PASHA PIROOZMAND

SPARSE DATA-DRIVEN SIMULATION OF TURBULENT FLOWS

DISS. ETH NO. 28843

PASHA PIROOZMAND

SPARSE DATA-DRIVEN SIMULATION OF TURBULENT FLOWS

DISS. ETH NO. 28843

SPARSE DATA-DRIVEN SIMULATION OF TURBULENT FLOWS

A dissertation submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

PASHA PIROOZMAND
MSc, Sharif University of Technology

born on 23 July 1990
citizen of Iran

accepted on the recommendation of

Prof. Dr. Patrick Jenny, examiner
Prof. Dr. Siddhartha Mishra, co-examiner
Prof. Dr. Heng Xiao, co-examiner

2022

Pasha Piroozmand: *Sparse Data-driven Simulation of Turbulent Flows*, © 2022

DOI: 10.3929/ethz-b-000583380

To Amir Ashrafi and other 175 victims of flight PS752

ABSTRACT

Data has always played an important role in the modeling and simulation of turbulent flows. In the last decade, with the advancement and abundance of measurement techniques, computational resources, and mathematical algorithms, data has been systematically used to develop and improve turbulence models. Experimental measurements, which are generally sparse or of low resolution, can be incorporated into models using mathematical techniques such as variational data assimilation to enable data-driven simulation of turbulent flows. In this project, we investigate different techniques to incorporate sparse data into the turbulence models to obtain the most accurate results possible. Such inverse problems are typically severely ill-posed and the solutions are non-unique. Therefore, in addition to sparse measurements, further information, expert knowledge, or physical constraints should be added to the problem to obtain a reconstructed solution that is smooth, accurate, and physical. We, therefore, impose the linear eddy viscosity (LEV) assumption to reduce the degrees of freedom of the problem and the constraint of positivity of the eddy viscosity. The optimization problem is defined as reducing the discrepancy between the LEV RANS model's solution and the sparse measurement data by tuning a corrective parameter field that results in an optimal eddy viscosity field. The discrete adjoint method is implemented in OpenFOAM to compute the gradients. The case study of flow over periodic hills is chosen for the investigation.

We see that the LEV assumption without including further information would lead to irregular, jagged velocity profiles, due to the ill-conditioning and non-uniqueness nature of the problem. Regularization is then introduced to promote regularity and smoothness to the solution. The L_2, total variation, and Sobolev gradient regularization methods were employed. We find that physical velocity profiles can be reconstructed using these methods. However, we observe that with each of these regularization methods we cannot accurately reconstruct wall shear stresses, which are important quantities of interest, even if sparse wall shear stress measurement data is assimilated. Therefore, we propose a method called piecewise linear dimension reduction (PLDR). In this method, the parameter field is constrained to be piecewise linear to avoid, in a controlled way, any noise and large fluctuations in the gradients. The results suggest that the PLDR method

can provide accurate and smooth velocity profiles as well as wall shear stresses. Next, we investigate a scenario where only sparse wall shear stress measurements are available. We observe that incomplete and not uniformly distributed data cannot be assimilated properly resulting in deteriorated solutions in the free shear flow region. Therefore, we augmented the available data by using the solution of a high-fidelity but computationally efficient model. A loosely coupled hybrid LES/RANS method is proposed, in which an under-resolved LES is coupled with a steady-RANS model. We show that such a method is easy to implement and that it provides an accurate velocity field with less accurate wall shear stresses. The solution of the proposed hybrid LES/RANS method in combination with the sparse wall shear stress measurements was successfully assimilated into a RANS model providing both accurate velocities and wall shear stresses. We can conclude that accurate sparse data-driven simulation is possible if the inverse problem is properly defined such that any available knowledge is incorporated, from adding data from a higher fidelity simulation, constraining the Reynolds stress tensor, and imposing a hard constraint on the positivity of the eddy viscosity, to penalizing noise and non-smoothness, etc. The project thus shows a promising path towards using sparse data-driven simulation for practical applications.

ZUSAMMENFASSUNG

Daten haben schon immer eine wichtige Rolle bei der Modellierung und Simulation turbulenter Strömungen gespielt. In den letzten zehn Jahren wurden mit dem Fortschritt und der zunehmenden Menge an Messverfahren, Rechenressourcen und mathematischen Algorithmen systematisch Daten zur Entwicklung und Verbesserung von Turbulenzmodellen verwendet. Experimentelle Messungen, die im Allgemeinen spärlich sind oder eine geringe Auflösung haben, können mit Hilfe mathematischer Techniken wie der Datenassimilation durch das Variationsverfahren in Modelle integriert werden, um eine datengesteuerte Simulation turbulenter Strömungen zu ermöglichen. In diesem Projekt untersuchen wir verschiedene Techniken zur Einbeziehung spärlicher Daten in die Turbulenzmodelle, um möglichst genaue Ergebnisse zu erzielen. Solche inversen Probleme sind in der Regel sehr schlecht gestellt und die Lösungen sind nicht eindeutig. Daher sollten zusätzlich zu den spärlichen Messungen weitere Informationen, Expertenwissen oder physikalische Randbedingungen in das Problem einfliessen, um eine glatte, genaue und physikalische Lösung zu rekonstruieren. Wir setzen daher die Annahme der linearen Wirbelviskosität (LEV) voraus, um die Freiheitsgrade des Problems und die Einschränkung der Positivität der Wirbelviskosität zu reduzieren. Das Optimierungsproblem besteht darin, die Diskrepanz zwischen der Lösung des LEV-RANS-Modells und den spärlichen Messdaten zu verringern, indem ein korrigierendes Parameterfeld eingestellt wird, das zu einem optimalen Wirbelviskositätsfeld führt. Die diskrete adjungierte Methode ist in OpenFOAM implementiert, um die Gradienten zu berechnen. Für die Untersuchung wird als Fallbeispiel die Strömung über periodische Hügel gewählt.

Es zeigt sich, dass die LEV-Annahme ohne Einbeziehung weiterer Informationen zu unregelmäßigen, gezackten Geschwindigkeitsprofilen führen würde, was auf die schlechte Konditionierung und die Nicht-Eindeutigkeit des Problems zurückzuführen ist. Die Regularisierung wird eingeführt, um die Regelmäßigkeit und Glätte der Lösung zu fördern. Im Speziellen wurden die Methoden L_2, Totalvariation und Sobolev-Gradient-Regularisierung eingesetzt. Dabei hat sich gezeigt, dass die physikalischen Geschwindigkeitsprofile mit diesen Methoden rekonstruiert werden können. Jedoch werden mit jeder dieser Regularisierungsmethoden die Wandschubspannungen, die von besonderem Interesse sind, nicht genau rekonstruiert,

selbst wenn spärliche Wandschubspannungsmessdaten assimiliert werden. Daher schlagen wir eine Methode namens Piecewise Linear Dimension Reduction (PLDR) vor. Bei dieser Methode muss das Parameterfeld stückweise linear sein, um Rauschen und große Schwankungen in den Gradienten kontrolliert zu vermeiden. Die Ergebnisse zeigen, dass die PLDR-Methode genaue und glatte Geschwindigkeitsprofile sowie Wandschubspannungen liefern kann. Als Nächstes untersuchen wir ein Szenario, bei dem nur spärliche Wandschubspannungsmessungen verfügbar sind. Wir stellen fest, dass unvollständige und nicht gleichmäßig verteilte Daten nicht richtig assimiliert werden können, was zu schlechteren Lösungen im Bereich der freien Scherströmung führt. Daher haben wir die verfügbaren Daten durch die Lösung eines sehr realitätsnahen, aber rechnerisch effizienten Modells ergänzt. Es wird eine hybride LES/RANS-Methode vorgeschlagen, bei der eine unteraufgelöste LES mit einem stationären RANS-Modell lose gekoppelt wird. Wir zeigen, dass eine solche Methode einfach zu implementieren ist und, dass sie ein genaues Geschwindigkeitsfeld mit weniger genauen Wandschubspannungen liefert. Die Lösung der vorgeschlagenen hybriden LES/RANS-Methode in Kombination mit den spärlichen Wandschubspannungsmessungen wurde erfolgreich in ein RANS-Modell integriert, das sowohl genaue Geschwindigkeiten als auch Wandschubspannungen liefert. Wir können schlussfolgern, dass eine genaue datengetriebene Simulation möglich ist, wenn das inverse Problem richtig definiert ist, so dass jedes verfügbare Wissen einbezogen wird, angefangen von der Hinzufügung von Daten aus einer Simulation mit höherer Genauigkeit, der Beschränkung des Reynolds-Spannungstensors und der Auferlegung einer harten Beschränkung für die Positivität der Wirbelviskosität bis hin zur Bestrafung von Rauschen und Nicht-Glätte, usw. Das Projekt zeigt somit einen vielversprechenden Weg zur Nutzung der datengesteuerten Simulation für praktische Anwendungen auf.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Prof. Patrick Jenny. During my four years on this project, I received excellent scientific support from him. Many ideas originated from our brainstorming in meetings and Prof. Jenny's vision always helped me take appropriate steps. His patience and understanding provided healthy working conditions for me to experiment with ideas and make try-and-error tests. He is an excellent example of an optimal supervisor who is kind and rational to deal with and scientifically expert in his field.

Second, I want to thank my colleague Oliver Brenner. We worked together on this topic. His solid programming and mathematical skills greatly improved our work. Most of the in-depth understandings came out of our discussions. His kind support during my difficult days amid the Ph.D. is greatly appreciated. His discipline was an inspiration for me and he practically showed me what a Swiss quality performance means.

Third, I want to thank Mahshid, my wife, for all her love, support, and patience. Without her, I could not carry on and finished this Ph.D. I am very lucky to have her in my life.

I want to thank Prof. Togni who was the vice-rector of ETHZ. His kind support enabled me to move to the IFD lab. He is one of the most amazing people I have ever seen and without his support this Ph.D. would have been nonexistent.

I thank Prof. Mishra for accepting to be my co-examiner and co-supervisor. I enjoyed our scientific discussions and appreciate his time reading and evaluating my thesis. I also thank Prof. Xiao, for being my co-examiner and for his insightful comments on the thesis. I particularly thank him for sharing his code on the hybrid method which greatly accelerated my project.

I thank my family. I thank my mother and father for growing me up and for their great care and sacrifice. I owe them for whatever I have achieved. I thank Pourya, my brother, for being my first role model. His kindness and support have always been with me. I thank Parya, my sister, for being a source of love and care. Their support and encouragement made me believe in myself and have the courage to study abroad.

I thank my "second family", Mr. and Mrs. Motie for being so supportive and kind toward me. I thank Mehrshad and Shiva for their kind support and for being real friends. I am very lucky to have you as my family member.

I would like to thank our group members. I thank Robert for his mental support, Justin for our fruitful discussions and the German translation of the abstract, Daniel, for always being kind to me, Ranit and Sun, for our cultural proximity, Michael and Giulia for our fun discussions, and all other members: Hossein, Valentin, Kristof, Heinrich, Lukas, Nemanja, Rajdeep, Franca, Stephan, Philip. I also thank my former group members known as "Mud Crabs". We made it together. Which can happen! Thanks, Jiggar, Christina, Lento, Laure, Hamed, Giorgia, Ishaan, Gianluca, Nilesh, Giovanni.

I thank my students Francesca, Fabiano, Coleman, and Severin who have done their projects with me. I learned a lot from them and hope that they also learned something from their projects.

I thank Dr. Daniel Meyer for letting me assist him in the course computational methods, which significantly improved my CFD knowledge.

I thank all IFD members for being such nice colleagues to me. I thank Bianca and Maria for their great and kind administrative support.

I thank ETH and Empa board for their great support, especially Prof. Bona, Prof. Springman, and Dr. Richner. I thank Prof. Carmeliet and Dr. Jonas Allegrini. I also thank the ETH foundation for their financial support.

Finally, I want to thank all my Iranian friends in Switzerland, Iran, and all around the world whose friendship is the most joyful thing in my life: Mahyar, Amirreza, Danial, Nima, Mohaddeseh, Shokoofeh, Sina, Ehsan, Fereshteh, Abdi, Hossein, Mohammad and Rezvan, Erfan, Maryam, Ahmad, Shima, Karim, Saeed, Ardalan, Sarah, Manouchehr, Elias, Parisa, Mohammad and Fatemeh, Ehsan Akrami, Elian, Tania, Omid, Ali ...

The picture of the cover of the thesis was designed by Vitaliiwoo1X - Freepik.

CONTENTS

List of Figures xv
List of Tables xix
List of Symbols xx
1 INTRODUCTION 1
1.1 Governing equations of fluid flow 1
1.2 Turbulence and Turbulent flows 3
1.3 Why study turbulent flows? 5
1.4 How study turbulent flows? 6
1.5 Turbulence Modeling 7
1.5.1 Reynolds-averaged Navier Stokes (RANS) 8
1.5.2 Conclusions on RANS models 15
1.5.3 Large eddy simulation (LES) 16
1.5.4 Conclusions on LES 17
1.5.5 Hybrid LES/RANS models and wall-modeled LES 18
1.5.6 Conclusions on turbulence modeling 21
1.6 Data and Turbulence Modeling 21
1.6.1 Data-driven turbulence modeling 23
1.6.2 Sparse data-driven simulation 25
1.7 Motivation 35
1.8 Outline 36
2 SPARSE DATA-DRIVEN RANS SIMULATION 39
2.1 Formulation of the forward problem 39
2.2 Formulation of the inverse problem 41
2.3 Sensitivity calculation: discrete adjoint method 41
2.4 Implementation 44
2.4.1 Implementation of the gradients computations in OpenFOAM 45
2.4.2 Implementation of the optimization procedure in Python 46
2.5 Optimization 46
2.5.1 Computation of the learning rate 46
2.5.2 Positivity constraint of the eddy viscosity 48
2.6 Case setup 48
2.7 Verification of the gradient computations and data assimilation 49
2.8 Assimilation of sparse data: results without regularization 54

3 REGULARIZATION 59
3.1 Conventional Regularization Methods 59
3.1.1 L_2 regularization 59
3.1.2 Total variation regularization 60
3.1.3 Sobolev gradient regularization 60
3.2 Piecewise Linear Dimension Reduction 61
3.2.1 Verification of contraction and expansion 63
3.3 Regularization Effect on a Single Optimization Step 65
3.4 Results and discussion 66
3.4.1 PLDR 66
3.4.2 Improved PLDR 67
3.4.3 Comparison with other regularization methods 68
3.5 Conclusions 71
4 SPARSE WALL DATA-DRIVEN HYBRID LES/RANS SIMULATION 75
4.1 Summary 75
4.2 Introduction 75
4.2.1 Data-driven RANS simulation 76
4.2.2 Wall-modeled LES and hybrid LES/RANS models 77
4.3 Methods 79
4.3.1 Tightly coupled dual-mesh hybrid LES/RANS 79
4.3.2 Loosely coupled dual-mesh hybrid LES/RANS method 82
4.3.3 Data assimilation 83
4.3.4 Case setup 85
4.4 Results and Discussion 87
4.5 Conclusions and Outlook 93
5 SUMMARY AND OUTLOOK 95

BIBLIOGRAPHY 99

LIST OF FIGURES

Figure 1.1 The Barycentric triangle [17]. It defines all realizable states of the Reynolds stress anisotropy. Any position in the Barycentric triangle represents a realizable anisotropy state of the Reynolds stress. The vertices of the triangle correspond to the limiting one, two, and three-component (isotropic) states 10

Figure 1.2 Turbulent boundary layer and the law of the wall. The plot is in semi-log format. Adapted from [40] 19

Figure 2.1 Periodic hills geometry with boundary conditions. All length scales are expressed with respect to the hill height H. 49

Figure 2.2 The gradients computed by FD for α_i in the cells i are marked by black crosses (bottom), which are compared with the gradients computed by the approximate adjoint method. Results are shown for the case of the periodic hills with full synthetic reference data for both velocity components with the initial value of unity for α and without any regularization. The figure is taken from reference [119]. 50

Figure 2.3 The gradients of all cells computed by FD compared with the gradients computed by the approximate adjoint method. Results are shown for the case of the periodic hills with full synthetic reference data for both velocity components with the initial value of unity for α and without any regularization [119]. 51

Figure 2.4 Process of using synthetic reference data to check the inverse solver performance [119]. 52

Figure 2.5 Comparison of the prescribed, synthetic parameter field α^{ref} (top) and the reconstructed field α obtained by the optimization process (bottom; data range [0.21, 1.95]). The full U_x reference field was used and no regularization was applied. Taken from reference [119]. 53

Figure 2.6 The synthetic reference U_x-profiles (black, dashed), initial (green, dotted), and optimized (red, solid) data. Length scales are normalized by the hill-crest height H and the velocity by the bulk velocity U_b above the hillcrest. Taken from reference [119]. 54

Figure 2.7 Distributions of measurement data (black dots) and testing data (gray crosses). Testing data are not used for data assimilation, but to estimate the generalization error. 55

Figure 2.8 Horizontal velocity component u_1 (top) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot). Results for the baseline $k-\varepsilon$ model (green), for the unregularized data-driven method (red), and for the reference LES data (black) are shown. 56

Figure 2.9 The evolution of the cost function during the optimization at the reference points (red curve) and the test points (blue curve) 57

Figure 3.1 The basis functions (blue, green, and red polyhedrons) span the solution inside the dashed yellow triangle. The locations of measurement points are depicted by black dots, and a few elements are shown by white triangles. 62

Figure 3.2 Flow chart of the data assimilation procedure using the PLDR method. 63

Figure 3.3 Verification of the dimensionality reduction process. 64

Figure 3.4 Sensitivity $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ contours after an intermediate optimization step without (a) and with (b-e) regularization. For PLDR, the sensitivities are obtained by projecting $\frac{\mathrm{d}f}{\mathrm{d}\beta}$ using the projection matrix. 65

Figure 3.5 Velocity profiles (top plot) and wall shear stress (friction coefficient) at the lower wall (bottom plot) using the PLDR method. 67

Figure 3.6 Velocity profiles (top) and wall shear stress (friction coefficient C_f) at the lower wall (bottom) obtained with the improved PLDR method. 69

Figure 3.7 The cost function evolution of DA with TV regularization during the optimization iterations. The red curve corresponds to the cost function at reference points and the blue curve is for the test points. 70
Figure 3.8 The comparison of stream-wise velocity profiles u_1 (left) and the friction coefficients C_f (right) for different regularization methods. 72
Figure 3.9 Optimal α fields without (a) and with (b-e) regularization. The same color bar scaling is used for all plots. 73
Figure 4.1 Periodic hills geometry and the predefined LES and RANS regions. All length scales are expressed with respect to the hill height H. 87
Figure 4.2 The location of sampling points is shown as crosses ($\times$) and the location of measurement points at the walls is shown by black circles ($\bullet$). 88
Figure 4.3 Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for the standalone cases of steady RANS (S-RANS) and underresolved LES (UR-LES). 89
Figure 4.4 Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for the tightly coupled LES/RANS model. Note that the friction coefficient of the tightly coupled model is derived from its U-RANS solution. 90
Figure 4.5 Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for the loosely coupled LES/RANS model. Note that the friction coefficient of the loosely coupled model is derived from its S-RANS solution. 91
Figure 4.6 Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for data assimilation of sparse wall shear stress data on S-RANS. 92

Figure 4.7 Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for data assimilation of sparse wall shear stress and the solution of the loosely coupled model on S-RANS. 94

LIST OF TABLES

Table 3.1 Hyperparameter values for regularization and optimization methods. 66

Table 3.2 Summary of the errors for different regularization methods. Training error is the final cost function value f normalized by the number of measurement points. Generalization error is the final cost function value at testing points (f_{test}) normalized by the number of testing points. The C_f training error is the mean absolute error of C_f at wall-adjacent measurement points for the bottom wall. Drag error (Skin friction drag error) is the relative error with respect to the skin friction drag of LES data for the bottom wall. 74

LIST OF SYMBOLS

Abbreviations

Re	Reynolds number
AD	automatic differentiation
ARSM	algebraic Reynolds stress model
CFD	computational fluid dynamics
DA	data assimilation
DNS	direct numerical simulation
EKF	extended Kalman Filter
EnKF	Ensamble Kalman Filter
EWA	exponentially weighted averaged
FD	finite difference
KF	Kalman Filter
LES	large eddy simulation
LEV	linear eddy viscosity
LHS	the left-hand side of an equation
ML	machine learning
NLEV	non-linear eddy viscosity
PDE	partial differential equation
PDF	probability density functions
PINN	physics-informed neural network
PIV	particle image velocimetry
PLDR	piecewise linear dimension reduction

RANS Reynolds-averaged Navier-Stokes

RHS the right-hand side of an equation

ROM reduced order modeling

RSM Reynolds stress model

RSTM Reynolds stress transport model

SIMPLE Semi-Implicit Method for Pressure Linked Equations

TKE turbulent kinetic energy

TV total variation

Greek letters

α corrective parameter

α' intermediatory corrective parameter

$\boldsymbol{\beta}$ reduced dimension corrective parameter

$\boldsymbol{\Phi}$ piecewise linear projection matrix

Δ learning rate or the optimization step size

δ very small number for DEMON Adam

δ_{ij} Kronecker delta, second-order identity tensor

ϵ diagonal preconditioning matrix in DEMON Adam, perturbation size

η the nominal step size in DEMON Adam

γ DEMON Adam variable

κ the bulk viscosity

κ weight of the decaying momentum in DEMON Adam

κ' discount factor in DEMON Adam

κ_0 the momentum initial value in DEMON Adam

λ Lamés coefficient

λ	adjoint variables
μ	dynamic viscosity
ν	kinematic viscosity
ν_t	turbulent eddy viscosity
ω_{L2}	the regularization hyperparameter of L_2 method
$\omega_{TV,\beta}$	the regularization hyperparameter of TV for β parameter
ω_{TV}	the regularization hyperparameter of TV method
ρ	density
σ	total stress tensor
σ_{HYD}	hydrostatic stress tensor
τ	shear stress tensor
τ_{ij}	Reynolds stress tensor
ε	turbulent dissipation rate

Roman letters

$\hat{\boldsymbol{V}}$	the vector of grid cells' volume
$\mathbf{H}$	Linear observation operator
$\boldsymbol{A}$	block system matrix of the residual equations
$\boldsymbol{b}$	the right hand side of the residual equations
$\boldsymbol{c_w}$	weight factor for wall-adjacent measurement
$\boldsymbol{J}$	matrix of ones
$\boldsymbol{R}$	vector of flow residuals
$\mathcal{B}_m$	the set of neighboring cells of the grid cell m
$\mathcal{B}_r$	the set of neighboring nodes of the measurement point m
$\mathcal{I}$	the identity matrix
C_μ	RANS model coefficient

C_f	skin friction coefficient
f	cost function
H^1	Sobolev space
i, j, k, l, m, o, r, t	indices
l_{sob}	the filtering lengthscale of the Sobolev gradient method
M	number of the parameters
N	the size of the computational grid
p	pressure
p^*	turbulent pressure
R	number of measurement points
t	the iteration number of the optimization
T_{max}	the maximum number of the optimization iterations
U	state variables
U_b	bulk velocity
u_i	velocity
u_i'	velocity fluctuation
x_i	spatial coordinates
y^+	dimensionless wall distance
k	turbulent kinetic energy
S_{ij}	the rate of strain tensor
K	Kalman gain
Q	drift force

1

INTRODUCTION

1.1 GOVERNING EQUATIONS OF FLUID FLOW

The word *fluid* is attributed to any material that continuously deforms in reaction to shear stresses or external forces. The act of continuous deformation can be called *flow* [1]. Applying the principles of the conservation of momentum and the conservation of mass to fluid flow, we can describe the *fluid dynamics* in a mathematical framework. These *governing equations* of fluid flow are called Navier–Stokes (NS) equations. The NS momentum equation is derived from the Cauchy momentum equation, which reads

$$\rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g}, \tag{1.1}$$

where $\boldsymbol{u}$ is the velocity vector, ρ is the density, $\boldsymbol{\sigma}$ is the stress tensor, and $\boldsymbol{g}$ is the body acceleration acting on the continuum. The mass conservation or continuity equation is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0. \tag{1.2}$$

The simple interpretation of the Cauchy momentum equation is that the rate of change of momentum of the fluid (the LHS term) is equal to the surface forces (the first RHS term) and the body forces (the second RHS term) experienced by the fluid. So it is simply a representation of Newton's second law in the Eulerian specification of the flow field[1].

We can decompose the stress tensor $\boldsymbol{\sigma}$ into two components: the hydrostatic stress and the deviatoric stress. The hydrostatic stress is related to volume change. It is equal to the average of the three normal stress components $\sigma_{HYD} = (\sigma_{11} + \sigma_{22} + \sigma_{33})/3$ (for a Cartesian coordinate system). Under any coordinate transformation, the hydrostatic stress remains the same. The negative of the mean normal stress is the pressure $p = -\sigma_{HYD}$. The deviatoric stress causes the shear stresses and is related to shape change.

1 The Eulerian specification of the flow field is a way of looking at fluid motion focusing on specific locations in the space through which the fluid flows at each instant during the motion [2].

It is defined simply as the difference between the pressure and the total stress tensor $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}$ where $\mathbf{I}$ is the identity tensor.

The deviatoric stress is non-zero only if there is a fluid motion. We can derive a relation that links the deviatoric stress to the deformation of the fluid. Here we narrow down the scope of this work by considering only *Newtonian Fluids*. For a Newtonian fluid, there is a linear relationship between the deviatoric stress and the strain rate [3]:

$$\boldsymbol{\tau} = 2\mu\frac{1}{2}[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T] + [(\lambda + \kappa)\nabla\cdot\boldsymbol{u}]\mathbf{I}, \tag{1.3}$$

where μ is the dynamic viscosity, κ is the bulk viscosity, which is negligible in practice, and λ is the Lamés coefficient. Using *Stoke's assumption* implicit in the definition of Newtonian fluids we can assume $\lambda = -\frac{2}{3}\mu$. The expression $\frac{1}{2}[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T]$ is called the pure deformation rate tensor $\mathbf{S}$ and it is indeed the symmetric part of the velocity gradient while the skew-symmetric part is called the rotation tensor [4]. The constitutive equation of Newtonian fluids, which relates stress to strain, is defined as

$$\boldsymbol{\sigma} = -\left(p + \frac{2}{3}\mu\nabla\cdot\boldsymbol{u}\right)\mathbf{I} + 2\mu\mathbf{S}. \tag{1.4}$$

We can now insert equation (1.4) into the Cauchy momentum equation to obtain the NS momentum equation. Before doing so, we further constrain our problem. We assume the flow is *incompressible*, i. e., the density within a parcel of fluid that moves with the flow velocity is always constant. This will reduce the continuity equation to $\nabla\cdot\boldsymbol{u} = 0$ which simplifies the constitutive equation (1.4) as well. In addition, the energy equation is decoupled from the momentum equation and need not be solved to obtain the velocity field. If we neglect the body acceleration due to the gravity (buoyancy force) and other body forces, the final NS equation for incompressible and Newtonian fluid with constant viscosity reads

$$\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\frac{1}{\rho}\nabla p + \nu\,\nabla^2\boldsymbol{u}, \tag{1.5}$$

where $\nu = \mu/\rho$ is the kinematic viscosity. Here we have four state variables and three equations. Since the pressure is decoupled from the energy equation and the equation of state, it cannot be calculated from these equations. In fact, this pressure is not a thermodynamic variable anymore [5]. Instead, the pressure is estimated such that the continuity equation $\nabla\cdot\boldsymbol{u} = 0$ is satisfied.

