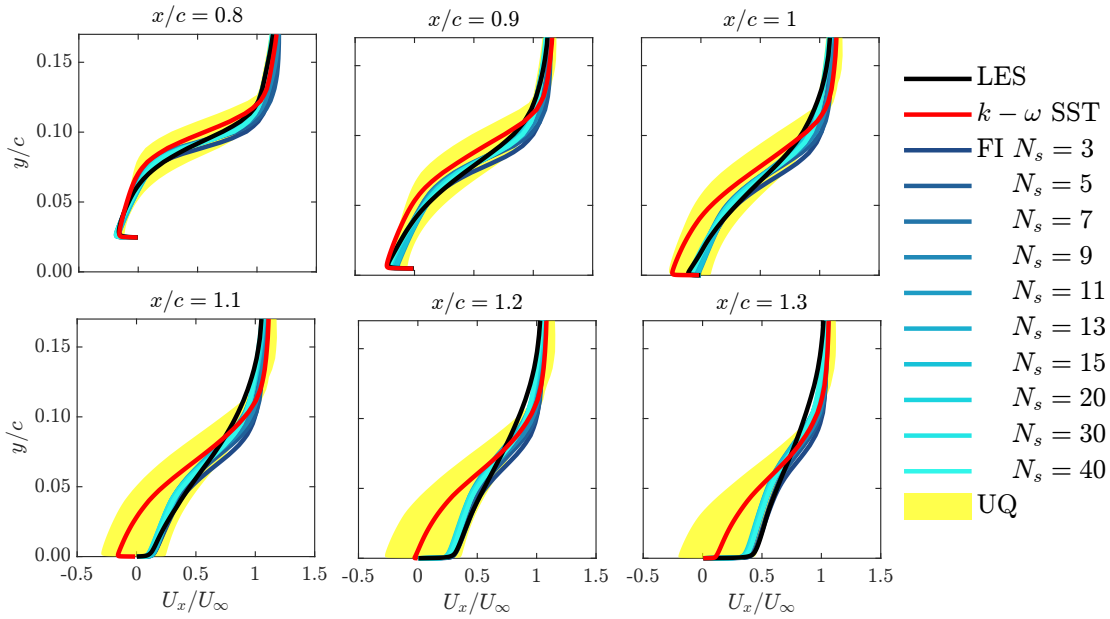


Figure 4.14: Normalised streamwise velocity profiles comparison for the 2D NASA hump case before and after data assimilation for different number of sensors.

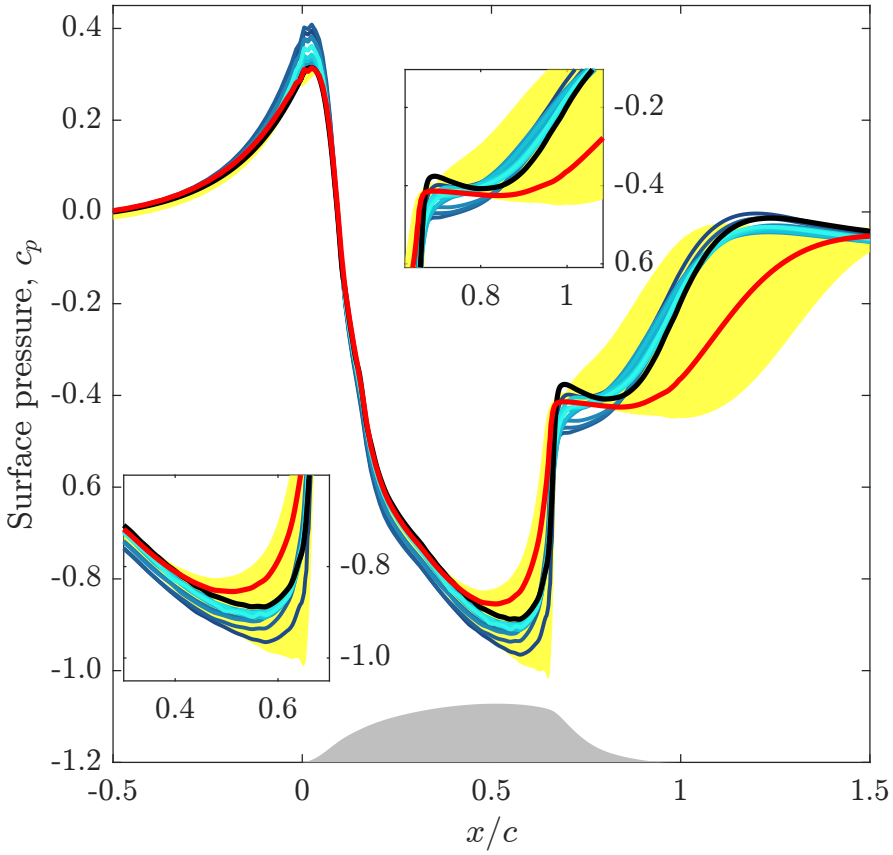


Figure 4.15: Surface pressure predictions on the 2D NASA hump wall. For the legend refer to Fig. 4.14.

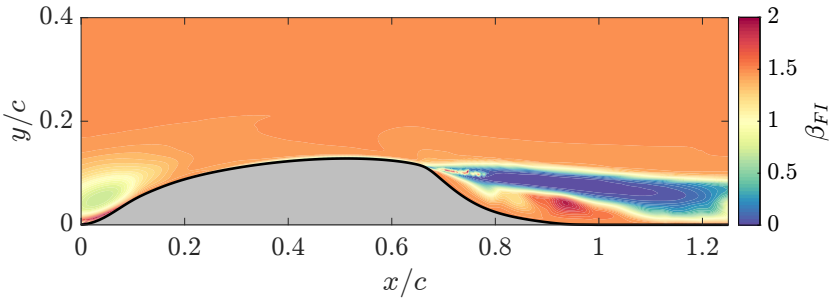


Figure 4.16: The corrective field β for the 2D NASA hump case, modifying the SST transport equation after data assimilation. This is for the case with $N_s = 33$.

be used as a reasonable guide to approximating the number of sensors *a priori* (i.e. by setting the number of sensors around the values when $\mathcal{M}_2$ levels off). However, there is no guarantee of optimality. In addition, as previously mentioned, the number of sensor may also be dictated by other constraints, (e.g. operational budget, etc.).

Next, we present some more detailed results in terms of normalised streamwise velocity profiles (Fig. 4.14) and surface pressure distribution on the hump wall (Fig. 4.15). The velocity profiles show that all field inversion scenarios are able to considerably reduce the errors in the separated shear layer. Broadly, increasing the number of sensors leads to an improved reconstructed quantity, however, as previously mentioned, using a very large dataset does not mean a proportionally similar error reduction. Encouragingly, Fig. 4.15 shows that using relatively sparse velocity data also leads in better reconstruction of the surface pressure distribution, notably in and around the separation zone, $0.5 \lesssim x/c \lesssim 1.5$.

Finally, the corrective field $\boldsymbol{\beta}$ is shown in Fig. 4.16. Highly non-linear $\boldsymbol{\beta}$ distribution is observed near the hump wall, especially in the shear layer. However, corrections are also applied in the farfield (i.e. $\boldsymbol{\beta} \neq 1$ for $y/c \gtrapprox 0.2$), where the baseline model predictions are satisfactory. Ideally, the $\boldsymbol{\beta}$ value should be close to unity in these regions. This can be because the optimisation is not well-regularised, i.e. the constant of regularisation λ in Eqn. 3.10 is too small, to avoid huge deviations from the baseline model.

4.4.2 Converging-diverging channel

The second test case is the converging-diverging channel, Fig. 3.10, previously used in Ch. 3.5.2. The eigenspace perturbation-based uncertainty map for the streamwise velocity field is shown in Fig. 4.17. The regions of highest uncertainty is aft of the channel bump apex, particularly concentrated in the separated shear

layer. We show comparison of the streamwise velocity profiles in Fig. 4.18. The baseline $k-\omega$ SST model significantly over-predicts the flow separation and the recirculation zone. For data assimilation, we prescribe $N_s = 15$ sensors, and place these using the proposed algorithm. The entire domain is assumed to be potential sites for sensors, i.e. $\chi = \boldsymbol{X} \in \mathrm{R}^{N_{\text{cells}}}$, with $N_{\text{cells}} = 98,700$. The initial radius is selected using the metric, $\mathcal{M}_1$ (Eqn. 4.13) leading to $r_{\text{initial}} = 0.53H$.

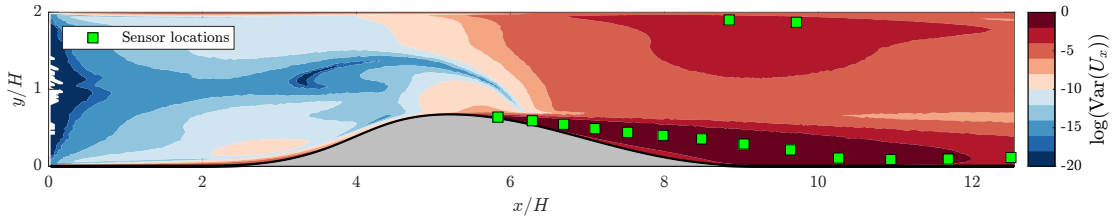


Figure 4.17: The streamwise velocity uncertainty map for the converging-diverging channel.

The resulting sensor locations are shown in Fig. 4.17. Reconstructed flow result with this configuration demonstrates significant improvements to the baseline predictions, especially in the separation and flow recovery region near the lower wall. As the velocity profiles demonstrate, the flow reconstruction with the streamwise velocity data in the entire flow field, leads to a marginally better field inversion result, also summarised in terms of the root-mean-square error reduction in Table 4.1.

Similar to the NASA hump case, the corrective field $\boldsymbol{\beta}$ distribution in the

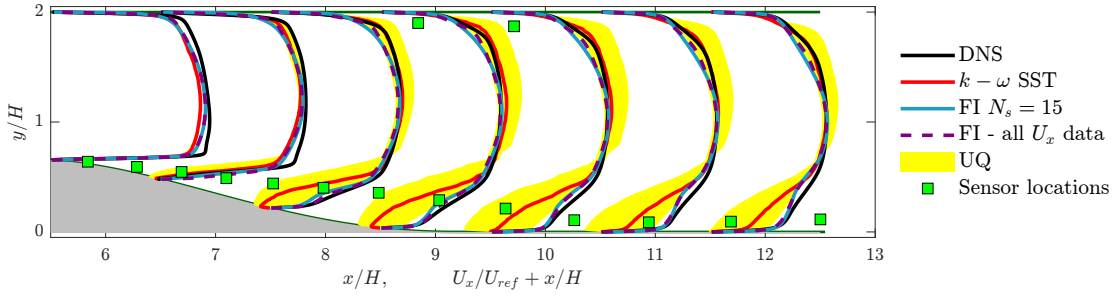


Figure 4.18: Velocity profiles comparison for the converging-diverging channel.

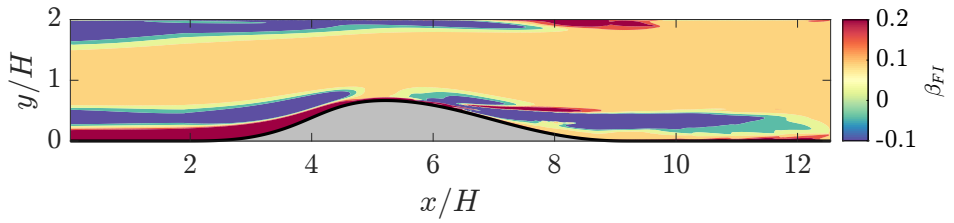


Figure 4.19: The corrective field β for the converging-diverging channel flow, using $N_s = 15$.

Table 4.1: Data assimilation root-mean-square error reduction for the converging-diverging channel flow.

Case	N_s	ΔRMSE(U_x), %	ΔRMSE(U_y), %	Average, %
ESP	15	51.4	58.4	54.9
All U_x	98,700	57.3	59.5	58.4

farfield is not equal to unity. Again, this may be due to poor regularisation.

4.4.3 Periodic hill

The final test case is the separated periodic hill flow, Fig. 3.15, previously used in Ch. 3.5.3. The uncertainty map from eigenspace perturbations are shown in Fig. 4.20. The uncertainty map is based on the combined variances of the streamwise and wall-normal velocity components, and the pressure. The maximum regions of uncertainty are concentrated near the upper and lower walls in general, and around the hills, in particular. Again, we place 15 sensors, with the initial exclusion domain radius, $r_{\text{initial}} = 0.68H$, which is selected iteratively using the metric $\mathcal{M}_1$ (Eqn. 4.13).

We present flow reconstruction results using $N_s = 15$ sensors placed using the proposed framework, and using the entire streamwise velocity field data (i.e. $N_s \equiv N_{cells} = 14,751$). The root-mean-square errors for the velocity components in the two field inversion scenarios are reported in Table 4.2. We note that the

flow reconstruction with just 15 sensors is marginally inferior to the case when all the streamwise velocity data is used.

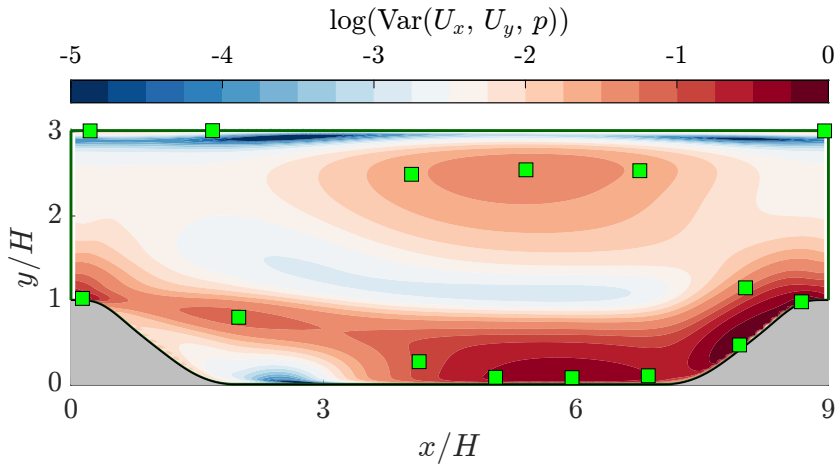


Figure 4.20: The uncertainty map for the periodic hill case. The variance is based on a combination of the streamwise and wall normal velocity components, and the pressure.

Streamwise velocity profiles are shown in Fig. 4.21. The baseline SST model is inaccurate near the walls, especially in the separated shear layer near the lower wall, as captured in the uncertainty ranges from the eigenspace perturbations. The baseline model over-predicts the size of the separation bubble. Both field inversion scenarios are able to improve the predictions, however, some discrepancies remain in the region close to the second hill. Once again, the improvements using only 15 data points vs. the entire streamwise velocity field is noteworthy.

Table 4.2: Data assimilation root-mean-square error reduction for the periodic hill flow.

Case	N_s	ΔRMSE(U_x), %	ΔRMSE(U_y), %	Average, %
ESP	15	42.5	50.1	46.7
All U_x	14,751	55.6	43.3	49.4

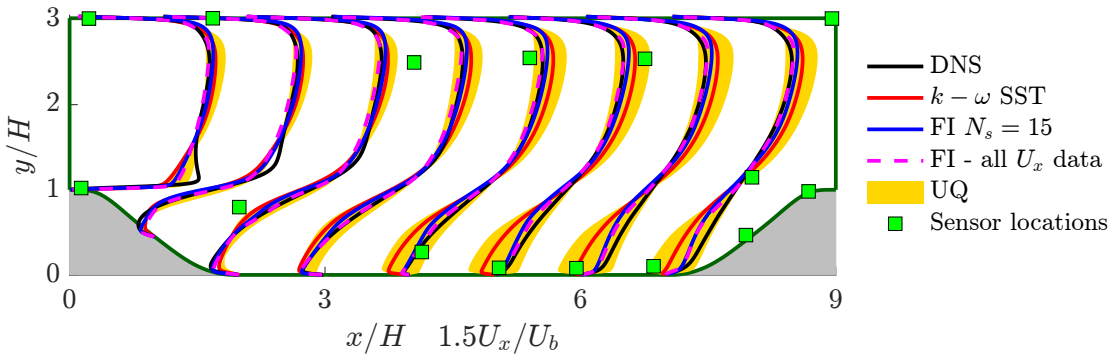


Figure 4.21: The streamwise velocity profiles comparison for the periodic hill case.

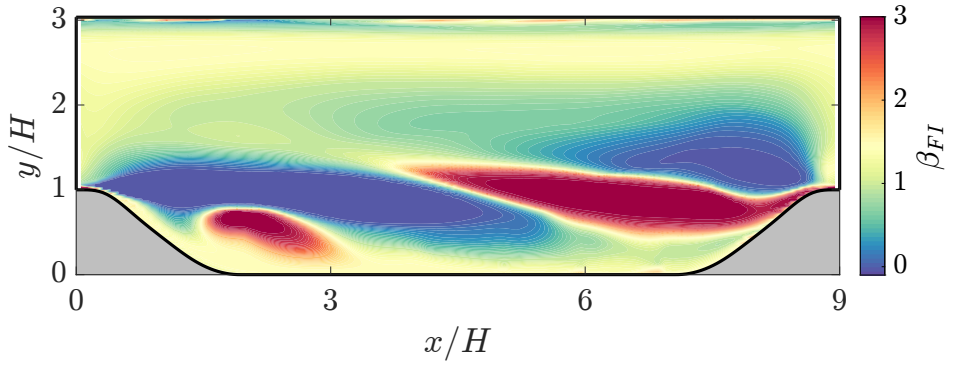


Figure 4.22: The corrective field β for the separated periodic hill flow.

4.5 Summary

We presented a novel strategy to sensor placement for data assimilation in the context of RANS-based turbulent flow reconstruction. The framework relied on generating a spatial uncertainty map through systematic perturbations of the Reynolds stress tensor. An optimisation-based greedy search was proposed to place sensors. Data assimilation was performed using a variational (adjoint-based) approach.

The proposed framework was tested on three two-dimensional wall-bounded fully turbulent flows with Reynolds numbers ranging from 5.6×10^3 to 9.36×10^5. All flows involved mild to massive separation which were over-predicted by the baseline $k-\omega$ SST model. Data assimilation were performed using streamwise ve-

locity data. Average root-mean-square error reduction in the velocity predictions (compared to the baseline SST model) after data assimilation are summarised as follows:

1. 2D NASA wall-mounted hump: the error was reduced by 61% just using 3 sensor data points using the proposed placement method. For comparison, using 33 sensor points with uniform sensor placement led to an error reduction of 38%, while using the entire available streamwise velocity data (30.7×10^3 data points) achieved 74% error reduction.

2. Converging-diverging channel: average error reduction of 55% with 15 sensor points using the proposed method, compared to a 58% reduction when using all the streamwise velocity field (98.7×10^3 data points).

3. Periodic hill flow: average velocity error reduction of 47% with 15 sensor points, compared to 49% when using all the data (14.7×10^3 data points).

CHAPTER 5

Multi-sensor data for data assimilation

5.1 Introduction

Following on from chapter 4, we further investigate the synergy of experimental measurement data and RANS-based simulations in this chapter. This chapter addresses the problem of using multiple types of sensor data in adjoint-based field inversion. This is an important gap in the literature because different types of sensor data can be collected for use in data assimilation and ideally, a computational model should accurately describe all the data. In practice, there can be a trade-off in optimising the model to describe different types of sensor data, making this a challenging problem. The novel contribution of this chapter is the development of a multi-objective optimisation scheme for field inversion that can handle multiple sensor types, which has been published in [19].

Zhang *et al.* [175] have investigated the use of ensemble-based data assimilation with multiple data sources simultaneously. They employed a regularised ensemble Kalman method for three incompressible flows: one-dimensional channel flow, 2D flow over a flat plate, and the 2D periodic hill flow. Types of data used included velocity profiles in the flow field, and wall measurements (pressure at the walls, surface friction, and lift and drag forces). Results presented show that the accuracy of reconstructed mean flow quantities such as velocity and pressure can be further enhanced when using multi-sensor data.

Waschkowski *et al.* [160] developed interpretable algebraic closures using a gene expression programming (GEP) algorithm. They use high-fidelity simulation data to perform a multi-objective optimisation with two different physical quantities in the loss function. The multi-objective optimisation is based on the non-dominated sorting genetic algorithm (NSGA-II) [34], which employs the concept of Pareto dominance. The framework was tested on two flows: a one-dimensional vertical natural convection flow, and the periodic hills flow. In both cases, only volume data (profiles in the flow domain) were used, as opposed to volume and wall quantities in the aforementioned work by Zhang et al. [175]. For the natural convection flow the quantities used were the streamwise velocity and the temperature profiles, while for the periodic hill case it was the streamwise velocity and the turbulent kinetic energy profiles. It was shown that using multi-sensor data results in better mean flow reconstruction of not only the quantities used during training, but also other mean flow variables.

Most applications of field inversion using adjoint-based methods has been limited to using only a single source of data. The types of single source data used have included volume data (e.g. velocity profiles [119]), surface data (e.g. skin friction [39], and pressure coefficient [12]), and integral data (e.g. lift coefficient [139])—listed in terms of lowest to highest data sparsity. Using only a single source of data is a limitation because multiple sources can be used simultaneously to train the model and in theory improve the accuracy of the model and ensure that it fits all available data correctly.

The aim of this chapter is to extend the adjoint-based field inversion framework, a popular data assimilation alternative to the two techniques discussed above, to make use of multiple data sources for enhanced turbulent mean flow reconstruction. The approach we take is to use a composite objective function with least squares error terms representing the data for multiple sensors and sources—

leading to a weighted least squares optimisation problem. We investigate the multi-objective nature of this optimisation by systematically varying the weighting terms to recover a Pareto-optimal front of the trade-off resulting from differences in weighting the sources of data.

5.2 Weighted sum multi-objective optimisation

The one-equation Spalart–Allmaras (S-A) model is chosen as the baseline turbulence model, and modified by the corrective field $\boldsymbol{\beta}$ as described in chapter 3.2. The weighted-sum objective function for the optimisation can be written as,

$$\min_{\boldsymbol{\beta}} \quad \mathcal{J} = \left(\sum_{i=1}^{N_d} w_i \frac{1}{\mathcal{J}_{0,i}} \|\mathcal{G}_i\left(\boldsymbol{\beta}\right) - \boldsymbol{d}_i\|_2^2 \right) + \lambda \|\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{prior}}\|_2^2, \tag{5.1}$$

where $\|.\|_2$ is the $L2$ norm; the index $i \in \mathbb{R}^{N_d}$ represents the different data quantities (e.g. velocity, and pressure); $\boldsymbol{d}_i$ represents the specific quantity from high-fidelity data, with $\mathcal{G}_i(\boldsymbol{\beta})$ representing the RANS model equivalent; w_i are the weights; $\mathcal{J}_{0,i}$ are the least-square errors between the baseline model and data; λ is a regularisation parameter, and $\boldsymbol{\beta}_{\text{prior}}$ is typically assumed to be 1, to bias the solution closer the baseline model. The formulation would be ill-posed without the regularisation term because of noisy data and the high degree of freedom in the model compared to the number of data points. We systematically vary the weights w_i to perform a multi-objective optimisation and investigate the trade-off between weighting different sources of data and obtain the Pareto-optimal front describing this trade-off.

5.3 Results

We present the results for two separated flow cases: 1) the 2D NASA wall-mounted hump [57], with available experimental data from the NASA Turbulence Modelling

Table 5.1: Summary of cases, where M, Re, C_p, C_f are the Mach number, Reynolds number, surface pressure, and skin friction respectively.

Geometry	Flow conditions	Data	Data type	Data source
2D hump	$M_{\text{ref}} = 0.1$	U_x/U_{ref}	Volume	Experiment
	$Re_c = 936,000$	C_p	Surface	
Periodic hill	$M_{\text{ref}} = 0.2$	U_x/U_b	Volume	LES
	$Re_H = 10,595$	C_f	Surface	

Resource database, and 2) flow over a periodic hill, with detailed LES results from [55]. Both cases involve complex flow features due to separation resulting in discrepancy between eddy viscosity model predictions compared to high-fidelity results, as shown in the preceeding chapters. The cases are summarised in Table 5.1.

5.3.1 NASA wall-mounted hump

This case involves flow separation, as a result of adverse pressure gradients, over a smooth hump surface, shown in Fig. 5.1.

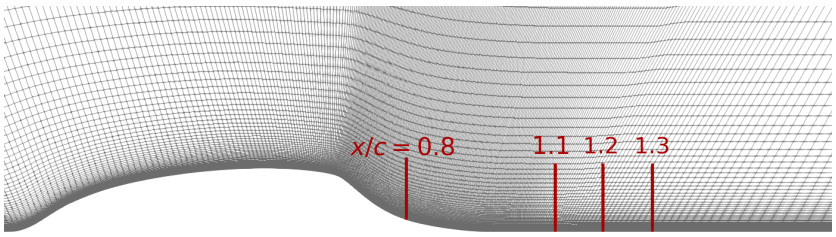


Figure 5.1: Close-up of the NASA wall-mounted hump.

Linear eddy viscosity models, such as Spalart–Allmaras, are known to poorly predict the separation, reattachment and boundary recovery by over-predicting the size of the separation bubble due to under-predicted turbulent shear stress

in the separated region, as also shown for the $k-\omega$ SST model in chapter 4.4.1. The available data include experimental surface pressure, and velocity profiles near the hump. For field inversion we use the surface pressure data in the region $-0.8 \leq x/c \leq 2.1$, and four streamwise velocity profiles as shown in Fig. 5.1. The objective function is:

$$\mathcal{J} = \underbrace{\frac{w_u}{\mathcal{J}_{0,u}} \sum_{i=1}^{N_d^u} \left[U_{x,i}\left(\boldsymbol{\beta}\right) - U_{x_{\text{data}},i}\right]^2}_{\mathcal{J}_u} + \underbrace{\frac{w_p}{\mathcal{J}_{0,p}} \sum_{j=1}^{N_d^p} \left[C_{p,j}\left(\boldsymbol{\beta}\right) - C_{p_{\text{data}},j}\right]^2}_{\mathcal{J}_p} + \lambda \sum_{k=1}^{N_m} \left[\boldsymbol{\beta}_k - 1\right]^2, \tag{5.2}$$

where $w_u + w_p = 1$, and the regularisation constant λ is set to 10^{-6}. To simulated the flow, we solve the incompressible, two-dimensional, steady Navier–Stokes equations on a structured mesh, shown in Fig. 5.1.