The analytical solution of the NS system of equations is only feasible for simple geometries and flow configurations. So instead, the PDEs are solved using numerical methods. This branch of fluid mechanics is called *Computational Fluid Dynamics (CFD)*.

1.2 TURBULENCE AND TURBULENT FLOWS

Assuming the characteristic length scale L and the velocity scale U we can non-dimensionalize independent variables (time and space) and dependent variables (velocity and pressure) to get a non-dimensional NS equation [6]

$$\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2\boldsymbol{u}, \tag{1.6}$$

where $\mathrm{Re} = UL/\nu$ is a dimensionless number called *the Reynolds number* and $\boldsymbol{u}$ and p here are scaled velocity and pressure. Re is the ratio of the magnitude of the inertial forces to the viscous forces. The larger the Reynolds number, the less power the dissipative viscous forces would have compared to inertial forces. There are two implications. First, equation (1.6) implies that if Re is the same for two configurations, the scaled velocity and pressure would be the same. This property is called Reynolds-number similarity [6]. Second, Re can be a quantitative measure of the *regime* of the flow. It turns out that correlating with Re the flow can exhibit different "behavior" namely *laminar flows* and *turbulent flows*. We call a flow laminar if the fluid particles follow smooth paths in layers with no or little lateral mixing. Such a flow is more *ordered* with dominant momentum diffusion but low convective momentum transfer. Examples of laminar flows are the smooth flow of a viscous liquid through a tube or pipe, blood flow through capillaries, water flow from a slightly opened tap, fountains, etc. A turbulent flow is characterized by high fluctuations of velocity and pressure. The flow is unsteady, irregular, seemingly random, chaotic, and much more apparently *disordered* [7]. Such chaotic motions arise additional transverse motion of fluid particles that enhances the momentum and energy exchange rate. Numerous examples of turbulent flows exist in nature, including blood flow in arteries, atmospheric and ocean currents, smoke from a chimney, waterfalls, etc. It is also ubiquitous in engineering applications, such as the flow of liquid and gases in pumps, compressors, pipes, flow over vehicles, mixing of fuel and air in engines, boilers, etc [6].

It should be mentioned that there is no sharp distinction between laminar flows and turbulent flows in reality. A transition region can also be defined in which the flow behavior is something between laminar and turbulent.

Compared to other regimes, transient flows are less understood. However, intermittent behavior is less pronounced with increasing Re. In this work, we instead focus on high Reynolds numbers and fully turbulent flows.

Recognizing the particular behaviors of turbulent flows and considering their numerous occurrences and applications, the next question would be why turbulence occurs in the first place. Turbulence arises due to perturbation in the fluid flow's initial condition, boundary conditions, or material properties. The perturbation itself is not responsible for turbulence, but when Re is high, such variations cannot be damped by viscous forces. Due to these nonlinear instabilities, turbulence happens. From another perspective, when Re is high, the flow's kinetic energy is very high. The viscous forces cannot act upon such large eddies to dissipate their energy to heat and force the flow into equilibrium. Therefore, the flow starts to reduce its energy by successively breaking each energetic eddy into smaller parts. What is observed is then the interaction of larger eddies with smaller ones and the chaotic behavior of the flow as a whole.

We can better explain the turbulence by referring to two themes of *energy cascade* proposed by Richardson (1922) [8] and *Kolmogorov hypothesis* (1941) [9]. First, energy cascade is the process of successively transferring the energy to smaller and smaller eddies until the Reynolds number is so small that the eddy motion is stable, and molecular viscosity is effective in dissipating the kinetic energy [6]. This implies a one-way inviscid transfer of energy from large energetic eddies to the smaller eddies and, more importantly, places the dissipation of energy at the end of the process, whose rate is defined by the beginning of the process of large eddies breakup.

The Kolmogorov hypothesis states that at a sufficiently high Reynolds number, the small-scale turbulent motions are statistically isotropic. In addition, the statistics of the small-scale motions have a universal form that is uniquely determined by molecular viscosity ν and the dissipation rate ε. Furthermore, the statistics of the motions of the intermediate bandwidth of small eddies (the inertial sub-range) have a universal form uniquely determined by ε independent of ν.

Like any other theory, Kolmogorov's theory provides a model, which is, as any model, "wrong" but useful [10]. There exist many situations in which the hypotheses are not strictly valid. For example, the theory is bound to cases with very high Reynolds numbers; otherwise, the assumption of statistically isotropic dissipative scales is not correct. The one-way direction of energy transfer is also not always actual. However, for the purpose

of this work, which is the simulation of turbulent flows for engineering applications, such models provide a solid mathematical basis upon which the turbulence models are built.

The multiscale nature of turbulent flows (both in time and space) indicates an essential factor in terms of the computational aspect of simulations. We need to consider small motions in the order of Kolmogorov microscales up to the characteristic length scale of the problem. For example, for atmospheric motions, long-lasting large eddies have length scales in the order of kilometers, while short-living eddies are in the order of 0.1 to 10 millimeters. This multiscale nature of turbulent flows implies the difficulty of their simulation in terms of accuracy and computational costs.

1.3 WHY STUDY TURBULENT FLOWS?

Now that we have some intuitive and mathematical understanding of turbulent flows, we state why and how we study them. As mentioned, turbulent flows are ubiquitous in nature and industry. In nature, they exist on different scales, from interior biological cells to geophysical flows such as oceans, atmospheres, subsurface flows, and to even super-galactic scales. In industry, they have applications in wind flow in a city, aerodynamic analyses, continuous casting, quenching process, cooling/heating systems, etc. [11].

We can also categorize study motivations into discovery, modeling, and control [6]. First, we want to know why phenomena such as vortex stretching occur. Regardless of the consequent applications, exploring and discovering what happens in nature or finding the causality of phenomena is one motivation. Such understanding can then lead to setting up models, experimentally or numerically, in order to predict the behavior of the flow. For example, what is the drag force of the flow on a structure, or how can we optimize the topology of sold bodies to increase the lift or reduce the energy consumption? Thousands of applications exist in modeling and simulating turbulent flows. Lastly, we would like to control the behavior of the flow to be beneficial for us, like how to increase the mixing in a combustion chamber or actively postpone the separation over an airfoil to avoid a stall, etc.

1.4 HOW STUDY TURBULENT FLOWS?

In studying fluid mechanics in general and turbulent flow in particular, two general approaches are numerical and experimental. Because of the complexity of turbulent flows, an analytical approach cannot directly be used to solve a real problem. However, in combination with some empiricism, they can significantly contribute to developing accurate and robust turbulence models.

The numerical approach is based on numerically solving the governing equations and finding the flow variables' true states. Note that knowing the states of flow variables in the whole domain is necessary, even if we only are interested in a general estimation or an integral quantity such as the drag of a solid body. Due to the complex, non-local, and multiscale nature of turbulent flows, we usually have to run a full CFD code for such problems.

On the other hand, the experiments can directly be done on the problem of interest, like field measurements of the surface layer of the atmospheric boundary layer. Alternatively, one can make a physical model of the problem on a different scale, such as a wind tunnel setup, and aim to study the problem using sensors or flow visualization techniques. One can then use the similarity in fluid flows and generalize the findings to the real problem.

Both approaches have their pros and cons. The numerical approaches are generally cheap because one does not need measurement devices or build a physical model. However, its accuracy is subject to the modeling part of the problem and the numerical techniques used. On the other hand, due to the flexibility of the numerical models, one can quickly test and design different configurations of a problem or study extreme conditions for which experiments are not feasible. Moreover, highly linked to the mathematical model, it provides a basis for concretely understanding the phenomenon. However, the modeling part, which is an inevitable aspect of the simulation, leads to errors and uncertainties in the results.

On the other hand, experiments can be very costly but sometimes necessary. It can follow an exact scenario in the real world, and the obtained measurements are easy to understand [12]. Any analytical or CFD model needs validation, and experiments can be used for this purpose. In addition, the reliability of experiments is more than that of simulations. The drawbacks are the limitation of scenarios to study, the error-prone nature of experiments, difficulty with replication of the real problem, the lengthy setup procedure, and, more importantly, the low resolution of the mea-

surements. One must use multiple sensors or visualization techniques with limited window view or resolution. Incorporating such low-resolution measurement data into the model to benefit both is an essential aspect of this project.

1.5 TURBULENCE MODELING

Having the equation (1.5) and the continuity equation, with some treatments, we can discretize these equations over a computational grid to get a linear algebraic system of equations and find the solution for the state variables velocity and pressure. The process seems straightforward unless the fact that, for an accurate solution, we need to resolve all the motions (variations) that exist in the domain. Therefore, we need a fine grid and small appropriate time integration. The former defines the number of equations in the system that has to be solved. The latter implies that the time step must be small enough to resolve the small motions of eddies, and it must be numerically robust to avoid divergence. Such a "direct" approach is called *Direct Numerical Simulation* (DNS). It provides one realization of the turbulent flow. The following concerns would arise: How much computational resource and memory are needed to run the code for only one realization? Does information of one realization is relevant for a particular application? The answer depends on a particular question we want to answer with our simulation. However, we can say that for most applications we are more interested in the statistics of the flow of interest, which demands multiple realizations, or following the assumption of ergodicity, it needs a large time window to average. In fact, more than 99.98% of the computational effort in DNS is dedicated to resolving the dissipation range of turbulence [6, 13]. Moreover, for a wall-bounded flow, which is of particular interest in this work, The computational costs roughly scale as $Re^{9/4}$ and mainly are devoted to resolving the inner layer of the turbulent boundary layer [14]. According to Pope [6] four approaches to reduce the costs are low-wave number forcing, large eddy simulation, hyperviscosity, and sparse-mode models. However, all these methods are, in fact, not direct methods, deviate from the exact NS equations, and involve some modeling.

Despite the computational barrier, DNS is a valuable tool for academic purposes. Fundamental questions on understanding turbulence can be answered with DNS. In addition, recently, DNS data has extensively been used for data-driven turbulence modeling [15]. Nonetheless, considering the limitations of DNS and the fact that for highly turbulent flows, one generally

is not interested in instantaneous information of only one realization, but rather in the flow statistics, one may model or parameterize the turbulence at least partially. Two main approaches rely on decomposing the state variables fields into mean and fluctuating parts (in RANS methods) or filtered components and the residual components (in LES methods). We first focus on RANS models in the following section and then introduce LES.

1.5.1 *Reynolds-averaged Navier Stokes (RANS)*

If we merely look for statistical quantities, why not directly apply the statistical operator to NS equations and derive the equations that govern the mean velocity field? We can decompose the instantaneous velocity field $u(x,t)$ into mean $\langle u(x,t)\rangle$ and fluctuation $u'(x,t)$, where $u(x,t) = \langle u(x,t)\rangle + u'(x,t)$, and then apply the decomposition to the continuity and NS equation. Since the mean operator is a Reynolds averaging operator, it has, by definition, some properties, e. g. the mean of the fluctuating quantity is equal to zero ($\langle u'(x,t)\rangle = 0$). Using these properties, with some simple mathematics, we can derive the Reynolds-averaged continuity and RANS equations as follow[2]:

$$\frac{\partial \langle u_i\rangle}{\partial x_i} = 0, \tag{1.7}$$

$$\frac{\partial \langle u_i\rangle}{\partial t} + \frac{\partial(\langle u_i\rangle \langle u_j\rangle)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \langle p\rangle}{\partial x_i} + \nu\frac{\partial^2 \langle u_i\rangle}{\partial x_j x_j} - \frac{\partial\left\langle u_i' u_j'\right\rangle}{\partial x_j}. \tag{1.8}$$

The last RHS term in equation 1.8 is extra compared to the NS equations and is called the unclosed term. It is the divergence of the covariance matrix between velocity components and is derived from applying the Reynolds averaging operator on the convective term. The covariance matrix is also called the Reynolds stress tensor. There is no direct mean information to characterize such a stress term so that it can be seen as a representation of lost information through the averaging process. However, it cannot be neglected as it dramatically impacts the flow qualitatively and quantitatively. Thus one has to somehow model or parameterize this term. Such a problem is called a closure problem and is still the subject of active research. There are generally two approaches to model the Reynolds stresses: introducing

2 From now on we occasionally shift to index notation as it is sometimes more convenient for our purposes

the concept of eddy viscosity or directly modeling the Reynolds stress components.

1.5.1.1 *Linear eddy viscosity models (LEV)*

The first approach, which is simpler, but not necessarily less "successful," is based on the eddy viscosity hypothesis. Recall the constitutive relation between stress and the strain rate. One can generalize such an approach and relate the turbulent stresses to the mean strain rate through the scalar eddy viscosity. We can decompose the Reynolds stress tensor into the hydrostatic and the deviatoric parts. The deviatoric part of the turbulent shear stress is approximated as

$$\boldsymbol{\tau_t} = -2\nu_t \langle \mathbf{S} \rangle = -2\nu_t \frac{1}{2}[\nabla \langle \boldsymbol{u} \rangle + (\nabla \langle \boldsymbol{u} \rangle)^T]. \tag{1.9}$$

We define half of the trace of the Reynolds stress tensor as the turbulent kinetic energy (TKE) $k = \frac{1}{2}\left(\langle u_1' \rangle^2 + \langle u_2' \rangle^2 + \langle u_3' \rangle^2\right)$. The remaining part of the Reynolds stress tensor, the hydrostatic part, contains the trace of the stress tensor and is $\frac{2}{3}k\mathbf{I}$. Therefore, the Reynolds stress approximation reads

$$\left\langle u_i' u_j' \right\rangle = \frac{2}{3}k\mathbf{I} - \nu_t[\nabla \langle \boldsymbol{u} \rangle + (\nabla \langle \boldsymbol{u} \rangle)^T]. \tag{1.10}$$

If we insert this relation into the RANS equation, for an incompressible flow, it reads

$$\frac{\mathrm{D}\langle u_i \rangle}{\mathrm{D}t} = -\frac{1}{\rho}\frac{\partial \langle p^* \rangle}{\partial x_i} + \frac{\partial}{\partial x_j}\left[2\left(\nu + \nu_t\right)\left\langle S_{ij} \right\rangle\right] , \tag{1.11}$$

where the turbulent pressure $\langle p^* \rangle$ also contains the hydrostatic part of the Reynolds stress $\langle p^* \rangle = \langle p \rangle + \frac{2}{3}k$ and $\left\langle S_{ij} \right\rangle$ is the mean strain rate. The goal of a linear eddy viscosity (LEV) turbulence model (or a turbulence model based on the Boussinesq eddy viscosity assumption) is to calculate the eddy viscosity accurately.

Before we introduce the common LEV turbulence models, it is valid to doubt the credibility of the LEV assumption itself. It is, in fact, an inaccurate assumption since it is a local model means that it only includes local information about the anisotropy of the turbulence [16]. In LEV models, eigenvectors of the Reynolds stress and strain rate tensors always are aligned, which in reality not always is the case. Among various states of turbulence, the fully aligned Reynolds stress tensor, plain-strain turbulence,

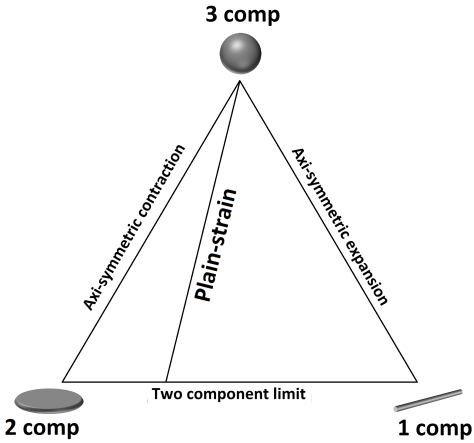


FIGURE 1.1: The Barycentric triangle [17]. It defines all realizable states of the Reynolds stress anisotropy. Any position in the Barycentric triangle represents a realizable anisotropy state of the Reynolds stress. The vertices of the triangle correspond to the limiting one, two, and three-component (isotropic) states

is only a portion of different realizable turbulence states. This can be shown in the barycentric triangle proposed by Banerjee et al. [17], which is depicted in figure 1.1. LEV models only produce an anisotropy state of the Reynolds stress that resides on the plain-strain line where at least one eigenvalue is zero. Any other position within the triangle is a realizable state of the Reynolds stress which is out of reach for LEV. For more information on the mathematical aspects of the Barycentric triangle, the reader is referred to Refs. [15, 17].

To have an idea about how to develop an LEV turbulence model, we note that the eddy viscosity has the dimensions of [velocity] $\times$ [length], suggesting that it can be modeled by $\nu_t = u_0 l_0$ where u_0 reflects the magnitude of velocity fluctuations and l_0 the size of turbulent eddies [18]. For simple configurations, we can indeed find, to some extent, simple universal relations to estimate both u_0 and l_0. For example, for wall-bounded flows, the velocity fluctuations can be characterized by friction velocity. For free shear flows, the velocity fluctuation can be related to the square root of the turbulent kinetic energy $k^{1/2}$. The length scale of the eddy for a wall-bounded flow is proportional to the distance to the boundary, and for free shear flow, l_0 is proportional to the width of the shear layer.

This approach is, more or less, the basis of algebraic (zero-equation) LEV turbulence models. They are mostly based on the mixing length hypothesis by Prandtl [19]. Mixing length (l_m) is the distance over which a hypothesized turbulent eddy retains its identity [20]. In such methods $l_0 = l_m$. The inherent problem of these models is their incompleteness; the mixing length scale has to be specified for each particular problem and depends on the flow type. The Cebeci-Smith model and the Baldwin-Lomax model are examples of such zero-equation models [21]. In addition to their limited applications, they fail for out-of-equilibrium flows such as separated flows. The Johnson-King model is more suited for the separated flows [22]. Since it solves an ordinary differential equation, it is sometimes called the half-equation model. In addition to this additional complexity, it introduces many empirical coefficients.

The one-equation models could circumvent some of the issues of the algebraic models to some extent. One-equation models that only solve the turbulent kinetic energy transport equation are still incomplete and need to estimate the turbulence length scale. The turbulent kinetic energy is a physical property, and its exact transport equation can be derived from the second moment derivation of the NS equation and it reads

$$\frac{\mathrm{D}k}{\mathrm{D}t} + \nabla \cdot T' = P - \varepsilon, \tag{1.12}$$

where $P = -\left\langle u_i' u_j' \right\rangle \frac{\partial \langle u_i \rangle}{\partial x_j}$ is called the production of turbulent kinetic energy, $\nabla \cdot T'$ is the turbulent transport of TKE and $\varepsilon = \nu \overline{\frac{\partial u_i'}{\partial x_j} \frac{\partial u_i'}{\partial x_j}}$ is the dissipation rate of TKE. In addition, $T_i' = \frac{1}{2}\left\langle u_i' u_j' u_j' \right\rangle + \left\langle u_i' p' \right\rangle / \rho - \nu \frac{\partial k}{\partial x_i}$, where the first term, the triple velocity correlation, describes turbulent convection, the second pressure transport, another form of turbulent transport resulting from the correlation of pressure and velocity fluctuations, and the third describes molecular diffusion of TKE [21]. Multiple unclosed terms need to be modeled to eventually close the unclosed Reynolds stress term. Generally, the equation is simplified through the following relation:

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = \tau_{ij} \frac{\partial u_i}{\partial x_j} - \varepsilon + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_T}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right], \tag{1.13}$$

where all the unclosed terms except ε are lumped into $\frac{\partial}{\partial x_j}\left(\frac{\nu_T}{\sigma_k}\frac{\partial k}{\partial x_j}\right)$ through the gradient-diffusion hypothesis, where $\sigma_k = 1$ is a coefficient. Since the length-scale needed for the eddy viscosity computation still is missing,

Prandtl proposed a relation for ε that links it to the mixing length as $\varepsilon = C_D \frac{k^{\frac{3}{2}}}{l}$, where $C_D = 0.08$ and thus $\nu_t = k^{\frac{1}{2}} l = C_D \frac{k^2}{\varepsilon}$. Such models, though to some degree successful, still are incomplete and fail for out-of-equilibrium flows like separated flows. Other one-equation models, such as the Baldwin-Barth model [23] and the Spalart-Allmaras model [24], use a transport equation for other than TKE. With the help of multiple empirical relations, they lead to complete sets of equations from which the eddy viscosity is estimated. Although debatable, one can argue that to reach more universal models, especially for separated flows, we need to consider the transport of both the velocity and length scales separately. Two-equation models were developed for this purpose. These models propose transport equations not only for the computation of k, but also for the turbulence length scale or something equivalent.

The most popular two-equation LEV model is the standard $k - \varepsilon$ [25]. The idea behind the $k - \varepsilon$ model is to estimate the eddy viscosity by solving a transport equation for TKE to estimate the velocity fluctuation $C_\mu{}^{1/4} k^{1/2}$ and solving a transport equation for ε. The latter is suggested to close the TKE equation and also to estimate the length-scale $l = C_\mu^{3/4} \frac{k^{\frac{3}{2}}}{\varepsilon}$. The eddy viscosity then can be estimated as

$$\nu_t = C_\mu \frac{k^2}{\varepsilon}, \tag{1.14}$$

where the default value for C_μ is 0.09. From the NS equations with some algebra the exact equation for ε can be derived;s [26]

$$\frac{\mathrm{D}\varepsilon}{\mathrm{D}t} = P_\varepsilon^{(1)} + P_\varepsilon^{(2)} + P_\varepsilon^{(3)} + P_\varepsilon^{(4)} + T_\varepsilon + V_\varepsilon + \pi_\varepsilon - \gamma_\varepsilon, \tag{1.15}$$

where $P_\varepsilon^{(1)} = \nu \left\langle \frac{\partial u_i'}{\partial x_m} \frac{\partial u_k'}{\partial x_m} \right\rangle \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} \right)$ is the mixed production, $P_\varepsilon^{(2)} = \nu \left\langle \frac{\partial u_i'}{\partial x_m} \frac{\partial u_i'}{\partial x_k} \right\rangle \left(\frac{\partial u_k}{\partial x_m} + \frac{\partial u_m}{\partial x_k} \right)$ is the production by mean velocity gradient, $P_\varepsilon^{(3)} = \nu \left\langle u_k' \frac{\partial u_i'}{\partial x_m} \right\rangle \frac{\partial^2 u_i}{\partial x_m \partial x_k}$ is the gradient production, $P_\varepsilon^{(4)} = 2\nu \left\langle \frac{\partial u_i'}{\partial x_m} \frac{\partial u_k'}{\partial x_m} \frac{\partial u_i'}{\partial x_k} \right\rangle$ is the turbulent production, $T_\varepsilon = \frac{\partial}{\partial x_k} \left(\nu \left\langle u_k' \frac{\partial u_i'}{\partial x_m} \frac{\partial u_i'}{\partial x_m} \right\rangle \right)$ is the turbulent diffusion, $V_\varepsilon = \nu \frac{\partial^2}{\partial x_k^2} \left(\left\langle \nu \frac{\partial u_i'}{\partial x_m} \frac{\partial u_i'}{\partial x_m} \right\rangle \right)$ is the viscous diffusion, $\pi_\varepsilon = -\frac{2}{\rho} \frac{\partial}{\partial x_k} \left(\left\langle \frac{\partial u_k'}{\partial x_m} \frac{\partial p'}{\partial x_m} \right\rangle \right)$ is the pressure diffusion, and $\gamma_\varepsilon = 2 \left(\left\langle \nu \frac{\partial^2 u_i'}{\partial x_m \partial x_k} \right\rangle \right)^2$ is the dissipation. As we see, this equation involves multiple unknown double and triple correlations whose values are almost impossible to measure. Thus it seems impossible to

derive a useful transport equation directly from equation 1.15. Therefore, an entirely empirical approach has been chosen to derive a modeled transport equation for ε; it reads

$$\frac{\partial \varepsilon}{\partial t} + u_j \frac{\partial \varepsilon}{\partial x_j} = C_{\varepsilon 1} \frac{\varepsilon}{k} \tau_{ij} \frac{\partial u_i}{\partial x_j} - C_{\varepsilon 2} \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right], \tag{1.16}$$

where $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, and σ_ε are the closure coefficients, whose values are tuned with respect to simple flows. As we see, the structure of the model transport equation of ε is very similar to the model TKE equation. Thus, there are no rigorous physical reasons that the equation above should work, and the success of the $k - \varepsilon$ model was a kind of a successful "try-on," as stated by Brian Spalding, the main developer of this model [27]. If such an amount of empiricism "works," there is no reason one would not suggest other quantities than ε to close the eddy viscosity model. The $k - \omega$ model [28] and its improved version, the $k - \omega$ SST model [29], are further successful models widely used in the community. All these models contain branches of further modifications, which have been successful for particular applications.

In general, the advantages of LEV models include [18]:

- They are simple to code.
- Since they increase the effective viscosity, they aid stability.
- They are, to some degree, theoretically supported for simple flows.
- More importantly, their accuracy is acceptable for many engineering flows.

The disadvantage of such models is the constraint of the Boussinesq eddy viscosity assumption on the Reynolds stress tensor. It dictates that the principal axes of the Reynolds stress tensor are aligned with the mean strain rate, which in general is not physical. A classical example is the logarithmic region of a fully-developed boundary layer, where the ratio of the normal stresses typically are $\left\langle u_1'^2 \right\rangle : \left\langle u_2'^2 \right\rangle : \left\langle u_3'^2 \right\rangle = 1.0 : 0.4 : 0.6$ [18]. However, all eddy viscosity models predict that all of these normal stresses are $\frac{2}{3}k$. The failure of LEV models occurs in flows with a sudden change in the mean strain rate, flow over curved surfaces, three-dimensional flows, etc. These major drawbacks motivate the development of two further classes of RANS models: nonlinear eddy viscosity models and Reynolds stress models.

1.5.1.2 *Non-linear eddy viscosity models (NLEV)*

We define the anisotropy tensor, the dimensionless and traceless form of the Reynolds stress as $a_{ij} = \frac{\left\langle u_i' u_j' \right\rangle}{k} - \frac{2}{3}\delta_{ij}$. For LEV models $a_{ij} = -2C_\mu s_{ij}$ where $s_{ij} = \frac{k}{\varepsilon} S_{ij}$ is the dimensionless mean strain. This relation can be generalized by considering the nonlinear effects that include higher orders of the mean strain and rotation tensors as follows

$$a_{ij} = -2C_\mu s_{ij} + \mathrm{NL}(s_{ij}, \omega_{ij}), \tag{1.17}$$

where $\omega_{ij} = \frac{1}{2}\frac{k}{\varepsilon}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$ is the dimensionless rotation tensor and NL accounts for the nonlinear terms, which can contain quadratic, cubic, etc. terms. Care should be taken in developing such models, e. g., respecting the realizability of the tensor. In general, nonlinear eddy viscosity models (NLEV) produce qualitatively-correct turbulent behavior in specific flows, and they introduce a slight computational overhead to linear eddy viscosity models. However, they do not accurately represent the actual production and advection processes and have a little theoretical foundation in complex flows [18]. They also introduce instability to solvers as they add nonlinear terms to the RANS equations.

1.5.1.3 *Reynolds stress models (RSM)*

Reynolds stress transport models (RSTM) solve transport equations for each stress component. The equations are derived by writing a momentum equation for the fluctuating component u_i' and multiplying it with u_j'. With some maths, one then obtains the equation for the Reynolds stress tensor. The equation reads[3]

$$\frac{\mathrm{D}\left\langle u_i' u_j' \right\rangle}{\mathrm{D}t} = D_{ij} + P_{ij} + \Pi_{ij} - \varepsilon_{ij}, \tag{1.18}$$

where the LHS is the rate of change of $\left\langle u_i' u_j' \right\rangle$ plus its convective transport, $D_{ij} = \frac{\partial}{\partial x_k}\left[\nu \frac{\partial}{\partial x_k}\left\langle u_i' u_j' \right\rangle - \frac{1}{\rho}\left\langle p'\left(u_i'\delta_{jk} + u_j'\delta_{ik}\right)\right\rangle - \left\langle u_i' u_j' u_k' \right\rangle\right]$ is the aggregate of molecular and turbulent diffusion of the stress tensor, $P_{ij} = -\left\langle u_i' u_k' \right\rangle \frac{\partial u_j}{\partial x_k} - \left\langle u_j' u_k' \right\rangle \frac{\partial u_i}{\partial x_k}$ is the mean strain production tensor, $\Pi_{ij} =$

3 effect of rotation and body forces are neglected

$\left\langle \frac{p'}{\rho}\left(\frac{\partial u_i'}{\partial x_j}+\frac{\partial u_j'}{\partial x_i}\right)\right\rangle$ is the pressure-rate-of-strain, and $\varepsilon_{ij}=-2\nu\left\langle \frac{\partial u_i'}{\partial x_k}\frac{\partial u_j'}{\partial x_k}\right\rangle$ is the dissipation rate. It seems that with fewer modeling constraints and more physics one can get more accurate results using these models. However, as we see, numerous unclosed terms need to be modeled. The production term and the LHS are exact, but the turbulent transport, dissipation, and pressure-rate-of-strain terms need modeling. Especially the pressure-rate-of-strain is challenging, as the stress redistribution and tendency to isotropy depend on this term. Turbulent transport can be simplified by the gradient-diffusion hypothesis. The dissipation rate tensor can be reduced to a scalar, i. e., $\varepsilon_{ij}=\frac{2}{3}\varepsilon\delta_{ij}$, following the Kolmogorov hypothesis.

Another branch of RSM models is the algebraic Reynolds stress model (ARSM), in which the transport terms are approximated by an algebraic expression so that the entire equation becomes algebraic. Thus the Reynolds stresses can be computed as functions of k, ε, and the mean velocity gradients [6].

As Spallart asserted, modeling the higher moments is very empirical, relying on dimensional analysis, invariance, intuition, and assumptions [16]. Therefore, it seems to add more physics to the model automatically is accompanied by the necessity of empirical modeling, thus introducing more uncertainties. The instability of RSM compared to LEV models and the additional computational costs are other reasons why such models have not been widely used in practice.

1.5.2 *Conclusions on RANS models*

We have seen that Reynolds averaging can significantly reduce the computational costs needed to simulate turbulent flows. However, the averaging nature of RANS equations induces an unclosed term, the Reynolds stress term, which needs modeling. From simple length-scale models to LEV models and NLEV models to ARSM models and RSTM models, we saw that no single RANS model universally works for all problems. Complexity, empirical relations, uncertain coefficients, computational cost, generalizability, robustness, and convergence are the factors one needs to consider when choosing a turbulence model. In fact, in recent years no new widely-accepted general-purpose RANS turbulence model has been devised [30]. However, recently, data-driven approaches have made some progress in turbulence modeling which will be discussed later.

1.5.3 *Large eddy simulation (LES)*

A remarkably different approach to turbulence modeling is the Large Eddy Simulation (LES). The idea is to decompose the instantaneous state variable ϕ into filtered and residual quantities, $\overline{\phi}$ and ϕ', respectively, through a filtering procedure. A general form of a spatial and temporal filtering operator reads

$$\overline{\phi(\boldsymbol{x},t)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi(\boldsymbol{r},\tau)G(\boldsymbol{x}-\boldsymbol{r},t-\tau)d\tau d\boldsymbol{r}, \tag{1.19}$$

where G is the filter convolution kernel. In practice, a more specific spatial filtering, e. g. a box or Gaussian filter is used. We can apply this filtering to the state variables and derive filtered continuity and NS equations. There are generally two important properties of filtering that differ from Reynolds averaging, namely $\overline{\overline{\phi(x,t)}} \neq \overline{\phi(x,t)}$ (a second filtering yields a different result from the first filtering) and $\overline{\phi'(x,t)} \neq 0$. Considering this, we derive filtered governing equations as follow [21]:

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0, \tag{1.20}$$

$$\frac{\partial \overline{u_i}}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{u_i u_j}\right) = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu\frac{\partial^2 \overline{u_i}}{\partial x_j x_j}, \tag{1.21}$$

where $\overline{u_i}$ is the filtered velocity and $\overline{p}$ is the filtered pressure. The convective flux is decomposed into the following components

$$\overline{u_i u_j} = \overline{u_i}\,\overline{u_j} + L_{ij} + C_{ij} + R_{ij}, \tag{1.22}$$

where $L_{ij} = \overline{\overline{u_i}\,\overline{u_j}} - \overline{u_i}\,\overline{u_j}$ is called the Leonard stress, $C_{ij} = \overline{\overline{u_i}u_j'} + \overline{\overline{u_j}u_i'}$ is the cross-term stress and $R_{ij} = \overline{u_i' u_j'}$ is the SGS Reynolds stress. L_{ij} represents the interactions among large scales, C_{ij} represents the cross-scale interactions between small and large scales, and R_{ij} represents interactions among the sub-filter scales. In practice, the sum of these three stress terms $\tau_{ij} = L_{ij} + C_{ij} + R_{ij}$ is modeled by a subgrid-scale (SGS) model. This modeling is challenging because the subgrid stress tensor τ_{ij} must account for interactions of all scales, including interactions between filtered scales and unfiltered scales. The SGS models can be based on a simple gradient-diffusion model [31], a one-equation model [32], or analogous to a second-order closure model [33]. Nonlinear stress-strain rate relationships have also been proposed [34]. The analog of the two-equation model has

not been postulated, because the filter width already serves as a readily-available length scale [21].

The most basic model is the Smagorinsky model [31], in which the residual stress τ_{ij} is related to the filtered rate of strain $\tau_{ij} = -2\nu_t \overline{S}_{ij}$. The eddy viscosity is then modeled as $\nu_t = {l_s}^2 \sqrt{\overline{S}_{ij}\overline{S}_{ij}} = (C_S\Delta)^2 \sqrt{\overline{S}_{ij}\overline{S}_{ij}}$, where C_S is the Smagorinsky coefficient. This equation is very similar to the mixing length formula with the mixing length as $C_S\Delta$ and is based on the assumption of an equilibrium of the energy production and dissipation of the small scales. This model performs poorly for wall-bounded flows because of the linear eddy viscosity assumption. In addition, in practice, it has been shown that the Smagorinsky coefficient spatially and temporally varies. It also does not produce "backscatter," in which a part of the turbulent kinetic energy returns to the large scales without being dissipated [20].

The dynamic Smagorinsky model [35] is a modification to the basic Smagorinsky model and allows the Smagorinsky coefficient to be a function of space and time. A second filtering operation termed a test filter is introduced and used to compute the unresolved stresses. The method was quite successful for many applications. However, some of the Smagorinsky model's drawbacks, such as the alignment of residual stress with the filtered strain rate, still exist. One can also consider the history and non-local effects through transport equations for the residual stress tensor. Since the filter width can always estimate the length scale, one may only solve the residual kinetic energy [36] or a simple modeled equation for the residual stress [37].

1.5.4 *Conclusions on LES*

Since the energy-containing part of the flow is directly resolved by LES, it is, in principle, more accurate than RANS, where all turbulent eddies are modeled. However, the computational burden on LES is still an issue, even with current computational resources. Due to their formulation, LES models cannot be conducted in a two-dimensional mode, which adds further cost for two-dimensional geometries. Furthermore, since most of the time we are interested more in flow statistics than one realization, LES models have to run for a considerable time to achieve statistics. Considering they should always run in a transient mode with $CFL < 1$, varying pseudo-time steps (like for steady RANS) are not possible. The computational cost of LES in free-shear flows is only weakly dependent on the Reynolds number. On the other hand, for wall-bounded flows, to resolve the near-wall region, the cost roughly scales with $Re^{1.8}$ [14].This computational barrier of LES

was a motivation to explore other approaches such as wall-modeled LES and hybrid LES/RANS methods. Since these two approaches sometimes overlap, we review them together.

1.5.5 *Hybrid LES/RANS models and wall-modeled LES*

There is a spectrum of methods, from applying a simple law of the wall to fully coupling LES with RANS models, all of which may introduce some inaccuracies, but reduce the computational cost. Since all these supplementary models to LES focus on the boundary layer, we first introduce this concept.

1.5.5.1 *Turbulent boundary layer*

Loosely speaking, the boundary layer is the region where the presence of the wall directly affects the fluid flow. The thickness of the boundary layer δ is often defined as the distance from the wall, where the stream-wise velocity in the boundary layer is $0.99U_e$ (U_e is the local free-stream speed) [38]. As depicted in figure 1.2, the turbulent boundary layer can be conceptually divided into the outer layer (the outermost 80-90%) and the inner layer (the innermost 10-20%). The inner layer can further be divided into three regions. The region closest to the wall is the viscous sublayer dominated by viscous effects, which exhibits a near linear velocity profile [39]. The next layer is the buffer layer between the viscous sublayer and the log-law layer and is the region where both dissipation and inertial effects are relevant. The log-law layer is the region in which the shear stress is constant, while the viscous effects are negligible; this results in a log profile for the mean velocity. It is also called the law of the wall.

1.5.5.2 *Wall-modeled LES*

In common LES, the grid should resolve even the viscous sublayer ($y^+ \approx 1$ or $y/\delta < 10^{-3}$). In addition, the cells' geometry must be close to isotropic; therefore, multiple cells are needed to resolve the wall-normal and wall-parallel directions of the inner layer. One may suggest that in the wall-normal direction, we place one cell within the log-law region and use the universal logarithmic relation to relate the local velocity to the wall shear stress. However, the law of the wall is based on multiple simplifying assumptions. If we start with the NS equation 1.5 and apply the equilibrium assumption, we remove the rate of change term on the left-hand side,

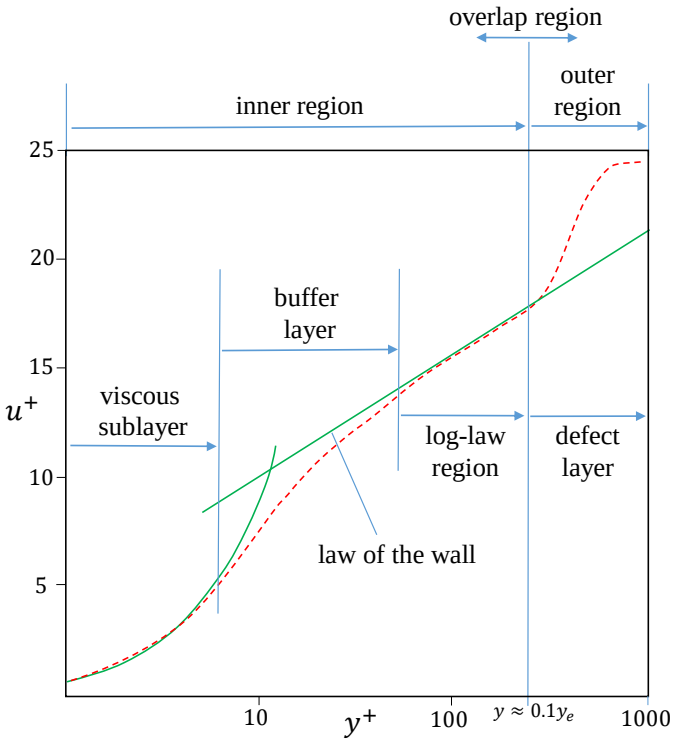


FIGURE 1.2: Turbulent boundary layer and the law of the wall. The plot is in semi-log format. Adapted from [40]

assuming the velocity vector is parallel to the wall, ignoring all spatial derivatives except in vertical direction, ignoring the viscous stresses and the pressure gradient, the only term remaining is $\frac{d\tau}{dz} = 0$, which means the stress is a constant vertical flux from which one can obtain a logarithmic profile for the mean velocity [41]. Of course, e. g., if the geometry is complex or the mean flow is not steady, most such assumptions are inaccurate. Therefore, in real applications, the law of the wall would "kill" the accuracy we seek from an LES simulation. It is important to grasp the fact that a low-resolution simulation is undesirable, not only because it contains less information (low resolution of information), but also because it leads to lower accuracy in the low-resolution region, in addition to the accuracy of the solution in the whole region. Some improvements to the logarithmic profile have been suggested. For instance, Yang et al. [42] proposed an algebraic profile that contains a viscous and a logarithmic layer with an additional linear term that takes into account the inertial and pressure gradient effects. Such non-equilibrium algebraic models are also common in the context of wall functions for RANS models [43–45]. A bit more complicated way is to solve an ODE in the wall-normal direction (solving the momentum equation without wall-parallel grid-connectivity), mostly needed when heat transfer is also involved [46]. Moreover, if one includes the wall-parallel grid connectivity, then a full boundary layer PDE might be solved, removing the necessity of a perfect balance between convection and the pressure-gradient [47].

1.5.5.3 *Hybrid LES/RANS models*

We saw in wall-modeled LES that a wall model receives information from LES at wall adjacent cells and solves algebraic or differential equations to compute the wall shear stress. In hybrid models, however, such communication can be generalized. A RANS model can be solved, which communicates with LES in different ways. In *segregated methods*, the domain is decomposed into two regions. Near the wall, RANS is solved, and LES is solved in the free shear flow region. Information is then exchanged at the boundaries of the two regions as boundary conditions [48]. In *unified methods*, one single mesh is chosen for the whole domain. The solver is switched between RANS and LES at specific cells, which can be statically specified beforehand or dynamically during the simulation. DES is one famous example of such an approach [49]. The RANS and LES regions can be divided sharply by interfaces, or softly as blended regions. Another approach is the dual-mesh method in which RANS and LES are solved on two different meshes of the

whole domain, and the information is exchanged in specific regions [14]. All these methods have their pros and cons. For example, the switching or blending of the models in unified methods can induce inconsistencies and errors. Coupling two models at interfaces of segregated methods needs special treatments. The computational overhead of dual-mesh methods can be larger.

1.5.6 *Conclusions on turbulence modeling*

The computational burden of direct numerical simulation of turbulent flows is the primary motivation for turbulence modeling. Two approaches of averaging and filtering can reduce the computational costs to some extent though introducing some inaccuracies. In the author's opinion, RANS closures more complex than two-equation LEV models may not be widely used in the near future, except for particular applications, due to their limitations such as coding complexity and convergence issues. In addition, considering the current computational power, wall-resolved LES models would still be prohibitive for industrial applications, especially if they involve iterative design or optimization. Thus hybrid LES/RANS models would probably be one of the main focuses of turbulence modeling developments [50]. However, there is a new research front on the topic of turbulent flow simulations which needs further consideration: data-driven modeling and data-driven simulation.

1.6 DATA AND TURBULENCE MODELING

Data in our context is defined as the available measured physical quantities by experimental setups or by DNS simulations. It is assumed that such data is the ground truth for that specific configuration and, therefore, can be used to inform turbulence models. Due to the advances in computational resources and measurement techniques, both data sources have become increasingly available in the last decade [51]. Specifically, in recent years, with the abundance of available databases and advances in data science, *data-driven* approaches have become very popular in fluid dynamics.

Before we proceed, however, one should note that data in the process of turbulence modeling has always been used for the most basic models as well as for the most complex ones. As we have seen, introducing Reynolds-averaging or filtering to the NS equations leads to exact equations, but with unclosed Reynolds or residual stresses that contain the effect of

unresolved quantities. The goal of turbulence models was to approximate these stresses using the mean or filtered quantities. Turbulence modelers introduce empirical knowledge based on dimensional analysis, invariance, asymptotic theories, intuition, and assumptions to close the terms [15, 16]. For example, in the $k - \varepsilon$ model introduction of the coefficient C_μ in the eddy viscosity definition, and the whole ε equation with its coefficients are examples of empiricism. Since empirical relations are not directly derived from exact equations, and thus they inevitably include some parameters whose values need to be defined a priori, using data is inevitable. Therefore, data in turbulence modeling was historically used to develop and calibrate the models. However, the development of turbulence models systematically guided by data has not been done until recently.

Concerning a particular problem, we divide datasets into four categories

1. Datasets of simple (canonical) flows, including homogeneous isotropic turbulence, homogeneous shear flow turbulence, different types of free shear flows like jets, the zero pressure-gradient turbulent boundary layer, turbulent channel, turbulent pipe flow, etc [52].

2. Datasets that include more complex phenomena such as separations or 3D effects, e. g. backward-facing steps, flow over periodic hills, 3D Bump-in-channel, 2D NACA 4412 airfoil trailing edge separation, etc. [53].

3. Data obtained from experimental measurements for a case with partially similar geometry or flow conditions.

4. Data obtained from experimental measurements exclusively for the problem of interest with identical geometry and flow conditions.

We note that, in general, the experimental measurements are incomplete in that they are of low resolution or sparse. In contrast, DNS data is complete. Roughly speaking, depending on the goal of a study, these datasets can be utilized in the following scenarios

- In development phase: Historically, the first category has been used for the development and *calibration* of turbulence models. Recently, attempts have been made to use *machine learning* (ML) to systematically guide the development of turbulence models (*data-driven modeling*) by using the first and second categories of datasets.

- In application phase: The second, third, and fourth categories of datasets are used for *validation* of the models. The fourth category of the dataset (which is incomplete by nature) is used to improve the predictions of the models using *data assimilation* (DA) techniques.

The role of data in calibration and validation is clear. The term *data-driven modeling* refers to an attempt to develop a turbulence model merely or partially based on data, primarily using the DNS datasets (datasets of categories one and two) but possibly also sparse experimental data (dataset of category three). ML (sometimes combined with DA) is the essential mathematical tool to obtain such models. The term *data-driven simulation* refers to an attempt to incorporate the experimental data of the problem of interest (fourth category of the dataset) to improve the accuracy of that particular simulation. DA is the primary mathematical technique for this aim. Since the experimental data is of low resolution or sparse, we also can call this *sparse data-driven simulation*. Before we focus on the topic of this thesis, "sparse data-driven simulation of turbulent flows," we briefly review data-driven modeling in the context of ML.

1.6.1 *Data-driven turbulence modeling*

Generally, ML models are used when the underlying mechanism of a process is not well understood. For example, one can assume the physics of the Reynolds or subgrid stresses is not fully known, and thus one may try to replace the whole or a part of these empirical-physical turbulence models with a purely data-driven machine learning model. The machine learning model is first trained by datasets of particular geometries and flow conditions through supervised learning. The hope is that the ML model, embedded in the governing equations, can predict correct solutions in unseen geometries or flow configurations.

Historically, the first attempt to use ML was to re-calibrate the parameters of existing turbulence models and quantify their uncertainties [54–58]. Then some researchers suggested reconstructing the discrepancy of Reynolds stresses through learning based on priori DNS data. The reconstructed discrepancy tensor was a function of space, so it lacks the generalizability feature, which was the main goal of data-driven models. Vollant et al. [59] and Tracey et al. [60] proposed a transformation of the model discrepancies from the physical space (spatial space) to mean flow feature space, such that the trained ML model can predict correct Reynolds/subgrid stresses for unseen cases solely from the information (state variables or secondary

variables) of the unseen case. However, it became apparent that these approaches did not satisfy the expectations, since the ML models were not informed enough of physical laws and constraints. In other words, the ML models were too free in inferring correct Reynolds stresses. Another issue with such models was the ill-conditioning nature of such explicitly data-driven models [61]. In addition to difficulty in convergence, a tiny error in the Reynolds-stress tensor estimation propagates through the equations and results in significant errors in the solutions [61]. So several works were focused on how to *represent* the ML model such that physical laws are better respected. Three general approaches have been followed:

- **Structural correction within an existing turbulence model**: In these approaches, the discrepancy is encoded in a part of the turbulence model. One of the early works is by Dow and Wang [62] where the optimal eddy viscosity field is inferred from DNS mean velocity data using discrete adjoint method. Duraisamy and co-workers put a multiplicative correction term in the Spalart-Allmaras model as the machine learning output. The ML model had some success in generalizability within a class of flows around airfoils [63].

- **Data-driven development**: A generic form of Reynolds stress is assumed, and ML is used to define the model's parameters. Ling et al. [64] proposed using neural networks to learn the coefficients of a tensor basis expansion of the Reynolds stress from flow features, as proposed by Pope [65], as follows

$$\tau = 2k\left(\frac{1}{3}\mathrm{I} + \left[\sum_{n=1}^{10}\delta^{(n)}(\tilde{\eta};w)\mathrm{T}^{(n)}(\tilde{S},\tilde{\Omega})\right]\right), \tag{1.23}$$

  where $\delta^{(n)}$ is the coefficient computed from the output of a neural network that has $\tilde{\eta}$ as flow features and w as the weight and biases of the neural network. $\mathrm{T}^{(n)}(\tilde{S},\tilde{\Omega})$ are the tensorial basis. This approach can be regarded as an approach between traditional turbulence modeling and data-driven modeling.

- **the eigendecomposition of the Reynolds stress anisotropy tensor**: to enforce some physical constraints such as Galilean invariance and realizability, one would eigen-decompose the Reynolds stress anisotropy tensor. The Reynolds stress tensor then reads [66]

$$\tau = 2k\left(\frac{1}{3}\mathbf{I} + \mathbf{V}\,\Lambda\,\mathbf{V}^{T}\right) \tag{1.24}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ and $\Lambda = diag[\lambda_1, \lambda_2, \lambda_3]$ are the orthonormal eigenvectors and eigenvalues of the Reynolds stress anisotropy tensor, respectively, indicating its shape and orientation. The structure grants the perseverance of Galilean invariance. In addition, the Lumley triangle [67] or the barycentric triangle [17] can be used to enforce the realizability of eigenvalues.

Despite some success of these approaches, it still is too early for a clear judgment on the future of data-driven models. There still are many shortcomings to overcome, such as [15]:

- The issue of inconsistency between the training of the model with the DNS dataset and the RANS or LES environment in *a priori* training techniques.
- The computational costs and issues with the ill-conditioning of inverse problems in model-consistent training.
- Proper choice of feature space.
- Difficulty of applying constraints on the input and output of the ML model.
- Generalizability of ML models to unseen geometries and flow conditions and their interpretability.

1.6.2 *Sparse data-driven simulation*

Most of the data-driven modeling approaches were focused on utilizing the full field of DNS data rather than sparse experimental data. A few have also used sparse experimental data, such as references [63, 68], where the extracted experimental information was used to correct a RANS model. However, inferring knowledge from incomplete data requires solving an under-determined inverse problem. Such a problem has a high degree of ambiguity and can easily lead to many solutions, most of which are unphysical. Therefore, extracting information to improve the predictions of that specific configuration is difficult, let alone obtaining information that has generalizability for other conditions. At least the data must first go through the "filter" of a RANS/LES model to obtain full-field data and then be used to train an ML data-driven model. Such a process sometimes is called field inversion [68, 69].

Nonetheless, the objective of a sparse data-driven simulation is not solely a step in data-driven modeling. Instead, it can be used to obtain accurate results by assimilating sparse data into the model, specific to that particular problem [70–79]. The mathematical framework for such an approach is data assimilation (DA). Below we briefly review DA before discussing its application to our problem.

1.6.2.1 *Inverse problems and data assimilation*

We first define the inverse problem. Two problems are inverse with respect to each other if the formulation of one involves the solution of the other [80]. The problem that is easy to solve and is generally well-posed is called the forward problem. The one that is more difficult to solve and generally ill-posed[4] is called the inverse problem.

Regarding turbulence modeling, the forward problem is the case when we solve RANS or LES equations based on initial and boundary conditions. The inverse problem is the one that gets the sparse measurements as input and infers the state variables in the whole domain as output.

Data assimilation is defined as the estimation of state variables of a system by combining the observation and a model of that system through tuning parameters. There are generally two approaches: variational DA and statistical DA. Both approaches compare the model output and measurements based on a form of optimization that looks for an optimal match between the model's output and the measurements.

The variational approach utilizes an optimization process based on a classical form that minimizes a cost function that measures the discrepancy between the model and the measurements. On the other hand, the statistical approach uses an optimization form that minimizes the variability or uncertainty of the model error based on statistical estimation theory [80].

1.6.2.2 *Statistical data assimilation*

If we deal with inexact data based on e. g. noisy sensor measurements, and if we want to include these variabilities in our solutions, a statistical DA approach is a proper choice. To explain the basics of such an approach, we start with a very simple system. Assume a system whose description only needs one time-independent state variable. The simplest statistical DA

4 A problem is well-posed if there exists a solution, the solution is unique, and the solution continuously depends on the input data. A problem that does not have these characteristics is an ill-posed problem

model for this system is called the best linear unbiased estimator (BLUE), which reads

$$x^a = x^b + \frac{1}{1+\alpha}\left(x^o - x^b\right), \tag{1.25}$$

where x^a is the analysis value, x^b is the model forecast (background value), and x^o is the measurement value. $\alpha = \sigma_0^2/\sigma_b^2$ is the square of the ratio of the observation variance and the model variance (that are uncorrelated).

We now can generalize such a statistical model to include temporal and spatial complex systems like turbulent flows by more advanced approaches such as Kalman Filter (KF) methods. The KF method is a sequential DA scheme with two steps: a prediction/forecast step and a correction/analysis step. A KF method iteratively computes an optimal a posteriori estimate $\boldsymbol{x}_k^a$ which is a linear combination of an a priori estimate $\boldsymbol{x}_k^f$ and a weighted difference between the measurement $\boldsymbol{y}_k$ and the model prediction $\mathrm{H}_k[\boldsymbol{x}_k^f]$ at measurement points [80]. Thus it reads

$$\boldsymbol{x}_k^a = \boldsymbol{x}_k^f + \mathrm{K}_k\left(\boldsymbol{y}_k - \mathrm{H}_k\boldsymbol{x}_k^f\right), \tag{1.26}$$

where K_k is the Kalman *gain* and $\left(\boldsymbol{y}_k - \mathrm{H}_k\boldsymbol{x}_k^f\right)$ is called *innovation*. The Kalman gain matrix K is iteratively updated to minimize a posteriori error covariance equation. With some maths, we can obtain the formulation of optimal Kalman gain at each correction step as follows [80]

$$\mathrm{K}_k = \mathrm{P}_k^f\mathrm{H}^T\left(\mathrm{H}\mathrm{P}_k^f\mathrm{H}^T + \mathrm{R}\right)^{-1}, \tag{1.27}$$

where P_k^f is the forecast error covariance matrix and R is the covariance matrix of measurement errors.

One should note that the propagation of an $n \times n$ covariance matrix at each time step is very costly. One may use some sub-optimal analytical schemes or follow ensemble approaches to remedy this. Another issue with regular KF models is that they do not work for nonlinear processes or measurement operators. Two solutions involve methods with some linearizations called *extended Kalman filter* (EKF) [81] or using *particle filter* [82] methods. *Ensamble Kalman filter* (EnKF) [83], though based on Gaussian assumptions, can handle nonlinearities and non-Gaussian statistics where an ensemble of time-dependent states describes the probability density functions (PDF).