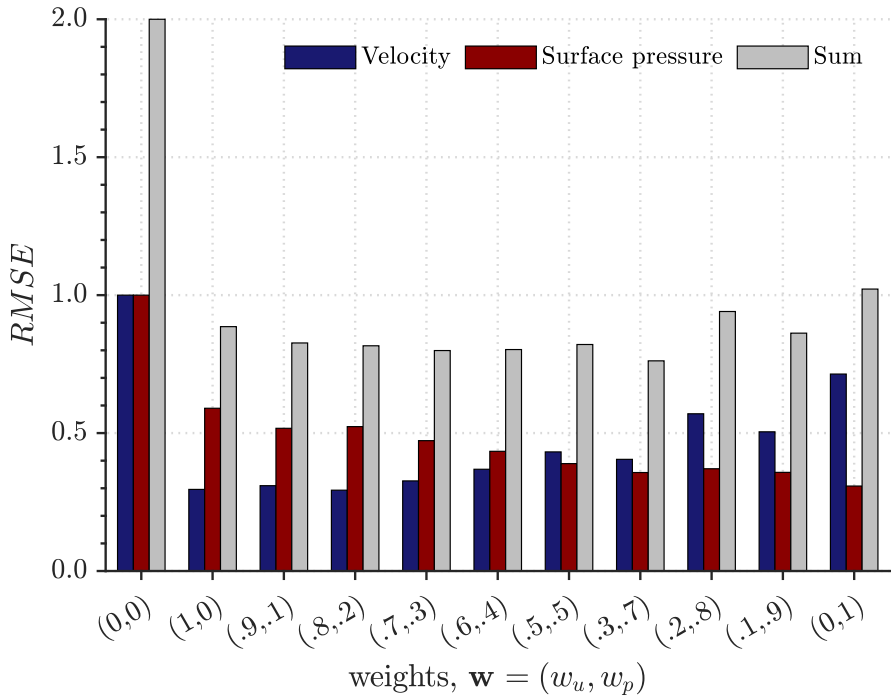


Figure 5.2: The root-mean-squared error for the 2D hump for the baseline model (labelled as $\boldsymbol{w} = (0, 0)$), and various field inversion scenarios. The RMSE values are normalised by the respective RMSE values of the baseline model.

Fig. 5.2 shows the root-mean-squared error in velocity and surface pressure predictions. It is clear that the total error can be significantly reduced by all field inversion scenarios. The least effective field inversion scenarios are when using data for only one quantity, as expected, although the reconstruction with velocity data is more effective than pressure data. Broadly, the field inversion scenarios with multi-sensor data lead to similar *total* error reduction, although there are clear differences in *individual* quantities. We observe that, generally, a larger weight for a particular quantity leads to a large error reduction for that quantity, and vice-versa, as intuitively expected.

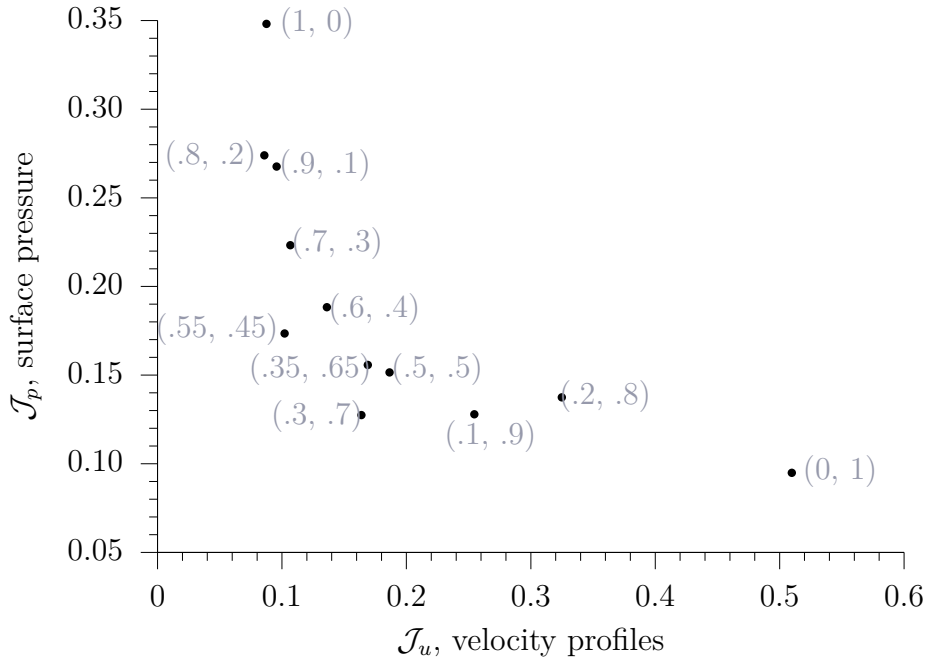


Figure 5.3: The Pareto front plot for the 2D hump. The weights are provided next to markers: weights, $\boldsymbol{w} = (w_u, w_p)$.

A comparison of the individual objective function terms $\mathcal{J}_u$ and $\mathcal{J}_p$ is shown in Fig. 5.3. It broadly shows an emerging Pareto front for the multi-objective optimisation, demonstrating the need for a trade-off between the two objectives: one of terms in the composite objective function degrades when the other is im-

proved. According to both figures, the best case scenario is with the weights, $(w_u, w_p) = (0.3, 0.7)$. However, this is not far superior to equally weighting the objective function terms, thus it could be argued that equal weights is sufficient, for this case.

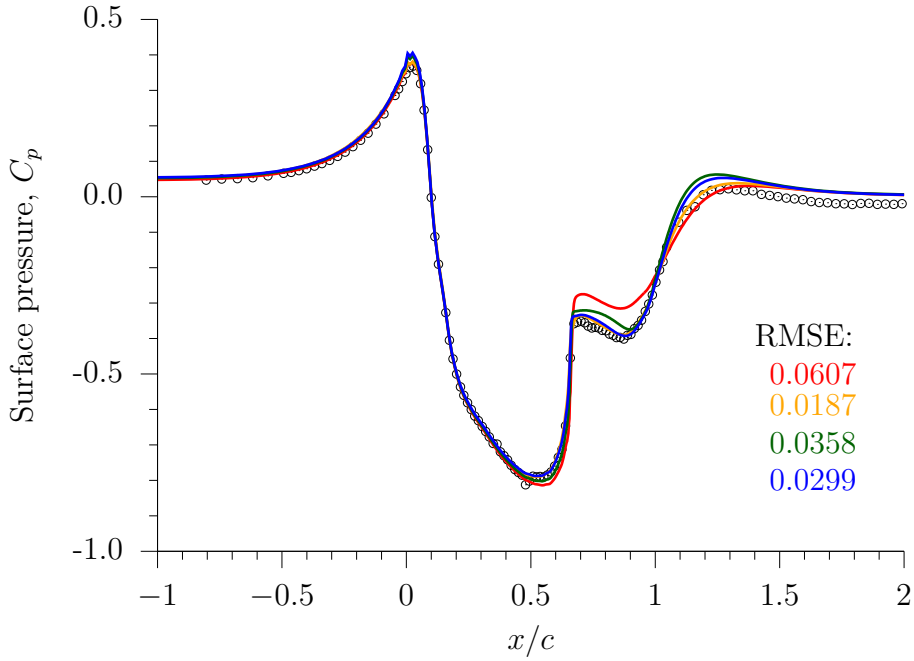


Figure 5.4: Comparison of surface pressure on the NASA wall-mounted hump wall. Legend: Experiment (○), Spalart–Allmaras (—); field inversion, C_p data (—); field inversion, U_x profile data (—); and field inversion, equally weighted C_p and U_x profile data (—). The RMSE label refers to the root-mean-squared error between the simulation and experimental data, with values colour-coded to match the legend.

Next, we will compare the baseline and field inversion results (with single and equally-weighted multi-sensor data) in more detail. Fig. 5.4 shows the surface pressure distribution. It is clear that the S-A model struggles to accurately predict the pressure in the separated shear layer, which is improved by all three field inversion scenarios. However, there is still some error in the pressure distribution when using velocity profiles alone.

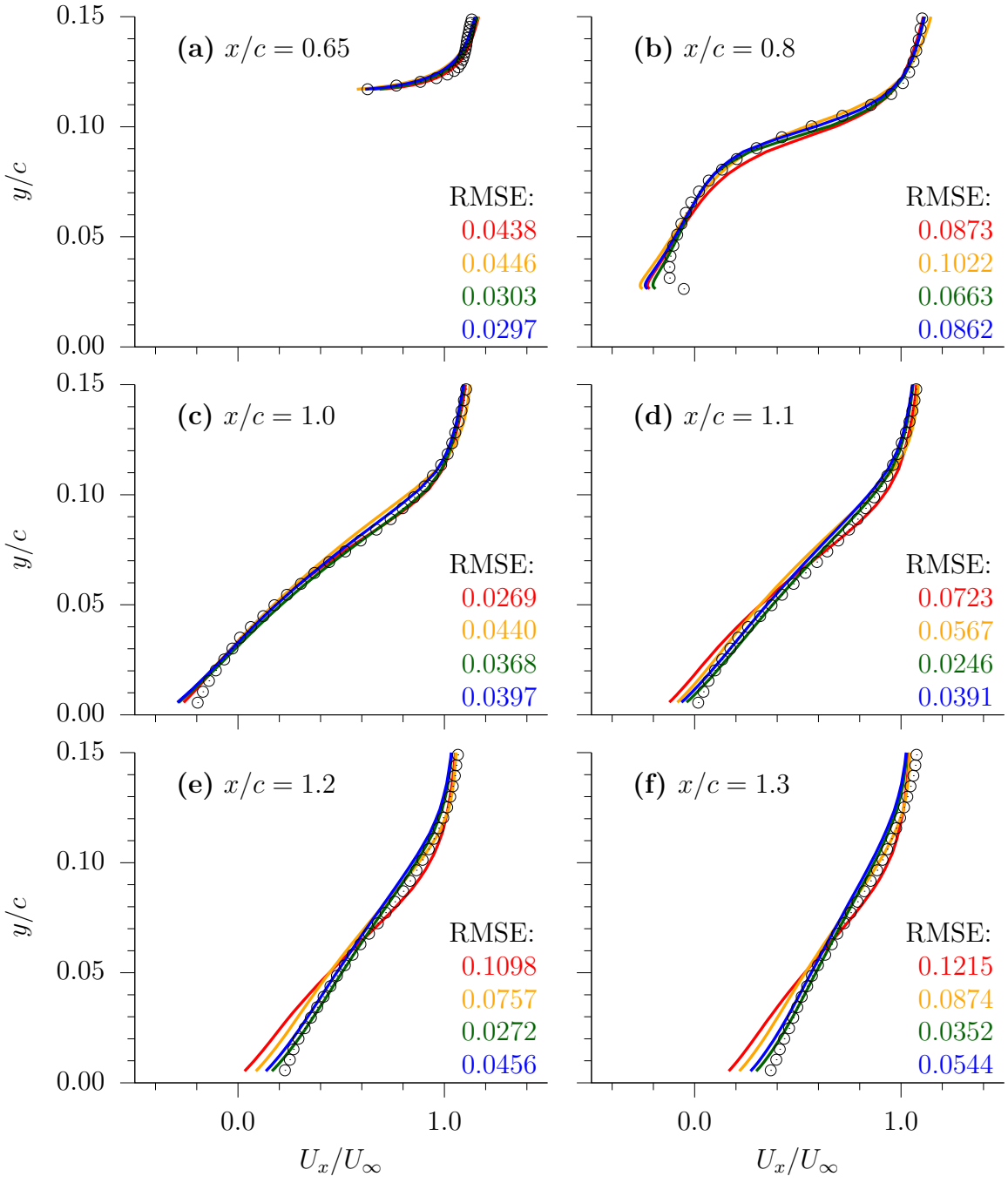
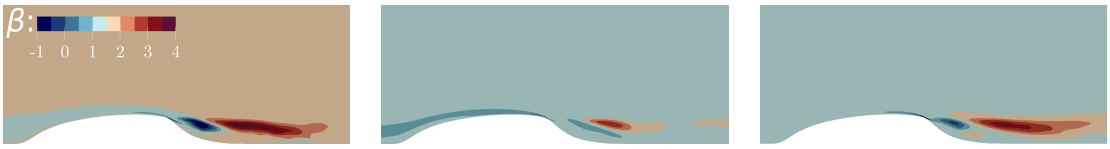
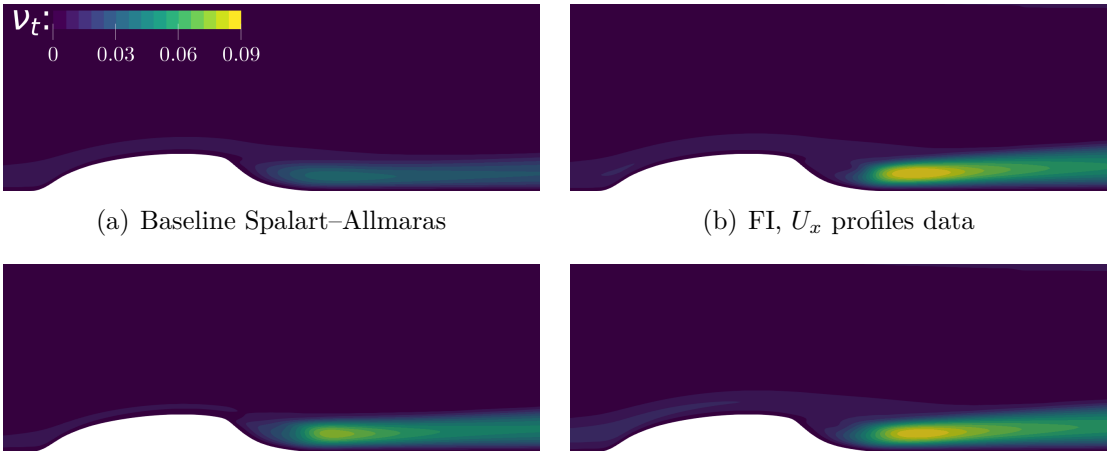


Figure 5.5: Comparison of the velocity profiles in the separated flow region of the NASA wall-mounted hump. Legend: Experiment (○), Spalart–Allmaras (—); field inversion, C_p data (—); field inversion, U_x profile data (—); and field inversion, equally weighted C_p and U_x profile data (—).



(a) Velocity profiles data (b) Surface pressure data (c) Equally-weighted U and C_p data

Figure 5.6: Comparison of the corrective field, $\boldsymbol{\beta}$, for the NASA wall-mounted hump with different field inversion scenarios.



(a) Baseline Spalart–Allmaras (b) FI, U_x profiles data

(c) FI, C_p surface data (d) Equally-weighted U and C_p data

Figure 5.7: Comparison of the eddy viscosity, ν_t, for the NASA wall-mounted hump case before and after modifications by $\boldsymbol{\beta}$ fields shown in Fig. 5.6.

The velocity profiles in Fig. 5.5 show that the baseline model under-predicts the streamwise velocity in the separated shear layer closer to the hump wall. This means that the baseline model predicts a larger separation bubble, and hence delayed reattachment, compared to the experimental data. Performing field inversion with velocity data leads to a more accurate characterisation of the separation, as shown in the velocity profiles. Additionally, it is clear that although the surface pressure data can lead to a very good improvement in the pressure distribution, it is less effective in reducing the error in the velocity predictions—supporting the case for flow reconstruction based on multi-sensor data.

The corrective fields modifying the baseline model to reconstruct the turbulent

mean flow is shown in Fig. 5.6. All cases show a complex distribution where the surrogate turbulence variable, $\tilde{\nu}$, in the S-A model is magnified or dampened, especially in and around the separation. Interpreting the $\boldsymbol{\beta}$ fields with reference to the eddy viscosity ν_t contours in Fig. 5.7 show that the various field inversion scenarios increase ν_t in the shear layer which reduces the size of the separation bubble. Negative values of $\boldsymbol{\beta}$ in the recirculation region and close to the hump wall lead to a reduction of eddy viscosity, hence a slightly delayed separation. The farfield $\boldsymbol{\beta}$ values for the field inversion case with C_p data and combined pressure and velocity data are close to unity (Fig. 5.6 (b-c)), which is expected since the baseline model predictions are satisfactory in these regions. However, for the case with velocity profile data the values are between 1 and 2. As discussed in chapter 4.4.1, this may be due to poor regularisation.

5.3.2 Periodic hill

The second flow is the periodic hill case, previously used in chapters 3.5.3 and 4.4.3. In this section, the LES dataset by Gloerfelt *et al.* [55] is used for flow reconstruction, with the flow conditions summarised in Table 5.1.

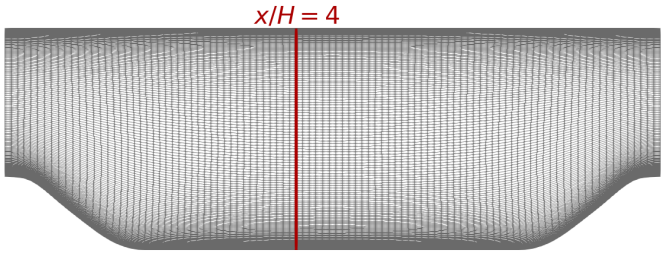


Figure 5.8: The structured periodic hill mesh, with the location of the velocity profile used for data assimilation.

Two sources of data are used for field inversion: 1) vertical continuous slice of streamwise velocity at $x/H = 4$ station (chosen as this is the region close to flow reattachment), and 2) the skin friction in the lower wall. The objective function

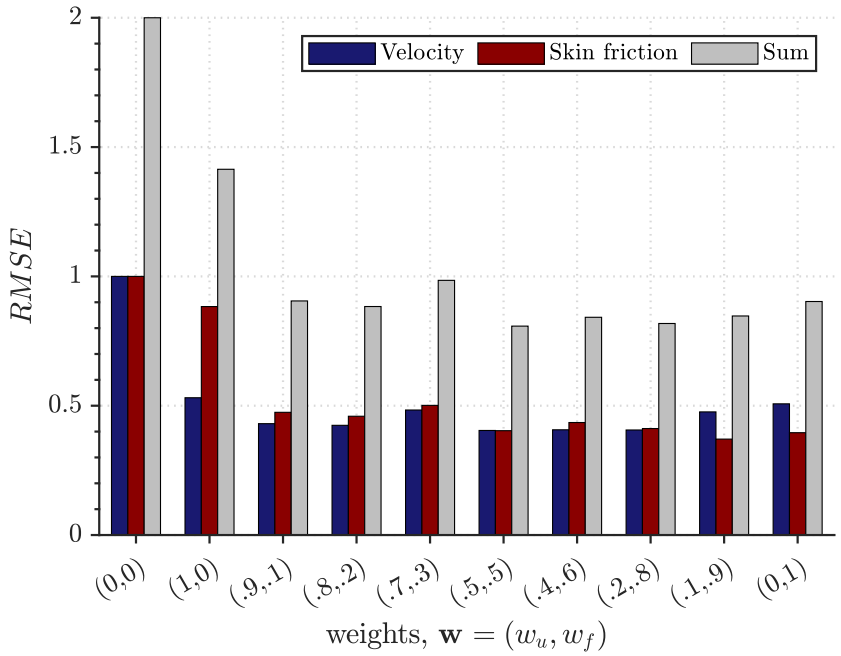


Figure 5.9: The periodic hill root-mean-squared error for the baseline model (labelled as $\boldsymbol{w} = (0, 0)$), and various field inversion scenarios. The RMSE values are normalised by the respective RMSE values of the baseline model.

is similar to the hump case, with the C_p term in Eqn. 5.2 replaced by C_f, and the regularisation constant is set to $\lambda = 10^{-10}$, a very small value to reflect high confidence in the data used for reconstruction.

Fig. 5.9 shows that all field inversion scenarios can considerably reduce the root-mean-squared errors for velocity and skin friction prediction compared to the baseline Spalart–Allmaras model. However, using skin friction data on the lower wall is significantly more effective compared to the velocity profile at $x/H = 4$. This is also reflected in the fact that the RMSE is lower for $w_f \geq 0.5$ (i.e. more biased towards fitting C_f data) compared to $w_u \geq 0.5$. The most effective scenario in terms of RMSE error reduction is when the individual data terms in the objective function (i.e. $\mathcal{J}_u$ and $\mathcal{J}_f$) are equally-weighted.

Fig. 5.10 shows a comparison of the individual objective terms, $\mathcal{J}_u$ and $\mathcal{J}_f$, for different weights. In contrast to the hump case, this case does not necessarily

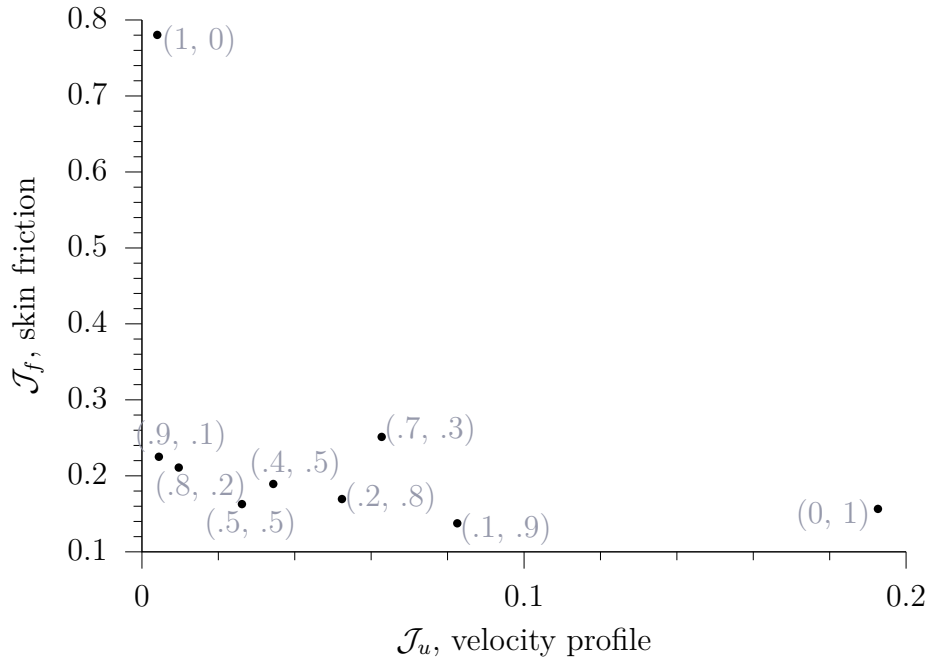


Figure 5.10: The Pareto front plot for the periodic hill. The weights are provided next to markers: weights, $\boldsymbol{w} = (w_u, w_p)$.

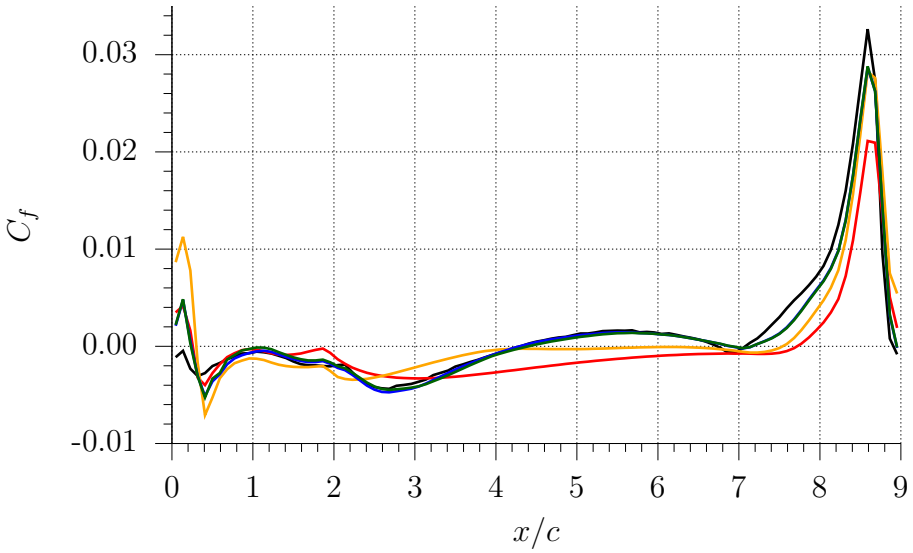


Figure 5.11: Comparison of skin friction on the lower wall for the periodic hill case. Legend: LES (—); Spalart–Allmaras (—); FI, U_x profile at $x/H = 4$ (—); FI, C_f (—); and FI, equally weighted with C_f and U_x (—).

show a clear conflict between minimising the individual terms in the objective function. While in the hump case the error reduction in an individual objective function term is proportional to the weight, in the periodic hill case this is not as clear. In fact, field inversion scenarios with multi-sensor data can lead to further error reduction in both objectives compared to data from a single source (unlike the hump case). This is expected as C_f is a function of the velocity derivative at the wall.

A comparison of the skin friction predictions by the baseline model and field inversion scenarios with C_f data, velocity profile, and both (equally-weighted) is shown in Fig. 5.11. The baseline model significantly over-predicts the size of the separation bubbles, and thus a very delayed flow reattachment. The field inversion scenarios with skin friction data, and equally weighted velocity profile and C_f data, significantly improve the prediction of separation and reattachment between $1 \leq x/H \leq 4$. Neither the baseline model, nor the different field inversion scenarios capture the slight separation at the initial hill crest, but all field inversion scenarios show improved C_f peak predictions on the second hill, compared to the baseline case. Flow reconstruction with a single velocity profile at $x/H = 4$ is insufficient to capture the complex flow characteristics in the near-wall region, although it does improve it compared to the baseline, by reducing the size of the separation bubble—a mild separation is still observable in and after the location the LES data suggests flow reattachment.