The issue with EKF is that the estimator is not optimal. In addition, the filter can easily diverge due to its linearization, if the process is not modeled correctly. The particle filter method also is a Monte Carlo algorithm that relaxes the assumption of linearity and Gaussian PDF and directly uses the recursive formula of the sequential Bayesian framework. It is very well suited for highly nonlinear problems such as turbulent flows. However, their value in practice is limited, since, like many sampling-based algorithms, their efficiency rapidly diminishes when the dimension of the state space increase, which is called "the curse of dimensionality" [84].

For practical cases of data-driven simulation of turbulent flows (our topic of interest), an explicit system covariance matrix is not available [85]. In addition, the process is highly nonlinear. Therefore, standard KF methods are not widely used. Instead, sample-based methods such as EnKF methods are preferable. However, if one wants to tune a large set of parameters to effectively improve the model prediction (having enough freedom to deviate from the baseline model towards the ground truth, e. g., tuning the eddy viscosity field, which has the same size as the computational grid), such sample-based methods will be very time-consuming. Nonetheless, KF methods have been used in the context of RANS models when the number of parameters is limited [86–91]. Recently, the Ensamble Kalman method was used to assimilate disparate data into an eddy viscosity model to improve the predictions and also for the field inversion [92, 93].

If one is not particularly interested in the higher statistical moments of the solution, such as PDFs or uncertainties and confidence intervals, and if the problem is not time-dependent, the variational approach is a better choice.

1.6.2.3 *Variational data assimilation*

Historically developed for numerical weather prediction models (NWP), variational DA is based on optimal control theory and is derived from the calculus of variations. Contrary to statistical methods, it minimizes a cost function instead of maximizing a PDF. First, it needs a proper definition of the cost function. Second, it involves numerical optimization techniques, which rely on the gradient of the cost function. Third, the gradient usually

is computed using *adjoint methods*. We define the following general form of a cost function (4D-Var) [94]

$$f(\boldsymbol{x}_o) = \frac{1}{2}\left(\boldsymbol{x}_0 - \boldsymbol{x}^b\right)^T \mathrm{B}^{-1}\left(\boldsymbol{x}_0 - \boldsymbol{x}^b\right) + \frac{1}{2}\sum_{i=0}^{N}\left(\mathrm{H}_i\boldsymbol{x}_i - \boldsymbol{y}_i\right)^T \mathrm{R}_\mathrm{i}^{-1}\left(\mathrm{H}_i\boldsymbol{x}_i - \boldsymbol{y}_i\right), \tag{1.28}$$

where $\boldsymbol{x}_o$, $\boldsymbol{x}^b$, and $\boldsymbol{y}_i$ are the model state at time t_0, the background state at time t_0, and measurements at time t_i, respectively. H_i is the observation matrix at time t_i, and R_i and B are the observation error covariance matrix at time t_i and the background error covariance matrix, respectively. The goal is then to minimize $f(x_0)$. For our purpose, sparse-data driven simulation of turbulent flows, we simplify the cost function as following [95]: we assume the problem is stationary (3D-Var), we neglect an explicit observation noise model, i. e., we set R_i to be the identity matrix, and we neglect prior modeling of the initial state resulting in the omission of the first term in equation (1.28). The simplified cost function reads

$$f(\boldsymbol{x}) = \frac{1}{2}\left(\mathrm{H}\boldsymbol{x} - \boldsymbol{y}\right)^T\left(\mathrm{H}\boldsymbol{x} - \boldsymbol{y}\right). \tag{1.29}$$

The minimization of f is based on a combination of an adjoint method for calculating the gradient ∇f with a descent algorithm in the direction of the gradient (gradient descent algorithms).

Derivative-free optimizations such as Genetic or PSO algorithms become very inefficient for a large number of parameters, as is the case for our problem. They may be useful if the cost function is not differentiable. Therefore, they are not the choice for DA. Gradient descent algorithms are essentially first-order iterative optimization algorithms for finding a local minimum of a differentiable function. Of course, due to its first-order nature, the convergence could be slow and easily trapped in local minima. Pure Newton's second-order methods can result in much faster convergence, but evaluating the second-order derivatives, the Hessian matrix, is very costly. Quasi-Newton methods such as BFGS are alternatives in which the Jacobian is directly computed, but the Hessian matrix is approximated using first-order derivative information, which can be an efficient trade-off between performance and speed. There are a variety of other first-order optimization algorithms that can provide a better convergence rate, mostly developed in the context of supervised machine learning. We will discuss these methods in more detail later. The general form of optimization is

$$\boldsymbol{\alpha}^{new} = \boldsymbol{\alpha}^{old} - \Delta \cdot \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}, \tag{1.30}$$

where $\boldsymbol{\alpha}$ is the parameter vector of the problem (parameters can be of any kind such as physical properties, flow properties, initial or boundary conditions, etc.), Δ defines the step size (can be either a full matrix, a diagonal matrix or a scalar value). The process described by equation 1.30 is repeated until a minimum is reached.

1.6.2.4 *Gradient computations and adjoint method*

The computation of the gradients is an important topic since it is by far the most computationally expensive part of the whole variational DA process. Before we explain adjoint methods, we briefly review the alternatives. The first approach is the brute-force finite difference (FD) method. For example, a first-order forward scheme can be used to evaluate the cost function gradient w.r.t. parameter α_i as follows

$$\left.\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}_i}\right|_{\mathrm{FD}}^{b} = \frac{f\left(\boldsymbol{\alpha}^{(i)}\right) - f\left(\boldsymbol{\alpha}^{b}\right)}{\epsilon}\,, \tag{1.31}$$

where $\boldsymbol{\alpha}^b$ is the baseline parameter vector (the current or initial parameter values), and $\boldsymbol{\alpha}^{(i)}$ is the parameter vector obtained from $\boldsymbol{\alpha}^b$ by replacing $\boldsymbol{\alpha}_i^b$ with $\boldsymbol{\alpha}_i^b + \epsilon$ (note that ϵ is a perturbation). $f\left(\boldsymbol{\alpha}^b\right)$ is the cost function computed by running the forward problem (e. g. RANS simulation) with the baseline parameters and $f\left(\boldsymbol{\alpha}^{(i)}\right)$ is the one computed with the perturbed parameter vector. As we see, for n parameters we need $n+1$ forward simulations, which would be prohibitive for RANS or LES simulations with many parameters. In addition, the FD method's accuracy is sensitive to the perturbation size. The complex-step method [96] can solve this issue, but the computational costs scale with the number of parameters similar to the FD method.

Another important approach is the use of Automatic Differentiation (AD). AD, similar to the FD method, can be used as a black-box approach, but the computational cost can be independent of the number of parameters. It is based on the systematic application of the differentiation chain rule to computer programs [97]. One approach to implement AD is through operator overloading, which involves a process known as "taping". Taping records all the partial derivatives in intermediate steps in a new user-defined type that is used instead of real numbers (including the value of the original variable as well as its derivative) and then performs the reverse mode calculations [98, 99]. Another approach is the source code transformation

which gets the original program and generates a new program to perform the calculations. Some applications of AD in CFD can be found in [100–102].

For our purpose, the forward problem involves an iterative solver, and because we need to save all operations in memory for reverse AD, such a naive use of AD would not be efficient. Instead, AD could be used for the computation of terms used in the adjoint method, resulting in low computational cost and memory efficiency at the same time [98]. Another issue with AD is that it is not built-in in most CFD codes.

A computationally efficient way to compute gradients is the adjoint method. It has been widely used in optimal control and design; both are relevant for CFD problems such as aerodynamic shape optimizations and data assimilation. There is a dichotomy between *discrete* adjoint method [99, 103] and *continuous* adjoint method [104, 105]. In the former, the governing equations are discretized, and then the adjoint equations are constructed and solved. In the latter, the adjoint equations are first analytically derived from the governing equations and then discretized and solved. The early works were mostly based on the continuous adjoint approach. The drawback of the continuous approach is low accuracy on coarser meshes and implementation challenges, especially for the boundary conditions [106]. Most of the recent works are based on discrete adjoint methods. The main advantage of discrete approaches is that the partial derivatives that appear in the equations need not be differentiated by hand and can be computed by other methods such as AD [103]. The boundary conditions also are automatically considered consistent with the implementation of the discretized form of the governing equations. Therefore, the discrete approach is more "automatic" than the continuous approach and thus easier to implement. drawbacks of the discrete approach are the computational and memory costs.

The main idea in the discrete adjoint method is to use the discretized PDE and the chain rule to set up a system of algebraic equations leading to the computation of the gradients. The associated cost is independent of the size of the parameters. The gradient can be expanded using the chain rule as follows

$$\frac{\mathrm{d}f}{\mathrm{d}\alpha} = \frac{\partial f}{\partial \alpha} + \frac{\partial f}{\partial U}\frac{\partial U}{\partial \alpha}, \tag{1.32}$$

where U is the state variables (velocity and pressure). The first right-hand side term is zero unless there is an additional term explicitly containing parameters in the cost function, e. g. a regularization term. $\frac{\partial f}{\partial U}$ can easily be computed analytically. $\frac{\partial U}{\partial \alpha}$, however, does not exist explicitly. However, the

state variables and parameters are implicitly related through the forward problem equations $R(U, \alpha) = 0$. Since $R(U, \alpha)$ is zero everywhere, its total derivative with respect to α is also zero everywhere, from which we obtain the following relation:

$$\frac{\mathrm{d}R}{\mathrm{d}\alpha} = \frac{\partial R}{\partial \alpha} + \frac{\partial R}{\partial U}\frac{\partial U}{\partial \alpha} = 0 \Rightarrow \frac{\partial U}{\partial \alpha} = -\left(\frac{\partial R}{\partial U}\right)^{-1}\frac{\partial R}{\partial \alpha}. \tag{1.33}$$

Thus the gradient can be formulated as

$$\frac{\mathrm{d}f}{\mathrm{d}\alpha} = \frac{\partial f}{\partial \alpha} - \lambda^T \frac{\partial R}{\partial \alpha}\,, \tag{1.34}$$

where

$$\left(\frac{\partial R}{\partial U}\right)^T \lambda = \left(\frac{\partial f}{\partial U}\right)^T\,. \tag{1.35}$$

The components of the vector λ are called adjoint variables, and the equation (1.35) can be solved iteratively in the same way as the forward problem. The term $\frac{\partial R}{\partial U}$ can be computed by solving the forward problem, and the term $\frac{\partial f}{\partial U}$ can be calculated analytically. The term $\frac{\partial R}{\partial \alpha}$, however, needs special treatment since, usually, the parameter does not explicitly appear in the governing equations. FD, AD, or other methods can be employed to compute this term. In the next chapters, we will discuss this issue and other aspects, such as the implementation of the code.

1.6.2.5 *Regularization*

Regularization is defined as incorporating any information or prior knowledge into the inverse problem to overcome the ill-posedness, prevent overfitting, and iregular solutions. We can divide the procedure into two ways: explicit and implicit regularization. The explicit procedures add priors, penalties, or constraints to the optimization problem. The most common way is to add penalties to the cost function. Implicit regularization consists of other forms of attempts to add information to the problem, such as early stopping. Regularization is extremely important in sparse data-driven simulation of turbulent flows since such an inverse problem is under-determined in that the number of parameters can be much larger than the number of measurements.

The ill-posedness implies that we may not arrive at any solution; even if so, there can be multiple solutions, all of them somewhat valid candidates. This is particularly the case for sparse-driven simulation, as we will see,

in that many combinations of parameter values would end up with the same cost function value (flat cost function landscape). However, we can refute many of these solutions with some expert knowledge. Specifically, any solutions that are not to some degree smooth and contain peaks and noises should be eliminated through regularization. Promoting local spatial correlation thus would be one attempt to get physical results (e. g. total variation regularization). Low pass filtering could be another option (Sobolev gradient method). Reduction of dimensionality of the parameters space is another way of adding prior knowledge. Constraining the problem through physical models, such as the Boussinesq eddy viscosity assumption, reduces the degree of freedom of the problem and can be seen as regularization. Introducing additional artificial or synthetic data can also be seen as a form of regularization. The concept of overfitting and irregular solutions is also somewhat relevant here since, as we will see, the optimizer mostly tends to match the solution close to the measurements at the cost of significant deviation in other regions. In that sense, regularization also avoids overfitting and irregular solutions in those regions.

1.6.2.6 *The dynamics between the availability of data and level of modeling and regularization*

One important aspect of data-driven simulations is that data availability dictates the level of modeling and the necessity of regularization. When only very sparse measurements are available, one needs further modeling and further regularization. This is not obvious beforehand, and one needs to decide based on trial and error or some expert experience. Of course, a low amount of data needs more constraints, leading to a form of *rigidity* [77] that does not allow the optimizer to thoroughly search the solution space and eventually get close to the ground truth. On the other hand, giving too much freedom to the optimizer without enough constraints increases the chance of trapping in local minima corresponding to unphysical results. In this sense, explicit regularization may help to avoid such an issue. More dense data may overcome the need for too much regularization and let us use a larger set of parameters, e. g., assuming the Reynolds stress tensor field is the parameter field instead of eddy viscosity in an LEV model. This aspect can be called *flexibility* [77]. As we see, this dynamic is not obvious prior to the process and needs special treatment and experimentation. A large part of this thesis is thus dedicated to these matters.

1.6.2.7 *Reduced order modeling (ROM) and Physics-informed neural network (PINN)*

For the sake of generality, we briefly mention two important approaches to tackle turbulent flow problems, into which data can easily be embedded. Reduced Order Modeling (ROM) is an approach that the original full-order model is replaced with a reduced-order model, which significantly reduces the dimensionality with a controlled loss of accuracy [107]. Proper orthogonal decomposition is used for projecting PDE dynamics to low-rank subspaces that can model the full spatio-temporal evolution of the governing complex system [108]. The dimension reduction is somehow similar to the goal of turbulence modeling, but here conducted in a different way, and usually, the dimension reduction is much more severe. Data can then easily be incorporated to construct such ROMs. Data-driven ROMs are extensively used in flow modeling and control [109–111].

Physics-informed Neural Network (PINN) is a new paradigm for solving forward and inverse problems that involve PDEs [112]. The idea is to train an ML model as a global approximator that has space and time as input and gives the state variables at that location and time as output. The training data here would be samples from the initial and boundary conditions. The cost function is defined as the error of the approximator at the samples of the initial and boundary conditions. The gradient of the cost function with respect to the weights and biases of the neural network is computed using efficient algorithms such as error backpropagation. However, it is virtually impossible to train a physically reasonable model merely based on initial and boundary conditions through such an inverse problem. Therefore, sampling should also be done within the domain and at different times. Since the state variables of these samples usually are not known (data is not available within the domain), instead, the residuals of the PDEs at those sample points are assumed to be zero. Therefore, in the cost function, a penalization term is added that estimates the sum of residuals at the sampling. AD here is used to compute the residual of the PDEs. A trained ML thus should satisfy the initial and boundary conditions as well as the PDEs within the domain. Therefore, we can say that PINN solves a forward problem in an inverse approach without a need for discretization schemes and the generation of a mesh. In this paradigm incorporating data would be very easy since any measurement information within the domain can be added to the cost function exactly the same way as samples of boundary and initial conditions. Parameters of a parametric PDE can also be tuned with the help of AD in the same way the weights and biases are tuned.

We see that data-driven simulation can be performed automatically in PINN without a need for implementing tedious codes of DA. However, there are challenges with PINN that need to be addressed. For example, solving the forward problem with an inverse approach, where the cost is minimized with first-order or quasi-Newton optimization methods, would be very difficult. The training phase would be very costly, and sometimes a satisfactory convergence may not be achieved. In addition, the sampling would be tricky and needs treatments similar to the issues with generating a proper mesh in a traditional CFD. The advantage of the discrete adjoint method to PINN is that the RANS equations are solved in a forward way more efficiently, and then the gradients are computed from the forward solution.

1.6.2.8 *Conclusions on data-driven approaches in turbulence modeling*

We have mentioned two general approaches, data-driven modeling, and data-driven simulation. Classical data assimilation and newer machine learning methods have been used to incorporate data into models. The main issues with ML-based data-driven models are generalizability and interpretability. For the case of sparse data, we have observed that the inverse problem is severely ill-posed and thus needs further modeling and physical insights to obtain correct solutions. We can conclude that both physics-informed and data-driven approaches are needed to obtain desired results. In addition, the problem of non-uniqueness is an important task that needs to be addressed.

1.7 MOTIVATION

The computational cost of DNS and LES for highly turbulent flows is still a major obstacle to their usage for many applications. RANS or hybrid LES/RANS methods will still be the only feasible options for practical applications in near future. We investigate the possibility of incorporating sparse or low-resolution experimental measurement data into to model to compensate for the inadequacies of these methods. If sparse data-augmented RANS or hybrid LES/RANS methods can reproduce accurate solutions, specifically velocity profiles and wall shear stresses, then such data-driven models can be used instead of costly LES and DNS for any applications for some measurement data are available.

However, we will see such an inverse problem is severely ill-posed and we have to deal with the problem of non-uniqueness solutions. The first

question to answer is how much prior modeling, e.g. linear eddy viscosity assumption, needs to be enforced to effectively reduce the degrees of freedom and thus the ill-conditioning of the problem. Then we need to impose different regularization strategies to avoid non-physical solutions, especially those jagged noisy over-fitted profiles. Therefore, we study how much smoothness would be necessary and how to impose them. We also investigate if dimensionality reduction can be used as a regularization strategy to avoid noisy solutions.

If we are successful, we can show that if a data-driven model with a specified physical constraint can reproduce correct solutions, in theory, that level of physical modeling constraint has the potential of providing correct solutions without the use of any data, if the physical model can correctly compute the parameters.

We also investigate the case where only sparse wall shear stress measurements are available. An example of such a condition would be a digital twin of an airplane where only near-the-wall measurements are accessible. Can we reconstruct a correct solution only based on a few wall shear stress data? if not, how can we overcome the problem? for example, does data augmentation in the regions where data is not available, using higher fidelity simulation, provide accurate solutions?

To summarize, we try to improve RANS models by assimilating sparse data. We investigate and propose different regularization strategies to solve the inverse problem and reach the most physical solution among many possible solutions.

1.8 OUTLINE

In this chapter, we briefly reviewed all relevant subjects of turbulence modeling and data-driven approaches. In the end, we shortly discussed the discrete adjoint method's role in variational data assimilation. In chapter 2, we formulate the problem more rigorously. First, we formulate the forward problem and then the inverse problem. Next, we provide background reasons why the inverse problem is formulated this way. Afterward, the computation of gradients is discussed in detail. The way the code is implemented in OpenFOAM is explained in detail here. Verification of the gradient computation and the whole procedure is done based on the full field of synthetic data. In chapter 3 we provide a more realistic case of sparse data and discuss the role of regularization techniques. We will see that wall shear stress, which is an essential quantity of interest, cannot be

physically reconstructed from sparse measurements if conventional regularization techniques are used. We, therefore, propose a dimensionality reduction approach as a regularization method to provide both velocity profiles and wall shear stresses at the same time. Finally, in chapter 4 we examine a case where only wall shear stress data is given, while accurate velocity profiles, as well as wall shear stresses, are sought. In this case, we use an accurate, but computationally efficient hybrid LES/RANS method to generate artificial data in regions where data is unavailable. We will see that such an approach is an efficient way to simulate turbulent flows with only a few wall shear stress measurements. Finally, conclusions are made, limitations of the work are discussed, and possible future works are proposed.

2

SPARSE DATA-DRIVEN RANS SIMULATION

In this chapter, the mathematical framework for RANS simulation driven by sparse measurements is explained. Specifically, we explain how we compute the gradient of the cost function, based on the discrete adjoint method. The implementation in OpenFOAM is discussed and the case setup for DA is described. The verification process is shown later. Finally, the results of the DA process without any regularization for synthetic and also real data are shown. The necessity for regularization is explained, which itself is then the topic of the next chapter. The discrete adjoint method in OpenFOAM and the optimization method in Python were developed in collaboration with Oliver Brenner. The post-processing scripts were developed by Oliver Brenner. Some results of this chapter have been published in:

Oliver Brenner, Pasha Piroozmand, and Patrick Jenny. Efficient assimilation of sparse data into RANS-based turbulent flow simulations using a discrete adjoint method. *Journal of Computational Physics* **471**, 111667 (2022).

2.1 FORMULATION OF THE FORWARD PROBLEM

As explained in the introduction, we formulate the data assimilation task as an optimization problem where the discrepancy between the RANS solution and the sparsely distributed measurement data is minimized under the constraint of the linear eddy viscosity RANS equations. We change the RANS equations into a parametric PDE form and find the optimal estimation of the parameters (here the eddy viscosity scalar field) in the optimization process. Einstein notation is used throughout this chapter. First, we introduce the steady-state RANS equations for an incompressible Newtonian fluid

$$\frac{\partial\left(\langle u_i\rangle\langle u_j\rangle\right)}{\partial x_j}+\frac{\partial\tau_{ij}}{\partial x_j}=-\frac{\partial\langle p\rangle}{\partial x_i}+\frac{\partial}{\partial x_j}\left(\nu\frac{\partial\langle u_i\rangle}{\partial x_j}\right), \tag{2.1}$$

where $\langle u_i\rangle$ are the mean velocity components, $\langle p\rangle$ is the mean kinematic pressure assuming a constant density ρ, and ν the constant kinematic viscosity. The Reynolds-stress tensor $\tau_{ij}=\left\langle u_i'u_j'\right\rangle$ is responsible for momentum

transfer by velocity fluctuations. This unclosed term needs modeling due to the loss of information through the averaging process. The above equations are solved together with the continuity equation

$$\frac{\partial \langle u_i \rangle}{\partial x_i} = 0 \tag{2.2}$$

to obtain the mean velocity and pressure fields. Due to the reasons explained in the introduction, we follow the Boussinesq hypothesis which constructs a linear constitutive relationship between the deviatoric part of the Reynolds stress and the mean flow strain rate, in particular

$$\tau_{ij} = \frac{2}{3} \langle k \rangle \delta_{ij} - 2\nu_t \langle S_{ij} \rangle \ , \tag{2.3}$$

where δ_{ij} is the Kronecker delta, $\langle S_{ij} \rangle = \frac{1}{2} \left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right)$ is the mean flow strain rate, and $\langle k \rangle = \frac{1}{2} \langle u_i' u_i' \rangle$ is the half-trace of the Reynolds stress tensor, called turbulent kinetic energy. The first term of the right-hand side is the hydrostatic part of the Reynolds stress and the second term is the deviatoric part. In practice, the hydrostatic part is absorbed into the pressure gradient term in equation (2.1) and the kinematic pressure $\langle p \rangle$ is replaced with the modified kinematic pressure $\langle p^* \rangle = \langle p \rangle + \frac{2}{3} \langle k \rangle$. Therefore, $\langle k \rangle$ need not be computed directly. The proportionality factor ν_t is the eddy viscosity that has to be evaluated locally in each cell. In our data-driven approach, we could compute ν_t directly from sparse measurement data. However, it is better to first initialize the ν_t field with a conventional LEV model, such as the $k - \varepsilon$ model, and then correct the computed eddy viscosity with a corrective field. This leads to better-conditioned inverse problems, where the initial correction factor α is unity [63]. It can also help to incorporate prior knowledge by using e. g. L_2 regularization where the deviation from the initial computation of the eddy viscosity is penalized leading to more physical results. The resulting governing parametric PDE reads

$$\frac{\partial \left(\langle u_i \rangle \langle u_j \rangle \right)}{\partial x_j} = -\frac{\partial \langle p^* \rangle}{\partial x_i} + \frac{\partial}{\partial x_j} \left[2 \left(\nu + \alpha \nu_t^b \right) \langle S_{ij} \rangle \right] \ , \tag{2.4}$$

where ν_t^b is the baseline eddy viscosity computed by an eddy viscosity RANS model. For simplicity of exposition, we will use a simplified notation in the remainder of this manuscript. In particular, we will refer to the modified kinematic pressure as pressure p, the mean velocity components as $\{u_1, u_2, u_3\}$, and the vector of state variables as $U = \{u_1, u_2, u_3, p\}$.

2.2 FORMULATION OF THE INVERSE PROBLEM

An efficient way to estimate the optimal values of $\boldsymbol{\alpha}$ is to follow a first-order iterative optimization procedure. The cost function f of the optimization problem is given by the discrepancy between the simulation results and the reference data at the measurement locations. This difference is defined as the squared Euclidean distance between the two as

$$f = \hat{\boldsymbol{V}} \cdot \sum_{k=1}^{4} \left(\mathbf{H}\boldsymbol{U}_k - \hat{\boldsymbol{U}}_k\right)^{\circ 2} , \tag{2.5}$$

where $\boldsymbol{U}_k$ is the vector of k-th state variables computed in each cell, $\mathbf{H}$ the linear observation operator [80] (a $R \times N$ matrix with R the number of measurement locations), $(\cdot) \cdot (\cdot)$ is the inner product, $(\cdot)^{\circ 2}$ is the Hadamard power of two (element-wise square of the vector), $\hat{\boldsymbol{U}}_k$ is the vector of k-th measured state variables and $\hat{\boldsymbol{V}}$ is the vector of cell volumes at specified reference locations. In practice, one may only measure a single component of the state variables, e. g. the horizontal velocity component u_1, resulting in

$$f = \hat{\boldsymbol{V}} \cdot \left(\mathbf{H}\boldsymbol{u}_1 - \hat{\boldsymbol{u}}_1\right)^{\circ 2} , \tag{2.6}$$

where the components of $\boldsymbol{u}_1$ and $\hat{\boldsymbol{u}}_1$ are the calculated and measured horizontal velocities, respectively.

Knowing the sensitivity of the cost function f with respect to the parameter vector $\boldsymbol{\alpha}$ allows us to iteratively optimize the parameter values with a gradient descent algorithm as

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - \Delta^{(t)} \left(\frac{\mathrm{d}f^{(t)}}{\mathrm{d}\boldsymbol{\alpha}}\right)^{T} , \tag{2.7}$$

where $t = 0, \ldots, T_{max}$ denotes the iteration number of the optimization, with T_{max} being the maximum number of iterations and Δ the learning rate or optimization step size. Next, we discuss how to obtain the cost function gradient $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$, while the choice of Δ, which can be either a scalar or a matrix, is discussed later.

2.3 SENSITIVITY CALCULATION: DISCRETE ADJOINT METHOD

The most straightforward way to compute the sensitivity of the cost function f with respect to the parameter vector $\boldsymbol{\alpha}$ is the brute-force finite difference

(FD) method, where each parameter is perturbed specifically while the change in the cost function is measured [113]. However, this requires M evaluations of the forward problem, where M is the number of parameters (length of the vector $\boldsymbol{\alpha}$). Note that M is equal to the number of cells N if each cell is allowed to attain an individual α value. In addition to the large computational cost of the method, the obtained sensitivities very much depend on the perturbation amplitude. The complex-step method [96] can circumvent this issue but the computational cost still scales with the number of parameters for both methods.

Another option is to apply automatic differentiation (AD) in a reverse mode to the forward problem solver code [114]. This way, the number of required forward problem evaluations is independent of the number of parameters. However, this approach is associated with an enormous memory requirement to store all intermediate steps of the iterative forward solver.