The velocity profiles shown in Fig. 5.12 show that field inversion with C_f data can highly improve the near wall (lower) velocity predictions. But, the effect away from the wall is diminished, notably towards the first-half of the domain. On the other hand, using the velocity profile can lead to improved velocity prediction through the majority of the domain, except some errors in the near wall region prior the second hill. It is clear that when combining both datasets, the velocity

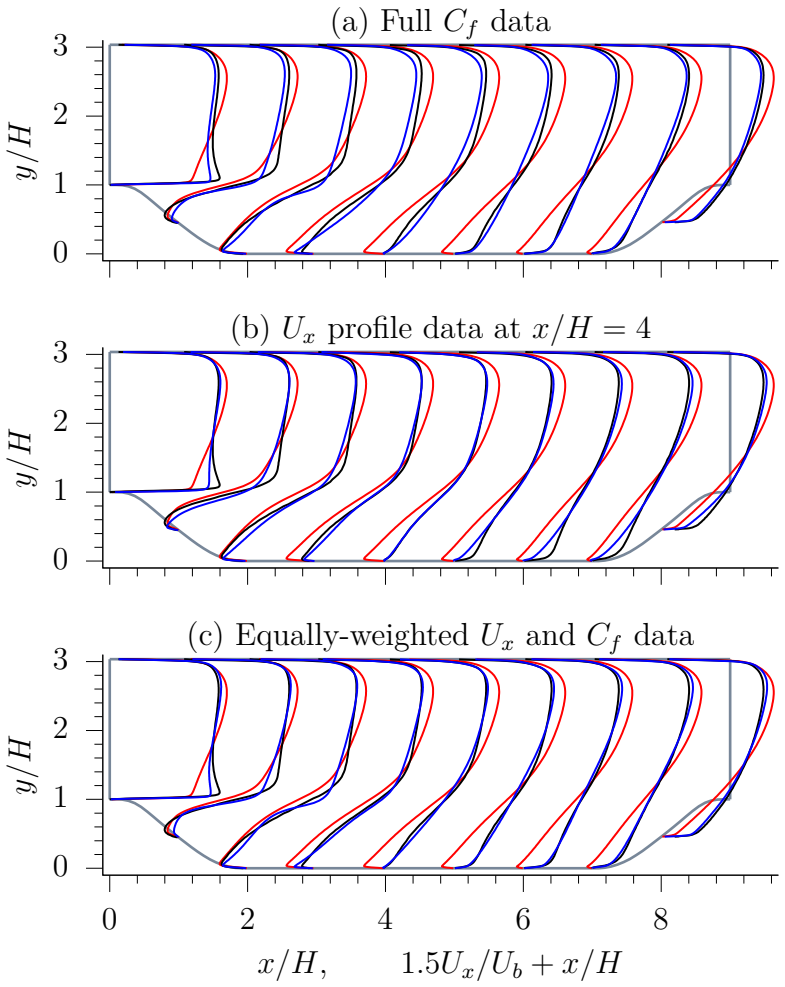
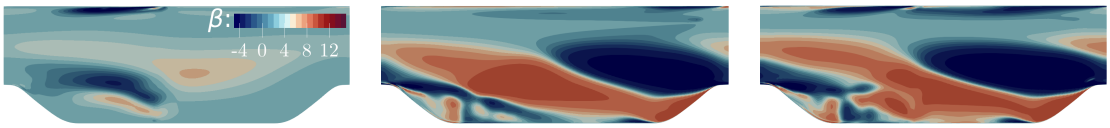
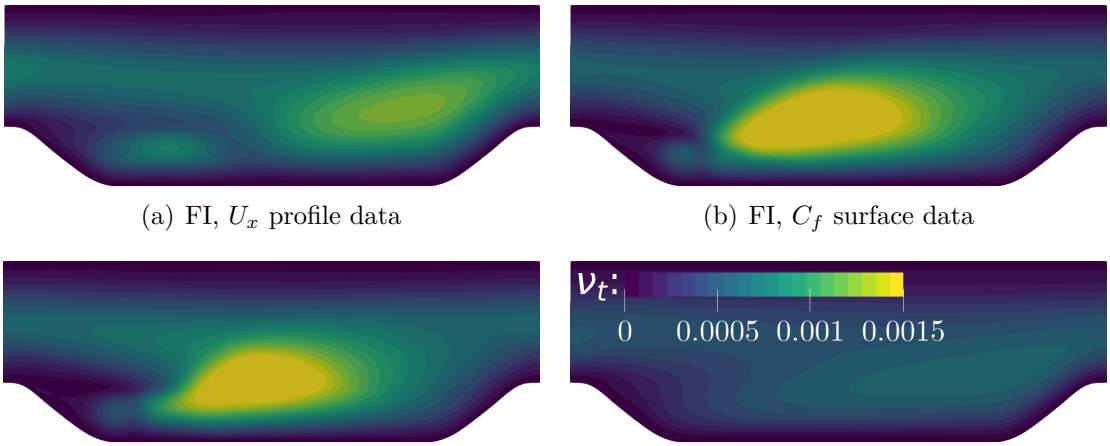


Figure 5.12: Comparison of velocity profiles for the periodic hill case. Legend: LES (—), Spalart–Allmaras (—), and field inversion (—).



(a) Velocity profiles data (b) Surface pressure data (c) Equally-weighted U and C_p data

Figure 5.13: Comparison of the corrective field, $\boldsymbol{\beta}$, for the periodic hill with different field inversion scenarios.



(a) FI, U_x profile data (b) FI, C_f surface data

(c) Equally-weighted U and C_f data (d) Baseline Spalart–Allmaras

Figure 5.14: Comparison of the eddy viscosity, ν_t, for the periodic hill before and after modifications by $\boldsymbol{\beta}$ fields shown in Fig. 5.13.

predictions can be significantly improved throughout the domain.

The corrective fields modifying the baseline S-A transport equation is shown in Fig. 5.13. All three field inversion scenarios show substantial changes to $\tilde{\nu}$ (thus ν_t) throughout the domain, with considerable variations of $\boldsymbol{\beta}$ in the separated shear layer. In the least effective case (field inversion with velocity profile data) the eddy viscosity (Fig. 5.14a) is increased in most of the domain, with reduction ($\boldsymbol{\beta} \leq 0$) above the recirculation zone, and the boundary layer in the top wall. In the other two cases, there are drastic magnifying of eddy viscosity in/surrounding the shear layer 5.14 b and c), while eddy viscosity is reduced around the crest of the second hill (Fig. 5.13 b and c).

Similar to the hump case, the baseline Spalart–Allmaras model under-predicts the eddy viscosity in the separated shear layer, causing a large separation bubble (hence, delayed flow reattachment). Given enough data, field inversion can remedy this issue, as demonstrated.

5.4 Summary

In this chapter the enhanced capability of turbulent mean flow reconstruction using multi-sensor data was demonstrated for two flows involving strong pressure gradients and flow separation. The variational/adjoint-based field inversion involved modification of the Spalart–Allmaras turbulence model through a spatial corrective field introduced in the model transport equation. The two cases tested showed that when using multi-sensor data, equally-weighting the individual terms in the objective function is appropriate. It was also shown that the baseline Spalart–Allmaras model under-predicts the eddy viscosity in the separated shear layer, which results in a larger separation bubble than observed in high-fidelity data. In the wall-mounted hump case we observed that velocity profiles were more effective in reconstructing turbulent flow compared to surface pressure. On the other hand, the periodic hill case was more sensitive to skin friction (velocity derivative) data, compared to the streamwise velocity profile.

CHAPTER 6

Development of a neural network-augmented turbulence model

6.1 Introduction

This chapter is the first part, paired with chapter 7, in moving toward addressing the problem of shape optimisation using field inversion methods. The methods developed up to this point in the thesis provide a computational framework to perform field inversion. This chapter uses these methods and augments them with the crucial second step, which is neural network correction of the physics model using results generated by field inversion. This chapter provides the groundwork for chapter 7 where the outputs from this chapter are used in shape optimisation. Together, chapters 6 and 7 address the research gap of using field inversion methods to improve separated turbulent flow predictions in the context of aerodynamic shape optimisation. This has been published in Ref. [17].

6.2 Field inversion and machine learning framework

Recall that field inversion approach modifies an existing turbulence model by introducing a *spatial* corrective field, $\boldsymbol{\beta}(\boldsymbol{x})$, to the model transport equation and then optimising the $\boldsymbol{\beta}$ field to reduce the error between the RANS output and the high-fidelity data . The spatial nature of the corrective field—defined for ev-

ery mesh cell—limits it to the case (i.e. flow conditions and geometry) for which the flow is reconstructed using the high-fidelity data. The corrective information through the data assimilation should then be converted into modelling knowledge to be used for predictions for different flows. Therefore, machine learning approaches such as deep neural networks (e.g. [139]) or Gaussian processes (e.g. [119]) are used to generalise the model corrections for unseen flow conditions/geometries for predictions. To achieve this, a number of field inversion solutions are required to gain sufficient corrective information for model training.

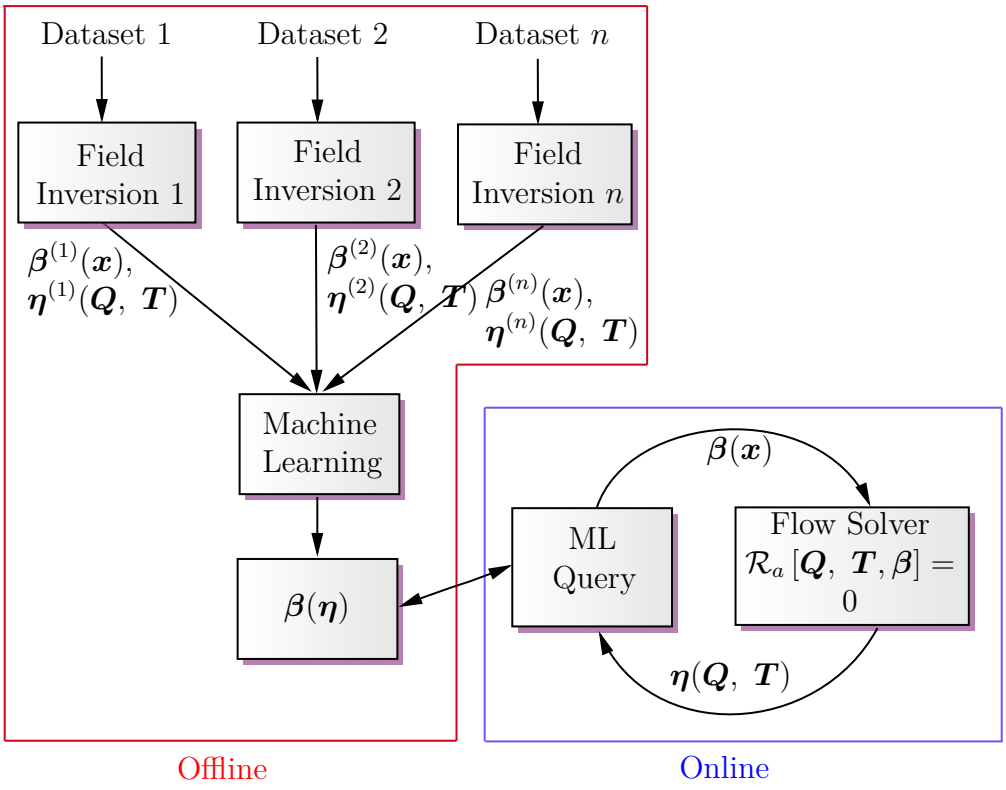


Figure 6.1: Graphical overview of the field inversion and machine learning framework.

The field inversion and machine learning framework, graphically represented in Fig. 6.1, is therefore as follows:

- Let $\mathcal{R}[\boldsymbol{Q}, \boldsymbol{T}] = 0$ represent the Reynolds-averaged Navier-Stokes (RANS)

equations, where $\boldsymbol{Q}$ are the mean flow quantities such as pressure, and velocity, and $\boldsymbol{T}$ are the secondary variables introduced by the turbulence model. The $k-\omega$ shear stress transport (SST) is used as the baseline turbulence model, which has been shown to perform poorly predicting complex turbulent flows (see chapter 3.5.3 and chapter 4.4). Thus the secondary variables are the turbulent kinetic energy (k) and the turbulence dissipation (ω).

- Let $i \in [1, n]$ represent a set of turbulent flows (i.e. different flow conditions, e.g. Reynolds numbers, and/or different geometries).

- The field inversion problem for turbulent flow i is then posed as follows (see chapter 3.2 for details),

$$\boldsymbol{\beta}^i(\boldsymbol{x}) \cdot \mathcal{P}(\boldsymbol{Q}, \boldsymbol{T}), \tag{6.1}$$

$$\min_{\boldsymbol{\beta}} \quad \mathcal{L}[Y_d, Y_m(\boldsymbol{\beta})] \quad \text{s.t.} \quad \mathcal{R}[\boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{\beta}] = 0, \tag{6.2}$$

  where $\mathcal{P}$ represent the production term in the transport equation for a turbulence variable $\boldsymbol{T}$, Y_d represent the flow quantity from data, Y_m represents the RANS-based flow quantity, and $\boldsymbol{\beta}$ is the corrective field. As described in chapter 3.2, the $\boldsymbol{\beta}$ field is introduced in the production term of the ω transport equation.

- Once the corrective fields $\boldsymbol{\beta}(\boldsymbol{x})$ are inferred for i flows, then neural networks are used to map the spatial fields to local flow features, $\boldsymbol{\eta} \in [1, j]$ (Table 6.2), which is another optimisation problem posed as,

$$\min_{\boldsymbol{w}} \quad \mathcal{L}[\boldsymbol{\beta}^i(\boldsymbol{x}), \boldsymbol{\beta}_m(\eta^j; \boldsymbol{w})], \tag{6.3}$$

  where $\boldsymbol{w}$ represents the neural network parameters (i.e. weights and biases).

- Following the neural network training and validation (to be discussed next

in chapter 6.3), the RANS equations with the neural network-augmented turbulence model used for predictions becomes

$$\mathcal{R}[\boldsymbol{Q}(\boldsymbol{\beta}_m(\eta)), \boldsymbol{T}(\boldsymbol{\beta}_m(\eta))]. \tag{6.4}$$

6.3 Neural network: theory and implementation

Fig. 6.2 illustrates a typical fully-connected deep neural network architecture, consisting of an input layer, hidden layer(s), and an output layer. Neural networks are useful to represent complex nonlinear dependencies of some output(s) to some input(s) to performed regression or classification tasks. For our purposes the input features are non-dimensional local flow variables based on physical quantities ($\boldsymbol{\eta}$), while the output is the corrective field, $\boldsymbol{\beta}$.

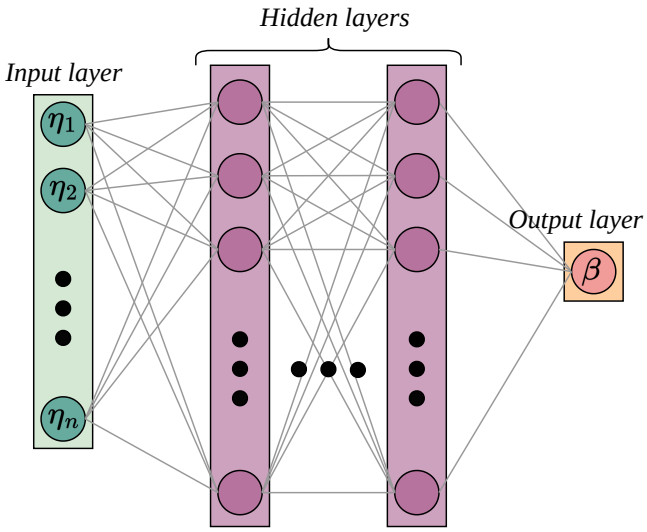


Figure 6.2: Neural network architecture.

Feature selection is critical for the neural network to be able to capture trends in the data. We use an often-employed set of features (Table 6.1 and 6.2) originally curated by Ling *et al.* [89] and subsequently used (in full or in part with additional features) in many studies, e.g. [159, 156, 63, 98]. The chosen features satisfy

a number of characteristics that are deemed essential: the features are locally non-dimensionalised which makes the learned corrective field applicable to flow conditions and/or geometries beyond the training data [37], and are Galilean-invariant, which requires that the physics of the fluid flow should be independent of coordinates frame of the observer [90].

Table 6.1: Description of features used as neural network inputs.

Feature	Description
η_1	Q-criterion
η_2	Turbulence intensity
η_3	Wall-distance based Reynolds number
η_4	Pressure gradient along streamline
η_5	Ratio of pressure normal stresses to shear stress
η_6	Non-orthogonality between velocity and its gradient
η_7	Ratio of total to normal Reynolds stress
η_8	Streamline curvature
η_9	Ratio of convection to production of TKE

The rationale behind some of the selected features such as the turbulent intensity (η_2), the ratio of total to normal Reynolds stress (η_7) and the ratio of convection to production of turbulent kinetic energy (η_9) are clear. These quantities are explicitly involved in turbulence closure modelling through the $k-\omega$ SST model, and hence appropriate input features for the neural network.

The Q-criterion (η_1) is a widely used feature in traditional CFD post-processing to identify vortex structures, and is the ratio of the magnitudes of the rotation-($\boldsymbol{\Omega}$) to strain-rate ($\boldsymbol{S}$). $\eta_1 > 0$ implies the rotational effects dominate shear effects [27]. The wall-distance based Reynolds number (η_3), also widely used in wall-functions in traditional turbulence modelling, is a marker that differentiates

Table 6.2: Formulae for features used as neural network inputs. L_c in η_8 is a characteristic length, which is also used in the Reynolds number calculation.

Feature	Formula	Normalisation
η_1	$\frac{1}{2}(\|\boldsymbol{\Omega}\|^2 - \|\boldsymbol{S}\|^2)$	$\|\boldsymbol{S}\|^2$
η_2	k	$\frac{1}{2}U_iU_i$
η_3	$\min\left(\frac{\sqrt{k}d}{50\nu}, 2\right)$	not applicable
η_4	$U_k\frac{\partial P}{\partial x_k}$	$\sqrt{\frac{\partial P}{\partial x_j}\frac{\partial P}{\partial x_j}U_iU_i}$
η_5	$\sqrt{\frac{\partial P}{\partial x_i}\frac{\partial P}{\partial x_i}}$	$\frac{1}{2}\rho\frac{\partial U_k^2}{\partial x_k}$
η_6	$\lvert U_iU_j\frac{\partial U_i}{\partial x_j}\rvert$	$\sqrt{U_lU_lU_i\frac{\partial U_i}{\partial x_j}U_k\frac{\partial U_k}{\partial x_j}}$
η_7	$\|\overline{u_i'u_j'}\|$	k
η_8	$\lvert\frac{D\Gamma}{Ds}\rvert$ where $\boldsymbol{\Gamma} \equiv \boldsymbol{U}/\lvert\boldsymbol{U}\rvert$, $Ds = \lvert\boldsymbol{U}\rvert Dt$	$\frac{1}{L_c}$
η_9	$U_i\frac{dU}{dx_i}$	$\lvert\overline{u_j'u_k'}S_{jk}\rvert$

between shear flows and boundary layers, hence important in RANS models [159]. The pressure gradient along streamline (η_4) is a measure of the alignment of the pressure gradient to the velocity. Positive values of η_4 indicate adverse pressure gradients which enhance turbulence, while negative values show favourable pressure gradients which diminish it [63]. η_5 is a measure of the dominance of pressure normal stresses to shear stresses, and vice versa. η_6 is a measure of the non-orthogonality between velocity and its gradient and was originally proposed by Gorlé *et al.* [56] to indicate flow deviation from 2D parallel shear flows. Most

baseline RANS models are calibrated to perform well for parallel shear flows (e.g. plane channel flow), however, they are known to perform poorly for flows that deviate from this, making η_5 an important feature [89, 159]. Finally, the streamline curvature (η_8) proposed by Wang *et al.* [159], is used because of its impact on the levels of turbulent shear stress, and thus its potential impact on the accuracy of turbulence models [21, 145].

The neural network output is constructed as a linear combination of inputs (the features for the first hidden layer, or the outputs from the preceding hidden layer), which are transformed through a nonlinear activation function. Thus, the output, $\boldsymbol{\beta}(\boldsymbol{\eta})$, can be expressed as follows:

$$\boldsymbol{\beta}(\boldsymbol{\eta}) = \boldsymbol{w}^{(l)}\sigma\left[\boldsymbol{w}^{(l-1)}\boldsymbol{\eta} + \boldsymbol{b}^{(l-1)}\right] + \boldsymbol{b}^{(l)}, \tag{6.5}$$

where l represents intermediate layers, σ is the activation function, $\boldsymbol{w}$ is the weights matrix, and $\boldsymbol{b}$ is the biases. The neural network training involves solving an optimisation problem (Eqn. 6.3) with the goal to estimate the values of the weights and biases. A number of activation functions are available. In this work we employed two commonly used functions: the rectified linear unit (ReLU) and the hyperbolic tangent (tanh) functions, and found the training algorithm to be insensitive to it. The particular loss function used to measure the training quality is the coefficient of determination, or simply R^2, function.

The two key hyper-parameters that the training was found to be particularly sensitive to were the number of hidden layers and the number of neurons per layer. Manual adjustments of these showed that many hidden layers and tens of neurons per layer are required to achieve reasonable accuracy when the neural network-augmented turbulence model is used for predictions. The chosen number of hidden layers was 10, with 100 neurons per layer. Fewer hidden layers and/or fewer neurons per layer resulted in poor NN predictions of the $\boldsymbol{\beta}$ field when tested

on the periodic hill test case in section 6.4. The training data (described in chapter 6.4) was spilt such that ninety percent was used to train the model, while ten percent is used for validation. Early-stopping was used to avoid over-fitting. The Adam optimiser is used, with the learning rate set to 10^{-3}. TensorFlow [1] was used for the neural network implementation, and coupled with OpenFOAM for CFD solutions, using a similar approach to Maulik et al. [102].

6.4 Training dataset and test cases

The neural network-augmented $k-\omega$ SST turbulence model is trained on the streamwise velocity field DNS data on a set of parameterised periodic hills, shown in Fig. 6.3. While the baseline geometry, i.e. $\alpha = 1.0$, is a canonical flow frequently used in data-driven turbulence modelling literature, Xiao *et al.* parameterised it by stretching/contracting the hill width in the streamwise direction as a function of α, where the total horizontal length L_x, normalised by the constant hill crest height, H, is given by [171]: $L_x/H = 3.858\alpha + 5.142$. The resulting geometries lead to flows with mild to massive separations. In this thesis, the baseline geometry ($\alpha = 1.0$ in Fig. 6.3) has previously been used as a test case in chapters 3.5.3, 4.4.3 and 5.3.2 to demonstrate the discrepancies in commonly used turbulence models, and the ability of field inversion to effectively reconstruct the turbulent mean flow from high-fidelity data.

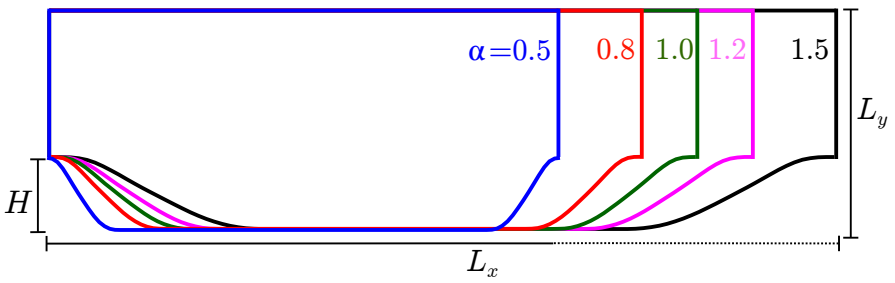


Figure 6.3: Parameterised periodic hill geometries.

Table 6.3: Geometries used for training and testing the neural network-augmented $k-\omega$ SST turbulence model.

Training cases	Periodic hills, $\alpha = \{0.5, 0.8, 1.2, 1.5\}$, $Re = 5,600$
Test cases	Periodic hill $\alpha = 1.0$, $Re = 5,600$
	Converging-diverging channel, $Re = 12,600$
	NASA-wall mounted hump, $Re = 936,000$

Table 6.3 outlines the four periodic hill geometries used for training, and the three cases for testing the neural network-augmented $k-\omega$ SST model. The test cases will not only investigate the interpolation capabilities of the neural network-augmented turbulence model (i.e. with the baseline periodic hill case), but also evaluate it for cases beyond the training flow configurations with very different geometries and Reynolds numbers.

6.5 Results

6.5.1 Periodic hills

This section compares the predictive capabilities of the neural network-augmented $k-\omega$ SST turbulence model, both *a priori*, i.e. $\alpha = \{0.5, 0.8, 1.2, 1.5\}$ where the field inversion data has been used for training the neural network model, and *a posteriori*, i.e. $\alpha = 1.0$ which was left out of the training dataset.

The error reduction (compared to the baseline $k-\omega$ SST model) are shown in Table 6.4. Field inversion is able to achieve significant error reductions in the velocity predictions. While the neural network predictions also reduce the errors in the baseline simulations to a large extent, these do not perform as well as the field inversion scenarios. Specifically, for all the shapes considered, the average root-mean-square error change ($\Delta\epsilon$) for field inversion and neural network predictions

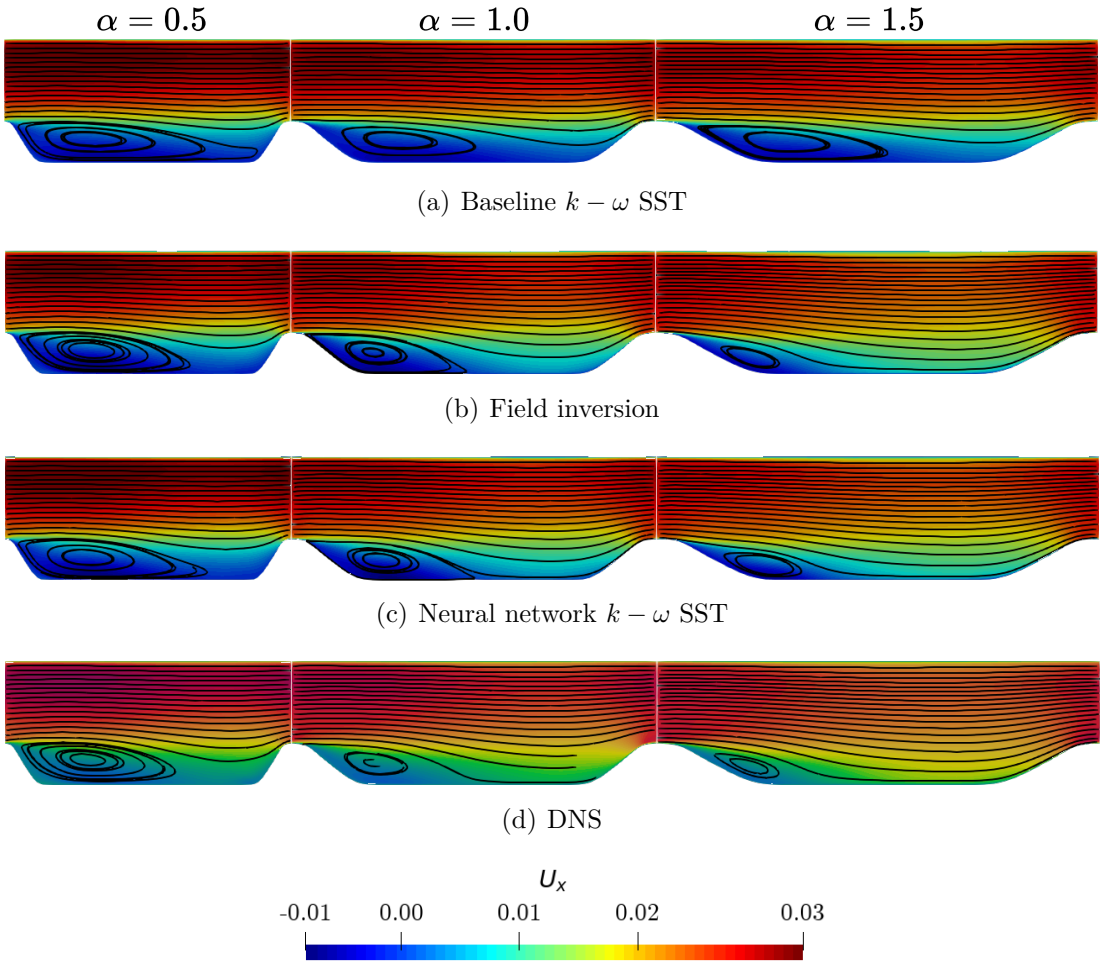
Table 6.4: Comparison of the root-mean-square error reduction ($\Delta\epsilon$) from field inversion (FI), and the neural network-augmented model (NN) for the periodic hill case. α, U_x, U_y, τ_{xx}, τ_{xy} represent the geometric parameter, the streamwise and wall-normal velocity components, and the normal and shear Reynolds stresses.

Case, α	Error reduction, $\Delta\epsilon = [1 - (\epsilon/\epsilon_{\text{baseline}})] \times 100$, %							
	$\Delta\epsilon(U_x)$		$\Delta\epsilon(U_y)$		$\Delta\epsilon(\tau_{xx})$		$\Delta\epsilon(\tau_{xy})$	
	FI	NN	FI	NN	FI	NN	FI	NN
0.5	58.2	17.0	53.2	30.0	22.3	10.5	28.4	9.9
0.8	69.5	47.5	71.5	51.7	37.0	20.7	36.3	26.1
1.2	62.4	60.4	62.5	53.5	22.9	16.5	27.3	23.5
1.5	82.0	65.3	83.2	63.0	36.0	19.6	34.6	24.2
1.0	71.9	66.7	73.8	60.1	40.0	24.7	34.9	28.5

compares as follows:

- streamwise velocity: field inversion: 68.8%; neural network 51.5%,
- wall-normal velocity: field inversion: 68.8%; neural network 51.7%,
- Reynolds stress (normal): field inversion: 31.6%; neural network 18.4%,
- Reynolds stress (shear): field inversion: 32.3%; neural network 22.4%.

The streamwise velocity predictions, overlayed by the streamlines, for the baseline ($\alpha = 1.0$), steepest ($\alpha = 0.5$), and least-steep ($\alpha = 1.5$) periodic hill geometries are shown in Fig. 6.4. The streamlines show that the steeper the hill, the larger the size of the separation bubble. As previously shown for the baseline geometry, the baseline $k - \omega$ SST model tends to massively over-predict the size of the separation bubble. These are significantly improved by both field inversion, and the neural-network augmented $k - \omega$ SST model.



(a) Baseline $k - \omega$ SST

(b) Field inversion

(c) Neural network $k - \omega$ SST

(d) DNS

Figure 6.4: Comparison of the streamwise velocity contours using the different RANS-based models against the DNS data for the periodic hill cases.

The skin friction comparison for the baseline shape ($\alpha = 1.0$) is shown in Fig. 6.5. The baseline $k-\omega$ SST model fails to capture to flow reattachment, as shown in previous chapters. Both the field inversion and the neural network-augmented model predictions are nearly identical, and are able to largely match the DNS data from Krank *et al.* [76].

Fig. 6.6 shows the corrective field, $\boldsymbol{\beta}$, contour for the baseline shape ($\alpha = 1.0$) from field inversion and the neural network prediction. As previously noted, the field inversion results was not included in the neural network training dataset.

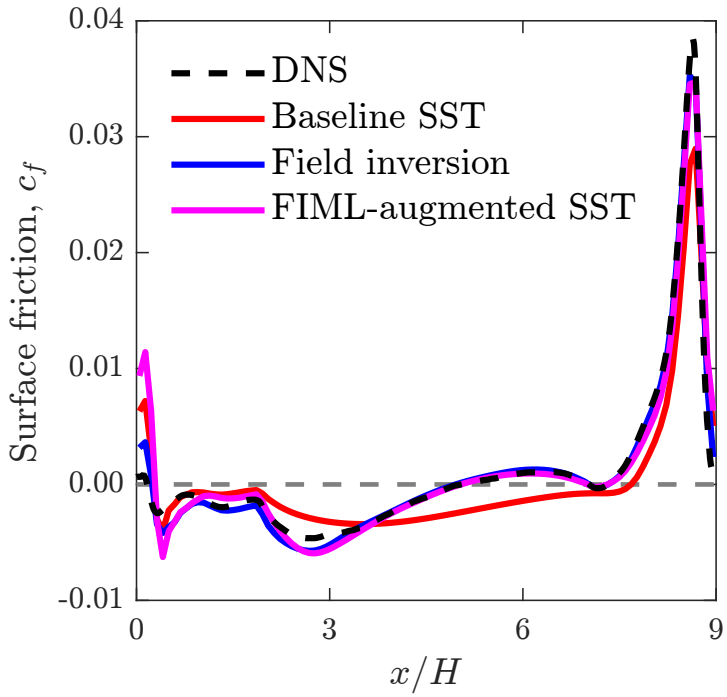


Figure 6.5: Comparison of the skin friction for the baseline periodic hill.

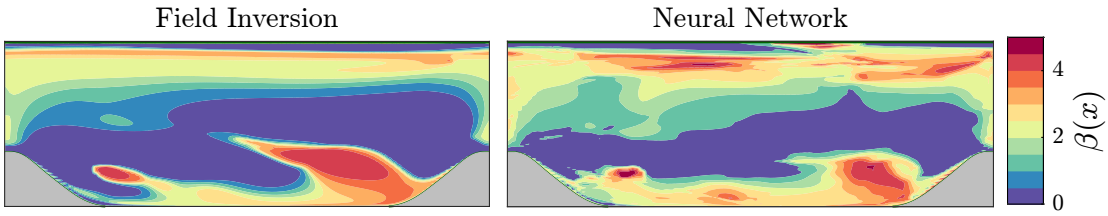
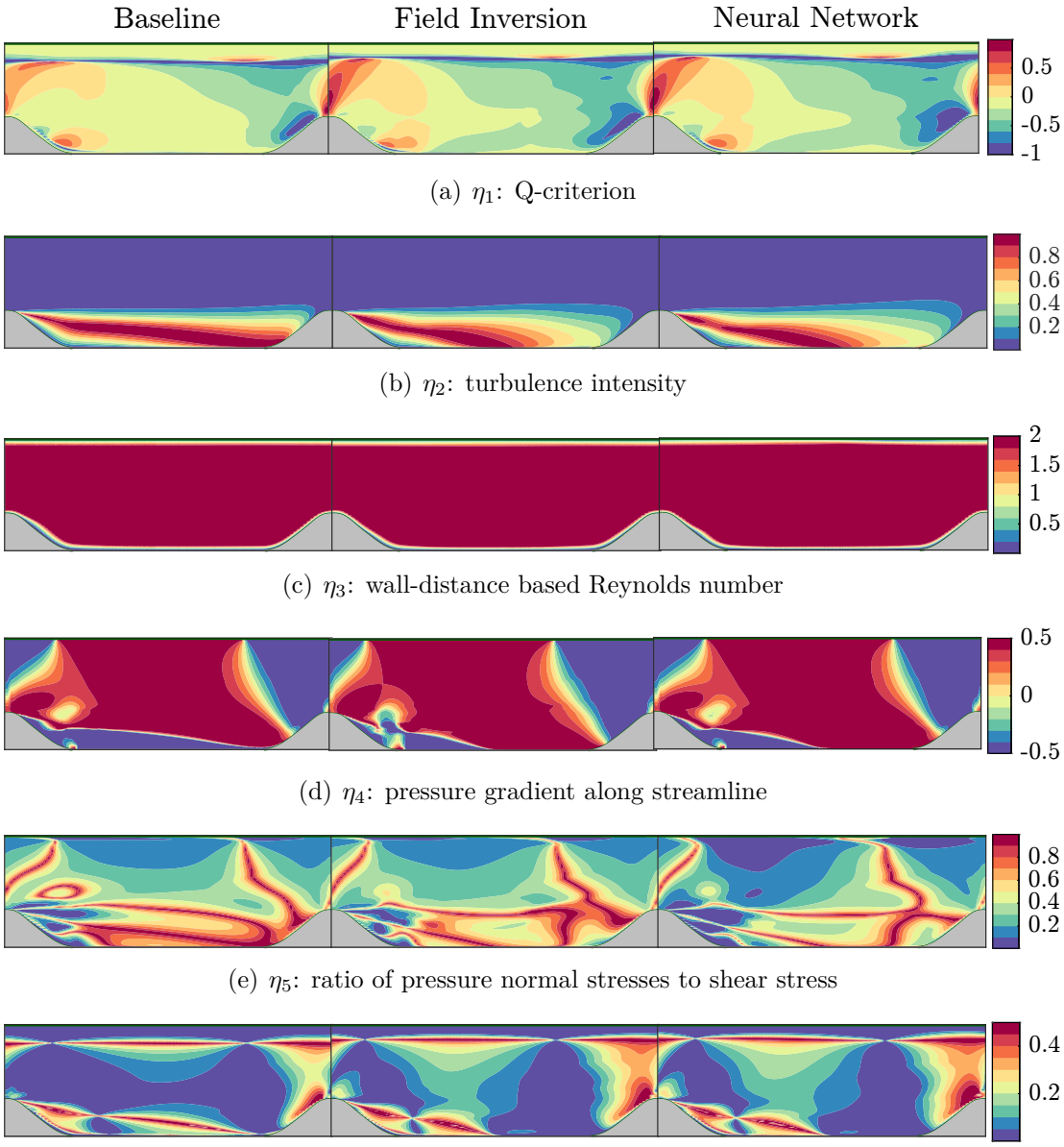


Figure 6.6: The corrective field $\boldsymbol{\beta}$ from field inversion vs. the neural-network prediction for the baseline periodic hill geometry, $\alpha = 1.0$.

The neural network $\boldsymbol{\beta}$ predictions is qualitatively similar to the field inversion case for the majority of the flow domain, however, it is not as smooth. This could be as a result of sub-optimal neural network training.

On the other hand, the flow features ($\boldsymbol{\eta}$) are less sensitive to the $\boldsymbol{\beta}$ field differences discussed above, as illustrated in Figs. 6.7 - 6.8. Comparisons of the input features also illustrate the impact of the corrective field in modifying the turbulence model, and the significant changes required to match the flow predictions



(a) η_1: Q-criterion

(b) η_2: turbulence intensity

(c) η_3: wall-distance based Reynolds number

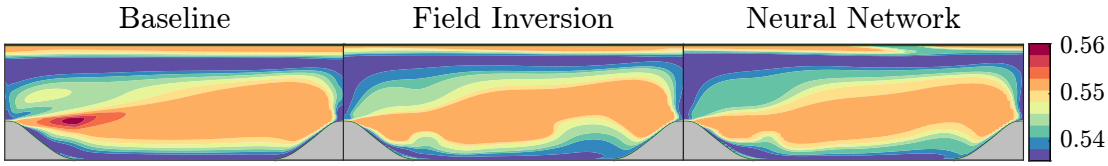
(d) η_4: pressure gradient along streamline

(e) η_5: ratio of pressure normal stresses to shear stress

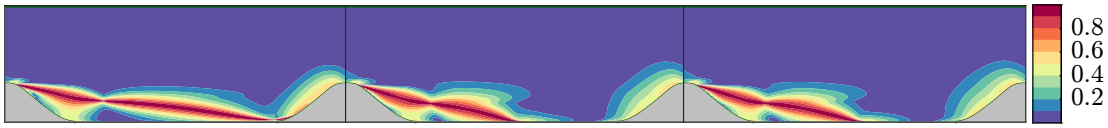
(f) η_6: non-orthogonality between velocity and its gradient

Figure 6.7: $\eta_1 - \eta_6$: comparison of the features in Table 6.1 for the baseline periodic hill case, $\alpha = 1.0$. Left to right columnwise: baseline, field inversion, and neural network (FIML) predictions.

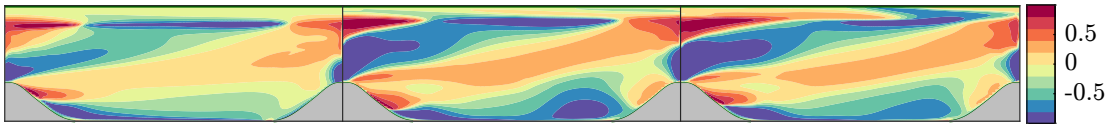
to the high-fidelity data. In addition, it is notable that despite difference in field inversion and neural network predicted $\boldsymbol{\beta}$ fields, the input features for the two scenarios are largely similar. The neural network model struggles to achieve the



(a) η_7: Ratio of total to normal Reynolds stress



(b) η_8: Streamline curvature



(c) η_9: ratio of convection to production of TKE

Figure 6.8: $\eta_7 - \eta_9$: comparison of the features in Table 6.1 for the baseline periodic hill case, $\alpha = 1.0$. Left to right columnwise: baseline, field inversion, and neural network (FIML) predictions.

same correction in the pressure gradient along streamline (η_4) feature, near the first hill crest (Fig. 6.7(d)). As previously mentioned, this may be due to poor training of the neural network hyper-parameters and ineffective regularisation. More broadly, the inherent decoupled nature of the FIML approach (i.e. field inversion and machine learning training being separate steps) means there are no guarantees that the field inversion corrections are fully learnable.

6.5.2 Converging-diverging channel

This case is used to further assess the extrapolation capabilities of the neural network-augmented $k - \omega$ SST model. It is not only significantly different in terms of the geometry from the cases used for neural network training, but is at a higher Reynolds number and has a milder flow separation, as discussed in chapter 3.5.2, compared to the training dataset, making it a challenging case for the augmented turbulence model.

Fig. 6.9 compares the streamwise velocity profiles for the baseline and augmented $k-\omega$ SST model, against the available DNS data. The augmented model significantly improves the flow predictions beyond $x/H \approx 7.5$, where the flow is re-attached based on the DNS data, whereas the baseline SST model predicts a large separation. Similarly, Fig. 6.10 shows that the pressure predictions by the NN-augmented SST model is much closer to the DNS results beyond the separated shear layer. Overall, the neural network-augmented $k-\omega$ SST model is effective in extrapolating to this case.

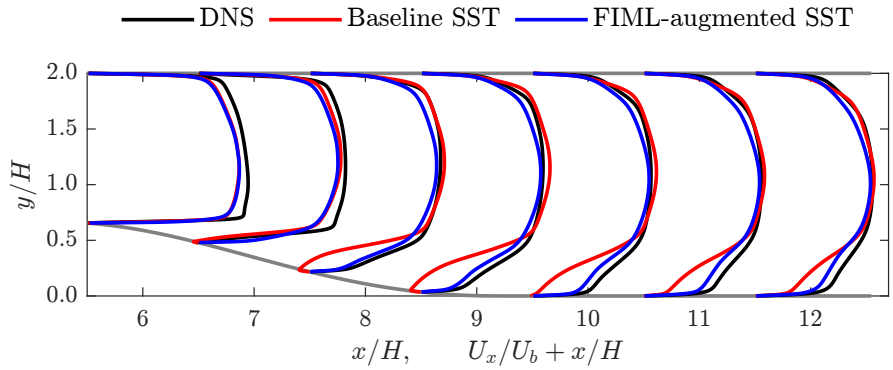


Figure 6.9: Comparison of the streamwise velocity profiles with the neural network-augmented turbulence model for the CDC case.

6.5.3 NASA wall-mounted hump

The final extrapolation test is the NASA wall-mounted hump, which again has a significantly different geometry, and a much higher Reynolds number compared to the neural network training dataset.

The streamwise velocity profiles from around the hump apex (locations shown in Fig. 5.1) are shown in Fig. 6.11. Again, the baseline $k-\omega$ SST model over-predicts the flow separation. The neural-network augmented turbulence model struggles to match the LES reference data when the flow reverses, in the range $0.65 \lesssim x/c \lesssim 0.8$, especially very close to the lower wall, however beyond this

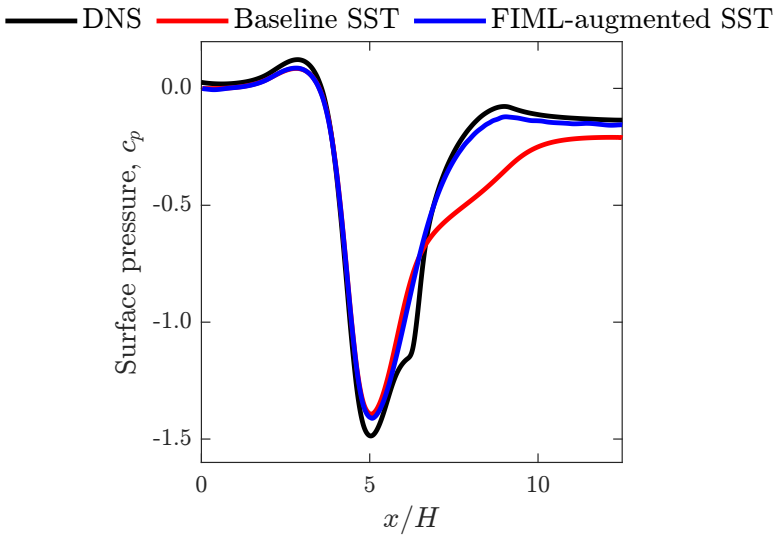


Figure 6.10: Surface pressure comparison with the neural network-augmented turbulence model for the CDC case.

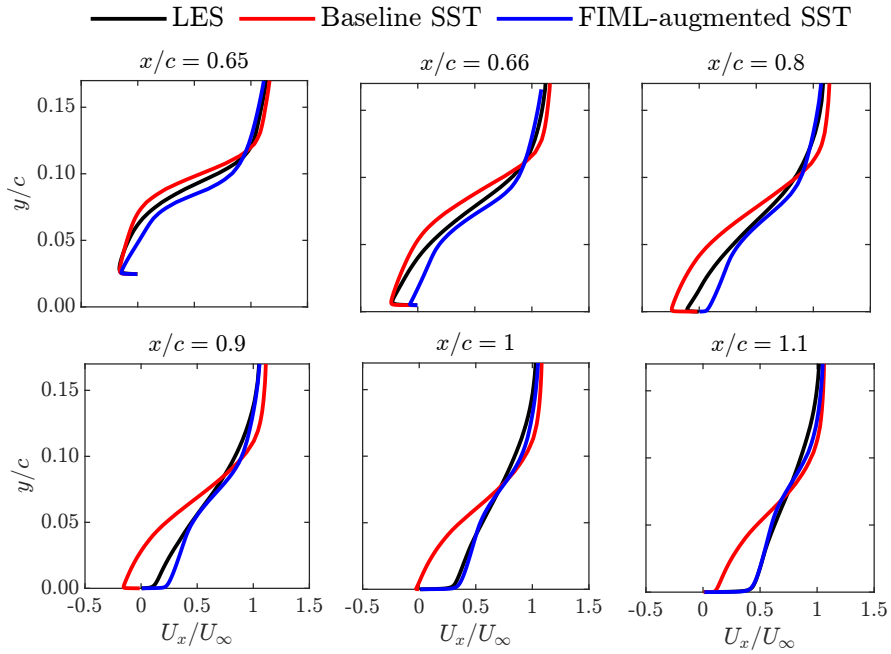


Figure 6.11: Comparison of the streamwise velocity profiles for the NASA wall-mounted hump case.

range the augmented SST matches the reference LES data fairly well. It under-predicts the size of the separation bubble, which it "learned" on the training

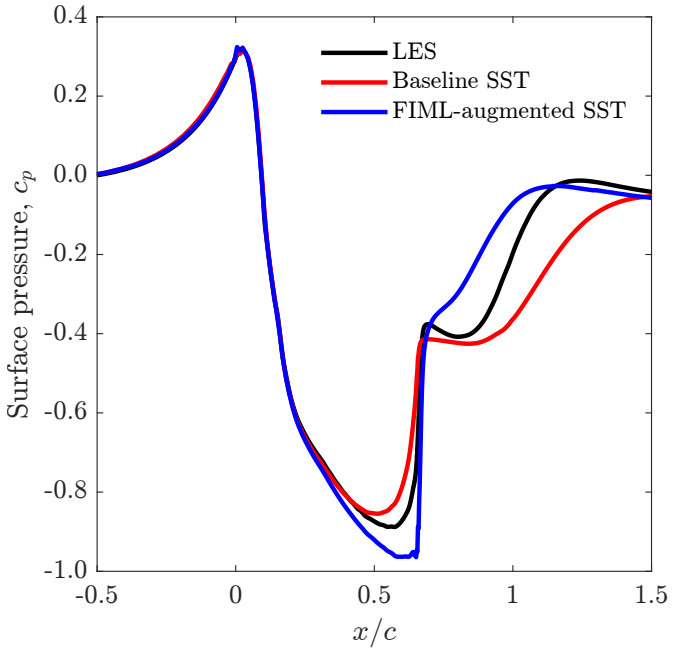


Figure 6.12: Comparison of the surface pressure predictions for the NASA wall-mounted hump case.

dataset which had much larger separation bubbles (in some periodic hill geometries this is several times the size of the characteristic length). However, since the separation is milder for this case, the NN-augmented model is over-correcting the baseline model. This is not unexpected, due to the significantly different geometry, Reynolds number, and hence flow characteristics of the hump case compared to the parameterised periodic hills used for training.

Finally, the surface pressure predictions on the hump wall are shown in Fig. 6.12. The NN-augmented turbulence model performs poorly overall, by over-correcting the baseline $k-\omega$ SST model in the separated shear layer. Overall, the extrapolative capabilities of the neural network-augmented SST model for this case are limited. This can be both a result of poor neural network training, over-fitting to the training data and a lack of generalisation. Neural network training issues may be partially remedied by performing a more systematic op-

timisation of the network structure and hyper-parameters. Over-fitting may be avoided by using early stopping more effectively and adding dropout layers in the neural network training. It might also be possible to reduce the network size by performing a feature importance analysis and considering entirely different set of features. In addition, generalisation can be also improved by training on a more varied dataset. However, there might be an inherent limitation within the original FIML framework, where the field inversion and neural network training steps are decoupled. As discussed in chapter 2.3.1, there are not guarantees that the field inversion corrections are learnable when the neural network is trained on static data. This may be improved by employing the FIML-direct approach proposed by Holland *et al* [65]. Finally as previously argued, and also shown in literature such as Refs. [132, 147], data-driven augmented turbulence models that are universally applicable are very difficult to achieve. Instead, it is more pragmatic to develop customised models for specific flows, which will be demonstrated in the following chapter where the augmented model developed here is applied for aerodynamic shape optimisation of a periodic hill flow, a case for which the augmented model performs very well as we have demonstrated in this chapter.

6.6 Summary

In this chapter a RANS-based baseline turbulence model—$k-\omega$ shear stress transport (SST)—was augmented using the field inversion and machine learning (FIML) approach. The two-step process involved flow reconstruction through inverse modelling (an adjoint-based optimisation where the goal is to minimise the errors between some high-fidelity data and the baseline RANS model by applying a correction field, $\boldsymbol{\beta}$, to the production term of the turbulent transport equation), and the generalisation of the $\boldsymbol{\beta}$ field using neural networks, used for predictions beyond the training dataset. Key findings are summarised as follows:

- We augmented the baseline $k-\omega$ SST model using the FIML approach. The augmented model was trained on DNS streamwise velocity fields for a set of parameterised periodic hills involving mild to massive flow separations.

- *A priori* test of the NN-augmented SST model on the periodic hill case showed that the baseline SST model was able to significantly reduce the errors in most of the quantities of interest, and was shown to match reference DNS data well.

- *A posteriori* tests on flow cases with significantly different geometries and Reynolds numbers showed that the NN-augmented model can improve the baseline predictions (significantly in for the converging-diverging channel flow, and to some degree for the NASA wall-mounted hump case). However, there is a risk of over-fitting, as seen in the deficiencies in the surface pressure predictions for the hump case.

CHAPTER 7

Aerodynamic shape optimisation with neural network-augmented turbulence modelling

7.1 Introduction

This chapter addresses the problem of improving aerodynamic shape optimisation using the field inversion and machine learning framework (FIML). There is a research gap in leveraging the potential benefits of the FIML in aerodynamic shape optimisation using high-fidelity data – this is an important opportunity because ideally the FIML approach should lead to improved turbulence models, which in turn should enable improved aerodynamic shape designs. This chapter provides the novel contribution of using the FIML approach to improve aerodynamic shape optimisation, bringing together the contributions from the previous chapters in order to solve an important problem in aerospace design optimisation. The work has been published in Refs. [17, 14].