Instead of these *black-box* approaches, one can use the discrete adjoint method to calculate the sensitivity based on the chain rule at low computational cost and memory requirements. The product rule for the sensitivity reads

$$\underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}}_{1\times M} = \underbrace{\frac{\partial f}{\partial \boldsymbol{\alpha}}}_{1\times M} + \underbrace{\frac{\partial f}{\partial \boldsymbol{U}}}_{1\times 4N} \underbrace{\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}}_{4N\times M} , \tag{2.8}$$

where $\boldsymbol{U}$ is the vector of all four state variables in all N cells. The term $\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}$ is a $4N \times M$ matrix that qualifies the sensitivities of the state variables in each cell (rows) with respect to the parameters in each cell (columns). While vectors $\frac{\partial f}{\partial \boldsymbol{\alpha}}$ and $\frac{\partial f}{\partial \boldsymbol{U}}$ can be explicitly computed, there is no explicit relation between the state variables and the parameters to obtain $\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}$. However, $\boldsymbol{U}$ and $\boldsymbol{\alpha}$ are implicitly related through the forward problem equations $\boldsymbol{R}(\boldsymbol{U}, \boldsymbol{\alpha}) = \boldsymbol{0}$. Since by definition $\boldsymbol{R}$ is identically zero, and so is its total derivative with respect to $\boldsymbol{\alpha}$. Expanding the total derivative of $\boldsymbol{R}$ thus yields

$$\underbrace{\frac{\mathrm{d}\boldsymbol{R}}{\mathrm{d}\boldsymbol{\alpha}}}_{4N\times M} = \underbrace{\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}}_{4N\times M} + \underbrace{\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}}_{4N\times 4N} \underbrace{\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}}_{4N\times M} = 0 . \tag{2.9}$$

These terms are rearranged to obtain an expression for $\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}$, which is then substituted into equation (2.8) to give

$$\underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}}_{1\times M} = \underbrace{\frac{\partial f}{\partial \boldsymbol{\alpha}}}_{1\times M} - \overbrace{\underbrace{\frac{\partial f}{\partial \boldsymbol{U}}}_{1\times 4N}\underbrace{\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}^{-1}}_{4N\times 4N}}^{\boldsymbol{\lambda}^T}\underbrace{\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}}_{4N\times M} = \underbrace{\frac{\partial f}{\partial \boldsymbol{\alpha}}}_{1\times M} - \underbrace{\boldsymbol{\lambda}^T}_{1\times 4N}\underbrace{\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}}_{4N\times M}, \tag{2.10}$$

with the adjoint variables, or Lagrangian multipliers, $\boldsymbol{\lambda}$. These are computed by evaluating the so-called *adjoint* equations

$$\underbrace{\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right)^T}_{4N\times 4N}\underbrace{\boldsymbol{\lambda}}_{4N\times 1} = \underbrace{\left(\frac{\partial f}{\partial \boldsymbol{U}}\right)^T}_{4N\times 1}. \tag{2.11}$$

This system of equations is solved iteratively, similar to the solution of the forward problem. Note that the system matrix and the right-hand side are both independent of $\boldsymbol{\alpha}$. Therefore, the computational cost of solving the adjoint equations is independent of the number of parameters $\boldsymbol{\alpha}$ and in the order of a single linearized forward problem evaluation. This is a great advantage compared to, e. g., the finite difference method which needs M forward problem evaluations.

As stated, derivative vectors of $\frac{\partial f}{\partial \boldsymbol{\alpha}}$ and $\frac{\partial f}{\partial \boldsymbol{U}}$ can be computed analytically, but the matrices $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}$ and $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}$ need to be evaluated in a different way. The reverse mode of AD can be used for this purpose. Naive use of AD, however, leads to large memory usage. There are numerous approaches to circumvent this issue [100, 106, 115]. Here, instead of using AD, we follow a simpler approach with some approximating assumptions. We use the fact that the discretized equations, e. g. by applying the finite volume method, have the form

$$\underbrace{\boldsymbol{R}(\boldsymbol{\alpha}, \boldsymbol{U})}_{4N\times 1} = \underbrace{\boldsymbol{A}}_{4N\times 4N}\underbrace{\boldsymbol{U}}_{4N\times 1} - \underbrace{\boldsymbol{b}}_{4N\times 1} = \underbrace{\boldsymbol{0}}_{4N\times 1}, \tag{2.12}$$

where $\boldsymbol{A}$ is the system matrix of the linearized and discretized residual. Therefore, the Jacobian $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}$ is simply equal to $\boldsymbol{A}$.

Notice that $\boldsymbol{A}$ is a $4N \times 4N$ matrix whose elements account for the implicit influence of each state variable on the corresponding residual equations. Such terms also arise from the velocity-pressure coupling in the forward problem. Such a coupling is often treated by segregated methods such as the *SIMPLE* algorithm. Segregated methods do not consider all

off-diagonal components $\boldsymbol{A}$. The coupling is accounted for in a *successive guess-and-correct* fashion [116]. Here, we use a fully coupled solver for both forward and inverse problems by which all off-diagonal elements of $\boldsymbol{A}$ are computed [117]. A further advantage of coupled solvers is faster convergence since the iterations are only needed for the nonlinear convective terms; albeit at the cost of higher memory requirements.

The computation of the remaining term $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}$ is tricky if one does not use AD or FD approaches. To address this issue, having converged forward solutions, we use implicit operators provided by *OpenFOAM* [118] to discretize and linearize the residual equations with respect to $\boldsymbol{\alpha}$. The resulting system matrix corresponds to $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}$ [119]. One should note that $\boldsymbol{\alpha}$ only appears explicitly in the momentum equations and not in the pressure equation. Therefore, the effect of the term $\frac{\partial \boldsymbol{R}_4}{\partial \boldsymbol{\alpha}}$ on $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$ is neglected. Brenner et al. [119] showed that this approximation introduces inaccuracies in the magnitude of the computed sensitivities, but in general the sign is correct, which is crucial for successful gradient-based optimization. It should be noted that such an approach cannot be used for other applications where the design variables are not explicitly present in the governing equations such as shape optimization problems.

2.4 IMPLEMENTATION

The computation of the gradients, which is the key part of the DA approach, is implemented in OpenFOAM. OpenFOAM is an open-source CFD package written in C++. The first reason for choosing OpenFOAM is that the computation of gradients necessarily involves solving the forward CFD problem thus CFD software is required. Second, there is no discrete adjoint method for DA already implemented in any CFD packages. Due to its open-source nature, OpenFOAM is one of the best-suited software to add and develop codes for the adjoint method. Particularly object-oriented programming paradigm of OpenFOAM makes developments much easier than other coding structures.

Once the gradients are computed, they are written to a file and are read by a Python script in which optimization methods were developed. The interactions between OpenFOAM codes and Python scripts were handled by PyFoam, which is a Python script acting as an interface between the two and especially makes the processing of OpenFOAM-style files much easier. The Python script reads the gradients and updates the parameters in the

direction of reducing the cost function. This process is repeated until some stopping criteria are satisfied.

2.4.1 *Implementation of the gradients computations in OpenFOAM*

It is common practice that one starts from an already existing OpenFOAM solver for a similar problem and then extends the code for the problem of interest. As our problem is limited to steady-state incompressible RANS flows, we start with `pUCoupledFoam` in *foam-extend-4.0*. The main parts of the forward problem code are briefly described below:

The flow and adjoint fields are defined in the file `createFields.H`. Then the block system matrix for velocity and pressure variables is defined and named `UpEqn`. The discretized form of momentum equations excluding the coupled term, the pressure gradient, is defined in the file `UEqn.H`. Likewise, the pressure equation excluding the coupled term, the divergence of velocity, is defined in the file `pEqn.H`. In the file `couplingTerms.H`, the coupled terms are then discretized and inserted into the block system matrix `UpEqn`. The block system matrix is solved and then the corresponding field is retrieved and the face flux field in the file `phi` is updated. This process is repeated in a `while` loop in the main file `pUCoupledFoam.C` and continues until the convergence is achieved.

We now can extend this code to compute also the gradients. The process is as follows: We define adjoint variables `Ua`, `pa`, and the parameter field `alpha` in the file `createFields.H` and a new file `frozenTurbulence.H`. A new file named `adjointSetup.H` is generated, in which the system matrix of the converged discrete residual equation of the forward problem is reconstructed. Then, each 4×4 sub-matrix is transposed by calling the function `T()`. In addition, each sub-matrix in the block system matrix is transposed. This is done by switching the elements of the upper-diagonal part (by calling the function `upper()`) with the lower-diagonal parts (by calling the function `lower()`). Another important point here is that OpenFOAM separates and stores the effect of boundary conditions in different arrays called `internalCoeffs` and `boundaryCoeffs`. `internalCoeffs` are the ones that affect the diagonal elements of the system matrix. In block systems, however, the `internalCoeffs` is already merged in `diag()` except for coupled boundaries including periodic boundaries. Therefore, this has to be transferred and transposed from the forward problem system matrix. From constructing this `UpaEqn` we actually obtain $\frac{\partial R}{\partial \overline{u}}$, the crucial element of the adjoint equations. The right-hand side of the adjoint equations, $\frac{\mathrm{d}f}{\mathrm{d}\overline{u}}$,

is simply obtained from the existing fields and added to the right-hand side of `UpaEqn` in its `source()`. The adjoint equation is then constructed and can be solved likewise. Next, the matrix $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}$ should be estimated and then multiplied to the adjoint variables to compute the gradient $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$. As explained, we ignore $\frac{\partial \boldsymbol{R}_p}{\partial \boldsymbol{\alpha}}$. The only term in $\boldsymbol{R}_{\boldsymbol{u}}$ containing the parameter α is [119]

$$\nabla \cdot (\alpha \nu_t \nabla U)\,, \tag{2.13}$$

which leads to

$$\frac{\partial R_U}{\partial \alpha} = -\frac{\partial}{\partial \alpha}\left(\nabla \cdot (\alpha \nu_t \nabla U)\right) = -\frac{\partial}{\partial \alpha}\left(\nabla \cdot \left((\nu_t \nabla U)\,\alpha\right)\right)\,. \tag{2.14}$$

Therefore, we need an implicit divergence operator acting on the scalar field $\boldsymbol{\alpha}$ with a tensor flux. The only existing implicit divergence operator in *foam-extend-4.0* is acting on a vector field (`volVectorField`), usually the velocity, with a scalar field (`surfaceScalarField`), usually `phi` the mass flow through the cell faces. It is called `vectorGaussDivScheme` and takes the form `fvm::div(phi, U)`. Here we extend this code to get a scalar field (`volScalarField`) and a tensor face field (`symmTensorSurfaceField`) in the form of `fvm::div(tensorField, scalarField)`, where `tensorField` is $\nu_t \nabla U$ interpolated to the faces and `scalarField` is α. Finally, the system matrix $\frac{\partial R_U}{\partial \alpha}$ is multiplied with λ to obtain the gradients.

2.4.2 *Implementation of the optimization procedure in Python*

A Python script based on the SciPy Python library was developed where the file containing the gradient `dfdA` is read and the updated α is computed based on the optimization algorithm. The script has the flexibility to use different optimization algorithms.

2.5 OPTIMIZATION

2.5.1 *Computation of the learning rate*

In equation (2.7), step size Δ can either be a scalar or a $R \times R$ matrix, depending on the optimization method. The simplest case is the vanilla gradient descent method, where Δ is a constant scalar, whose value should be tuned by trial and error. If the optimization method uses second-order derivatives (or a Hessian matrix), like Newton's method, Δ is a matrix.

Such methods lead to fewer optimization iterations. However, due to high computational costs, in general, Hessian matrices are not directly computed. The next approach would be the family of quasi-Newton methods such as BFGS, where the Hessian matrix is approximated using the gradient information [120].

In this work we use an adaptive learning rate method, where Δ is a diagonal matrix, i. e., for each element of the parameter vector, at each iteration, a specific optimization step size is employed. No correlations between parameters are assumed (zero off-diagonals). Based on our empirical tests we use the DEMON Adam [121] method. The default Adam method uses larger learning rates for those parameter elements that were less frequently updated with larger gradients. In addition, a decaying momentum (DEMON) term is added to accelerate convergence [122]. In summary, equation (2.7) is replaced by

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - \boldsymbol{\Delta}^{(t)}\, \boldsymbol{\gamma}^{(t)}\,, \tag{2.15}$$

where

$$\boldsymbol{\gamma}^{(t)} = \left(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}\right)^{T} + \kappa^{(t)}\, \boldsymbol{\gamma}^{(t-1)}\,. \tag{2.16}$$

The variable $\kappa^{(t)}$ is the weight of the decaying exponential moving average of past gradients (momentum) which is computed for each iteration as

$$\kappa^{(t)} = \kappa_0 \frac{\left(1 - \frac{t}{T}\right)}{(1-\kappa_0) + \kappa_0 \left(1 - \frac{t}{T}\right)}\,, \tag{2.17}$$

where κ_0 is the initial momentum value and T is the maximum number of iterations. The optimization step size of the r-th parameter of the vector of α is computed as

$$\boldsymbol{\Delta}_{r,r}^{(t)} = \frac{\eta}{\sqrt{\boldsymbol{\epsilon}_r^{(t+1)} + \delta}}\,, \tag{2.18}$$

where η is the nominal step size, $\delta > 0$ is a small constant, and

$$\boldsymbol{\epsilon}^{(t+1)} = \kappa' \boldsymbol{\epsilon}^{(t)} + \left(1 - \kappa'\right) \left(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}^{(t)} \circ \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}^{(t)}\right)^{T} \tag{2.19}$$

acts as a diagonal preconditioning matrix. Parameter κ' is the discount factor and $\circ$ stands for the Hadamard product (element-wise multiplication).

2.5.2 *Positivity constraint of the eddy viscosity*

Since the computed eddy viscosity should by definition be positive, it makes sense to incorporate this constraint into the optimization process. Brenner et al. [119] showed that imposing a hard constraint, e. g., by overriding negative α values, impairs the performance of the optimizer. Instead, they suggested using a mapping function that acts as a soft constraint to avoid negative values. The mapping reads

$$\alpha\left(\alpha'\right) = \begin{cases} \alpha' + 1, & \alpha' > 0 \\ \exp\left(\alpha'\right), & \alpha' \leq 0\,, \end{cases} \tag{2.20}$$

where α' is the *intermediatory parameter* and α is the *physical parameter*. The former can reach any real value, while the latter always is positive. Consequently, the sensitivities $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$ are modified through the chain rule as

$$\frac{\mathrm{d}f}{\mathrm{d}\alpha'} = \frac{\mathrm{d}f}{\mathrm{d}\alpha}\,\frac{\mathrm{d}\alpha}{\mathrm{d}\alpha'}\,. \tag{2.21}$$

The regularizations thus affect the *intermediatory parameter* α' rather than the *physical parameter* α.

2.6 CASE SETUP

The framework was developed in *foam-extend-4.0*, a version of *OpenFOAM* with implemented coupled solvers [118]. The proposed method was tested for flow over periodic hills, for which numerous numerical and experimental reference data are available at various Reynolds numbers. The problem, although considers a rather simple two-dimensional geometry, features separation and reattachment which are complex phenomena for eddy viscosity models. We are specifically interested in an accurate wall shear stress profile using data assimilation. The geometry of the problem is depicted in Figure 2.1. The fluid periodically flows from the left boundary over the hill and exits over the right boundary over the next hill. A mean bulk velocity U_b is maintained by imposing a pressure drop from the left to the right boundary; no-slip boundary conditions are applied at the top and bottom walls.

We consider flow at Reynolds number $Re = 10595$, based on the hillcrest height H and the bulk velocity U_b at the inflow boundary. Sparsely sampled averaged LES data by Gloerfelt and Cinnella [123] is considered as a reference.

The computational grid consists of 180×130 cells in streamwise (x) and spanwise (y) directions, respectively. A dimensionless wall distance of $y^+ \approx 1-5$ was maintained for the wall-adjacent cell center locations. The baseline eddy viscosity was computed using the standard $k-\varepsilon$ model with default coefficient values. The turbulent viscosity was fixed to zero at wall boundaries (low Reynolds boundary condition). Second-order schemes were used for the discretization of the convective and diffusive terms. A bi-conjugate gradient stabilized linear solver with a Cholesky preconditioner was used to solve the coupled system.

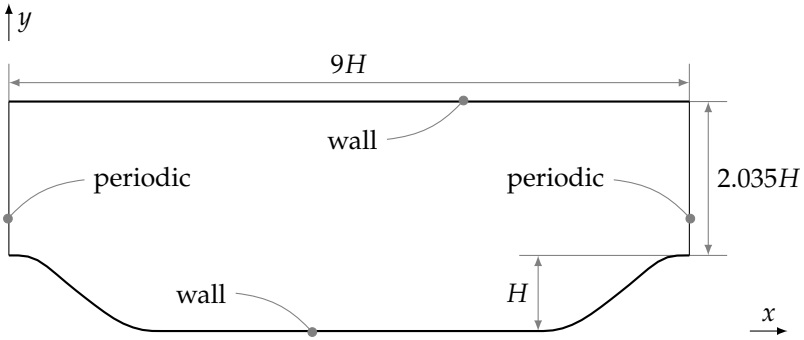


FIGURE 2.1: Periodic hills geometry with boundary conditions. All length scales are expressed with respect to the hill height H.

It should be noted that the large value for the maximum iteration number T (cf. Table 3.1) did not result in high computational cost, since α only slightly changed between optimization steps, and the solution fields were initialized with the results from the previous step. In addition, we chose a rather large value of 1e-4 for the forward problem convergence criterion which accelerates the process with a minor effect on the quality.

2.7 VERIFICATION OF THE GRADIENT COMPUTATIONS AND DATA ASSIMILATION

There are two important questions; whether the sensitivities are computed accurately and second whether the whole DA results in a correct solution. The first question can be answered, if the gradients are computed based on a different method that we are sure is correct. We can use the finite difference method in this regard. Note that the computational cost of FD is very high but for verification of only one optimization step it is feasible. First, we

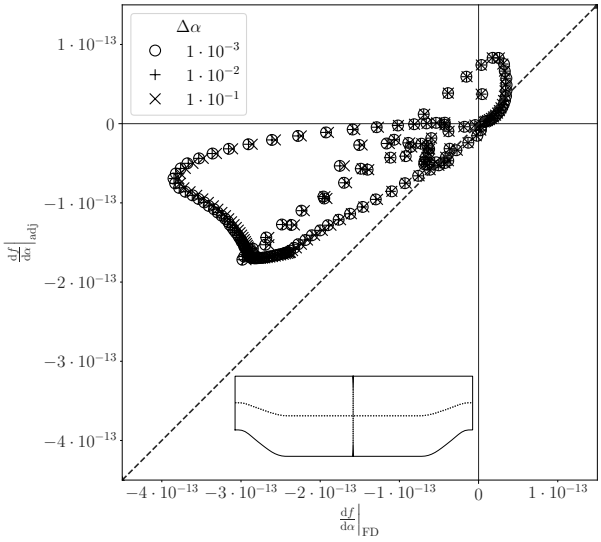


FIGURE 2.2: The gradients computed by FD for α_i in the cells i are marked by black crosses (bottom), which are compared with the gradients computed by the approximate adjoint method. Results are shown for the case of the periodic hills with full synthetic reference data for both velocity components with the initial value of unity for α and without any regularization. The figure is taken from reference [119].

apply the method for different step sizes. A too large step size for such a first-order scheme would result in inaccurate gradients and a too small step size would end up with round-off errors. First, we find an optimal step size of a finite difference approximation by performing the analysis with perturbation sizes $\epsilon \in \left[1 \cdot 10^{-3}, 5 \cdot 10^{-3}, 1 \cdot 10^{-2}, 5 \cdot 10^{-2}, 1 \cdot 10^{-1}\right]$ [119]. Brenner et al. [119] chose a subset of cells to reduce the computational costs. The sensitivity study and the location of the cells are shown in figure. 2.2. The gradients are evaluated in every second cell along the horizontal and vertical center lines of cells. We see that the adjoint gradients are in the same order as the FD gradients, however, the deviation can be seen for many cells. Nonetheless, what is more important for DA is the sign of the gradients since the optimization step size is a hyper-parameter and heavily influences the optimization anyway. We will see that the accuracy here is

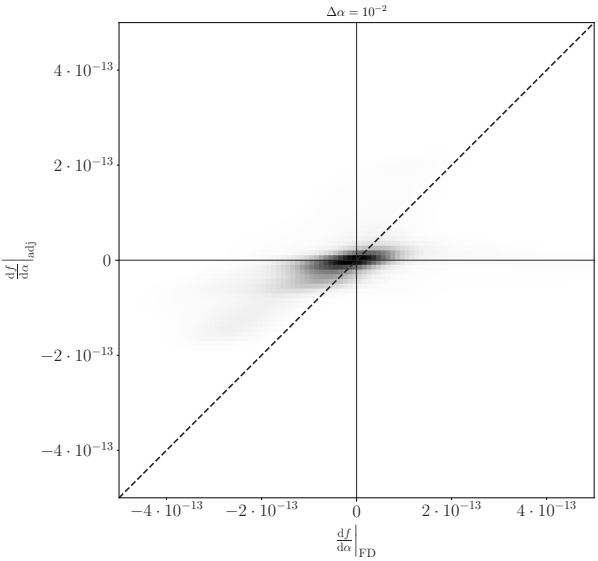


FIGURE 2.3: The gradients of all cells computed by FD compared with the gradients computed by the approximate adjoint method. Results are shown for the case of the periodic hills with full synthetic reference data for both velocity components with the initial value of unity for α and without any regularization [119].

enough for our DA process. To have a better view of the accuracy, the FD gradient with the optimal step size is performed for the whole field. The results are shown in figure 2.3. The points are compacted through a kernel to better visualize the plot using a Gaussian kernel from Scipy [119]. The same conclusions can be drawn. The order of magnitudes matches quite well. Quantitatively the adjoint gradients are generally underestimated. The sign of the gradients is mostly correct.

The next step is to verify that the optimization can recover the correct state variables. However, we consider that due to the nature of such an inverse problem multiple combinations of parameters can result in the same state variables. Therefore, we set up a case in which the α field that generates the reference velocity field is known. This way, the non-uniqueness issue is also

better shown. The algorithm that generates this case is shown in figure 2.4. First, we define an arbitrary α field as follows

$$\alpha^{ref}(x,y) = \begin{cases} 1 + 2\sin^2\left(\frac{\pi}{2.035}(y-1)\right)\sin^2\left(\frac{\pi}{9}x\right), & \text{for } 1 \leq y \\ 1, & \text{for } y < 1. \end{cases}$$

Then we run the forward solver with the given α field to get the solution. The obtained horizontal velocity field is assumed to be the reference for DA. DA is then performed to retrieve the reference horizontal velocity field and the optimal parameter field that is obtained is compared with the originally given one. The retrieved velocity profiles are shown in figure 2.6. It can be seen that the optimizer could perfectly match the horizontal velocity profiles. However, according to figure 2.5 the retrieved parameter field slightly deviates from the reference parameter field. This shows the degree of the issue of non-uniqueness, even when complete data is given. The deviation can also be due to inaccuracies in the adjoint gradients.

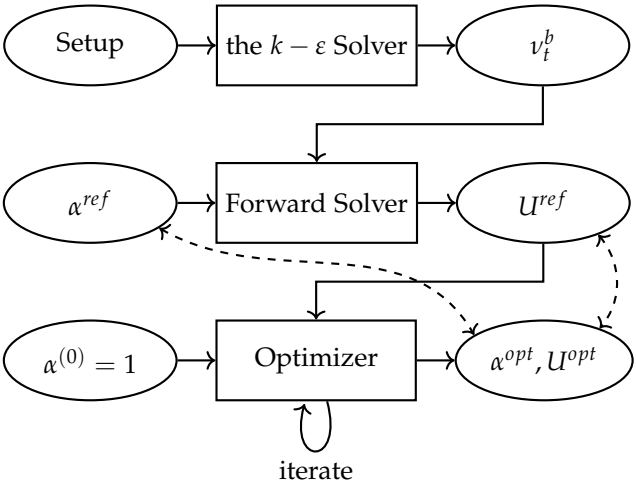


FIGURE 2.4: Process of using synthetic reference data to check the inverse solver performance [119].

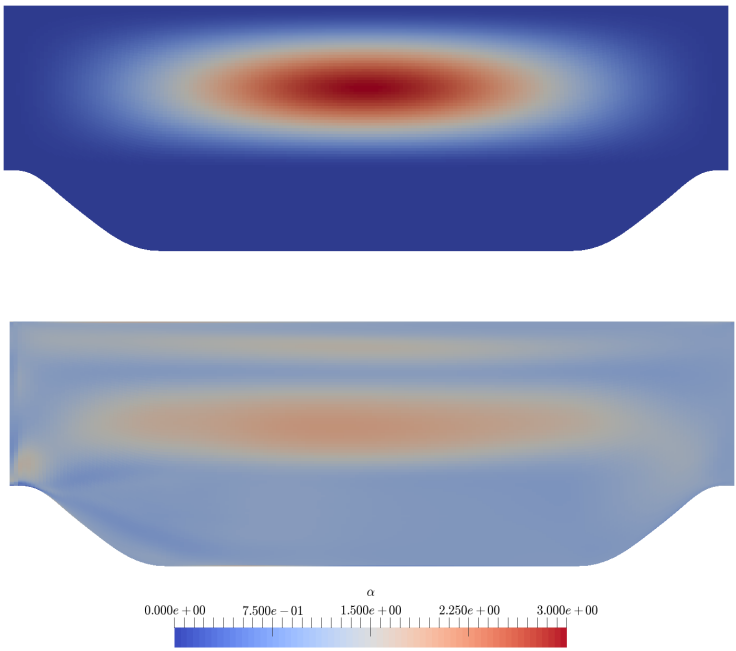


FIGURE 2.5: Comparison of the prescribed, synthetic parameter field α^{ref} (top) and the reconstructed field α obtained by the optimization process (bottom; data range [0.21, 1.95]). The full U_x reference field was used and no regularization was applied. Taken from reference [119].

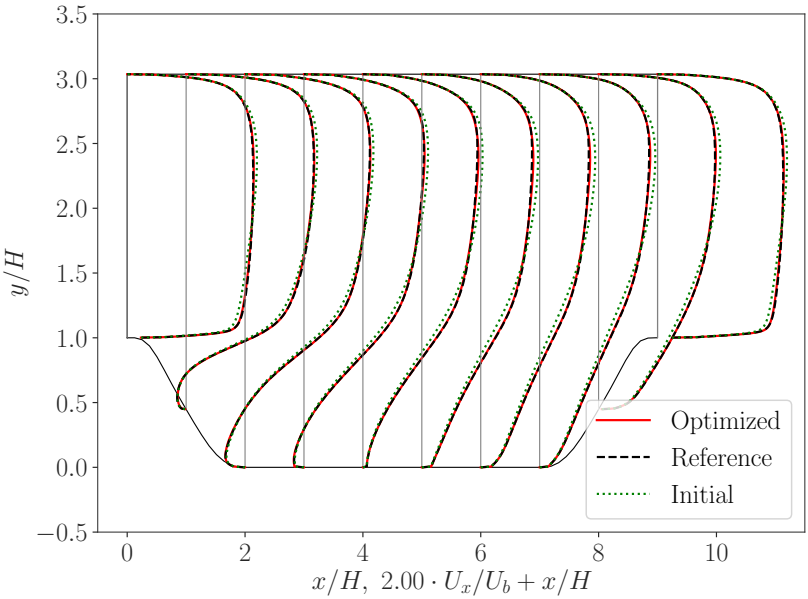


FIGURE 2.6: The synthetic reference U_x-profiles (black, dashed), initial (green, dotted), and optimized (red, solid) data. Length scales are normalized by the hillcrest height H and the velocity by the bulk velocity U_b above the hillcrest. Taken from reference [119].