Aerodynamic shape optimisation based on computational fluid dynamics (CFD) analyses is a vibrant research field and is expected to play a significant role in industry [141, 100], for applications such as modern aerospace vehicles design, more efficient renewable energy solutions (e.g. wind turbines), etc. Turbulence is a ubiquitous flow phenomenon in these applications, which can be numerically simulated using techniques with various levels of fidelity. Low fidelity ap-

proaches such as potential-flow or Euler equations ignore viscous effects, while turbulence can be partially or fully resolved in large eddy simulations (LES) and direct numerical simulations (DNS), respectively. As aerodynamic shape optimisation is an iterative process, that requires many function evaluations, LES and DNS are computationally prohibitive. Alternatively, simulations based on the Reynolds-averaged Navier-Stokes (RANS) equations are considered adequate for CFD-based on-design aerodynamic shape optimisation due to their *relative* simplicity, low computational cost, and robustness. However, RANS simulations have well-documented deficiencies for complex flows (e.g. [148]), such as those involving separation, which means the models are inadequate in these scenarios, especially for off-design conditions. In this work, we explore aerodynamic shape optimisation in the presence of a massive flow separation, simulated by a machine learning-augmented RANS-based turbulence model.

Like many areas of science the use of machine learning techniques in aerodynamic shape optimisation has received noticeable attention. Main application areas as identified by Li *et al.* in the recent review are [85]: geometric parameterisation of the design space to exclude aerodynamic shapes which are deemed abnormal, thus allowing the use of fewer design variables; predictions of aerodynamic coefficients leveraging machine-learning-based predictive simulations; and new optimisation architectures, such as replacing CFD-based optimisation with surrogate-based optimisation. However, the use of machine learning techniques in aerodynamic shape optimisation for more accurate turbulence flow predictions in the RANS setting is less well-explored.

Investigation of aerodynamic shape optimisation sensitivity to the choice of turbulence model has shown that while it is less sensitive in on-design conditions with relatively simple flows (e.g. small or no separation, no shock boundary layer interactions, etc.), it can be sensitive in off-design conditions in deterministic

optimisation setups (i.e. when uncertainty due to turbulence modelling is unaccounted) [157]. Cook *et al.* investigated performing shape optimisation while accounting for structural uncertainties in turbulence closure modelling through the eigenspace perturbation approach [33]. They concluded that performing design optimisation under turbulence model uncertainty can lead to more robust solutions, and avoids sub-optimal designs where turbulence model sensitivities are unaccounted.

Inspired by the FIML approach, Fidkowski recently investigated aerodynamic shape optimisation for unsteady turbulent flows [45]. The training data was generated using 2D (unsteady) detached eddy simulations and used to correct steady RANS-based flow simulations. The design objective and constraints were time-averaged quantities. The results achieved demonstrated that accounting for unsteady flow behaviour leads to improved designs, compared to steady flow simulations. In this work we augment an existing steady turbulence model ($k-\omega$ SST) trained on time-averaged high-fidelity simulation data from DNS.

7.2 Optimisation framework

An overview of the proposed methodology for aerodynamic shape optimisation is provided in Fig. 7.1, involving three steps:

1. *Baseline geometry*: we choose the periodic hill flow as a proof-of-concept case, also used in the previous chapters. As previously stated, this case is chosen due to the availability of the high-fidelity DNS data for the set of parameterised periodic hills (Fig. 6.3), making it an appropriate choice to investigate aerodynamic shape optimisation with neural network-augmented turbulence modelling.

2. *RANS-based aerodynamic shape optimisation*: we perform aerodynamic shape

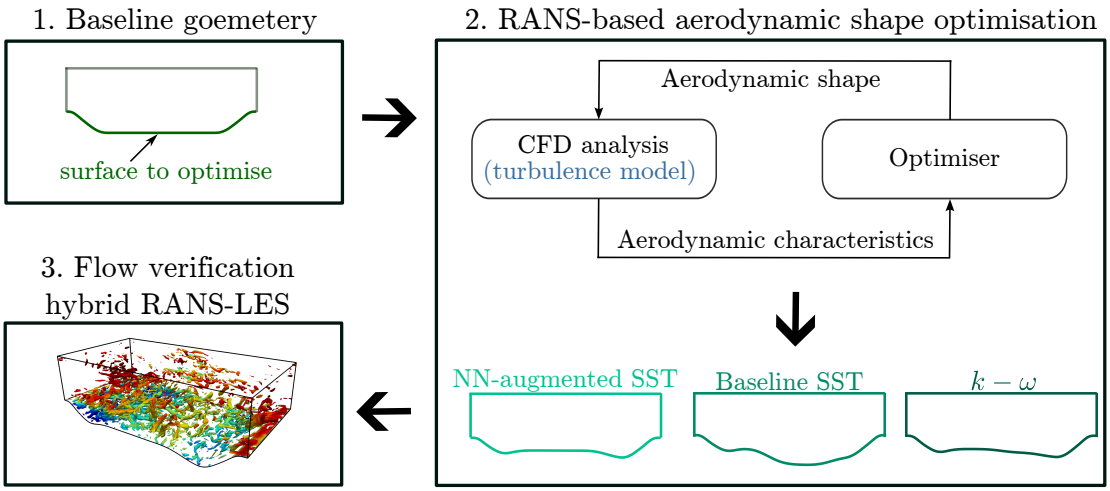


Figure 7.1: Overview of the aerodynamic shape optimisation of the periodic hill case.

optimisation using the neural network-augmented $k-\omega$ SST model, developed and tested in chapter 6). In order to investigate the sensitivity of the aerodynamic shape optimisation to the turbulence model, we will also employ two existing turbulence models: the baseline $k-\omega$ SST, and the Wilcox $k-\omega$. The shape optimisation problem will be formulated in chapter 7.2.1, with the implementation details in chapter 7.2.2.

3. *Flow verification*: Once the aerodynamic shape optimisation is performed, it is important to validate the predictive capabilities of the different RANS-based turbulence models using a higher-fidelity method. In this work, we use the improved delayed-detached eddy simulation (IDDES) hybrid RANS-LES approach, which will be detailed and validated in chapter 7.3.

7.2.1 Problem formulation

We solve an aerodynamic shape optimisation problem where the goal is to optimise the lower wall for minimum drag. The objective function formulation for this particular case is more of a mathematical construct than an application of

aerodynamic engineering design optimisation. We stress that the goal is to assess the proposed framework in the presence of complex flow structures which are encountered in engineering applications, making this proof-of-concept case a suitable one, nonetheless.

The objective functions and constraints are defined as follows:

$$\min_{\Delta y} \quad C_{d,\text{lower wall}} = \frac{1}{\frac{1}{2}\rho U_b^2 S}\left(\iint_S \tau_w(\hat{t}\cdot\hat{i})\mathrm{d}s + \iint_S p(\hat{n}\cdot\hat{j})\mathrm{d}s\right),$$

subject to:

$$\begin{aligned} &-0.4 \leq \Delta y/H \leq 0.4, \\ &\nabla p = \nabla p_{\text{prescribed}}, \\ &T.B. = \sum_{i=1}^{N_{cells}} \phi V_i \nu_{t,i} \geq T.B._{\text{prescribed}}, \quad \phi = 10^3, \end{aligned} \tag{7.1}$$

where Δy (representing the manipulation of the lower wall in the y-direction) is the design variable, and C_d is the drag coefficient. The unit vectors $\hat{i}$, $\hat{j}$, $\hat{t}$ and $\hat{n}$ represent the directions parallel and perpendicular to the bulk velocity, U_b. τ_w, p, ρ, and S are the wall shear stress, wall pressure, density, and surface area. The constraints are the upper and lower bounds for the possible y-coordinates values, the prescribed pressure gradient to achieve a constant bulk flow, and the sum of turbulent viscosity over the entire flow field. The last constraint, $T.B.$, is used to penalise the optimiser tendency to essentially turn the lower wall into a flat wall to reduce the drag by eliminating the flow separation. This is to ensure that a certain turbulence level continues to characterise the flow. The constant $\phi = 10^3$ is chosen to scale the $T.B.$ constraint to have a value close to unity.

7.2.2 MACH-Aero

The aerodynamic shape optimisation problem is solved using the open-source MACH-Aero framework*. The framework relies on a gradient-based approach to perform design optimisation, which is computationally cheaper—in terms of function evaluations required—compared to gradient-free approaches, especially for cases of industrial interest, e.g. airfoil [123], or wing [96] optimisation. The approach is illustrated in Fig. 7.2 using the extended design structure matrix (XDSM) [78].

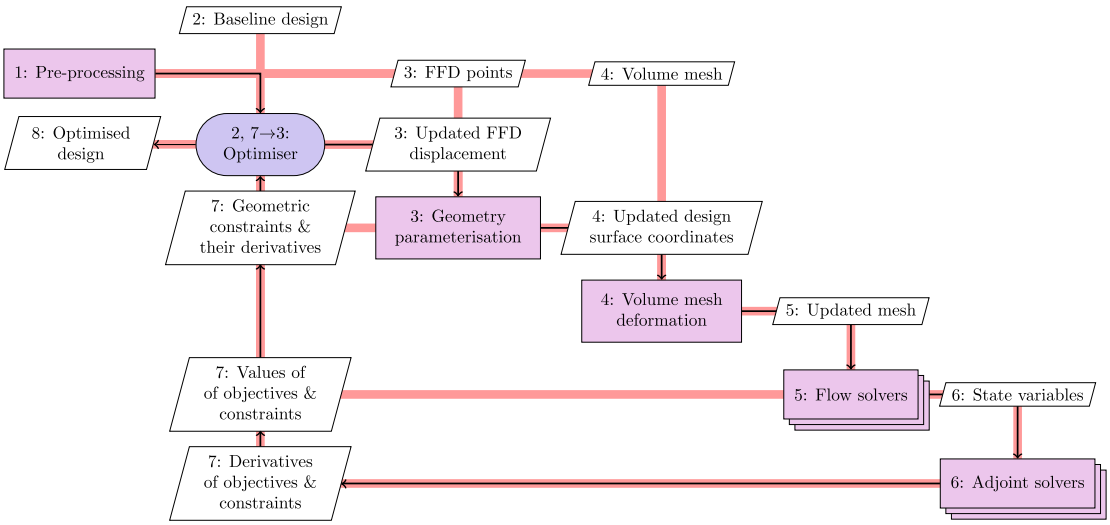
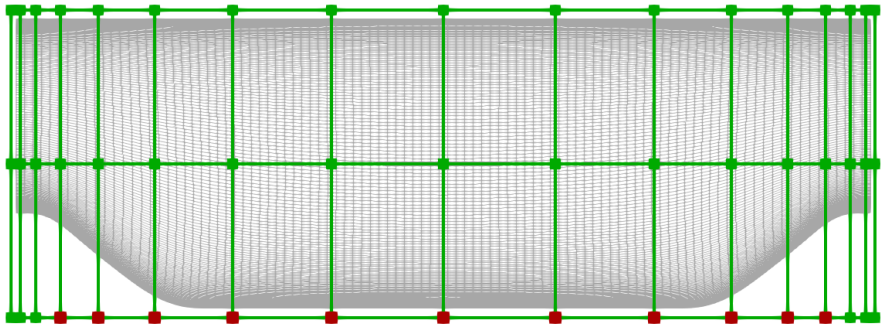


Figure 7.2: Extended design structure matrix (XDSM) diagram [78]. Diagonal rectangles represent processes/modules (software), while off-diagonal parallelograms represent data input and output. Thick lines illustrate data flow between modules, while the black lines show the process flow for the adjoint solver (for derivative computation). Rounded rectangle(s) denote an iterative process, also called the 'driver'.

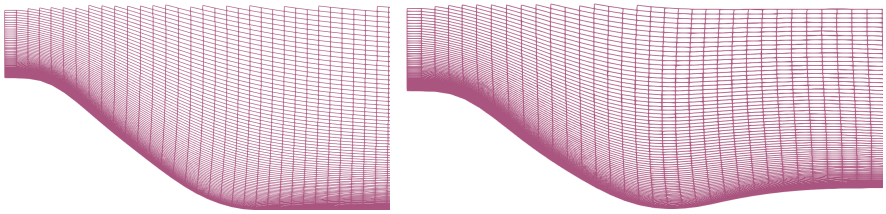
The shape optimisation procedure as depicted in numerical order in Fig. 7.2 is as follows:

1. Generate a volume mesh for the geometry, and an initial free-form deformation (FFD) box with volumetric control points, Fig. 7.3. The FFD technique

*https://github.com/mdolab/MACH-Aero



(a) Baseline mesh, overlayed by the FFD points.



(b) Close-up of mesh, left: baseline, right: deformed during optimisation.

Figure 7.3: The periodic hill mesh and the FFD points.

is used for morphing the design surface, and is described in step 3.

2. Provide a set of baseline design variables to the optimiser. The optimisation problem is formulated using pyOptSparse [168]. The optimiser updates the design variables and outputs them to the geometry parameterisation module. The optimiser used in the current work is based on the sequential least squares programming (SLSQP) algorithm, which was found to be sufficient for the number of design variables involved. The two key parameters for the SLSQP optimiser in pyOptSparse is the termination criterion (convergence accuracy, set to 10^{-12}) and the maximum number of optimisation iterations (set to 100) [†]. Fig. 7.4 shows the optimisation histories for the optimisation results in section 7.4.

3. Deform the design surface based on the updated design variables and the

[†] https://tinyurl.com/594sykfw

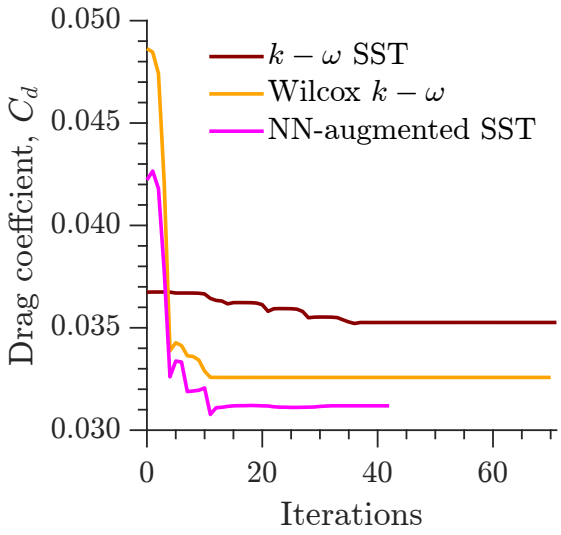


Figure 7.4: Objective function history with the different turbulence models during the optimisations. The maximum number of iterations is set to 100, and the convergence accuracy is set to 1×10^{-12}.

FFD points. The geometry parameterisation uses the free-form deformation technique whereby the design surfaces are embedded in an FFD volume which maps the surface coordinates to the FFD control points using a set of B-splines curves. The design surface can than be deformed by moving the FFD points.

4. Update the volume mesh based on the deformed design surface(s). An example of the original and deformed mesh is shown in Fig. 7.3 (b).

5. Run RANS-based flow simulations on the updated geometry/mesh. As outlined in chapter 3.4, in this work we use DAFoam for flow predictions.

6. Run the adjoint computations, using the discrete adjoint approach, detailed in chapter 3.3. All the turbulence models, including the NN-augmented $k-\omega$ SST, are differentiated using the reverse-mode algorithmic differentiation approach utilising CoDiPack [97].

7. Compute the derivatives of the objective and constraint functions with respect to the design variables, and supply these to the optimiser.

8. Repeat steps 2-7, until the objective function satisfies a predefined metric.

7.3 Hybrid RANS-LES verification framework

Hybrid methods utilise a closure model in the near-wall region of the flow, while performing eddy-resolving simulation away from the wall. The predictions with these methods can be significantly more accurate than RANS models, while requiring a coarser mesh than required for wall-resolved LES, or DNS [51, 28]. In this work we use the Spalart-Allmaras IDDES model. The surrogate eddy viscosity $\tilde{\nu}$ is defined as [136]:

$$\frac{\partial \tilde{\nu}}{\partial t} + U_i \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}\tilde{S}\tilde{\nu} + \frac{1}{\sigma}\left[\nabla \cdot (\tilde{\nu}\nabla\tilde{\nu}) + c_{b2}\left(\nabla\tilde{\nu}\right)^2\right] - c_{w1} f_w \left(\tilde{r}\left(\frac{\tilde{\nu}}{l_{IDDES}}\right)^2\right), \quad (7.2)$$

where the turbulent eddy viscosity is defined as $\nu_t = f_{v1}\tilde{\nu}$. The functions f_{v1} and f_w are for near-wall corrections, $\tilde{S}$ is the strain rate tensor, the non-dimensional term $\tilde{r}$ is defined as $\nu_t/(\tilde{S}\kappa^2 d_w^2)$, where κ is the von Kármán constant, d_w is the distance from the wall, and $\{\sigma, c_{b1}, c_{b2}, c_{w1}\}$ are model constants. The modified length scale l_{IDDES} is used to switch the transition from the unsteady RANS to scale-resolving LES, and along with the intermediate variables and functions, which are defined as:

$$l_{IDDES} = \tilde{f}_d\left(1 + f_e\right) d_w + \left(1 - \tilde{f}_d\right) l_{LES}, \quad (7.3)$$

$$l_{LES} = C_{DES}\psi\Delta, \quad C_{DES} = 0.65, \quad (7.4)$$

$$\Delta = \min(\max([C_w d_w, C_w h_{max}, h_{wn}])), \quad (7.5)$$

$$\tilde{f}_d = \max(1 - f_d, f_B), \quad (7.6)$$

$$f_d = 1 - \tanh\left[8\left(r_d^3\right)\right], \quad (7.7)$$

$$f_B = \min\left[2\exp\left(-9\alpha^2\right), 1.0\right], \tag{7.8}$$

$$\alpha = 0.25 - \frac{d_w}{h_{max}}, \tag{7.9}$$

$$f_e = \max\left[(f_{e1} - 1), 0\right] \psi f_{e2}. \tag{7.10}$$

For detailed discussions refer to Shur *et al.* [136].

7.3.1 Simulation setup

IDDES simulations are performed on all the periodic hill geometries. The flow is driven by a constant pressure gradient which is established by adding a source term in the momentum equation. The Reynolds number is $Re_H = 5,600$ for all the parameterised geometries. Cyclic boundary conditions are applied at the inlet, outlet, and the spanwise directions, while no-slip is imposed at the hills and top wall. Two different mesh resolutions are used. For $\alpha = \{0.5, 0.8, 1.0\}$ the number of cells is $n_x \times n_y \times n_z = 200 \times 160 \times 80 = 2.56 \times 10^6$, while for $\alpha = \{1.2, 1.5\}$, the number of cells is $n_x \times n_y \times n_z = 400 \times 220 \times 80 = 7.04 \times 10^6$. The mesh are deliberately dense to reduce mesh related inaccuracies. The geometry in the spanwise direction is extended by half the streamwise length of the baseline geometry, i.e. $L_z = 4.5H$ following the DNS dataset. The time-step used is $\Delta t = 0.0004$, and the data is collected over a time period of $T = 350H/U_b$.

The simulations are run in OpenFOAM [69], based on the finite-volume method. The convective term in the momentum equation is discretised using the hybrid scheme of Travin *et al.* [153] blending a second-order central difference scheme for the LES regions, and a second-order upwind scheme for the RANS regions. All other terms were discretised using a second-order limited linear differencing scheme. The PIMPLE algorithm is used for the pressure-velocity coupling, which combines the Pressure-Implicit with Splitting of Operators (PISO) and the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithms [129].

7.3.2 IDDES predictions vs. DNS

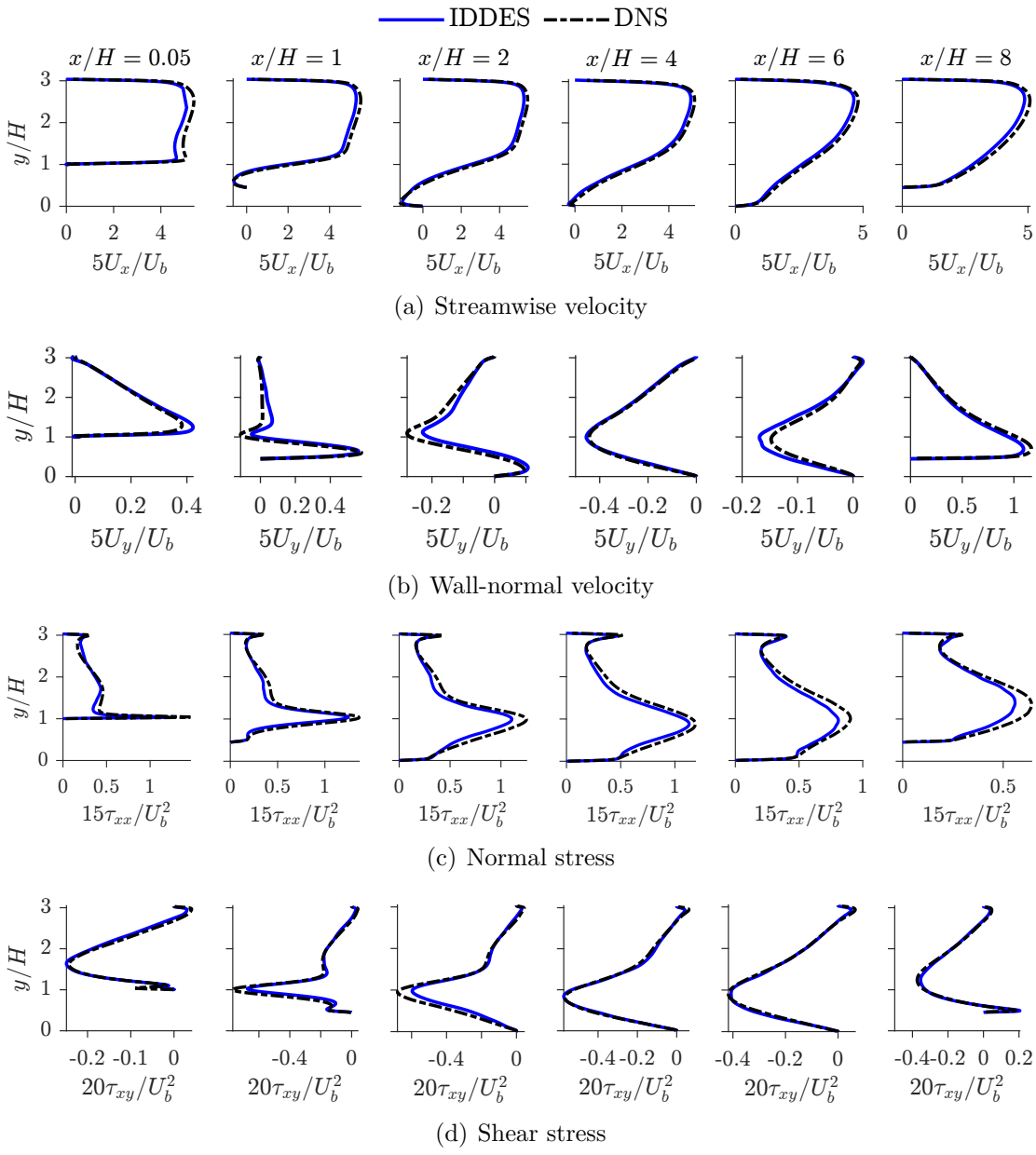
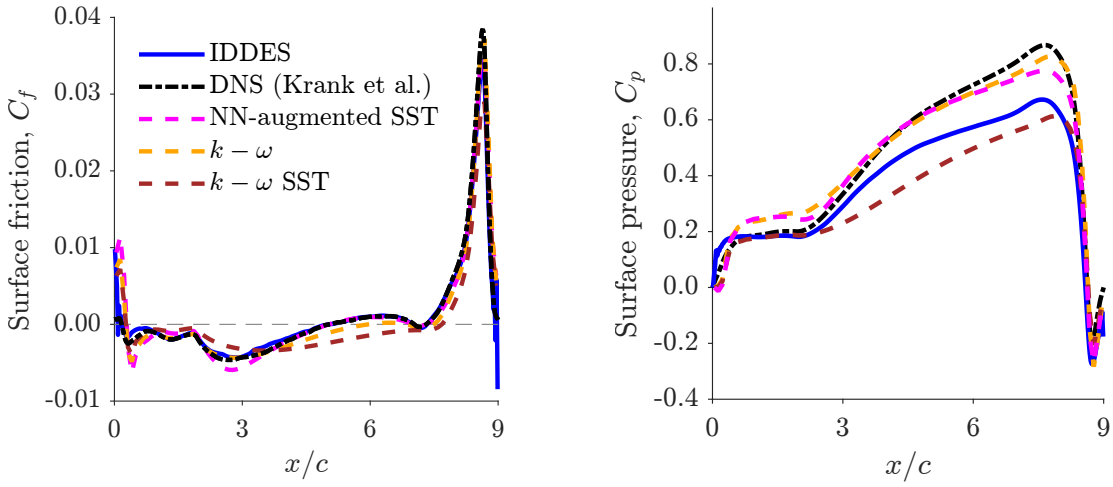


Figure 7.5: IDDES vs. DNS vs. RANS predictions of velocity and Reynolds stresses for the baseline shape, $\alpha = 1.0$.

In this section we will verify the IDDES predictions against the DNS data from Xiao *et al.* [171]. For brevity, the results will primarily focus on the baseline

periodic hill geometry ($\alpha = 1.0$).

Profiles of streamwise and wall-normal velocity components, and the normal and shear Reynolds stresses are plotted in Fig. 7.5. The IDDES predictions for the baseline geometry show excellent agreement with the DNS data.



(a) Skin friction

(b) Surface pressure

Figure 7.6: Comparison of the skin friction and surface pressure for the baseline geometry, $\alpha = 1.0$.

The surface quantities on the hill wall is plotted in Fig. 7.6. In addition, predictions from the Wilcox $k-\omega$, $k-\omega$ SST, and the neural network-augmented $k-\omega$ SST are shown for comparison. The wall quantities are not available from the DNS dataset by Xiao *et al.* [171], and are instead a comparison to the DNS data from Krank *et al.* [76] for the same geometry and flow condition. The skin friction predictions from IDDES are in excellent agreement with the DNS reference data. However, for the surface pressure, there is a considerable discrepancy between the IDDES prediction and the DNS data for $x/H \geq 3$. It is unclear what causes this discrepancy. However, the pressure predictions by the neural-network augmented SST and the Wilcox $k-\omega$ models are in good agreement to the reference DNS data. Fig. 7.7 compares the drag coefficient prediction on the hill wall for the different

geometries, with the different models. It is clear that the drag rises as the hill crests become steeper, causing larger separation. The IDDES and baseline SST model have comparable drag predictions. However, it is important to note that the pressure drag is the dominant component (compared to the viscous drag) in the total drag, and as previously discussed, the IDDES and baseline SST pressure predictions are least accurate compared to the reference DNS data (for the baseline shape, $\alpha = 1.0$).

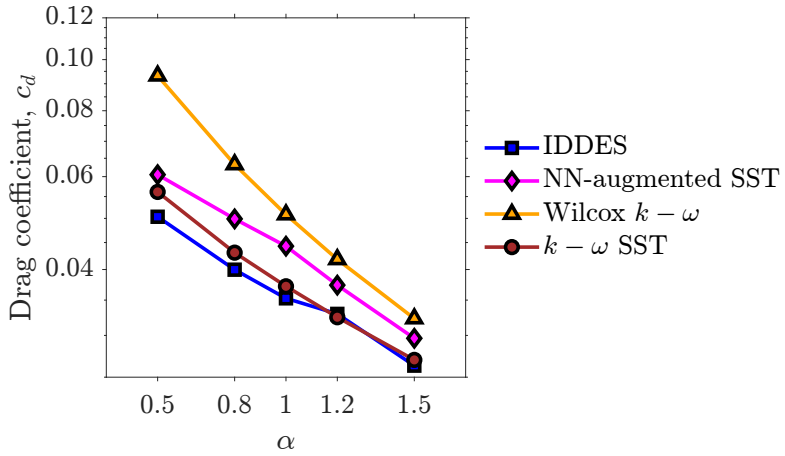


Figure 7.7: Comparison of the drag coefficient predictions for the different periodic hill geometries.

Finally, comparisons of the root-mean-squared errors for the IDDES and various RANS-based simulations against the DNS dataset for all the geometries are shown in Fig. 7.8. For the four quantities considered (streamwise and wall normal velocity components, and the normal and shear stresses), the IDDES results show considerably lower errors compared to the baseline $k-\omega$ SST model predictions for all the geometries. The IDDES predictions are also more accurate compared to all the RANS-based predictions for the wall normal velocity and Reynolds stresses. However, the neural network-augmented SST model has the least error in the streamwise velocity predictions. This is because the model has been trained on the streamwise velocity data from DNS. Similarly, by using the

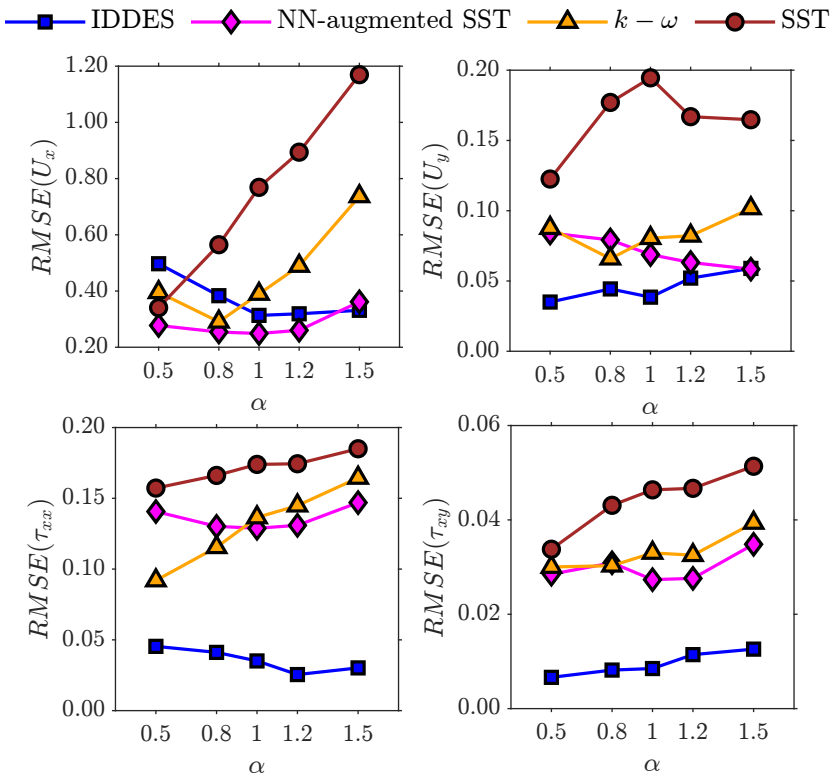


Figure 7.8: Root-mean-squared error comparison of the IDDES results compared to three RANS-based results. α, U_x, U_y, τ_{xx}, τ_{xy} represent the five periodic hills, the streamwise and wall-normal velocity components, and the normal and shear Reynolds stresses, respectively.

corresponding high fidelity data for the other quantities, even higher accuracy in these quantities may be possible. This will be investigated further in future works. The Wilcox $k-\omega$ and the neural network-augmented SST models are consistently more accurate than the baseline $k-\omega$ SST model for all four quantities considered.

Having verified the accuracy of the IDDES predictions compared to DNS predictions, we will now proceed to presenting the shape optimisation results, along with IDDES verification of the flow predictions.

7.4 Optimisation results

The aerodynamic shape optimisation results using the three turbulence models considered, $k-\omega$ SST, FIML-augmented $k-\omega$ SST, and the Wilcox $k-\omega$ model will be presented in this section.

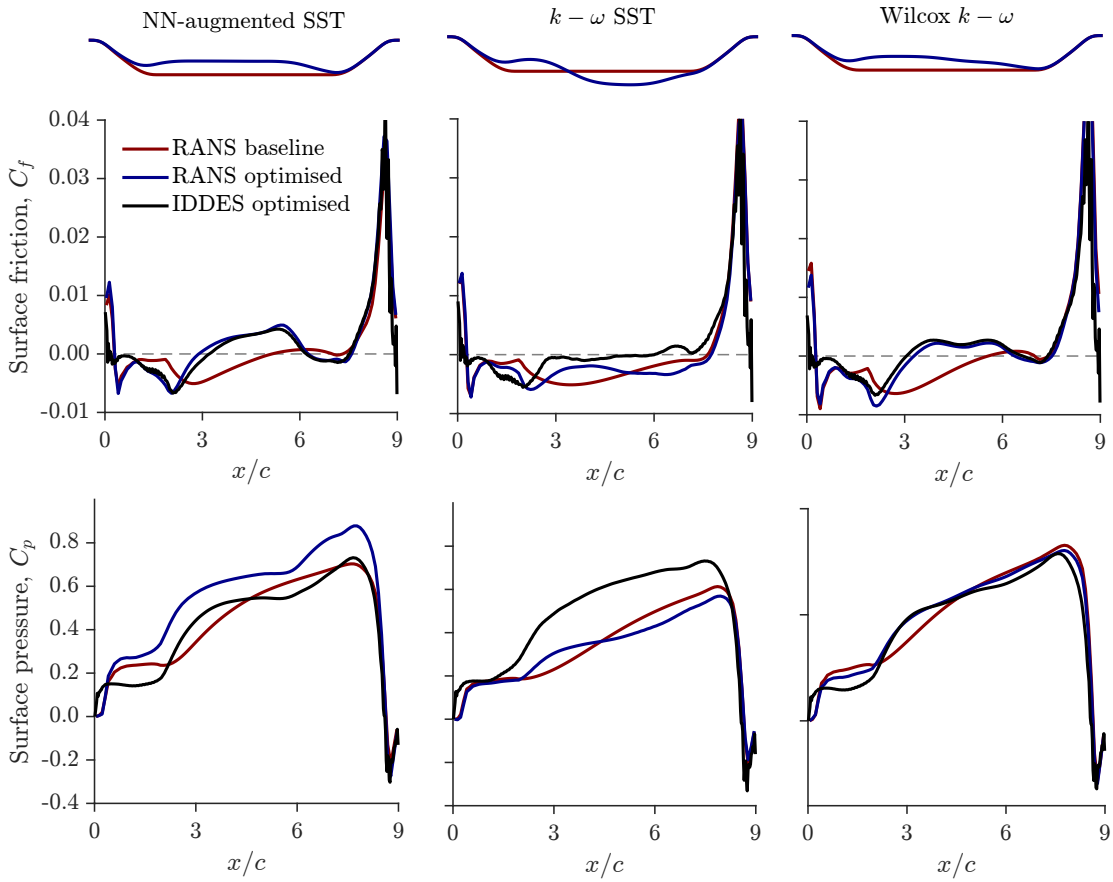


Figure 7.9: The baseline shape overlaid with the optimised shapes for the three RANS-based turbulence models. The lower wall surface friction and surface pressure for the baseline and optimised shapes are also shown using the three respective turbulence models in each column. In addition, IDDES predictions for the optimised shapes are shown for comparison.

The optimised shapes, along with lower wall pressure and skin friction distributions are shown in Fig. 7.9. Tables 7.1 and 7.2 show comparisons of the drag coefficients. Contours of streamwise velocity, and the streamlines for the RANS

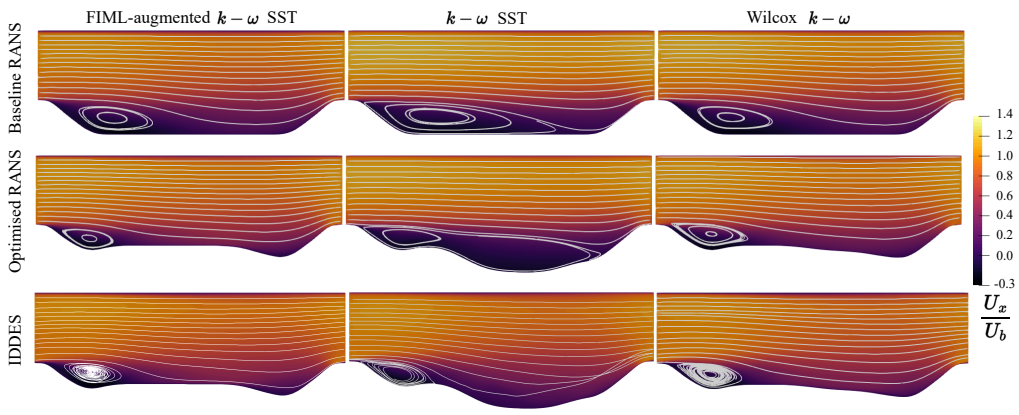


Figure 7.10: The baseline and optimised streamlines with normalised streamwise velocity. The first two rows show the predictions from the respective RANS-based turbulence models used during the shape optimisation, while the final row shows the IDDES predictions of the optimised shapes.

Table 7.1: Drag coefficient predictions for the optimised hill walls using the different models. C_d RANS column is the drag prediction by the turbulence model used during aerodynamic shape optimisation, i.e. the model in column one. ΔC_d is the percentage drag reduction compared to the baseline shape.

Optimisation case	C_d RANS	C_d IDDES	Error	ΔC_d RANS	ΔC_d IDDES
$k-\omega$ SST	0.035371	0.033291	6.25%	6.53%	4.52%
NN-augmented SST	0.031191	0.026043	19.77%	20.81%	25.30%
Wilcox $k-\omega$	0.033695	0.027963	20.50%	35.30%	19.80%

models and IDDES results are shown in Fig. 7.10, with velocity profiles comparisons in Fig. 7.11. Finally, turbulent flow structures are illustrated through Q-criterion contours from IDDES in Fig. 7.12. The key findings are summarised as follows:

1. The choice of turbulence model for flow predictions clearly impacts the aero-

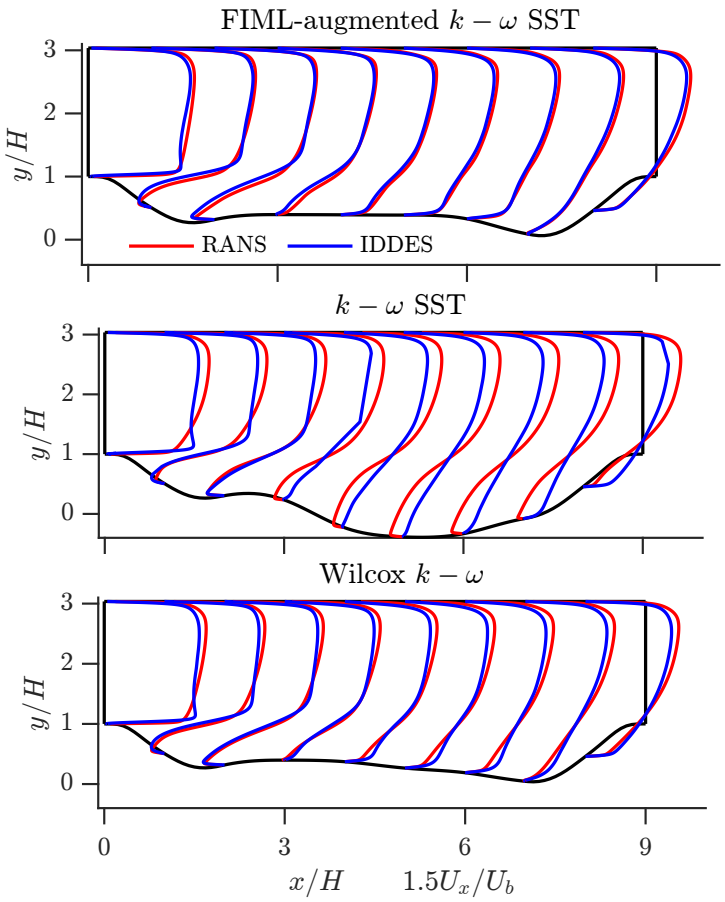


Figure 7.11: Streamwise velocity profiles for the optimised shapes using the three different RANS models compared against the IDDES results.

Table 7.2: Predictions of drag components for the three turbulence model optimisation scenarios. $C_{d,p}$ and $C_{d,v}$ refer to the pressure and viscous (friction) forces.

Optimisation case	$C_{d,p}$ RANS	$C_{d,p}$ IDDES	$C_{d,v}$ RANS	$C_{d,v}$ IDDES
$k-\omega$ SST	0.03514	0.03211	0.00024	0.00119
FIML-augmented SST	0.02867	0.02423	0.00252	0.00181
Wilcox $k-\omega$	0.03132	0.02608	0.00237	0.00188

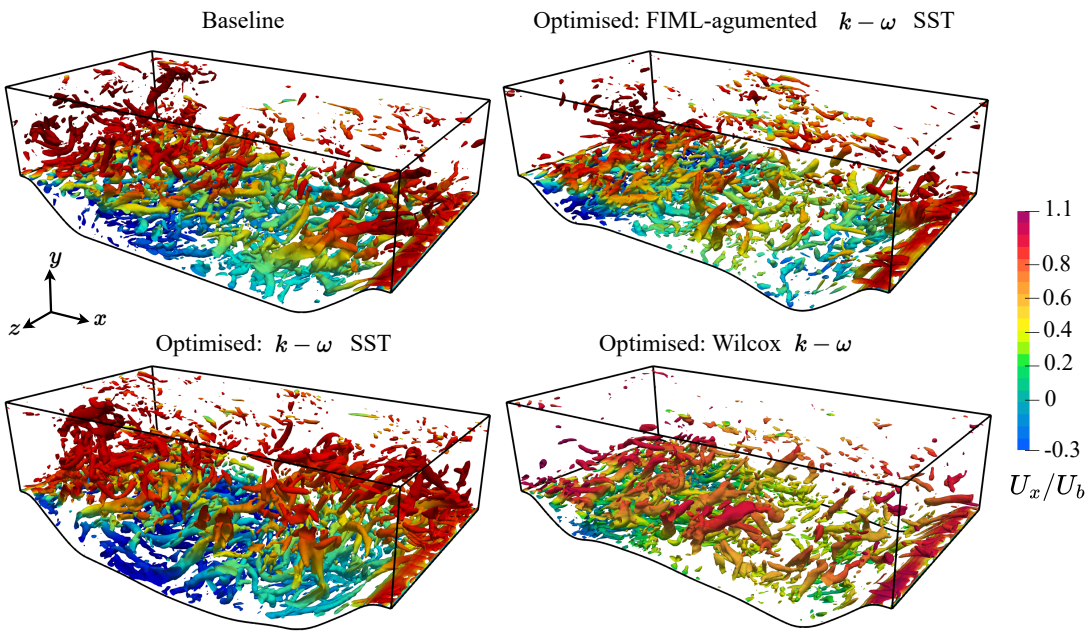


Figure 7.12: Visualisation of the turbulent flow structures for the baseline and optimised shapes, using IDDES flow predictions. Contours show the Q-criterion, coloured by the normalised streamwise velocity.

dynamic shape optimisation outcome, as the three models result in three distinct "optimal" shapes. The FIML-augmented $k-\omega$ and the Wilcox $k-\omega$ share some similarities which we will discuss shortly.

2. Table 7.1 shows that the baseline $k-\omega$ SST model achieves the least drag reduction–based both on the RANS predictions and the IDDES results. The FIML-augmented $k-\omega$ SST and the Wilcox $k-\omega$ achieve considerably higher reductions. However, there are considerable errors between the RANS and IDDES predictions. This could be as a result of the issue in pressure predictions in the IDDES results.

3. All three models attempt to reduce the drag in the baseline shape by decreasing the flow separation. The FIML augmented $k-\omega$ SST and the Wilcox $k-\omega$ achieve this by raising the lower wall. This results in a smaller

separated shear layer compared to the baseline shape, as demonstrated by the earlier flow reattachment locations on the skin friction plots (Fig. 7.9), and the streamlines in Fig. 7.10. On the other hand, while the baseline $k-\omega$ SST reduces the size of the recirculation zone after the first hill, it introduces a second larger recirculation zone adjacent to the first one (based on the RANS predictions).

4. In terms of the skin friction predictions, the baseline $k-\omega$ SST model continues to be the most erroneous, similar to the baseline shape. Both the FIML-augmented $k-\omega$ SST and the Wilcox $k-\omega$ model predictions are similar to the IDDES.

5. For the surface pressure distributions, there are considerable discrepancies between the SST RANS models (both baseline and FIML-augmented), while the Wilcox $k-\omega$ predictions are similar. Again, further investigation is required to discern how accurate the IDDES predictions are before using these for benchmarking the RANS predictions.

6. The baseline $k-\omega$ SST model continues to over-predict the flow separation in the shear layer, in comparison to the IDDES results, as shown in the velocity contours in Fig. 7.10 and the streamwise velocity profiles in Fig. 7.11. The Wilcox $k-\omega$ model is somewhat more accurate in velocity predictions, however, discrepancies exist near the walls. On the other hand, the FIML-augmented $k-\omega$ SST match the IDDES velocity predictions well (Fig. 7.11).

7. Finally, turbulent flow structures are visualised using the Q-criterion in Fig. 7.12. The baseline case has much larger turbulent structures near the hills. These are reduced by various degrees in the different optimised cases. Large regions of flow reversal (negative U_x/U_b values) in the optimisation case with

the baseline $k-\omega$ SST model also illustrate the higher drag prediction for this geometry.

7.5 Summary

This chapter presented an aerodynamic shape optimisation framework in which a RANS-based baseline turbulence model—$k-\omega$ shear stress transport (SST)—is augmented using the field inversion and machine learning (FIML) approach, presented in chapter 6. The framework was applied to a proof of concept periodic hill case. Key steps and findings are summarised as follows:

- The adjoint-based aerodynamic shape optimisation was formulated as that of reducing the surface drag on the lower wall of the periodic hill geometry.
- In order to demonstrate the inherent sensitivity of the shape optimisation outcome on the choice of RANS-based turbulence model for flow predictions, we also used the Wilcox $k-\omega$ model to perform shape optimisation. We also demonstrated that of the two existing (baseline) turbulence models considered, the Wilcox $k-\omega$ model was overall more accurate in terms of flow prediction comparisons to the reference DNS data.
- We also proposed using a higher-fidelity approach to validate RANS-based flow predictions for the optimised shapes. For this, the Spalart-Allmaras based improved delayed detached eddy simulations (IDDES) was used. We validated the IDDES results against the available reference DNS data. While the IDDES results were much more accurate compared to the baseline RANS-based predictions in most quantities (velocity, Reynolds stress components, and skin friction), it failed to match the reference DNS pressure distribution. The cause of this discrepancy is subject to future investigations. Additionally, future work can entirely rely on the IDDES data for

training the neural network-augmented turbulence model (instead of the DNS dataset used in the present work), which will ensure that the training and validation high-fidelity framework is consistent.

- The three different RANS-based turbulence models—baseline $k-\omega$ SST, FIML-augmented $k-\omega$ SST, and the Wilcox $k-\omega$—resulted in different optimal shapes, highlighting the sensitivity of the final outcome to the turbulence model.

- The drag reductions were quantified both based on the respective RANS-based flow predictions of the baseline and optimised shapes, and the IDDES-based predictions, Table 7.1.

- The baseline $k-\omega$ SST model led to the least drag reduction: 4.5%. The FIML-augmented $k-\omega$ SST model achieved a drag reduction of 20.8%. The Wilcox $k-\omega$ model achieved an error reduction of 19.8%.

- In comparison to the IDDES predictions, the baseline $k-\omega$ SST model was the least accurate in terms of velocity and skin friction predictions. Both the Wilcox $k-\omega$ and the FIML-augmented $k-\omega$ SST model matched the skin friction and velocity predictions from IDDES fairly well.

- The performance of the baseline $k-\omega$ model is similar to the NN-augmented $k-\omega$ SST, naturally, leading one to argue that the latter is unnecessary. However, the use of the Wilcox $k-\omega$ model in this work is to illustrate the sensitivity of aerodynamic design optimisation to the turbulence model in complex flows. The choice to augment the SST model in this work has been due to: 1) the knowledge that this model performs poorly for the separated periodic hill flows, and 2) to investigate the application of FIML to augment turbulence models for aerodynamic design optimisation, i.e. a

proof-of-concept study. The methodology presented is equally applicable to other turbulence models, including the Wilcox $k-\omega$, which while effective for the periodic hills, can perform poorly for other flows.

- Arguably, the $T.B.$ constraint (Eqn. 7.1) is artificial—a mathematical construct for a proof-of-concept test case. Future work will explore other constraints. For example, a minimum integrated area between the hills apex (i.e. $y/H \leq 1$, with an appropriate range constraint for x/H, e.g. $1 \leq x/H \leq 8$). Similar to the $T.B.$ constraint this should avoid the optimiser turning the lower wall into a flat plate in order to reduce the drag. Additionally, we will also consider entirely different flow cases, e.g. aerofoils.

CHAPTER 8

Conclusions and future work

8.1 Conclusions

Reynolds-averaged Navier-Stokes (RANS) equations with turbulence models continue to be the workhorse for predictions of complex flows despite their limitations. In recent years a number of data assimilation techniques have been introduced for improved RANS turbulence modelling. While some rely on huge datasets of high-fidelity data, such as those available from direct numerical simulations (DNS), other frameworks have been shown to achieve considerable improvements using limited, experimentally measurable, data. One such framework is field inversion: an approach that involves perturbations of the turbulence model transport equation through a spatial field and the iterative optimisation of this field such that the error between model prediction and data is minimised.

In this thesis we develop a novel sensor placement approach for generating sparse experimental data for data assimilation. It leverages systematic eigenspace perturbations of the Reynolds-stress tensor to identify regions of flow where RANS turbulence modelling assumptions are most erroneous and concentrate measurements in these areas using a novel greedy algorithm. The method is *a priori*, directly addressed structural errors in the existing model, and is computationally cheaper compared to many existing approaches in literature. Our results for various test cases show that the error reduction using our method for sensor placement

is close in accuracy to the instances where the entire flow field data are used for flow reconstruction, which are two to three orders of magnitude more data points than the placed sensors.

We investigate fusing data from various quantities (e.g. velocity and surface pressure measurements) to perform data assimilation using multi-objective optimisation. Results demonstrate that multi-sensor data leads to enhanced flow reconstruction. In addition, the tests results show that equally weighting the different terms in the composite multi-objective loss function is appropriate. In addition we investigate the relative importance of various physical quantities on the flow reconstruction. It is quantitatively shown that skin friction data can achieve overall error reduction comparable to using the streamwise velocity for the entire domain. On the other hand, using surface pressure data and discrete velocity profiles lead to similar overall error reductions.

Another area of interest for us is adjoint-based aerodynamic shape optimisation at off-design conditions where the flows exhibit complex behaviour, such as separation, poorly predicted by RANS models. Using higher fidelity methods (e.g. DNS, LES), or even unsteady RANS simulations, for iterative shape optimisation is prohibitively expensive as many function evaluations are required. To tackle this challenge, we propose the use of machine learning techniques trained on high-fidelity data for a relatively small parameter space to enhance the RANS-based models and then using the augmented model for shape optimisation, at a computational cost similar to the existing RANS-based approaches. We perform field inversion for geometries/flow conditions sufficiently representative of the design space. Next, neural networks are employed to predict the corrective fields for unseen geometries and/or flow conditions. We perform preliminary hypothesis testing on a proof-of-concept case based on a parameterised set of periodic hills which involve mild to massive flow separations with a drag reduction objective

function. Investigations highlight the role of the turbulence model on the optimal shapes achievable, with the data-driven model achieving approximately 4-times more drag reduction compared to the baseline model.

8.2 Future work

The following directions are identified for potential future research.

Sensor placement

While the framework was tested with adjoint-based data assimilation, it is directly applicable to other methods, such as the ensemble-based field inversion, since the placement strategy is decoupled from the data assimilation process. Future work can investigate this and compare the data assimilation outcomes between adjoint- and ensemble-based data assimilation approaches.

The framework was tested on well-established two-dimensional cases, however, the same methodology outlined in chapter 4 is directly applicable to three-dimensional flows, which will be investigated in the future. In addition, future work will investigate sensor placement for surface measurements (the thesis having only focused on velocity measurements within the flow domain). Especially for three dimensional flows the proposed strategy can be used to generate uncertainty maps for the surface pressure and skin friction predictions and then apply the placement algorithm to concentrate sensors at the regions of highest uncertainty.