2.8 ASSIMILATION OF SPARSE DATA: RESULTS WITHOUT REGULARIZATION

Now, we set up the main case. We follow a scenario where sparse data are available and a correct solution through DA is sought. The sparse data is actually a sampling of a time-averaged solution of a high-resolution LES simulation. We define two categories of measurement points: training and test data (unseen data). The training data is used in the DA process, while the test data is only used to test the performance of DA. Note that generally a smaller set of data is chosen for testing the generalization (e. g. 30%). The locations of these points are shown in figure 2.7.

Figure 2.8 shows the corresponding horizontal velocity profiles and the wall shear stresses at the bottom wall. Without regularization, the velocity profiles and wall shear stresses are unphysical and unrealistic. The solution is quite accurate at the measurement points (cf. Figure 2.7), but severely

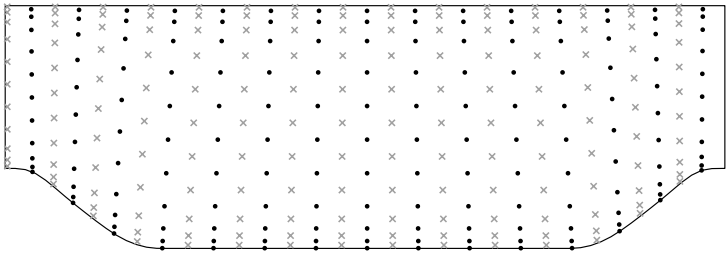


FIGURE 2.7: Distributions of measurement data (black dots) and testing data (gray crosses). Testing data are not used for data assimilation, but to estimate the generalization error.

deviates at other locations. This is also reflected in Table 3.2 (in the next chapter), where for the unregularized results the training error is very low, but the generalization error is high. We will shortly see that the jagged velocity profiles are due to a very noisy parameter field. We can conclude that without regularization the DA process will not result in realistic results. Therefore, the optimizer needs more information to arrive at a good solution, which can be done by introducing regularization schemes. The problem of over-fitting and low generalizibility can also be seen in the evolution of the cost function in Figure 2.9. The cost function at the reference points goes to a very low value after a couple of iterations but the cost function at test points is stuck to a rather large value.

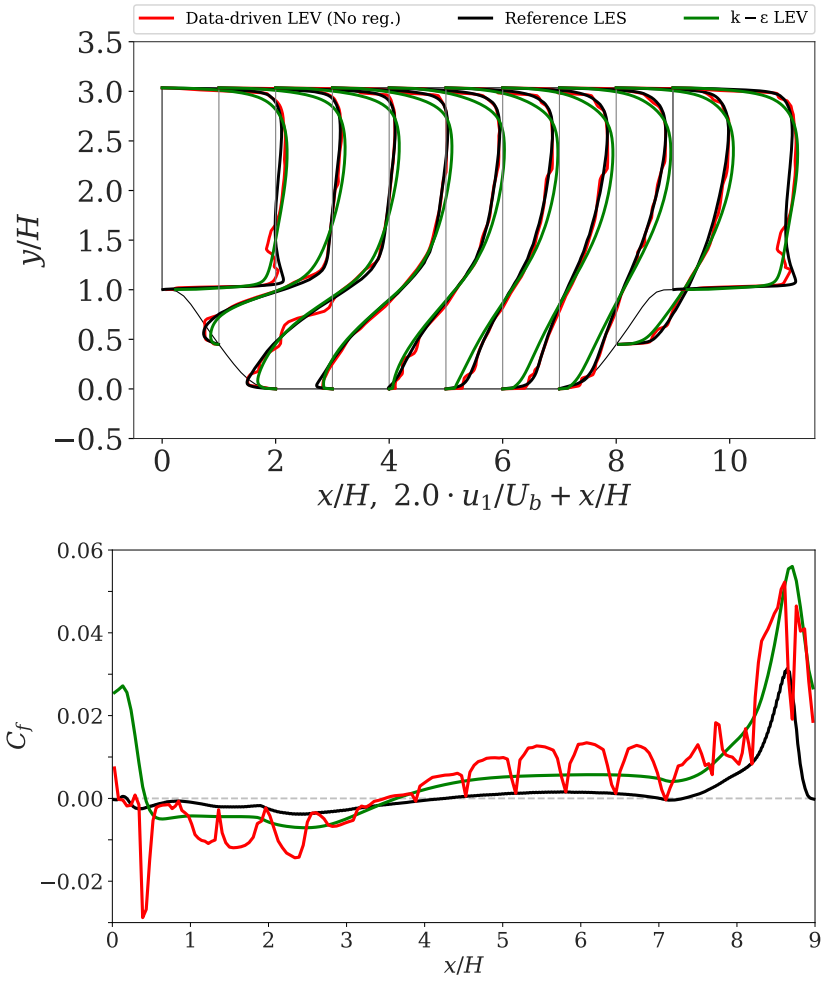


FIGURE 2.8: Horizontal velocity component u_1 (top) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot). Results for the baseline $k-\varepsilon$ model (green), for the unregularized data-driven method (red), and for the reference LES data (black) are shown.

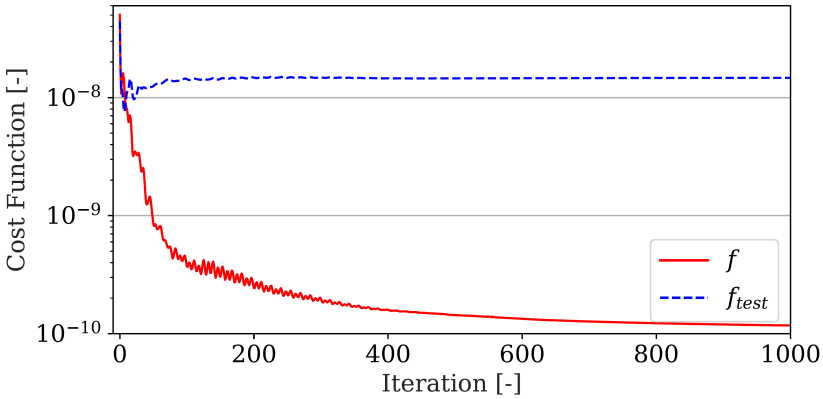


FIGURE 2.9: The evolution of the cost function during the optimization at the reference points (red curve) and the test points (blue curve)

3

REGULARIZATION

As discussed in the introduction, merely minimizing the distance function without any constraints cannot result in meaningful results. That was the reason for the choice of a linear eddy viscosity model as a constraint for the inverse problem. However, it can be shown that for the case where only sparse measurements are available, such a physical constraint still is not enough to avoid overfitting at the measurement points and irregular solutions. The process of adding more information or physical constraints to overcome the ill-conditioning and to avoid unphysical solutions can be called regularization. Here we first discuss three common regularization methods and then propose the PLDR approach. The L_2, TV, and Sobolev gradient regularization methods were developed in OpenFOAM in collaboration with Oliver Brenner. The post-processing scripts were developed by Oliver Brenner. The analysis of the results was done in collaboration with Oliver Brenner.

3.1 CONVENTIONAL REGULARIZATION METHODS

3.1.1 *L_2 regularization*

The first method is called L_2 regularization and is in our case defined as a term that penalizes the deviation of $\boldsymbol{\alpha}$ from its base values $\boldsymbol{\alpha}^b$, which is unity here. This automatically enforces some smoothness in the optimized parameter field since it limits the changes in $\boldsymbol{\alpha}$ and avoids irregular solutions. Alternatively, it can be interpreted as the level of trust put in the baseline eddy viscosity model. To implement the method the squared L_2 norm penalty term is added to the cost function in equation (2.6) as

$$f = \hat{\boldsymbol{V}} \cdot (\mathbf{H}\boldsymbol{u}_1 - \hat{\boldsymbol{u}}_1)^{\circ 2} + \omega_{L2} \left\| \boldsymbol{\alpha} - \boldsymbol{\alpha}^b \right\|^2 , \tag{3.1}$$

where ω_{L2} is a regularization hyper-parameter. The first term penalizes the misfit of the model's output with respect to the measurement values and the second term penalizes overfitting in order to reduce generalization errors. Mostly overfitting occurs in cells that are close to the measurement

points and the generalization errors tend to be larger in cells that are away from the measurement points.

In practice, the effect of the regularization contributes to the cost function gradient (cf. equation (2.12)) since

$$\underbrace{\frac{\partial f}{\partial \boldsymbol{\alpha}}}_{1\times M} = 2\,\omega_{L2}\,\underbrace{\left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{b}\right)}_{1\times M}\,. \tag{3.2}$$

3.1.2 *Total variation regularization*

The second method is called total variation (TV) regularization which is defined in our case as a sort of penalization of the local spatial gradient of $\boldsymbol{\alpha}$ [124]. This type of regularization aims to remove strong local variations or noise due to overfitting and irregular solutions; based on the intuition that the optimized eddy viscosity field should be smooth. Equation (2.6) is replaced by

$$f = \hat{\boldsymbol{V}} \cdot \left(\mathbf{H}\boldsymbol{u}_1 - \hat{\boldsymbol{u}}_1\right)^{\circ 2} + \omega_{TV} \sum_{m=1}^{M} \left[\frac{1}{|\mathcal{B}_m|} \sum_{o\in\mathcal{B}_m} \left(\alpha_m - \alpha_o\right)^2 \right], \tag{3.3}$$

where ω_{TV} is the regularization hyperparameter, m the index looping over all parameters, $\mathcal{B}_m$ is the set of adjacent parameters to the parameter with index m, o the index looping over the set $\mathcal{B}_m$, and $|\mathcal{B}_m|$ is the number of adjacent parameters. Similar to the L_2 method, the TV method effectively influences the optimization by correcting the sensitivities by

$$\frac{\partial f}{\partial \alpha_m} = 2\,\omega_{TV} \sum_{o\in\mathcal{B}_m} \left[\left(\frac{1}{|\mathcal{B}_m|} + \frac{1}{|\mathcal{B}_o|} \right) \left(\alpha_m - \alpha_o\right) \right]. \tag{3.4}$$

3.1.3 *Sobolev gradient regularization*

The third method is the Sobolev gradient regularization. The idea is to project the sensitivity field $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$ which is defined in L^2 space onto the more regular Sobolev space H^1 [71]. The projection is done by solving the Poisson equation

$$\underbrace{\left(\frac{\mathrm{d}f^{H^1}}{\mathrm{d}\boldsymbol{\alpha}}\right)^{T}}_{M\times 1} = \underbrace{\left(\frac{1}{1 + {l_{sob}}^2} \left(\mathbb{I} - l_{sob}^2 \nabla^2 \right) \right)^{-1}}_{M\times M} \underbrace{\left(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}\right)^{T}}_{M\times 1}, \tag{3.5}$$

where ∇^2 is the Laplace operator, l_{sob} a regularization parameter, and $\mathbb{I}$ the $M \times M$ identity matrix. This equation can be solved by the same finite volume method as used for the forward problem. The l_{sob} parameter can be considered as a low-pass filter width below which the fluctuations of the sensitivity field are smoothed.

While ω_{L2} and ω_{TV} are hyperparameters without physical meaning that must be adjusted by trial and error, the Sobolev gradient parameter l_{sob} has a physical meaning and can therefore be defined more easily [71].

3.2 PIECEWISE LINEAR DIMENSION REDUCTION

We propose the piecewise linear dimension reduction (PLDR) method based on the observation that the regularization methods discussed above, even with optimal hyperparameter values, fail to provide physical and smooth wall shear stresses. Noisy adjoint sensitivities and consequently noisy $\boldsymbol{\alpha}$ fields manifest themselves in unphysical and irregular wall shear stresses and velocities, which are both important quantities of interest. The data assimilation process should thus reproduce both quantities equally well.

The basic idea of the PLDR method is to reduce noise and high spatial variability in the parameter field by decreasing its degrees of freedom. More specifically, the $\boldsymbol{\alpha}$ field is represented as a piecewise linear function with linear Lagrange elements that are much coarser than the computational grid. For now, we define the elements to be triangles (for two-dimensional cases) whose vertices coincide with the measurement point locations, i. e., the $\boldsymbol{\alpha}$ field is represented as

$$\alpha_m = \phi_{mr}\, \beta_r \,, \tag{3.6}$$

with parameters $\boldsymbol{\beta}$ and the piecewise linear projection matrix $\boldsymbol{\Phi}$. Subscripts $m = 1, \ldots, M$ and $r = 1, \ldots, R$ are the indices of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ vectors, respectively. For now, we assume that R is equal to the number of measurement points. The components of each row of $\boldsymbol{\Phi}$ are all zero, except for three components corresponding to the three vertices of the associated triangular element. The sum of each row of $\boldsymbol{\Phi}$ is unity and every α_m is a linear combination of three corresponding nonzero nodal values β_r.

It should be noted that the triangular elements may not cover the whole domain, e. g. a few cells close to the convex parts of the boundaries of the periodic hills may reside outside of any triangular element. To solve this issue we add dummy nodes far away from the boundary outside of the domain. This acts as a Neumann boundary condition for the reconstructed

$\boldsymbol{\alpha}$ field. An example of piecewise linear basis functions is depicted in figure 3.1.

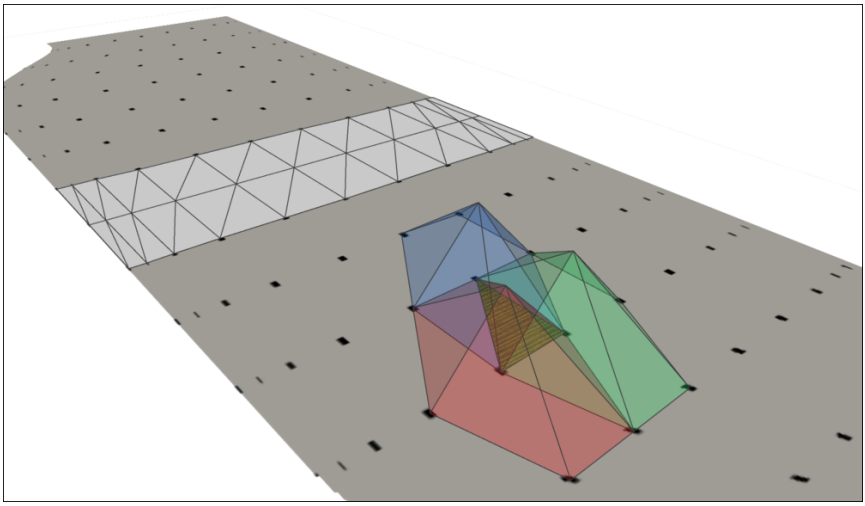


FIGURE 3.1: The basis functions (blue, green, and red polyhedrons) span the solution inside the dashed yellow triangle. The locations of measurement points are depicted by black dots, and a few elements are shown by white triangles.

In practice, PLDR regularizes the inverse problem through the reduction of the number of dimensions of the sensitivities from $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$ with size M to $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}$ with size R, where $R \ll M$. The chain rule yields

$$\underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}}_{1\times R} = \underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}}_{1\times M}\underbrace{\frac{\mathrm{d}\boldsymbol{\alpha}}{\mathrm{d}\boldsymbol{\beta}}}_{M\times R} = \underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}}_{1\times M}\underbrace{\boldsymbol{\Phi}}_{M\times R} . \tag{3.7}$$

The values of $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}$ are normalized by the sum over each corresponding column of $\boldsymbol{\Phi}$ to be consistent with $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$. Based on $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}$, the parameter vector $\boldsymbol{\beta}$ is updated as

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \Delta \left(\frac{\mathrm{d}f^{(t)}}{\mathrm{d}\boldsymbol{\beta}} \right)^{T} , \tag{3.8}$$

and equation (3.6) subsequently is used to reconstruct the physical parameter vector $\boldsymbol{\alpha}$. The optimization process continues by evaluating the forward

and adjoint problems that yield the sensitivities, which, in turn, are used to update $\boldsymbol{\beta}$ again. The flow chart of the data assimilation procedure is presented in figure 3.2.

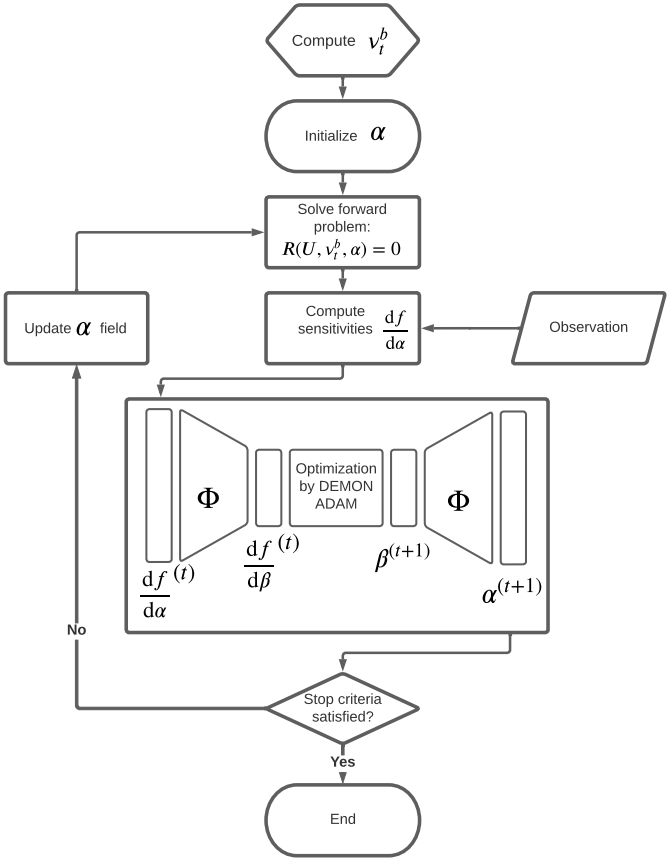
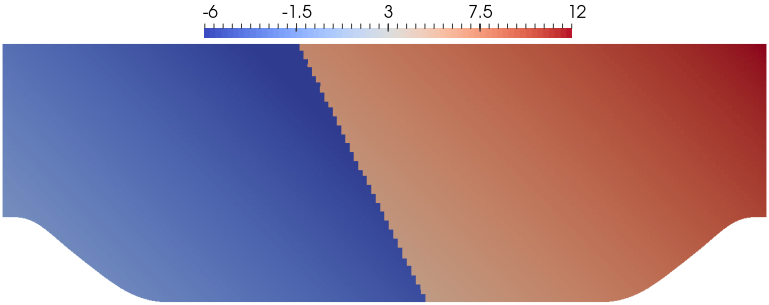


FIGURE 3.2: Flow chart of the data assimilation procedure using the PLDR method.

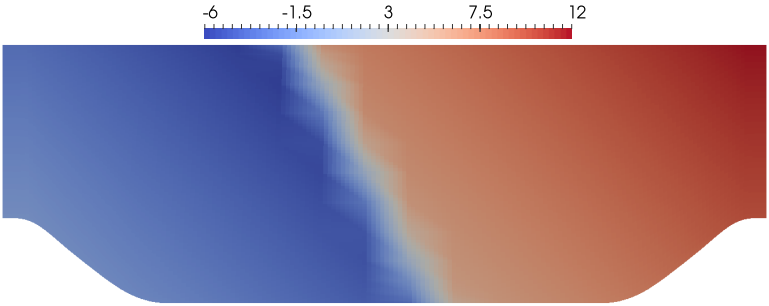
3.2.1 *Verification of contraction and expansion*

To make sure that the process of mapping $\frac{df}{d\alpha}$ to $\frac{df}{d\beta}$ (contraction or restriction) and then β to α (expansion or prolongation) is properly implemented, we test the code for a synthetic $\frac{df}{d\alpha}$. If we assume $\Delta = 1$ and $\beta = 0$ multiply-

ing the synthetic $\frac{df}{d\alpha}$ with the projection matrix results in $\frac{df}{d\beta}$. Furthermore, based on equation (3.8) the resulting α should be very similar to $\frac{df}{d\alpha}$, except the sharp changes must be smoothed out because of the piecewise linear projection. The synthetic $\frac{df}{d\alpha}$ used here is a linearly varying field in both x and y direction with a sharp change in the middle as depicted in figure 3.3a. The resulting α field is shown in figure 3.3b. As we see, PLDR preserves linearity, and values of $\frac{df}{d\alpha}$ are the same everywhere, except near the abrupt change. This is a desirable feature of PLDR as the goal is to avoid sharp changes in the sensitivities while preserving the local linearity of the field.



(a) The synthetic $\frac{df}{d\alpha}$ field

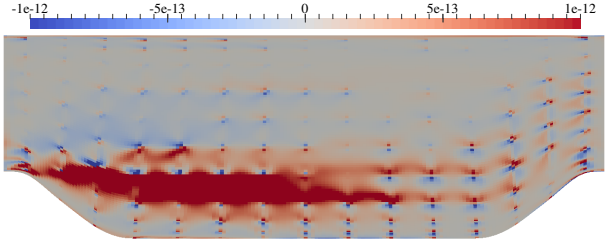


(b) The resulting α field from the synthetic $\frac{df}{d\alpha}$

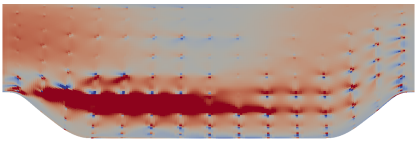
FIGURE 3.3: Verification of the dimensionality reduction process.

3.3 REGULARIZATION EFFECT ON A SINGLE OPTIMIZATION STEP

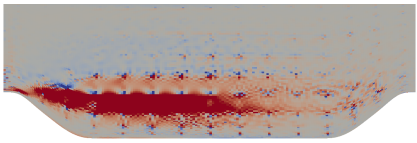
Before we proceed with presenting the results of PLDR and the other methods, we discuss the effect of each regularization method on the sensitivities $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ for a single initial optimization step. The L_2, TV, and Sobolev gradient methods directly act on $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$, while PLDR works indirectly. To compare the PLDR sensitivity $\frac{\mathrm{d}f}{\mathrm{d}\beta}$ to the other methods, it is projected to $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ by multiplying it with the linear projection matrix. Figure 3.4 shows the corresponding results of each method. It can be seen that PLDR is the most successful method in removing noise and the peaks at the measurement points (cf. figure 2.7), especially near the walls.



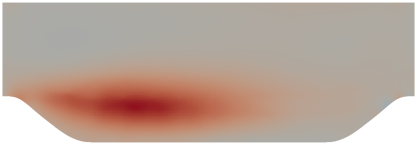
(a) No regularization.



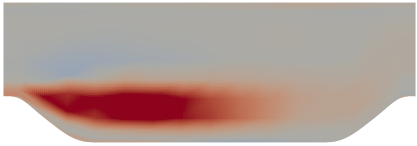
(b) L_2 regularization.



(c) TV regularization.



(d) Sobolev gradient regularization.



(e) PLDR regularization.

FIGURE 3.4: Sensitivity $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ contours after an intermediate optimization step without (a) and with (b-e) regularization. For PLDR, the sensitivities are obtained by projecting $\frac{\mathrm{d}f}{\mathrm{d}\beta}$ using the projection matrix.

Hyperparameter values			
Regularization		Optimization	
ω_{L2}	1e-12	κ_0	0.900
ω_{TV}	5e-12	κ'	0.999
l_{sob}	5e-1	η	0.010
$\omega_{TV,\beta}$	1e-13	T	1000

TABLE 3.1: Hyperparameter values for regularization and optimization methods.

3.4 RESULTS AND DISCUSSION

The optimal hyperparameter values used in this case study are summarized in Table 3.1.

3.4.1 *PLDR*

we saw in the previous chapter in figure 2.8 that a data-driven model without regularization results in severe irregular and unphysical solutions. The solution is quite accurate at the measurement points (cf. figure 2.7) but severely deviates at other locations which means the training error is very low, but the generalization error is high.

When PLDR is used, however, the results are much smoother and closer to the reference profiles. This is more pronounced for the wall shear stresses (cf. figure 3.5, bottom). For example, the peak at the second hill is now much lower and closer to the reference value. The reattachment length is closer to the true value, and at the inflow, the wall shear stress is quite accurate. However, there is an oscillation near the first hill that does not match the reference profile. With regard to the velocity profiles, a substantial improvement is observed in all regions. Over predictions are observed close to the upper wall and slight deviations are seen close to the hillcrest. The oscillations in the wall shear stress and the deviations in the velocity profiles motivate further improvements to the PLDR method, which are discussed below.

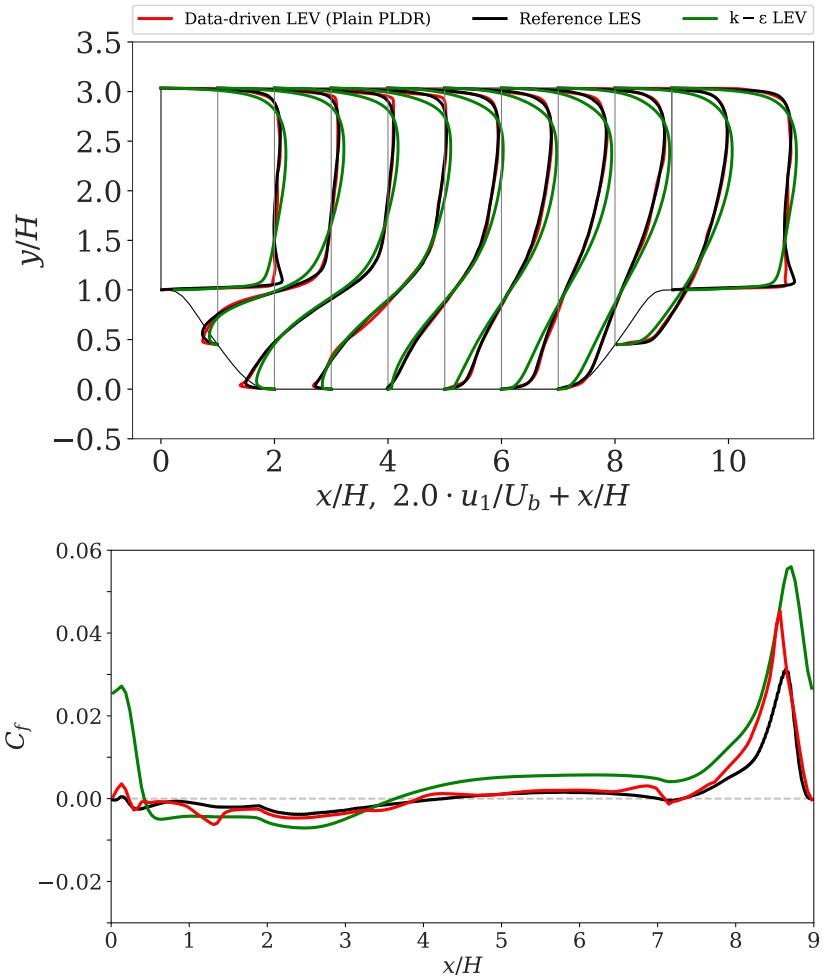


FIGURE 3.5: Velocity profiles (top plot) and wall shear stress (friction coefficient) at the lower wall (bottom plot) using the PLDR method.

3.4.2 *Improved PLDR*

Looking at the optimal α field obtained with the PLDR method (figure 3.5), we can see deviations near the top wall which likely result from the high gradients of α in those regions. One idea to reduce these discrepancies is

to apply a concept similar to TV regularization to the β field. This leads to improved predictions of the quantities of interest, as we will see below. The adapted cost function reads

$$f = \hat{\boldsymbol{V}} \cdot (\mathbf{H}\boldsymbol{u}_1 - \hat{\boldsymbol{u}}_1)^{\circ 2} + \omega_{TV,\beta} \sum_{r=1}^{R} \left[\frac{1}{|\mathcal{B}_r|} \sum_{l \in \mathcal{B}_r} (\beta_r - \beta_l)^2 \right] , \tag{3.9}$$

where $\omega_{TV,\beta}$ is the weight factor of the β-TV regularization, $\mathcal{B}_r$ the set of neighborhood nodes, and l denotes the index of the adjacent nodes. The sensitivity formulation ($\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}$) will change accordingly.

Moreover, the wall shear stress estimation can be improved by adding extra weight $\boldsymbol{c_w}$ to the discrepancy function near the wall, i. e.,

$$f = \left(\boldsymbol{c_w} \circ \hat{\boldsymbol{V}}\right) \cdot (\mathbf{H}\boldsymbol{u}_1 - \hat{\boldsymbol{u}}_1)^{\circ 2} + \omega_{TV,\beta} \sum_{r=1}^{R} \left[\frac{1}{|\mathcal{B}_r|} \sum_{l \in \mathcal{B}_r} (\beta_r - \beta_l)^2 \right] . \tag{3.10}$$

The weight factor vector $\boldsymbol{c_w}$, in our case, is set to 10 at wall adjacent reference locations and 1 otherwise.

Figure 3.6 shows that the above improvements reduce the deviations of both the optimized velocity profiles and the wall shear stresses. Tiny undershoots near the inlet remain in both quantities of interest. These might be explained by the lack of reference points in that region, sub-optimally placed sensors, or inaccuracies in the computation of the sensitivities. Another explanation could be that the triangular elements cannot properly represent the curved shape of the hill, which affects the optimized parameter field and consequently the quantities of interest.

3.4.3 *Comparison with other regularization methods*

The significance of the PLDR method in producing physically sound and smooth profiles is highlighted by comparison with conventional regularization methods. Here, we compare PLDR with L_2, Total Variation, and Sobolev gradient regularization. Figure 3.8 shows the horizontal velocity profiles and wall shear stresses resulting from applying different regularization approaches.

Regarding the horizontal velocity profiles, qualitatively all regularization methods were able to prevent overfitted and irregular profiles. The L_2 method, however, shows more inaccuracies in the first hill regions while the TV method is the most accurate almost everywhere. To further examine the methods quantitatively, we compared their training (the error at the

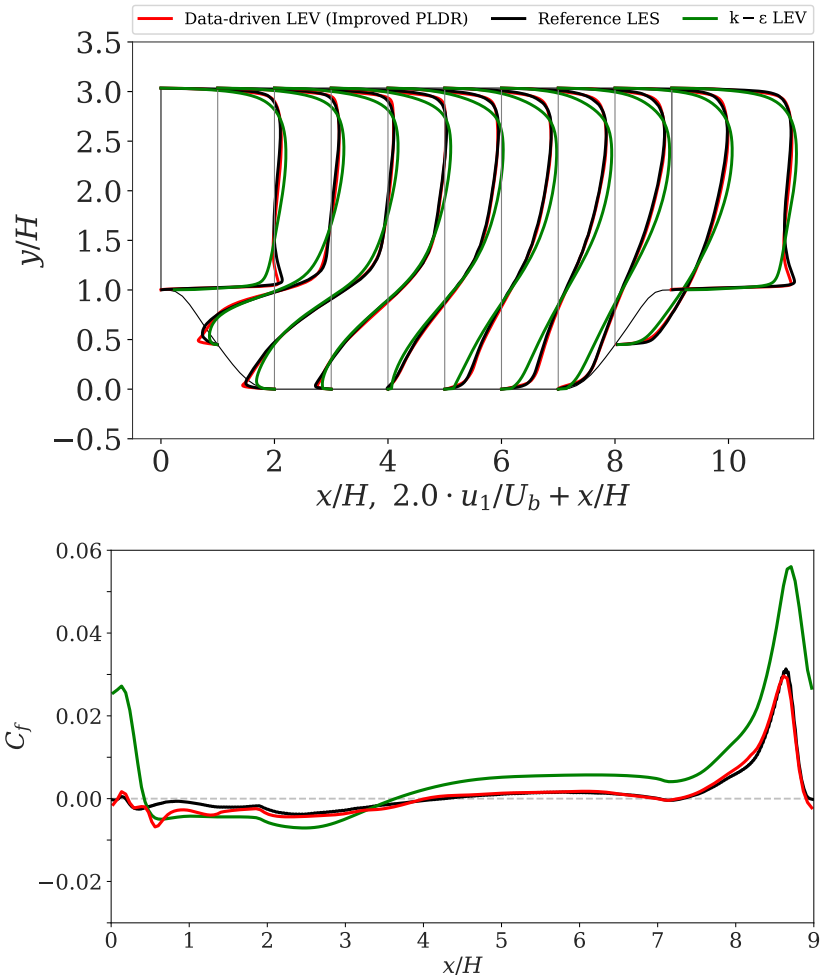


FIGURE 3.6: Velocity profiles (top) and wall shear stress (friction coefficient C_f) at the lower wall (bottom) obtained with the improved PLDR method.

measurement points) and generalization errors (the error at testing points) in Table 3.2. The training error is one to two orders of magnitude smaller when data assimilation is applied. The lowest training error is observed for data-driven LEV without regularization. This comes at the cost of a very large generalization error. The lowest generalization error is observed

for the TV method and the largest for DA without regularization. The PLDR method works quite well in both aspects (cf. figure 3.6). It should be noted that the optimal values for the regularization hyperparameters are determined on a trial-and-error basis. While there is an optimal value for the TV method, where the cost function and generalization errors are both low, we could not find such an optimal value for the L_2 and Sobolev gradient methods. Small parameter values resulted in noise and overfitting, while large values lead to overly smooth results. The excellent performance of the TV method is reflected in the cost function evolution at reference points and test points, as shown in Figure 3.7. This is especially clear when it is compared to the cost function evolution of the DA without regularization (Figure 2.9), highlighting the importance of regularization.

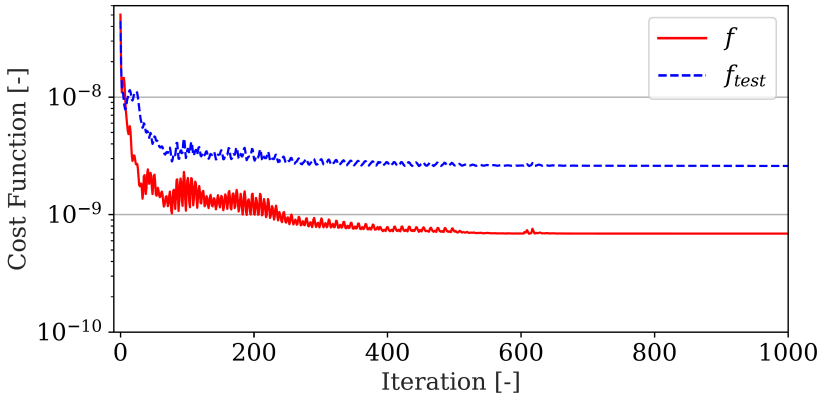


FIGURE 3.7: The cost function evolution of DA with TV regularization during the optimization iterations. The red curve corresponds to the cost function at reference points and the blue curve is for the test points.

With regards to the wall shear stresses, it is seen that except for the PLDR method all methods fail to prevent overfitted and irregular solutions. Referring to Table 3.2, the advantage of PLDR is pronounced in the case of reconstructing wall shear stress and computation of the skin friction drag. The C_f training error is the lowest for PLDR and TV while the generalization error is highest for no regularization and the L_2 method. The TV method, which performs best for the internal field, is poor in the skin friction drag estimation with around 200% error. The Sobolev gradient method only has an 8.74% error which is a great advantage mostly due to the correct estimation of the peak at the second hill. The PLDR method nonetheless introduces only a 1.38% error in the skin friction drag

computation, which indeed is very accurate considering the limitations of the linear eddy viscosity model. The advantage is more pronounced when we consider that the Sobolev gradient method resulted in a bumpy unphysical profile while PLDR leads to a physically reasonable smooth profile.

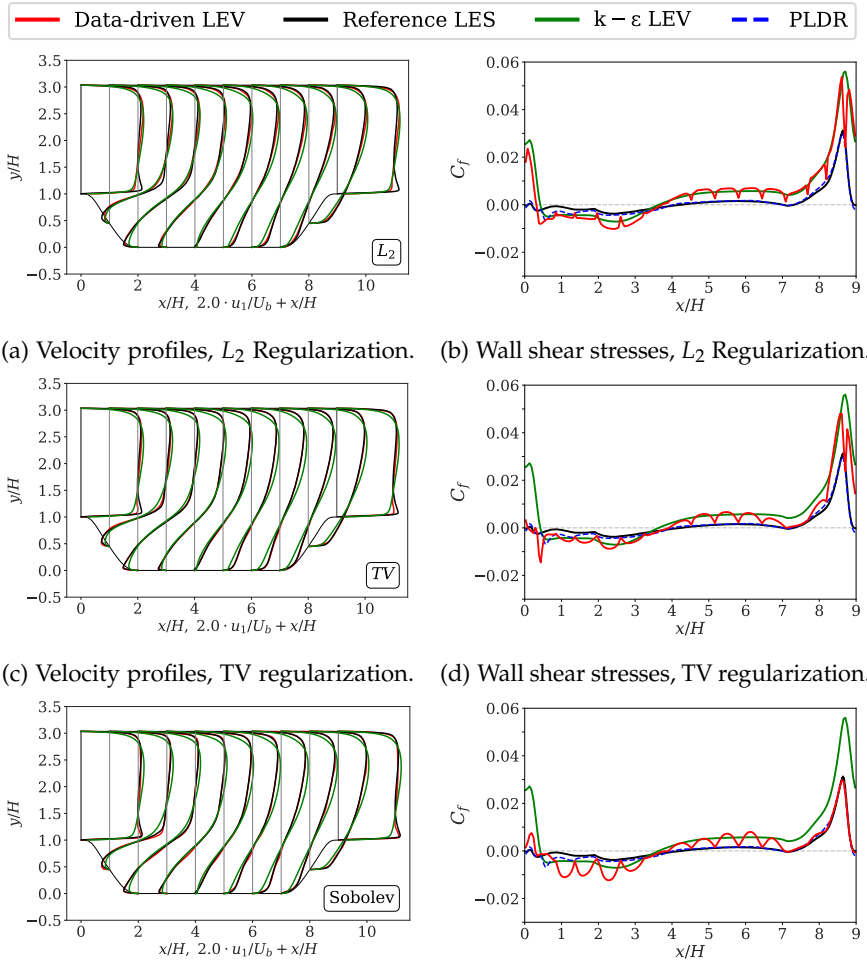
Figure 3.9 shows the reconstructed optimized parameter fields α for the different regularization methods and without regularization. It can be seen that the mixing effect of the Sobolev gradient method is not able to diffuse the α variations at the wall adjacent measurement points. The same is true for the L_2 method. The TV method leads to smooth α fields with a small range of values, but the issues with boundary values persist. Finally, the PLDR method succeeds in removing noise in the near-wall region simply by constraining the variations to be piecewise linear.

It seems that the linearity of α for elements based on the measurement points is sufficient to represent a physical eddy viscosity and finally the velocities. This works specifically well for the problem of correcting the eddy viscosity based on sparsely distributed reference data in the presented case. However, it shows the general regularization effect of dimensionality reduction to avoid overfitting and irregular solutions.

It should be noted that different scenarios with varying distributions and numbers of measurement points were considered. For the PLDR method, we examined whether to locate the nodes at positions other than the measurement points (staggered) or to add more nodes than the measurement points. In summary, the best results were obtained when the nodes of the linear elements coincided with the reference data locations.

3.5 CONCLUSIONS

A data-driven RANS model was developed to obtain solutions that are consistent with sparsely distributed measurements within the simulation domain. It was argued that for data assimilation the sparsity of the measurement data forces the modeler to voluntarily constrain the inverse problem to avoid unphysical solutions due to the ill-posed nature. The RANS equations are closed by the linear eddy viscosity model and the eddy viscosity was chosen as a parameter to be tuned in the data assimilation process. In particular, a baseline value for the eddy viscosity is computed by a physics-based model like the $k-\varepsilon$ model and then multiplied with the parameter field that is tuned. We further emphasized the need for regularization to avoid overfitting and irregular solutions and introduced a new regularization



(a) Velocity profiles, L_2 Regularization.

(b) Wall shear stresses, L_2 Regularization.

(c) Velocity profiles, TV regularization.

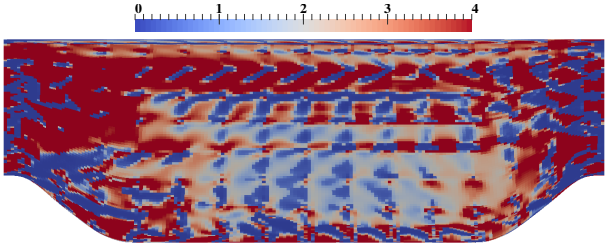
(d) Wall shear stresses, TV regularization.

(e) Velocity profiles, Sobolev gradient regularization.

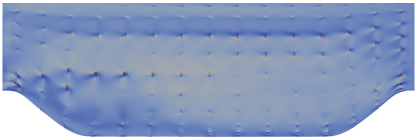
(f) Wall shear stresses, Sobolev gradient regularization.

FIGURE 3.8: The comparison of stream-wise velocity profiles u_1 (left) and the friction coefficients C_f (right) for different regularization methods.

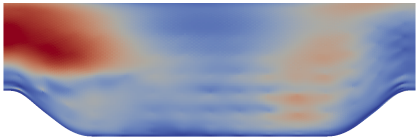
method called piecewise linear dimensional reduction (PLDR). The goal of PLDR was to further constrain the correction field to vary piecewise linearly so that overfitting and unphysical results are avoided. This type of regularization is suited particularly for reconstructing wall shear stresses



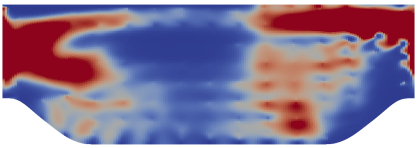
(a) No regularization; data range [0.0, 85.6].



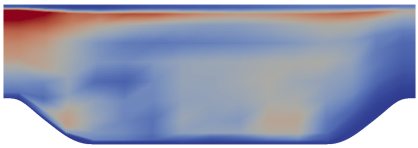
(b) L_2 regularization; data range [0.003, 8.94].



(c) TV regularization; data range [0.001, 4.2].



(d) Sobolev gradient regularization; data range [0.0, 11.2].



(e) PLDR regularization; data range [0.0, 5.1].

FIGURE 3.9: Optimal α fields without (a) and with (b-e) regularization. The same color bar scaling is used for all plots.

where conventional regularization schemes like the TV or Sobolev gradient methods fail. We tested the new method for the case of flow over periodic hills and compared it to other regularization methods. The results show that the PLDR method is able to reconstruct an accurate internal velocity field. More importantly, PLDR is the only regularization method among the other candidates that could provide smooth and accurate wall shear stress profiles with only a 1.38% error in the estimation of the skin friction drag force on the bottom wall. There are, however, limitations in employing the PLDR method for general purposes that remain to be investigated.

Here, the PLDR method was only used for a sparse data-driven RANS model for two-dimensional flow over periodic hills. The sparsity of the measurement points is important in well representing the optimal corrective field. It cannot be expected in general that the piecewise bilinear approach

Method	Errors			
	Training	Generalization	C_f training	Drag [%]
$k-\varepsilon$ model	5.04e-08	4.45e-08	5.12e-06	454.87
Data-driven, no reg.	1.17e-10	1.47e-08	1.60e-07	404.65
Data-driven, L_2	3.75e-09	8.12e-09	1.99e-06	356.87
Data-driven, TV	6.89e-10	2.60e-09	1.38e-06	200.06
Data-driven, Sobolev	3.68e-10	8.01e-09	1.98e-06	8.74
Data-driven, PLDR	3.06e-09	6.26e-09	1.39e-06	1.38

TABLE 3.2: Summary of the errors for different regularization methods. Training error is the final cost function value f normalized by the number of measurement points. Generalization error is the final cost function value at testing points (f_{test}) normalized by the number of testing points. The C_f training error is the mean absolute error of C_f at wall-adjacent measurement points for the bottom wall. Drag error (Skin friction drag error) is the relative error with respect to the skin friction drag of LES data for the bottom wall.

suffices to predict the optimal field reasonably well. For such cases, it could be helpful to add some *dummy* points to increase the degrees of freedom of the corrective field. It would also be beneficial to use quadratic or even higher-order elements. It can be difficult to impose boundary conditions with coarse triangular elements for complex geometries since the elements must not overlap with the boundaries.

Another future work could extend and test the method for 3D cases, where tetrahedral elements would be used instead of triangular ones. In addition, it should be noted that while the piecewise linear method works well for the eddy viscosity approach, it may not work for other data-driven models where, e. g., Reynolds stresses are tuned directly. Nonetheless, it can be argued that the piecewise linear dimension reduction, or any other dimension reduction method, can be seen as an attractive tool to improve ill-posed data assimilation of sparse measurements and to avoid overfitted and irregular solutions.

4

SPARSE WALL DATA-DRIVEN HYBRID LES/RANS SIMULATION

4.1 SUMMARY

We investigate a scenario where only sparse wall shear stress measurements are available and accurate wall shear stress and velocity profiles are sought. The work was motivated by the observation that by solely assimilating near-wall measurements using a discrete adjoint-based approach while leading to accurate wall shear stresses, a RANS model generally does not produce accurate velocity fields in the free shear flow regions. Therefore, we devise a method based on additional reference data away from walls produced by a relatively cheap hybrid LES/RANS model and integrate them into the assimilation process. To achieve this, we modified the dual-mesh hybrid LES/RANS framework proposed by Xiao and Jenny [14] such that the mean LES fields only get loosely coupled with corresponding steady RANS solutions. The framework was developed in OpenFOAM and tested for flow over a periodic hill with $Re = 10595$. Results show that the framework outperforms the conventional dual-mesh hybrid LES/RANS and standalone sparse wall-data assimilated RANS models in terms of accuracy and computational cost. The development of the discrete adjoint code, the optimizer, and analyzing the results were done in collaboration with Oliver Brenner. The post-processing scripts were developed by Oliver Brenner.

4.2 INTRODUCTION

Accurate simulation of high Reynolds number turbulent flows is a computationally demanding task since a wide range of eddy sizes has to be resolved. This is one of the reasons why Direct Numerical Simulation (DNS) is prohibitively expensive for most practical applications.

Large Eddy Simulation (LES) is computationally more efficient since only the larger eddies are resolved, while the effect of the small ones is taken into account via a statistical model. However, for turbulent wall-bounded flows, the required grid resolution has to be fine enough to capture the

inner layer, which is only 10-20% of the boundary layer but inhabits very small energetic eddies. Therefore, the computational cost of such an LES roughly scales with $Re^{1.8}$ [14], and this gets too high for very large Reynolds numbers

Three alternatives to LES would be to use Reynolds-averaged Navier-Stokes models (RANS), to model the wall stress in LES (wall-modeled LES), or to somehow combine LES with RANS (coupled or hybrid LES/RANS). RANS models, often based on the Boussinesq eddy viscosity assumption, are computationally cheap and numerically robust, but their accuracy is limited for complex flows. Here we propose to combine experimental data with a hybrid LES/RANS method, but first, we briefly review data-driven RANS approaches.

4.2.1 *Data-driven RANS simulation*

There are two approaches to improving linear eddy viscosity (LEV) RANS models. The first one is a *physics-driven* approach [125], in which either the eddy viscosity is computed by solving more accurate transport equations [29], or to relax the eddy viscosity assumption resulting in non-linear eddy viscosity models [126], or to directly model the Reynolds stress components using algebraic or differential equations [127, 128]. It has been observed that more complex models tend to be less general and robust and they require tuning of more parameters

In *data-driven* approaches, on the other hand, the RANS models are directly corrected based on available experimental data (typically from sparse measurements). Depending on the availability of data, one needs to add more physical constraints to the problem to avoid ill-conditioning and non-uniqueness inherent in inverse problems [77]. In fact, even if one has full DNS data and uses it for assimilation, retrieving correct Reynolds stress tensors is a difficult task [61]. In the case of sparse data, which increases the ill-conditioning and non-uniqueness, including a model describing the Reynolds stresses is a necessity. For this reason, having sparse measurement data, Brenner et al. [119] kept the Boussinesq assumption, and instead of Reynolds stresses, they sought optimal eddy viscosity fields that lead to an agreement with the experimental data. Such an approach is successful if the sparse data is regularly distributed in the whole domain. Lack of data in one part of the domain can easily lead to unphysical profiles in the rest, which is the case when only sparse wall shear stress measurements are available.

To cope with such a lack of reference data, we devised a new method that employs reference data in the free shear flow regions produced by a computationally affordable high-fidelity model i. e. a loosely coupled dual-mesh LES/RANS method. The generation of such data can be seen as a form of regularization of data assimilation, which is defined as introducing additional information to solve an ill-posed problem, to prevent overfitting and irregular solutions. The rest of this section reviews a background on wall-modeled LES and hybrid LES/RANS approaches.

4.2.2 *Wall-modeled LES and hybrid LES/RANS models*

To avoid the necessity of a dense and isotropic grid near walls, one can use wall-modeled LES (WMLES) in which an algebraic relation or an ODE is solved separately to provide wall shear stress estimations [129]. The wall stress model gets information from the LES data in the logarithmic layer and provides boundary conditions for LES at the wall surface.

Some algebraic wall models are based on a simple logarithmic law of the wall, while some analytical wall models also account for non-equilibrium effects such as pressure gradient, convection, etc. [43–45, 130]. More advanced wall models solve RANS-like ODE in the wall-normal direction, where the wall-parallel velocity is obtained from LES, while a simplified momentum equation is solved to provide wall shear stress estimations [46].

One also can solve 3D RANS PDFs on another grid that overlaps with the LES grid in the inner-layer region (20% of the boundary layer thickness) and exchange information with LES via the wall stress and the boundary condition [131]. In another approach, instead of providing only the wall shear stress, a forcing term can be added to nudge the LES solution to the RANS solution [132]. Such a model may not be categorized strictly as WMLES. From a computational point of view, WMLES can relax the grid resolution requirements in the wall-normal direction but in the wall-parallel direction, the usual LES grid requirement is needed [133].

In some hybrid LES/RANS methods, the computational domain is decomposed into two subdomains, where RANS is solved only in the near wall region and LES in the remaining domain, and information is transferred in both ways across the interface. Such hybrid models are called segregated or zonal models [48].

Perhaps the most popular hybrid LES/RANS model is detached eddy simulation (DES) [49], which can be categorized as a unified hybrid model. In unified hybrid models, there is only one mesh and one solver, which

switches or blends between LES and RANS in each grid cell (either predefined or dynamic based on the local flow features [133]). Taking advantage of the fact that the structure of unsteady RANS and LES equations is the same, the main task of a hybrid model is to compute the Reynolds stresses (or the eddy viscosity, if both models are based on the Boussinesq assumption) by blending or switching the two. In such models, the wall-normal direction should be resolved at least up to the buffer layer (for RANS to resolve the inner layer), but in the wall-parallel direction the grid only resolves the geometry and the mean flow (in contrast to WMLES) [133].

The choice between these methods is a trade-off between accuracy, computational cost, and ease of implementation and use. The main drawback of WMLES is the strong dependence of accuracy on the wall model, which not always is accurate for complex flows and geometries. If the wall model solves ODEs or PDEs, the implementation is challenging, which is also the case for segregated models. The unified hybrid models suffer from inherent inconsistencies between the filtered nature of LES and the ensemble or time-averaged nature of the RANS equations [14]. They are also highly dependent on the grid quality [134]. Grid ambiguity can result in erroneous switching from LES to RANS near the wall, which reduces the modeled turbulent stresses where LES has not created enough fluctuations. This phenomenon is called modeled stress depletion (MSD), which is responsible for different issues, such as grid-induced early separation, log-layer mismatch in the gray zone, spurious buffer layer, artificial super-streaks, etc. [48, 135–138]. Additional treatment, therefore, is needed to avoid MSD, e. g by adding additional forcing near the interface to sustain the velocity fluctuations, which may result in more inconsistencies [139, 140].

To avoid the above problems, Xiao and Jenny proposed an alternative approach called dual-mesh dual-solver hybrid models [14, 141, 142]. In this method, two different grids are generated for the entire domain. The RANS grid resolves the near-wall region but is rather coarse in the free shear flow region. The LES grid is coarse and under-resolved, but rather isotropic everywhere without high resolution close to the wall. The two sets of equations are solved separately but simultaneously. Through nudging, the exponentially weighted time-averaged LES solution is relaxed towards the RANS velocity field in the near wall region and the RANS solution is relaxed towards the exponentially weighted time-averaged Reynolds averaged LES velocity in the free shear flow region. This ensures consistency between Reynolds averaged LES and RANS quantities and avoids problems such as

MSD and additional forcing. It also is independent of the choice of RANS closure models, in contrast to models such as DES.

This approach has been shown to be successful in various cases [143, 144], but two solvers have to run simultaneously and exchange information every couple of steps. The latter was especially difficult and time-consuming when two different codes were used. In addition, the URANS time step has to be reduced to be consistent with the LES simulation time step. Here we simplify the procedure by solving steady RANS instead of URANS [145], which results in a sequential, loose coupling of the two solvers, that is much easier to implement. For example, due to the lower exchange frequency, it is efficient enough to simply transfer the necessary information via files. In addition, the RANS solver converges much faster than URANS using under-relaxation techniques.

The drawback of both the tightly (previous) and loosely (proposed) hybrid methods is the inability to predict accurate wall shear stresses if the LES grid is not fine enough [141]. Such an issue can be solved with data assimilation of sparse wall stress data, which is the second component of the proposed framework.

The rest of the chapter is structured as follows. First, in section 4.3 the original version of the dual-mesh hybrid model is explained followed by the description of the new loosely coupled method. Then, the data assimilation procedure is described. In section 4.4, results of the new hybrid method are presented and it is explained how it can be combined with data assimilation of sparse wall shear stress measurements. Finally, in section 4.5, conclusions are provided.