As previously discussed, once the uncertainty maps are generated using eigenspace perturbations, sensors are placed using a novel greedy algorithm introduced in chapter 4.3.2—these two sequential steps being decoupled from one another. The decoupled nature of the strategy may be a limitation as we do not consider the sensitivity of the RANS model states with respect to the regions of uncertainty

during the sensor placement procedure. Our assumption is that the uncertainty map—variance for any *one* given quantity to be measured (e.g. U_x or p) based on the five predictions from eigenspace perturbations—can sufficiently inform the sensor placement. However, the sensitivity of that quantity to other flow variables are not necessarily taken into account when placing sensors. The advantage of this is the substantially lower computational cost of the placement strategy. However, future work can investigate the use of deep learning methods to incorporate the insights from the eigenspace perturbations into a single step with sensor placement. For instance, neural networks can be used to develop a reduced order model of the sensitivity of the flow variables to spatial regions of the flow, and incorporate this in an optimisation problem similar to the deep neural network used in Ref. [35]. This may make the heuristic minimum distance constraint to avoid sensor clustering in the greedy algorithm introduced in chapter 4.3.2 redundant.

Multi-sensor data

We demonstrated enhanced flow reconstruction when using data for multiple quantities when performing adjoint-based field inversion. As noted throughout the thesis, neural network approaches are required to generalise the field inversion corrections in order to use the augmentations beyond the cases used for training. Future work will investigate if the enhanced field inversion capability using multi-sensor data can also extend to enhanced capabilities when training the neural network-augmented turbulence model for flows beyond the training cases.

Aerodynamic shape optimisation

While we demonstrated the FIML utility in aerodynamic shape optimisation for a proof-of-concept case (due to the available high-fidelity data for the parameterised geometry)—the periodic hill flow—future work will apply the methodology to

aerospace flows of interest. In addition, we used commonly employed features/inputs for the neural networks. In future, the importance of these features will be investigated using methods such as SHAP (SHapley Additive exPlanations) analysis [95], leading to addition or deletion of features where appropriate. In addition, we will further investigate if other sets of features can be used to improve the neural-network training and prediction performance. In particular, the set of features we used required very large neural-network architectures, as discussed in chapter 6.3. It would be beneficial to note if another set of features can be selected that require a smaller neural network such as the one used in Ref. [45]. Finally, we will investigate if the FIML-direct approach as discussed in chapter 2.3.1 can lead to improved neural network training.

The black-box neural network approach can be replaced with closed-form corrections [68], or symbolic regression [165], to develop interpretable augmentations of turbulence models. A closed-form augmented model with simple algebraic expressions, such as in Ref. [68] can also reduce the computational overhead. Finally, we will investigate how the training data for any particular application can be generated systematically. It is critical that the training dataset sufficiently covers the parameter space to allow maximum flexibility and accuracy in predictions during the aerodynamic shape optimisation process. To achieve this, approaches such as Latin hypercube sampling and novel deep learning approaches will be investigated [140, 86].

Open-source implementation

The open-source FIML was implemented on three commonly used linear eddy viscosity turbulence models, and tested on incompressible and low Mach number compressible, two-dimensional steady flows. Future works can extend the capabilities to more complex flows, such as those in the transonic and supersonic

regime. The implementation can be extended to more sophisticated turbulence models, such as non-linear eddy viscosity models, and transitional models. In addition, the topics investigated in this thesis can be incorporated into the new capability for time-accurate unsteady flows in DAFoam [43]—the platform where we implemented the original open-source FIML capability.

Appendix A

Transport equations for baseline turbulence models

A.1 Spalart-Allmaras

The transport equation for the surrogate variable, $\tilde{\nu}$ along with the intermediate functions, and model constants are as follows [143]:

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}\left(1 - f_{t2}\right)\tilde{S}\tilde{\nu} - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right], \quad \text{(A.1)}$$

where the intermediate functions are

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \qquad \chi = \frac{\tilde{\nu}}{\nu}, \qquad f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}, \qquad g = r + c_{w2}(r^6 - r),$$
$$r = \frac{\tilde{\nu}}{\tilde{\Omega}\kappa^2 d^2}, \qquad \tilde{\Omega} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \qquad c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}, \quad \text{(A.2)}$$

with Ω representing the rotation rate, and d the wall distance. The model constants are as follows: $c_{b1} = 0.1355$, $c_{v1} = 7.1$, $\sigma = 2/3$, $c_{b2} = 0.622$, $\kappa = 0.41$, $c_{w2} = 0.622$, and $c_{w3} = 2.0$.

A.2 Wilcox $k-\omega$

The transport equation for the two-equation turbulence model solving for the turbulent kinetic energy and dissipation rate, along with the model constants are outlined below [163]:

$$\frac{(\partial k)}{\partial t} + U_j \frac{\partial (k)}{\partial x_j} = \tau_{ij} \frac{\partial U_i}{\partial x_j} - \beta^\star \omega k + \frac{\partial}{\partial x_j} \left[\left(\mu + \sigma^* \frac{k}{\omega} \right) \frac{\partial k}{\partial x_j} \right], \tag{A.3}$$

$$\frac{\partial \omega}{\partial t} + U_j \frac{\partial \omega}{\partial x_j} = \alpha \frac{\omega}{k} \tau_{ij} \frac{\partial U_i}{\partial x_j} - \beta \omega^2 + \frac{\partial}{\partial x_j} \left[\left(\nu + \sigma \frac{k}{\omega} \right) \frac{\partial \omega}{\partial x_j} \right], \tag{A.4}$$

where

$$\tau_{ij} = \nu_t \left(2S_{ij} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} k \delta_{ij}, \tag{A.5}$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{A.6}$$

and the model constants are as follows: $\alpha = 5/9$, $\beta = 3/40$, $\beta^\star = 9/100$, $\sigma = 1/2$, and $\sigma^* = 1/2$.

A.3 $k-\omega$ shear stress transport

The $k-\omega$ shear stress transport (SST) [105, 106] model blends the $k-\omega$ model in regions close to walls, and $k-\epsilon$ in the farfield. The transport equations for the turbulent variables are

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = P_k - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right], \tag{A.7}$$

$$\frac{\partial \omega}{\partial t} + U_j \frac{\partial \omega}{\partial x_j} = \alpha S^2 - \beta \omega^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \nu_t) \frac{\partial \omega}{\partial x_j} \right] + 2\left(1 - F_1\right) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, \quad \text{(A.8)}$$

where

$$F_2 = \tanh\left[\left[\max\left(\frac{2\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right) \right]^2 \right], \quad \text{(A.9)}$$

$$P_k = \min\left(\tau_{ij} \frac{\partial U_i}{\partial x_j}, 10\beta^* k\omega \right), \quad \text{(A.10)}$$

$$F_1 = \tanh\left\{ \left\{ \min\left[\max\left(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \frac{4\sigma_{\omega 2} k}{CD_{k\omega} y^2} \right] \right\}^4 \right\}, \quad \text{(A.11)}$$

$$CD_{k\omega} = \max\left(2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-10} \right), \quad \text{(A.12)}$$

with the following model constants: $\alpha_1 = 5/9$, $\alpha_2 = 0.44$, $\beta_1 = 3/40$, $\beta_2 = 0.0828$, $\beta^* = 9/100$, $\sigma_{k1} = 0.85$, $\sigma_{k2} = 1$, $\sigma_{\omega 1} = 0.5$, and $\sigma_{\omega 2} = 0.856$.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: A system for large-scale machine learning*. 2016. arXiv: 1605.08695 [cs.DC].

[2] Adel Abbas-Bayoumi and Klaus Becker. "An industrial view on numerical simulation for aircraft aerodynamic design". In: *Journal of Mathematics in Industry* 1.1 (2011), p. 10. DOI: 10.1186/2190-5983-1-10.

[3] Shady E. Ahmed, Suraj Pawar, Omer San, Adil Rasheed, Traian Iliescu, and Bernd R. Noack. "On closures for reduced order models—A spectrum of first-principle to machine-learned avenues". In: *Physics of Fluids* 33.9 (Sept. 2021). ISSN: 1089-7666. DOI: 10.1063/5.0061577.

[4] Giancarlo Alfonsi. "Reynolds-Averaged Navier–Stokes Equations for Turbulence Modeling". In: *Applied Mechanics Reviews* 62.4 (June 2009). DOI: 10.1115/1.3124648.

[5] Naseem Ali, Marc Calaf, and Raúl Bayoán Cal. "Clustering sparse sensor placement identification and deep learning based forecasting for wind turbine wakes". In: *Journal of Renewable and Sustainable Energy* 13.2 (Mar. 2021), p. 023307. DOI: 10.1063/5.0036281.

[6] Richard C Aster, Brian Borchers, and Clifford H Thurber. *Parameter Estimation and Inverse Problems*. Elsevier, 2019. DOI: 10.1016/c2015-0-02458-3.

[7] Jeffrey S Baggett. "On the feasibility of merging LES with RANS for the near-wall region of attached turbulent flows". In: *Annual Research Briefs* (1998), pp. 267–277.

[8] S. Banerjee, R. Krahl, F. Durst, and Ch. Zenger. "Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches". In: *Journal of Turbulence* 8 (Jan. 2007), N32. DOI: 10.1080/14685240701506896.

[9] Paul Bartholomew, Georgios Deskos, Ricardo A.S. Frantz, Felipe N. Schuch, Eric Lamballais, and Sylvain Laizet. "Xcompact3D: An open-source framework for solving turbulence problems on a Cartesian mesh". In: *SoftwareX* 12 (July 2020), p. 100550. ISSN: 2352-7110. DOI: 10.1016/j.softx.2020.100550.

[10] Andrea Beck and Marius Kurz. "A perspective on machine learning methods in turbulence modeling". In: *GAMM-Mitteilungen* 44.1 (Mar. 2021). DOI: 10.1002/gamm.202100002.

[11] S. Beetham and J. Capecelatro. "Formulating turbulence closures using sparse regression with embedded form invariance". In: *Physical Review Fluids* 5.8 (Aug. 2020). DOI: 10.1103/physrevfluids.5.084611.

[12] Zeno Belligoli, Richard P. Dwight, and Georg Eitelberg. "Reconstruction of Turbulent Flows at High Reynolds Numbers Using Data Assimilation Techniques". In: *AIAA Journal* 59.3 (Mar. 2021), pp. 855–867. DOI: 10.2514/1.j059474.

[13] Ismaïl Ben Hassan Saïdi, Martin Schmelzer, Paola Cinnella, and Francesco Grasso. "CFD-driven symbolic identification of algebraic Reynolds-stress models". In: *Journal of Computational Physics* 457 (May 2022), p. 111037. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2022.111037.

[14] Omid Bidar, Sean Anderson, and Ning Qin. "A Hybrid RANS-LES Dataset for Data-driven Turbulent Mean Flow Reconstruction". In: *Cambridge Unsteady Flow Symposium*. 2024.

[15] Omid Bidar, Sean Anderson, and Ning Qin. "A Priori Sensor Placement Strategy for Turbulent Mean Flow Reconstruction Using Parametric Model Perturbations". In: *AIAA SCITECH 2024 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2024. DOI: 10.2514/6.2024-1580.

[16] Omid Bidar, Sean Anderson, and Ning Qin. "Sensor placement for data assimilation of turbulence models using eigenspace perturbations". In: *Physics of Fluids* 36.1 (Jan. 2024). DOI: 10.1063/5.0182080.

[17] Omid Bidar, Ping He, Sean Anderson, and Ning Qin. "Aerodynamic Shape Optimisation Using a Machine Learning-Augmented Turbulence Model". In: *AIAA SCITECH 2024 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2024. DOI: 10.2514/6.2024-1231.

[18] Omid Bidar, Ping He, Sean Anderson, and Ning Qin. "An Open-Source Adjoint-based Field Inversion Tool for Data-driven RANS modelling". In: *AIAA AVIATION 2022 Forum*. American Institute of Aeronautics and Astronautics, June 2022. DOI: 10.2514/6.2022-4125.

[19] Omid Bidar, Ping He, Sean Anderson, and Ning Qin. "Turbulent Mean Flow Reconstruction Based on Sparse Multi-sensor Data and Adjoint-based Field Inversion". In: *AIAA AVIATION 2022 Forum*. American Institute of Aeronautics and Astronautics, June 2022. DOI: 10.2514/6.2022-3900.

[20] Sanjeeb T. Bose and George Ilhwan Park. "Wall-Modeled Large-Eddy Simulation for Complex Turbulent Flows". In: *Annual Review of Fluid Mechanics* 50.1 (Jan. 2018), pp. 535–561. ISSN: 1545-4479. DOI: 10.1146/annurev-fluid-122316-045241.

[21] Peter Bradshaw and AD Young. "Effects of streamline curvature on turbulent flow". In: Agard Paris. 1973.

[22] M. P. Brenner, J. D. Eldredge, and J. B. Freund. "Perspective on machine learning for advancing fluid mechanics". In: *Physical Review Fluids* 4.10 (Oct. 2019). ISSN: 2469-990X. DOI: 10.1103/physrevfluids.4.100501.

[23] M. Breuer, N. Peller, Ch. Rapp, and M. Manhart. "Flow over periodic hills – Numerical and experimental study in a wide range of Reynolds numbers". In: *Computers & Fluids* 38.2 (Feb. 2009), pp. 433–457. DOI: 10.1016/j.compfluid.2008.05.002.

[24] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. "Machine Learning for Fluid Mechanics". In: *Annual Review of Fluid Mechanics* 52.1 (Jan. 2020), pp. 477–508. DOI: 10.1146/annurev-fluid-010719-060214.

[25] Jared L. Callaham, Kazuki Maeda, and Steven L. Brunton. "Robust flow reconstruction from limited measurements via sparse representation". In: *Physical Review Fluids* 4.10 (Oct. 2019). DOI: 10.1103/physrevfluids.4.103907.

[26] Arthur Shiniti Cato, Pedro Stefanin Volpiani, Vincent Mons, Olivier Marquet, and Denis Sipp. "Comparison of different data-assimilation approaches to augment RANS turbulence models". In: *Computers amp; Fluids* 266 (Nov. 2023), p. 106054. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2023.106054.

[27] Pinaki Chakraborty, S. Balachandar, and Ronald J. Adrian. "On the relationships between local vortex identification schemes". In: *Journal of Fluid Mechanics* 535 (July 2005), pp. 189–214. ISSN: 1469-7645. DOI: 10.1017/s0022112005004726.

[28] Bruno Chaouat. "The State of the Art of Hybrid RANS/LES Modeling for the Simulation of Turbulent Flows". In: *Flow, Turbulence and Combustion* 99.2 (July 2017), pp. 279–327. ISSN: 1573-1987. DOI: 10.1007/s10494-017-9828-8.

[29] Sai Hung Cheung, Todd A. Oliver, Ernesto E. Prudencio, Serge Prudhomme, and Robert D. Moser. "Bayesian uncertainty analysis with applications to turbulence modeling". In: *Reliability Engineering & System Safety* 96.9 (Sept. 2011), pp. 1137–1149. DOI: 10.1016/j.ress.2010.09.013.

[30] Haecheon Choi and Parviz Moin. "Grid-point requirements for large eddy simulation: Chapman's estimates revisited". In: *Physics of Fluids* 24.1 (Jan. 2012). ISSN: 1089-7666. DOI: 10.1063/1.3676783.

[31] Kelly Cohen, Stefan Siegel, and Thomas McLaughlin. "A heuristic approach to effective sensor placement for modeling of a cylinder wake". In: *Computers & Fluids* 35.1 (Jan. 2006), pp. 103–120. DOI: 10.1016/j.compfluid.2004.11.002.

[32] Donald Coles and Alan J. Wadcock. "Flying-Hot-wire Study of Flow Past an NACA 4412 Airfoil at Maximum Lift". In: *AIAA Journal* 17.4 (Apr. 1979), pp. 321–329. ISSN: 1533-385X. DOI: 10.2514/3.61127.

[33] L. W. Cook, A. A. Mishra, J. P. Jarrett, K. E. Willcox, and G. Iaccarino. "Optimization under turbulence model uncertainty for aerospace design". In: *Physics of Fluids* 31.10 (Oct. 2019), p. 105111. DOI: 10.1063/1.5118785.

[34] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2 (Apr. 2002), pp. 182–197. DOI: 10.1109/4235.996017.

[35] Zhiwen Deng, Chuangxin He, and Yingzheng Liu. "Deep neural network-based strategy for optimal sensor placement in data assimilation of turbulent flow". In: *Physics of Fluids* 33.2 (Feb. 2021), p. 025119. DOI: 10.1063/5.0035230.

[36] Eric Dow and Qiqi Wang. "Quantification of Structural Uncertainties in the k -w Turbulence Model". In: *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. American Institute of Aeronautics and Astronautics, Apr. 2011. DOI: 10.2514/6.2011-1762.

[37] Karthik Duraisamy. "Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence". In: *Physical Review Fluids* 6.5 (May 2021). DOI: 10.1103/physrevfluids.6.050504.

[38] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. "Turbulence Modeling in the Age of Data". In: *Annual Review of Fluid Mechanics* 51.1 (Jan. 2019), pp. 357–377. DOI: 10.1146/annurev-fluid-010518-040547.

[39] Karthik Duraisamy, Anand-Pratap Singh, and Shaowu Pan. "Augmentation of Turbulence Models Using Field Inversion and Machine Learning". In: *55th AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, Jan. 2017. DOI: 10.2514/6.2017-0993.

[40] Karthikeyan Duraisamy, Ze J. Zhang, and Anand Pratap Singh. "New Approaches in Turbulence and Transition Modeling Using Data-driven Techniques". In: *53rd AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, Jan. 2015. DOI: 10.2514/6.2015-1284.

[41] Paul A. Durbin. "Some Recent Developments in Turbulence Closure Modeling". In: *Annual Review of Fluid Mechanics* 50.1 (Jan. 2018), pp. 77–103. DOI: 10.1146/annurev-fluid-122316-045020.

[42] Michael Emory, Johan Larsson, and Gianluca Iaccarino. "Modeling of structural uncertainties in Reynolds-averaged Navier-Stokes closures". In: *Physics of Fluids* 25.11 (Nov. 2013), p. 110822. DOI: 10.1063/1.4824659.

[43] Lean Fang and Ping He. "Field inversion machine learning augmented turbulence modeling for time-accurate unsteady flow". In: *Physics of Fluids* 36.5 (May 2024). ISSN: 1089-7666. DOI: 10.1063/5.0207704.

[44] Krzysztof Fidkowski. "Correcting an Algebraic Transition Model using Field Inversion and Machine Learning". In: *AIAA SCITECH 2024 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2024. DOI: 10.2514/6.2024-2739.

[45] Krzysztof J. Fidkowski. "Gradient-based shape optimization for unsteady turbulent simulations using field inversion and machine learning". In: *Aerospace Science and Technology* 129 (Oct. 2022), p. 107843. DOI: 10.1016/j.ast.2022.107843.

[46] Krzysztof J. Fidkowski. "Output-based error estimation and mesh adaptation for unsteady turbulent flow simulations". In: *Computer Methods in Applied Mechanics and Engineering* 399 (Sept. 2022), p. 115322. ISSN: 0045-7825. DOI: 10.1016/j.cma.2022.115322.

[47] Krzysztof J. Fidkowski and Guodong Chen. "Metric-based, goal-oriented mesh adaptation using machine learning". In: *Journal of Computational Physics* 426 (Feb. 2021), p. 109957. DOI: 10.1016/j.jcp.2020.109957.

[48] Krzysztof J. Fidkowski and David L. Darmofal. "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics". In: *AIAA Journal* 49.4 (Apr. 2011), pp. 673–694. ISSN: 1533-385X. DOI: 10.2514/1.j050073.

[49] Dimitry P. G. Foures, Nicolas Dovetta, Denis Sipp, and Peter J. Schmid. "A data-assimilation method for Reynolds-averaged Navier–Stokes-driven mean flow reconstruction". In: *Journal of Fluid Mechanics* 759 (Nov. 2014), pp. 404–431. DOI: 10.1017/jfm.2014.566.

[50] Jochen Frohlich, Christopher. P. Mellen, Wolfgang Rodi, Lionel Temmerman, and Michael A. Leschziner. "Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions". In: *Journal of Fluid Mechanics* 526 (Mar. 2005), pp. 19–66. ISSN: 1469-7645. DOI: 10.1017/s0022112004002812.

[51] Jochen Fröhlich and Dominic von Terzi. "Hybrid LES/RANS methods for the simulation of turbulent flows". In: *Progress in Aerospace Sciences* 44.5 (July 2008), pp. 349–377. DOI: 10.1016/j.paerosci.2008.05.001.

[52] T.B. Gatski and T. Jongen. "Nonlinear eddy viscosity and algebraic stress models for solving complex turbulent flows". In: *Progress in Aerospace Sciences* 36.8 (Nov. 2000), pp. 655–682. DOI: 10.1016/s0376-0421(00)00012-9.

[53] Mike Giles, Devendra Ghate, and Mihai Duta. "Using automatic differentiation for adjoint CFD code development". In: (2005).

[54] Philip E. Gill, Walter Murray, and Michael A. Saunders. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization". In: *SIAM Review* 47.1 (Jan. 2005), pp. 99–131. DOI: 10.1137/s0036144504446096.

[55] Xavier Gloerfelt and Paola Cinnella. "Large Eddy Simulation Requirements for the Flow over Periodic Hills". In: *Flow, Turbulence and Combustion* 103.1 (Mar. 2019), pp. 55–91. DOI: 10.1007/s10494-018-0005-5.

[56] C. Gorlé, J. Larsson, M. Emory, and G. Iaccarino. "The deviation from parallel shear flow as an indicator of linear eddy-viscosity model inaccuracy". In: *Physics of Fluids* 26.5 (May 2014). ISSN: 1089-7666. DOI: 10.1063/1.4876577.

[57] David Greenblatt, Keith B. Paschal, Chung-Sheng Yao, Jerome Harris, Norman W. Schaeffler, and Anthony E. Washburn. "Experimental Investigation of Separation Control Part 1: Baseline and Steady Suction". In: *AIAA Journal* 44.12 (Dec. 2006), pp. 2820–2830. DOI: 10.2514/1.13817.

[58] Fernando F Grinstein, Len G Margolin, and William J Rider. *Implicit large eddy simulation*. Vol. 10. Cambridge university press Cambridge, 2007.

[59] K. Hanjalić, M. Popovac, and M. Hadžiabdić. "A robust near-wall elliptic-relaxation eddy-viscosity turbulence model for CFD". In: *International Journal of Heat and Fluid Flow* 25.6 (Dec. 2004), pp. 1047–1051. DOI: 10.1016/j.ijheatfluidflow.2004.07.005.

[60] K. Hanjalic´. "Will RANS Survive LES? A View of Perspectives". In: *Volume 2, Parts A and B*. HT-FED2004. ASMEDC, Jan. 2004. DOI: 10.1115/ht-fed2004-56356.

[61] Chuangxin He, Yingzheng Liu, and Lian Gan. "A data assimilation model for turbulent flows using continuous adjoint formulation". In: *Physics of Fluids* 30.10 (Oct. 2018), p. 105108. DOI: 10.1063/1.5048727.

[62] Ping He, Charles A. Mader, Joaquim R. R. A. Martins, and Kevin J. Maki. "DAFoam: An Open-Source Adjoint Framework for Multidisciplinary Design Optimization with OpenFOAM". In: *AIAA Journal* 58.3 (Mar. 2020), pp. 1304–1319. DOI: 10.2514/1.j058853.

[63] Xiao He, Jianheng Tan, Georgios Rigas, and Mehdi Vahdati. "On the explainability of machine-learning-assisted turbulence modeling for transonic flows". In: *International Journal of Heat and Fluid Flow* 97 (Oct. 2022), p. 109038. ISSN: 0142-727X. DOI: 10.1016/j.ijheatfluidflow.2022.109038.

[64] Joel Ho and Alastair West. "Field Inversion and Machine Learning for turbulence modelling applied to three-dimensional separated flows". In: *AIAA AVIATION 2021 FORUM*. American Institute of Aeronautics and Astronautics, July 2021. DOI: 10.2514/6.2021-2903.

[65] Jonathan R. Holland, James D. Baeder, and Karthikeyan Duraisamy. "Field Inversion and Machine Learning With Embedded Neural Networks: Physics-Consistent Neural Network Training". In: *AIAA Aviation 2019 Forum*. American Institute of Aeronautics and Astronautics, June 2019. DOI: 10.2514/6.2019-3200.

[66] Jonathan R. Holland, James D. Baeder, and Karthikeyan Duraisamy. "Field Inversion and Machine Learning With Embedded Neural Networks: Physics-Consistent Neural Network Training". In: *AIAA Aviation 2019 Forum*. American Institute of Aeronautics and Astronautics, June 2019. DOI: 10.2514/6.2019-3200.

[67] Gianluca Iaccarino, Aashwin Ananda Mishra, and Saman Ghili. "Eigenspace perturbations for uncertainty estimation of single-point turbulence closures". In: *Physical Review Fluids* 2.2 (Feb. 2017). DOI: 10.1103/physrevfluids.2.024605.

[68] Florian Jäckel. "A Closed-Form Correction for the Spalart–Allmaras Turbulence Model for Separated Flows". In: *AIAA Journal* 61.6 (May 2023), pp. 2319–2330. ISSN: 1533-385X. DOI: 10.2514/1.j061649.

[69] Hrvoje Jasak, Aleksandar Jemcov, Zeljko Tukovic, et al. "OpenFOAM: A C++ library for complex physics simulations". In: *International workshop on coupled methods in numerical dynamics*. Vol. 1000. IUC Dubrovnik Croatia. 2007, pp. 1–20.

[70] Bo Jin, Simon J. Illingworth, and Richard D. Sandberg. "Resolvent-based approach for H_2-optimal estimation and control: an application to the cylinder flow". In: *Theoretical and Computational Fluid Dynamics* 36.3 (Apr. 2022), pp. 491–515. DOI: 10.1007/s00162-022-00608-z.

[71] Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations". In: *Journal of Computational Physics* 426 (Feb. 2021), p. 109951. DOI: 10.1016/j.jcp.2020.109951.

[72] Forrester T. Johnson, Edward N. Tinoco, and N. Jong Yu. "Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle". In: *Computers & Fluids* 34.10 (Dec. 2005), pp. 1115–1151. DOI: 10.1016/j.compfluid.2004.06.005.

[73] Mikael L.A. Kaandorp and Richard P. Dwight. "Data-driven modelling of the Reynolds stress tensor using random forests with invariance". In: *Computers & Fluids* 202 (Apr. 2020), p. 104497. DOI: 10.1016/j.compfluid.2020.104497.

[74] Niharika Karnik, Mohammad G. Abdo, Carlos E. Estrada Perez, Jun Soo Yoo, Joshua J. Cogliati, Richard S. Skifton, Pattrick Calderoni, Steven L. Brunton, and Krithika Manohar. "Optimal Sensor Placement with Adaptive Constraints for Nuclear Digital Twins". In: *arXiv e-prints* (June 2023). DOI: 10.48550/arXiv.2306.13637.

[75] Gaetan K.W. Kenway, Charles A. Mader, Ping He, and Joaquim R.R.A. Martins. "Effective adjoint approaches for computational fluid dynamics". In: *Progress in Aerospace Sciences* 110 (Oct. 2019), p. 100542. DOI: 10.1016/j.paerosci.2019.05.002.

[76] Benjamin Krank, Martin Kronbichler, and Wolfgang A. Wall. "Direct Numerical Simulation of Flow over Periodic Hills up to $Re = 10,595$". In: *Flow, Turbulence and Combustion* 101.2 (June 2018), pp. 521–551. DOI: 10.1007/s10494-018-9941-3.

[77] Pradeep Kumar, Yosef M. El Sayed, and Richard Semaan. "Optimized sensor placement using stochastic estimation for a flow over a 2D airfoil with Coanda blowing". In: *7th AIAA Flow Control Conference*. American Institute of Aeronautics and Astronautics, June 2014. DOI: 10.2514/6.2014-2101.

[78] Andrew B. Lambe and Joaquim R. R. A. Martins. "Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes". In: *Structural and Multidisciplinary Optimization* 46.2 (Jan. 2012), pp. 273–284. ISSN: 1615-1488. DOI: 10.1007/s00158-012-0763-y.

[79] Johan LARSSON, Soshi KAWAI, Julien BODART, and Ivan BERMEJO-MORENO. "Large eddy simulation with modeled wall-stress: recent progress and future directions". In: *Mechanical Engineering Reviews* 3.1 (2016), pp. 15-00418-15–00418. ISSN: 2187-9753. DOI: 10.1299/mer.15-00418.

[80] B. E. Launder, G. J. Reece, and W. Rodi. "Progress in the development of a Reynolds-stress turbulence closure". In: *Journal of Fluid Mechanics* 68.3 (Apr. 1975), pp. 537–566. ISSN: 1469-7645. DOI: 10.1017/s0022112075001814.

[81] B. E. Launder and N. D. Sandham, eds. *Closure Strategies for Turbulent and Transitional Flows*. Cambridge University Press, Jan. 2001. DOI: 10.1017/cbo9780511755385.

[82] B.E. Launder and D.B. Spalding. "The numerical computation of turbulent flows". In: *Computer Methods in Applied Mechanics and Engineering* 3.2

(1974), pp. 269–289. ISSN: 0045-7825. DOI: https://doi.org/10.1016/0045-7825(74)90029-2.

[83] Jean-Philippe Laval and Matthieu Marquillie. "Direct Numerical Simulations of Converging–Diverging Channel Flow". In: *ERCOFTAC Series*. Springer Netherlands, 2011, pp. 203–209. DOI: 10.1007/978-90-481-9603-6_21.

[84] Michael Leschziner. "Modelling Separation from Curved Surfaces with Anisotropy-Resolving Turbulence Closures". In: *Modelling Fluid Flow*. Springer Berlin Heidelberg, 2004, pp. 23–47. DOI: 10.1007/978-3-662-08797-8_2.

[85] Jichao Li, Xiaosong Du, and Joaquim R.R.A. Martins. "Machine learning in aerodynamic shape optimization". In: *Progress in Aerospace Sciences* 134 (Oct. 2022), p. 100849. ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2022.100849.

[86] Sen Li, Chuangxin He, and Yingzheng Liu. "A data assimilation model for wall pressure-driven mean flow reconstruction". In: *Physics of Fluids* 34.1 (Jan. 2022). DOI: 10.1063/5.0076754.

[87] Xiangdong Li and Jiyuan Tu. "Evaluation of the eddy viscosity turbulence models for the simulation of convection–radiation coupled heat transfer in indoor environment". In: *Energy and Buildings* 184 (Feb. 2019), pp. 8–18. DOI: 10.1016/j.enbuild.2018.11.043.

[88] J. Ling and J. Templeton. "Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty". In: *Physics of Fluids* 27.8 (Aug. 2015), p. 085103. DOI: 10.1063/1.4927765.

[89] J. Ling and J. Templeton. "Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty". In: *Physics of Fluids* 27.8 (Aug. 2015), p. 085103. DOI: 10.1063/1.4927765.

[90] Julia Ling, Reese Jones, and Jeremy Templeton. "Machine learning strategies for systems with invariance properties". In: *Journal of Computational Physics* 318 (Aug. 2016), pp. 22–35. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2016.05.003.

[91] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". In: *Journal of Fluid Mechanics* 807 (Oct. 2016), pp. 155–166. DOI: 10.1017/jfm.2016.615.

[92] Mario Lino, Stathi Fotiadis, Anil A. Bharath, and Chris D. Cantwell. "Current and emerging deep-learning methods for the simulation of fluid dynamics". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 479.2275 (July 2023). ISSN: 1471-2946. DOI: 10.1098/rspa.2023.0058.

[93] Shengqi Lu and George Papadakis. "Flow Reconstruction Around a Surface-Mounted Prism from Sparse Velocity and/or Scalar Measurements Using a Combination of POD and a Data-Driven Estimator". In: *Flow, Turbulence and Combustion* 110.4 (Apr. 2023), pp. 1059–1090. ISSN: 1573-1987. DOI: 10.1007/s10494-023-00417-2.

[94] John L. Lumley and Gary R. Newman. "The return to isotropy of homogeneous turbulence". In: *Journal of Fluid Mechanics* 82.1 (Aug. 1977), pp. 161–178. ISSN: 1469-7645. DOI: 10.1017/s0022112077000585.

[95] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI].

[96] Zhoujie Lyu, Zelu Xu, and Joaquim R.R.A. Martins. "Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization". In: *Proceedings of the 8th international conference on computational fluid dynamics. Chengdu, Sichuan, China*. 2014.

[97] N.R. Gauger M. Sagebaum T. Albring. "High-Performance Derivative Computations using CoDiPack". In: *ACM Transactions on Mathematical Software (TOMS)* 45.4 (Dec. 1, 2019). URL: https://dl.acm.org/doi/abs/10.1145/3356900. published.

[98] Hannes Mandler and Bernhard Weigand. "Feature importance in neural networks as a means of interpretation for data-driven turbulence models". In: *Computers & Fluids* 265 (Oct. 2023), p. 105993. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2023.105993.

[99] Krithika Manohar, Bingni W. Brunton, J. Nathan Kutz, and Steven L. Brunton. "Data-Driven Sparse Sensor Placement for Reconstruction: Demonstrating the Benefits of Exploiting Known Patterns". In: *IEEE Control Systems Magazine* 38.3 (2018), pp. 63–86. DOI: 10.1109/MCS.2018.2810460.

[100] Joaquim R.R.A. Martins. "Aerodynamic design optimization: Challenges and perspectives". In: *Computers & Fluids* 239 (May 2022), p. 105391. DOI: 10.1016/j.compfluid.2022.105391.

[101] PJ Mason. "Large-eddy simulation: A critical review of the technique". In: *Quarterly Journal of the Royal Meteorological Society* 120.515 (Jan. 1994), pp. 1–26. DOI: 10.1256/smsqj.51501.

[102] Romit Maulik, Himanshu Sharma, Saumil Patel, Bethany Lusch, and Elise Jennings. "Deploying deep learning in OpenFOAM with TensorFlow". In: *AIAA Scitech 2021 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2021. DOI: 10.2514/6.2021-1485.

[103] Ryley McConkey, Eugene Yee, and Fue-Sang Lien. "A curated dataset for data-driven turbulence modelling". In: *Scientific Data* 8.1 (Sept. 2021). DOI: 10.1038/s41597-021-01034-2.

[104] F. Menter. "Zonal Two Equation k-w Turbulence Models For Aerodynamic Flows". In: *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*. American Institute of Aeronautics and Astronautics, July 1993. DOI: 10.2514/6.1993-2906.

[105] F. R. Menter. "Two-equation eddy-viscosity turbulence models for engineering applications". In: *AIAA Journal* 32.8 (Aug. 1994), pp. 1598–1605. DOI: 10.2514/3.12149.

[106] Florian R Menter, Martin Kuntz, and Robin Langtry. "Ten years of industrial experience with the SST turbulence model". In: *Turbulence, heat and mass transfer* 4.1 (2003), pp. 625–632.

[107] Pedro M. Milani, Julia Ling, and John K. Eaton. "Turbulent scalar flux in inclined jets in crossflow: counter gradient transport and deep learning modelling". In: *Journal of Fluid Mechanics* 906 (Nov. 2020). DOI: 10.1017/jfm.2020.820.

[108] Pedro M. Milani, Julia Ling, and John K. Eaton. "Turbulent scalar flux in inclined jets in crossflow: counter gradient transport and deep learning modelling". In: *Journal of Fluid Mechanics* 906 (Nov. 2020). DOI: 10.1017/jfm.2020.820.

[109] Aashwin Ananda Mishra, Jayant Mukhopadhaya, Gianluca Iaccarino, and Juan Alonso. "Uncertainty Estimation Module for Turbulence Model Predictions in SU2". In: *AIAA Journal* 57.3 (Mar. 2019), pp. 1066–1077. DOI: 10.2514/1.j057187.

[110] Paritosh Mokhasi and Dietmar Rempfer. "Optimized sensor placement for urban flow measurement". In: *Physics of Fluids* 16.5 (May 2004), pp. 1758–1764. DOI: 10.1063/1.1689351.

[111] Vincent Mons, Jean-Camille Chassaing, and Pierre Sagaut. "Optimal sensor placement for variational data assimilation of unsteady flows past a rotationally oscillating cylinder". In: *Journal of Fluid Mechanics* 823 (June 2017), pp. 230–277. DOI: 10.1017/jfm.2017.313.

[112] Vincent Mons and Olivier Marquet. "Linear and nonlinear sensor placement strategies for mean-flow reconstruction via data assimilation". In: *Journal of Fluid Mechanics* 923 (July 2021). DOI: 10.1017/jfm.2021.488.

[113] J.-D. Müller and P. Cusdin. "On the performance of discrete adjoint CFD codes using automatic differentiation". In: *International Journal for Numerical Methods in Fluids* 47.8–9 (Jan. 2005), pp. 939–945. ISSN: 1097-0363. DOI: 10.1002/fld.885.

[114] Kumi Nakai, Keigo Yamada, Takayuki Nagata, Yuji Saito, and Taku Nonomura. "Effect of Objective Function on Data-Driven Greedy Sparse Sensor Optimization". In: *IEEE Access* 9 (2021), pp. 46731–46743. DOI: 10.1109/access.2021.3067712.

[115] Uwe Naumann. *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation*. Society for Industrial and Applied Mathematics, Jan. 2011. ISBN: 9781611972078. DOI: 10.1137/1.9781611972078.

[116] R. Nichols. "Applications of RANS/LES Turbulence Models". In: *41st Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, Jan. 2003. DOI: 10.2514/6.2003-83.

[117] B. R. Noack, D. Fernex, and R. Semaan. "Machine Learning for Reduced-Order Modeling". In: *Data-Driven Fluid Mechanics*. Cambridge University Press, Jan. 2023, pp. 287–303. ISBN: 9781108842143. DOI: 10.1017/9781108896214.021.

[118] Todd A Oliver and Robert D Moser. "Bayesian uncertainty quantification applied to RANS turbulence models". In: *Journal of Physics: Conference Series* 318.4 (Dec. 2011), p. 042032. DOI: 10.1088/1742-6596/318/4/042032.

[119] Eric J. Parish and Karthik Duraisamy. "A paradigm for data-driven predictive modeling using field inversion and machine learning". In: *Journal of Computational Physics* 305 (Jan. 2016), pp. 758–774. DOI: 10.1016/j.jcp.2015.11.012.

[120] S. B. Pope. "A more general effective-viscosity hypothesis". In: *Journal of Fluid Mechanics* 72.02 (Nov. 1975), p. 331. DOI: 10.1017/s0022112075003382.

[121] Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, Aug. 2000. DOI: 10.1017/cbo9780511840531.

[122] Mathieu Le Provost, Wei Hou, and Jeff Eldredge. "Deep learning and data assimilation approaches to sensor reduction in estimation of disturbed separated flows". In: *AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2020. DOI: 10.2514/6.2020-0799.

[123] Thomas Pulliam, Marian Nemec, Terry Holst, and David Zingg. "Comparison of evolutionary (genetic) algorithm and adjoint methods for multi-objective viscous airfoil optimizations". In: *41st Aerospace sciences meeting and exhibit*. 2003, p. 298.

[124] Maurizio Quadrio, Alessandro Chiarini, Jacopo Banchetti, Davide Gatti, Antonio Memmolo, and Sergio Pirozzoli. "Drag reduction on a transonic airfoil". In: *Journal of Fluid Mechanics* 942 (May 2022). ISSN: 1469-7645. DOI: 10.1017/jfm.2022.369.

[125] Jean Rabault, Feng Ren, Wei Zhang, Hui Tang, and Hui Xu. "Deep reinforcement learning in fluid mechanics: A promising method for both active flow control and shape optimization". In: *Journal of Hydrodynamics* 32.2 (Apr. 2020), pp. 234–246. DOI: 10.1007/s42241-020-0028-y.

[126] M. Raissi, P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (Feb. 2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045.

[127] Ch. Rapp and M. Manhart. "Flow over periodic hills: an experimental study". In: *Experiments in Fluids* 51.1 (Feb. 2011), pp. 247–269. ISSN: 1432-1114. DOI: 10.1007/s00348-011-1045-y.

[128] Osborne Reynolds. "IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion". In: *Philosophical Transactions of the Royal Society of London. (A.)* 186 (Dec. 1895), pp. 123–164. DOI: 10.1098/rsta.1895.0004.

[129] E. Robertson, V. Choudhury, S. Bhushan, and D. Walters. "Validation of OpenFOAM numerical methods and turbulence models for incompressible bluff body flows". In: *Computers &; Fluids* 123 (2015), pp. 122–145.

[130] Samuel Rudy, Alessandro Alla, Steven L. Brunton, and J. Nathan Kutz. "Data-Driven Identification of Parametric Partial Differential Equations". In: *SIAM Journal on Applied Dynamical Systems* 18.2 (Jan. 2019), pp. 643–660. DOI: 10.1137/18m1191944.

[131] Christopher L. Rumsey. "Recent Developments on the Turbulence Modeling Resource Website (Invited)". In: *22nd AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2015. DOI: 10.2514/6.2015-2927.

[132] Christopher L. Rumsey, Gary N. Coleman, and Li Wang. "In Search of Data-Driven Improvements to RANS Models Applied to Separated Flows". In: *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2022. DOI: 10.2514/6.2022-0937.

[133] Christopher L. Rumsey, Jeffrey P. Slotnick, and Anthony J. Sclafani. "Overview and Summary of the Third AIAA High Lift Prediction Workshop". In: *2018 AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, Jan. 2018. DOI: 10.2514/6.2018-1258.

[134] Benjamin Sanderse, Panos Stinis, Romit Maulik, and Shady E. Ahmed. *Scientific machine learning for closure models in multiscale problems: a review*. 2024. arXiv: 2403.02913 [math.NA].

[135] Martin Schmelzer, Richard P. Dwight, and Paola Cinnella. "Discovery of Algebraic Reynolds-Stress Models Using Sparse Symbolic Regression". In: *Flow, Turbulence and Combustion* 104.2-3 (Dec. 2019), pp. 579–603. DOI: 10.1007/s10494-019-00089-x.

[136] Mikhail L. Shur, Philippe R. Spalart, Mikhail Kh. Strelets, and Andrey K. Travin. "A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities". In: *International Journal of Heat and Fluid Flow* 29.6 (Dec. 2008), pp. 1638–1649. DOI: 10.1016/j.ijheatfluidflow.2008.07.001.

[137] Anand Pratap Singh. "A framework to improve turbulence models using full-field inversion and machine learning". PhD thesis. 2018.

[138] Anand Pratap Singh and Karthik Duraisamy. "Using field inversion to quantify functional errors in turbulence closures". In: *Physics of Fluids* 28.4 (Apr. 2016), p. 045110. DOI: 10.1063/1.4947045.

[139] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. "Machine-Learning-Augmented Predictive Modeling of Turbulent Separated Flows over Airfoils". In: *AIAA Journal* 55.7 (July 2017), pp. 2215–2227. DOI: 10.2514/1.j055595.

[140] S.N. Skinner and H. Zare-Behtash. "State-of-the-art in aerodynamic shape optimisation methods". In: *Applied Soft Computing* 62 (Jan. 2018), pp. 933–962. DOI: 10.1016/j.asoc.2017.09.030.

[141] Jeffrey Slotnick, Abdollah Khodadoust, Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, and Dimitri Mavriplis. "CFD vision 2030 study: a path to revolutionary computational aerosciences". In: *NASA Langley Research Centre* (2014). NASA/CR–2014-218178.

[142] D M Somers. *Design and experimental results for the S809 airfoil*. Tech. rep. Jan. 1997. DOI: 10.2172/437668.

[143] P. Spalart and S. Allmaras. "A one-equation turbulence model for aerodynamic flows". In: *30th Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, Jan. 1992. DOI: 10.2514/6.1992-439.

[144] P. R. Spalart and V. Venkatakrishnan. "On the role and challenges of CFD in the aerospace industry". In: *The Aeronautical Journal* 120.1223 (Jan. 2016), pp. 209–232. DOI: 10.1017/aer.2015.10.

[145] P.R. Spalart. "Strategies for turbulence modelling and simulations". In: *Engineering Turbulence Modelling and Experiments 4*. Elsevier, 1999, pp. 3–17. ISBN: 9780080433288. DOI: 10.1016/b978-008043328-8/50001-1.

[146] Philippe R. Spalart. "Detached-Eddy Simulation". In: *Annual Review of Fluid Mechanics* 41.1 (Jan. 2009), pp. 181–202. DOI: 10.1146/annurev.fluid.010908.165130.

[147] Vishal Srivastava, Christopher L. Rumsey, Gary N. Coleman, and Li Wang. "On Generalizably Improving RANS Predictions of Flow Separation and Reattachment". In: *AIAA SCITECH 2024 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2024. DOI: 10.2514/6.2024-2520.

[148] W.-D. Su, H. Zhao, Q.-D. Cai, A.-K. Xiong, and J.-Z. Wu. "A diagnosis of linear eddy-viscosity in turbulence modeling". In: *Physics of Fluids* 14.3 (Mar. 2002), pp. 1284–1287. DOI: 10.1063/1.1436495.

[149] Salar Taghizadeh, Freddie D Witherden, and Sharath S Girimaji. "Turbulence closure modeling with data-driven techniques: physical compatibility and consistency considerations". In: *New Journal of Physics* 22.9 (Sept. 2020), p. 093023. DOI: 10.1088/1367-2630/abadb3.

[150] Kunihiko Taira, Maziar S. Hemati, Steven L. Brunton, Yiyang Sun, Karthik Duraisamy, Shervin Bagheri, Scott T. M. Dawson, and Chi-An Yeh. "Modal Analysis of Fluid Flows: Applications and Outlook". In: *AIAA Journal* 58.3 (Mar. 2020), pp. 998–1022. DOI: 10.2514/1.j058462.

[151] Edward N. Tinoco, Olaf P. Brodersen, Stefan Keye, Kelly R. Laflin, John C. Vassberg, Ben Rider, Richard A. Wahls, Joseph H. Morrsion, and David Hue Pomeroy et al. "Summary Data from the Seventh AIAA CFD Drag Prediction Workshop". In: *AIAA AVIATION 2023 Forum*. American Institute of Aeronautics and Astronautics, June 2023. DOI: 10.2514/6.2023-3492.

[152] Brendan Tracey, Karthik Duraisamy, and Juan Alonso. "Application of Supervised Learning to Quantify Uncertainties in Turbulence and Combustion Modeling". In: *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, Jan. 2013. DOI: 10.2514/6.2013-259.

[153] A. Travin, M. Shur, M. Spalart Strelets, P. Physical, and Numerical Upgrades. "Detached-Eddy Simulation of Complex Turbulent Flows". In: *Advances In LES Of Complex Flows* (2002), pp. 239–254.

[154] Ali Uzun and Mujeeb R. Malik. "Large-Eddy Simulation of Flow over a Wall-Mounted Hump with Separation and Reattachment". In: *AIAA Journal* 56.2 (Feb. 2018), pp. 715–730. DOI: 10.2514/1.j056397.

[155] J. Viquerat, P. Meliga, A. Larcher, and E. Hachem. "A review on deep reinforcement learning for fluid mechanics: An update". In: *Physics of Fluids* 34.11 (Nov. 2022). ISSN: 1089-7666. DOI: 10.1063/5.0128446.

[156] Pedro Stefanin Volpiani, Morten Meyer, Lucas Franceschini, Julien Dandois, Florent Renac, Emeric Martin, Olivier Marquet, and Denis Sipp. "Machine learning-augmented turbulence modeling for RANS simulations of massively separated flows". In: *Physical Review Fluids* 6.6 (June 2021). DOI: 10.1103/physrevfluids.6.064607.

[157] Aslihan Vuruskan and Serhat Hosder. "Impact of Turbulence Models and Shape Parameterization on Robust Aerodynamic Shape Optimization". In: *Journal of Aircraft* 56.3 (May 2019), pp. 1099–1115. DOI: 10.2514/1.c035039.

[158] Andreas Wächter and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear program-

ming". In: *Mathematical Programming* 106.1 (Apr. 2005), pp. 25–57. ISSN: 1436-4646. DOI: 10.1007/s10107-004-0559-y.

[159] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data". In: *Physical Review Fluids* 2.3 (Mar. 2017). DOI: 10.1103/physrevfluids.2.034603.

[160] Fabian Waschkowski, Yaomin Zhao, Richard Sandberg, and Joseph Klewicki. "Multi-objective CFD-driven development of coupled turbulence closure models". In: *Journal of Computational Physics* 452 (Mar. 2022), p. 110922. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2021.110922.

[161] J. Weatheritt and R.D. Sandberg. "The development of algebraic stress models using a novel evolutionary algorithm". In: *International Journal of Heat and Fluid Flow* 68 (Dec. 2017), pp. 298–318. DOI: 10.1016/j.ijheatfluidflow.2017.09.017.

[162] David Wilcox. In: *29th Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, Jan. 1991. DOI: 10.2514/6.1991-615.

[163] David C Wilcox. *Turbulence modeling for CFD*. Vol. 2. DCW industries La Canada, CA, 1998.

[164] David C. Wilcox. "Reassessment of the scale-determining equation for advanced turbulence models". In: *AIAA Journal* 26.11 (Nov. 1988), pp. 1299–1310. ISSN: 1533-385X. DOI: 10.2514/3.10041.

[165] Chenyu Wu and Yufei Zhang. "Enhancing the shear-stress-transport turbulence model with symbolic regression: A generalizable and interpretable data-driven approach". In: *Physical Review Fluids* 8.8 (Aug. 2023). DOI: 10.1103/physrevfluids.8.084604.

[166] Jin-Long Wu, Heng Xiao, and Eric Paterson. "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework". In: *Physical Review Fluids* 3.7 (July 2018). DOI: 10.1103/physrevfluids.3.074602.

[167] Jinlong Wu, Heng Xiao, Rui Sun, and Qiqi Wang. "Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned". In: *Journal of Fluid Mechanics* 869 (Apr. 2019), pp. 553–586. DOI: 10.1017/jfm.2019.205.

[168] Neil Wu, Gaetan Kenway, Charles A. Mader, John Jasa, and Joaquim R. R. A. Martins. "pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems". In: *Journal of Open Source Software* 5.54 (2020), p. 2564. DOI: 10.21105/joss.02564.

[169] H. Xiao, J.-L. Wu, J.-X. Wang, R. Sun, and C.J. Roy. "Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier–Stokes simulations: A data-driven, physics-informed Bayesian approach". In: *Journal of Computational Physics* 324 (Nov. 2016), pp. 115–136. DOI: 10.1016/j.jcp.2016.07.038.

[170] Heng Xiao and Paola Cinnella. "Quantification of model uncertainty in RANS simulations: A review". In: *Progress in Aerospace Sciences* 108 (July 2019), pp. 1–31. DOI: 10.1016/j.paerosci.2018.10.001.

[171] Heng Xiao, Jin-Long Wu, Sylvain Laizet, and Lian Duan. "Flows over periodic hills of parameterized geometries: A dataset for data-driven turbulence modeling from direct simulations". In: *Computers & Fluids* 200 (Mar. 2020), p. 104431. DOI: 10.1016/j.compfluid.2020.104431.

[172] Muchen Yang and Zhixiang Xiao. "Improving the k—–Ar transition model by the field inversion and machine learning framework". In: *Physics of Fluids* 32.6 (June 2020). ISSN: 1089-7666. DOI: 10.1063/5.0008493.

[173] Chen Yi, Denggao Tang, Fanzhi Zeng, Yao Li, and Chao Yan. "Improvement of the algebraic stress model for separated flows based on field inversion and machine learning". In: *Physics of Fluids* 35.11 (Nov. 2023). ISSN: 1089-7666. DOI: 10.1063/5.0174257.

[174] B. Yildirim, C. Chryssostomidis, and G.E. Karniadakis. "Efficient sensor placement for ocean measurements using low-dimensional concepts". In: *Ocean Modelling* 27.3-4 (Jan. 2009), pp. 160–173. DOI: 10.1016/j.ocemod.2009.01.001.

[175] Xin-Lei Zhang, Heng Xiao, Guo-Wei He, and Shi-Zhao Wang. "Assimilation of disparate data for enhanced reconstruction of turbulent mean flows". In: *Computers & Fluids* 224 (June 2021), p. 104962. DOI: 10.1016/j.compfluid.2021.104962.

[176] Yaomin Zhao, Harshal D. Akolekar, Jack Weatheritt, Vittorio Michelassi, and Richard D. Sandberg. "RANS turbulence model development using CFD-driven machine learning". In: *Journal of Computational Physics* 411 (June 2020), p. 109413. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109413.