4.3 METHODS

4.3.1 *Tightly coupled dual-mesh hybrid LES/RANS*

The initial version of the dual-mesh hybrid LES/RANS method developed by Xiao and Jenny [14] is explained below. In this method, two sets of governing equations are solved on different meshes. For the LES solver, the

filtered momentum and pressure equations of an incompressible flow with constant density read

$$\frac{\partial \overline{u_i}}{\partial t} + \frac{\partial (\overline{u_i u_j})}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial^2 \overline{u_i}}{\partial x_j x_j} - \frac{\partial \tau_{ij}^{sgs}}{\partial x_j} + {Q_i}^L \tag{4.1}$$

$$\text{and} \quad \frac{\partial^2 \overline{p}}{\partial x_i \partial x_i} = -\frac{\partial^2}{\partial x_i \partial x_j}(\overline{u_i u_j} + \tau_{ij}^{sgs}) + \frac{\partial {Q_i}^L}{\partial x_i}, \tag{4.2}$$

where x_i, t, and ν are spatial coordinates, time, and kinematic viscosity, respectively. $\overline{u_i}$, $\overline{p}$, and τ_{ij}^{sgs} are filtered velocity, filtered kinematic pressure, and residual stresses, respectively. ${Q_i}^L$ is the drift force applied to the filtered equations, which ensures the statistical consistency between LES and RANS. The definition of this term is presented later; see equation (4.6).

Consequently, the Reynolds-averaged momentum and pressure equations are defined as

$$\frac{\partial \langle u_i \rangle}{\partial t} + \frac{\partial (\langle u_i \rangle \langle u_j \rangle)}{\partial x_j} = -\frac{\partial \langle p \rangle}{\partial x_i} + \nu \frac{\partial^2 \langle u_i \rangle}{\partial x_j x_j} - \frac{\partial \langle u_i' u_j' \rangle}{\partial x_j} + {Q_i}^R \tag{4.3}$$

$$\text{and} \quad \frac{\partial^2 \langle p \rangle}{\partial x_i \partial x_i} = -\frac{\partial^2}{\partial x_i \partial x_j}(\langle u_i \rangle \langle u_j \rangle + \langle u_i' u_j' \rangle) + \frac{\partial {Q_i}^R}{\partial x_i}, \tag{4.4}$$

where $\langle u_i \rangle$, $\langle p \rangle$, and $\langle u_i' u_j' \rangle$ are Reynolds-averaged velocity, kinematic pressure, and Reynolds stresses, respectively. The drift force Q_i^R is applied to the Reynolds-averaged equations; again to ensure the statistical consistency between RANS and LES. Its definition is provided later; see equation (4.9).

4.3.1.1 *Consistency*

In the next step, to achieve consistency, we need to define the drift terms. Before that, however, we first introduce the exponentially weighted averaging operator

$$\langle \phi \rangle^{\mathrm{EWA}}(t) = \frac{1}{T} \int_{-\infty}^{t} \phi_i(t') \mathrm{e}^{-(t-t')/T} \, \mathrm{d}t', \tag{4.5}$$

where t is the current time and T an averaging time scale.

It is a linear operator acting on instantaneous LES fields leading to the corresponding exponentially weighted averaged quantity. Considering that the physical interpretation of the filtered quantities in LES and the Reynolds-averaged quantities in RANS is different, we assume that for large values of T the exponentially weighted filtered- and the Reynolds-averaged quantities

approximately represent the same, i. e., that $\langle\phi\rangle^{\mathrm{EWA}} \approx \langle\phi\rangle$. Although this assumption is only rigorous for $T \to \infty$ and when filtering is applied to homogeneous directions of the flow with constant grid spacing, it is more consistent than comparing instantaneous filtered- with Reynolds-averaged fields as done in most other hybrid methods. Regions, where the LES mesh is adequately fine, are called LES regions (there the LES solution dominates); the remaining regions are called RANS regions (there the RANS solution is dominant). Consistency in these regions is achieved by applying drift terms, that is, ${Q_i}^L$ to the filtered momentum equation in all RANS regions and ${Q_i}^R$ to the Reynolds-averaged momentum equation in all LES regions.

4.3.1.2 *Drift terms*

Following the assumption of $\langle\phi\rangle^{\mathrm{EWA}} \approx \langle\phi\rangle$, the drift term in the filtered momentum equation is defined as

$$ {Q_i}^L = \begin{cases} (\langle u_i\rangle - \langle\overline{u_i}\rangle^{\mathrm{EWA}})/\tau_l + G_{ij}(\langle\overline{u_j}\rangle^{\mathrm{EWA}} - \overline{u_j})/\tau_g, & \text{in RANS regions} \\ 0 & \text{in LES regions,} \end{cases} \tag{4.6} $$

where

$$ G_{ij} = \frac{\langle\tau_{ij}\rangle^{\mathrm{EWA}} - \langle u_i u_j\rangle}{\langle\tau_{kk}\rangle^{\mathrm{EWA}} + \langle u_k u_k\rangle}. \tag{4.7} $$

The parameters τ_l and τ_g are the relaxation time scales for the EWA velocity and EWA turbulent stresses, respectively. The first term in equation (4.6) relaxes the averaged filtered velocity towards the RANS velocity, changing only the averaged velocity, not the fluctuations (for large T). The second term forces the turbulent stresses to relax towards the turbulent stresses, as predicted by the RANS model, and therefore only changes the fluctuations, but not the EWA velocity. According to our experience, since linear eddy viscosity RANS models cannot provide reliable Reynolds stresses (e. g. the $k-\epsilon$ model [146] does not properly describe the anisotropy of Reynolds stress tensors near walls), it is better to enforce consistency of the total turbulent kinetic energy in LES and RANS solutions ($\langle\tau_{ii}/2\rangle^{\mathrm{EWA}} \approx k^R$, where k^R is the turbulent kinetic energy in RANS) rather than between individual Reynolds stresses. Therefore, the term G_{ij} (equation (4.7)) in equation (4.6) is replaced by as

$$ G_{ij} = \frac{\langle\tau_{kk}/2\rangle^{\mathrm{EWA}} - k^R}{\langle\tau_{kk}/2\rangle^{\mathrm{EWA}} + k^R}\,\delta_{ij}, \tag{4.8} $$

which is an isotropic tensor. For the RANS equations, the drift term is defined as

$$Q_i{}^R = \begin{cases} (\langle u_i \rangle^{\mathrm{EWA}} - \langle u_i \rangle)/\tau_r, & \text{in LES regions} \\ 0 & \text{in RANS regions,} \end{cases} \tag{4.9}$$

where τ_r is the relaxation time scale for the RANS velocity. Similar to the momentum and pressure equations, drift terms are added to the k- and $\varepsilon-$ equations,i. e.,

$$Q^k = (\langle \tau_{ii}/2 \rangle^{\mathrm{EWA}} - k^R)/\tau_r, \tag{4.10}$$

and

$$Q^\epsilon = (\langle \epsilon \rangle^{\mathrm{EWA}} - \epsilon^R)/\tau_r, \tag{4.11}$$

where $\langle \epsilon \rangle^{\mathrm{EWA}}$ is the exponentially weighted averaged filtered total dissipation rate, i. e., th EWA of $\overline{\epsilon} = 2\nu \overline{S}_{ij}\overline{S}_{ij} - \tau_{ij}{}^{sgs}\overline{S}_{ij}$, where $\overline{S}_{ij}$ is the resolved rate-of-strain tensor.

4.3.2 *Loosely coupled dual-mesh hybrid LES/RANS method*

The problem with the above approach is the difficulty in implementing the method, in which information must be transferred between two solvers frequently, since they should run simultaneously. In addition, small step sizes are required for the URANS simulation, which can be computationally expensive. To address these issues, we propose periodically steady forcing in the RANS part. This allows to run LES and quasi-steady RANS sequentially, and experience shows that within a few iterations accurate solutions are achieved.

The first simplification would be to ignore time derivative terms in the RANS equations. The second change is to redefine the drift term $Q_i{}^R$ such that the RANS solution gets nudged to the LES velocity averaged over a period, which is consistent with the steady RANS solution, rather than to the EWA LES velocity. To further simplify the process, we ignore the source terms in the $k-$ and $\varepsilon-$ transport equations, and in the LES equations, we remove the turbulent stress relaxation term G_{ij}. Thus the modified source terms are:

$$Q_i{}^R = \begin{cases} (\langle u_i \rangle^{\mathrm{MA}} - \langle u_i \rangle)/\tau_r, & \text{in LES regions,} \\ 0 & \text{in RANS regions,} \end{cases} \tag{4.12}$$

and

$$Q_i^L = \begin{cases} (\langle u_i \rangle - \langle \overline{u_i} \rangle^{\mathrm{EWA}})/\tau_l, & \text{in RANS regions,} \\ 0 & \text{in LES regions,} \end{cases} \tag{4.13}$$

where $\langle u_i \rangle^{\mathrm{MA}}$ is the LES velocity averaged over a long time period.

4.3.3 *Data assimilation*

The main objective of the method is to obtain velocity profiles and wall shear stresses at the same time by assimilating only sparse measurements close to the wall. The data assimilation approach that we use to this end is based on a discrete-adjoint method [119]. In this method, a cost function is defined which measures the discrepancy between numerical results and measurements at specified locations. The sensitivity of the cost function with respect to the parameters, which shall be optimized to achieve agreement, is computed by solving so-called adjoint equations. The computational cost of the sensitivity computation is similar to that of solving the forward problem, and it is independent of the number of parameters.

The choice of parameters depends on the availability of data. Brenner et al. [119] showed that one needs to constrain the model to avoid ill-conditioning and non-uniqueness when only sparse measurement data is available. They corrected the eddy viscosity (instead of inferring the Reynolds stress tensor field directly) with a gradient-based optimizer. Further, the ambiguity of the problem was reduced by introducing a total variation (TV) penalty to the cost function, which acts as a regularization. However, Piroozmand et al. [147] showed that for cases where sparse data are located near the wall and where the wall shear stresses are sought, TV or Sobolev gradient regularization methods [71] fail to recover physical and accurate wall shear stresses. Instead, they suggested a reduction of the dimensionality of the parameter field, a method called Piecewise Linear Dimension Reduction (PLDR), as a regularization strategy to further constrain the problem. The method successfully avoids jagged wall shear stress and velocity profiles.

We follow this approach with the only difference that, in addition to the sparse wall shear stress measurement data, the internal velocity field of the loosely coupled hybrid model is used as synthetic data. This process can also be seen as a regularization strategy called data augmentation by artificial data, as it involves introducing additional information. Compared to the loosely coupled hybrid LES-RANS method, we also obtain more accurate

wall shear stresses as well as conservative solutions since the correction is made without adding source terms to the momentum equation. The tuned parameter α leads to a correction of the diffusive flux in the steady RANS equation

$$\frac{\partial(\langle u_i\rangle\langle u_j\rangle)}{\partial x_j} = -\frac{\partial\langle p\rangle}{\partial x_i} + \frac{\partial}{\partial x_j}\left[(\nu + \alpha\langle\nu_t\rangle)\,\frac{\langle u_i\rangle}{\partial x_j}\right] , \tag{4.14}$$

where $\langle\nu_t\rangle$ is the eddy viscosity computed by the RANS part of the hybrid model. DA corrects $\langle\nu_t\rangle$ through α instead of inferring the eddy viscosity directly, which helps the optimizer to narrow down the search space for an optimal eddy viscosity. In practice, the corrective field α distills the effect of the drift term in the hybrid equations into an optimal eddy viscosity and it compensates for the shortcomings of the RANS model (here the $k-\epsilon$ model) in incorporating near wall effects. Although complete internal data are provided by the loosely coupled hybrid method, we opt to only use sparse information in order to avoid irregular solutions at walls and to simplify the implementation.

As mentioned earlier, we use the PLDR method in which the α field is decomposed into piecewise linear elements whose nodes coincide with the measurement points. This helps the optimizer to search in a subspace instead and to reduce ill-conditioning and non-uniqueness. The nodal values, called β, are optimized instead of the cell values of α. The relation between β and α reads

$$\alpha_n = \phi_{nm}\,\beta_m , \tag{4.15}$$

where the subscripts $n = 1, \ldots, N$ and $m = 1, \ldots, M$ address cells and nodes, respectively, and ϕ is the piecewise linear projection matrix.

We then use a first-order gradient-based optimization to find the optimal parameter values based on the iterative procedure

$$\beta^{(t+1)} = \beta^{(t)} - \Delta^{(t)}\,\frac{\mathrm{d}f}{\mathrm{d}\beta}^{(t)} , \tag{4.16}$$

where $t = 0, \ldots, T_{max}$ denotes the iteration number of the optimization and T_{max} is the maximum number of iterations. The learning rate, or optimization step size, Δ can either be a scalar, a diagonal matrix, or a full matrix. The sensitivity $\frac{\mathrm{d}f}{\mathrm{d}\beta}$ of the cost function with respect to the parameters can easily be computed from $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ using the chain rule, while $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ itself is obtained from the discrete adjoint method. The cost function f is defined

as the discrepancy between the RANS results and the reference data at the measurement points. The readers are referred to our previous papers on adjoint-based data assimilation for more explanations [147]. Two important changes have been introduced compared to the previous method. The first is the mapping function that enforces the positivity of the α field and thus the eddy viscosity. The new mapping function reads

$$\alpha\left(\alpha'\right) = \begin{cases} \arctan(\alpha') + 1, & if \;\; \alpha' > 0 \\ \exp\left(\alpha'\right), & if \;\; \alpha' \leq 0\,, \end{cases} \tag{4.17}$$

which limits the maximum value of α to 2.57 in contrast to the previous version, where α was unbounded. The reason for this modification was the occurrence of large peaks that locally deteriorated the solution. The second change is a modification to the total variation penalty that is applied to the β values. The nodal distance was added to the denominator, since, in general, the nodes are not as uniformly distributed as grid cells. This implies that the total variation penalty is stronger where the nodes are closer to each other. The new penalty term added to the cost function thus reads

$$f_\beta = \omega_{TV,\beta} \sum_{m=1}^{M} \left[\frac{1}{|\mathcal{B}_m|} \sum_{o \in \mathcal{B}_m} \frac{(\beta_m - \beta_o)^2}{d_{mo}} \right], \tag{4.18}$$

where $\omega_{TV,\beta}$ is the weight factor of the penalty, $\mathcal{B}_m$ the set of neighborhood nodes, o denotes the adjacent nodes, and d_{mo} is the distance between the node m and its adjacent node o.

4.3.4 *Case setup*

The hybrid method and the data assimilation framework were developed based on *foam-extend-4.0*, which is a fork of OpenFOAM with implemented coupled solvers [118]. To demonstrate the proposed method, it was tested for the case of flow over periodic hills, which has been studied numerically [148] and experimentally [123] by various researchers. Although the geometry is simple, the flow exhibits separation and reattachment, which renders accurate predictions by CFD models difficult. We chose a rather high Reynolds number of $Re = 10595$ based on the bulk velocity U_B at the inflow and the height H of the crest. The geometry of the test case is depicted in Figure 2.1. At the walls no-slip and at in- and outflow as well in the span-wise direction periodic boundary conditions were applied. The bulk velocity is sustained by imposing a corresponding mean pressure gradient.

The computational grid for the URANS simulation of the tightly coupled hybrid method consists of $74 \times 37 \times 18$ cells in $x-$, $y-$, and $z-$ directions, which are aligned with the streamwise, wall-normal, and spanwise directions, respectively. The grid for the steady RANS simulation in the loosely coupled hybrid method and for data assimilation is two-dimensional and has $180 \times 130 \times 1$ cells in $x-$, $y-$, and $z-$ directions. The LES grid used for both the tightly and loosely coupled hybrid methods consists of $74 \times 37 \times 36$ cells and is close to uniform in all directions without attempting to resolve the boundary layer. The dimensionless wall distance of the wall-adjacent cells is $y^+ \approx 1 - 5$ for RANS, but much larger for LES.

The RANS regions are composed of all cells which are within $D = 0.2H$ from the nearest wall; the remaining cells are assigned to LES regions. The flow-through time is defined as $T_{thr} = 9H/U_b$, where $U_b = 0.52975$ and the averaging period of URANS and LES is $12 \times T_{thr}$. The time window of exponentially weighted averaging is $T = 4.24H/U_b$ for the tightly coupled approach and $T = 42.4H/U_b$ for the loosely coupled approach. The relaxation times for drift terms are the same for both approaches, specifically $\tau_l = 0.53H/U_b$, $\tau_g = 0.13H/U_b$, and $\tau_r = 0.53H/U_b$. The time step size for URANS and LES in the tightly coupled approach is $\Delta t = 2.65 \times 10^{-3} H/U_b$, while for the loosely coupled approach the LES time step is $0.0106H/U_b$.

The turbulence model used for the RANS simulations is the standard $k - \epsilon$ model with default coefficient values. A one-equation eddy viscosity model (OEE) is used with an additional transport equation for the subgrid-scale (SGS) turbulent kinetic energy [36]. The parameters used are provided in [14].

A low Reynolds number boundary condition was assumed for the eddy viscosity, which is based on a damping function that sets the eddy viscosity to zero at wall boundaries. For the data assimilation process, 16 uniformly distributed shear stress measurements at the lower and 16 at the upper wall were used. The measurement data are samples from a high-resolution LES simulation by Gloerfelt and Cinnella [123]. For the discretization of the convective and diffusive terms, second-order schemes were used. To solve the coupled system a bi-conjugate gradient stabilized linear solver with a Cholesky preconditioner was employed, and the optimization was based on the DEMON Adam method [121]. Further information on the configuration of the setup is provided in Ref. [147].

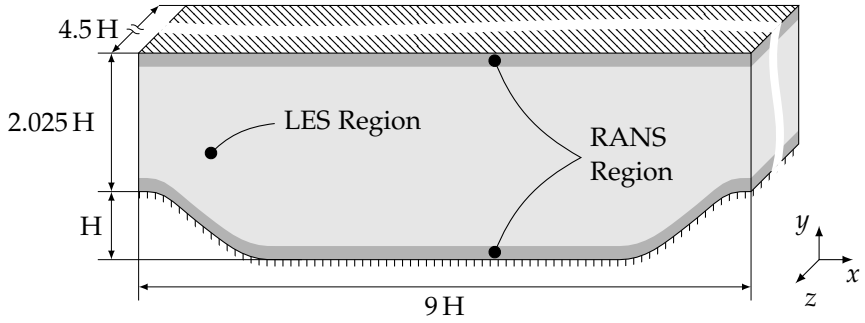


FIGURE 4.1: Periodic hills geometry and the predefined LES and RANS regions. All length scales are expressed with respect to the hill height H.

4.4 RESULTS AND DISCUSSION

In this section, we provide the results in the following order: First, the result of standalone steady RANS (S-RANS) and standalone under-resolved LES (UR-LES) are presented. Then, the results of the tightly coupled approach (hybrid UR-LES/U-RANS) are shown. Results of the loosely coupled approach (hybrid UR-LES/S-RANS) follow. The results of the data assimilation procedure are presented for the case where only sparse wall shear stresses are assimilated into an S-RANS (W-DA on S-RANS) and are compared to the case where the sampled velocity data from UR-LES of the loosely coupled hybrid model is added and assimilated (W-H-DA on S-RANS). All results are compared with a fully-resolved LES simulation (FR-LES) as a reference. The location of measurement and sampling points are shown in Figure 4.2. The streamwise velocity profiles and the wall shear stresses at the lower wall for S-RANS and UR-LES are shown in Figure 4.3. It can be observed that the velocity profiles of standalone UR-LES differ significantly from the reference, which shows the importance of resolving small eddies close to the wall. The standalone S-RANS shows better agreement with the reference FR-LES, but it deviates in the core of the free shear flow region. For the wall shear stress profiles, we can see that both approaches significantly deviate from the reference, especially the overprediction of S-RANS is evident. Note that wall shear stresses obtained from UR-LES may not be valid, since the first grid points lie outside of the viscous sublayer (their y^+ is in the range of 10).

Next, we look at the results of the tightly coupled approach. According to Figure 4.4 the streamwise velocity profiles show excellent agreement

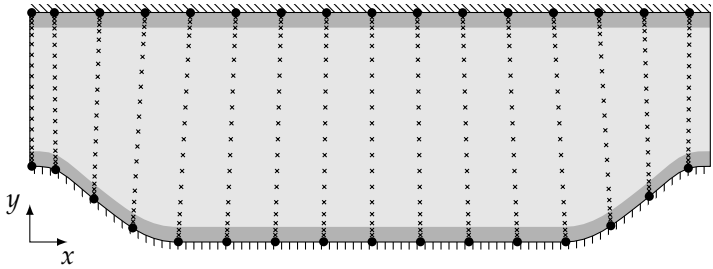


FIGURE 4.2: The location of sampling points is shown as crosses (×) and the location of measurement points at the walls is shown by black circles (•).

with the reference; especially when compared to stand-alone UR-LES. This emphasizes the importance of the U-RANS forcing applied to UR-LES and of the reciprocal feedback between UR-LES and U-RANS. However, the wall shear stress profiles of the tightly coupled approach do not show a substantial improvement compared to standalone U-RANS, although standalone U-RANS already provides more accurate results compared to S-RANS. The superiority of U-RANS in comparison with S-RANS could be due to the different meshes which were used. The mesh for U-RANS was highly resolved near the wall, but coarse in the free shear flow region. The S-RANS mesh was less resolved near the wall but much more uniform in the free shear flow region. Another explanation for the better performance of U-RANS may be the fact that the unsteadiness of the mean flow is resolved. Overall, the tightly coupled approach provides very accurate internal velocity profiles, but cannot improve wall shear stress computations.

We now take a look at the results of the loosely coupled approach. It can be seen from Figure 4.5 that the velocity profiles have significantly improved compared to the standalone UR-LES. However, the accuracy is a bit lower in comparison with the tightly coupled approach, which can be expected considering the assumption of steadiness. That is, the advantage of lower computational cost and ease of implementation seems to come with slightly lower accuracy. However, we can conclude that the loosely coupled approach still provides good results, especially when considering that it is the steady RANS that sequentially communicates with UR-LES. The assumption of steadiness in the near-wall region is therefore valid at least for the case of flow over periodic hills. The improvement in wall shear

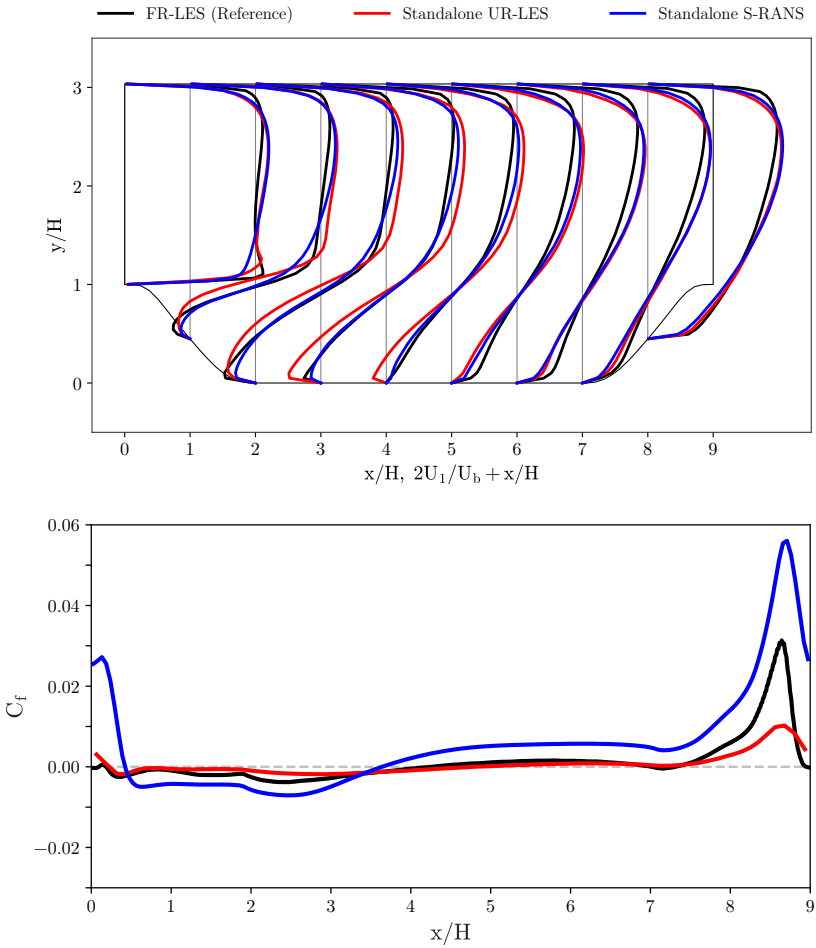


FIGURE 4.3: Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for the standalone cases of steady RANS (S-RANS) and underresolved LES (UR-LES).

stress estimations is limited, similar to the tightly coupling, which shows a general drawback of the dual-mesh hybrid approach.

The issue of inaccurate wall shear stresses of the hybrid approach can be circumvented by data assimilation if measurement data is available.

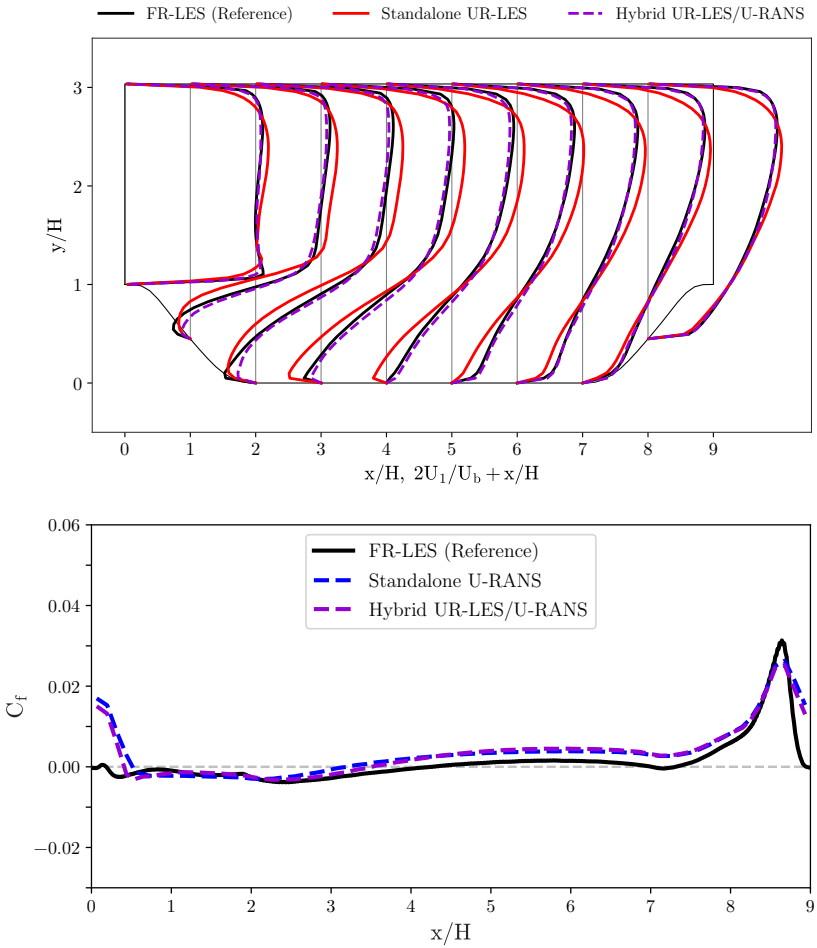


FIGURE 4.4: Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for the tightly coupled LES/RANS model. Note that the friction coefficient of the tightly coupled model is derived from its U-RANS solution.

However, before we proceed, let's look at the problem in another way. Assume we only have sparse wall shear stress measurements and that we want to assimilate this data set into an S-RANS model. The results in Figure 4.6 show that if we only assimilate sparse wall shear stress data,

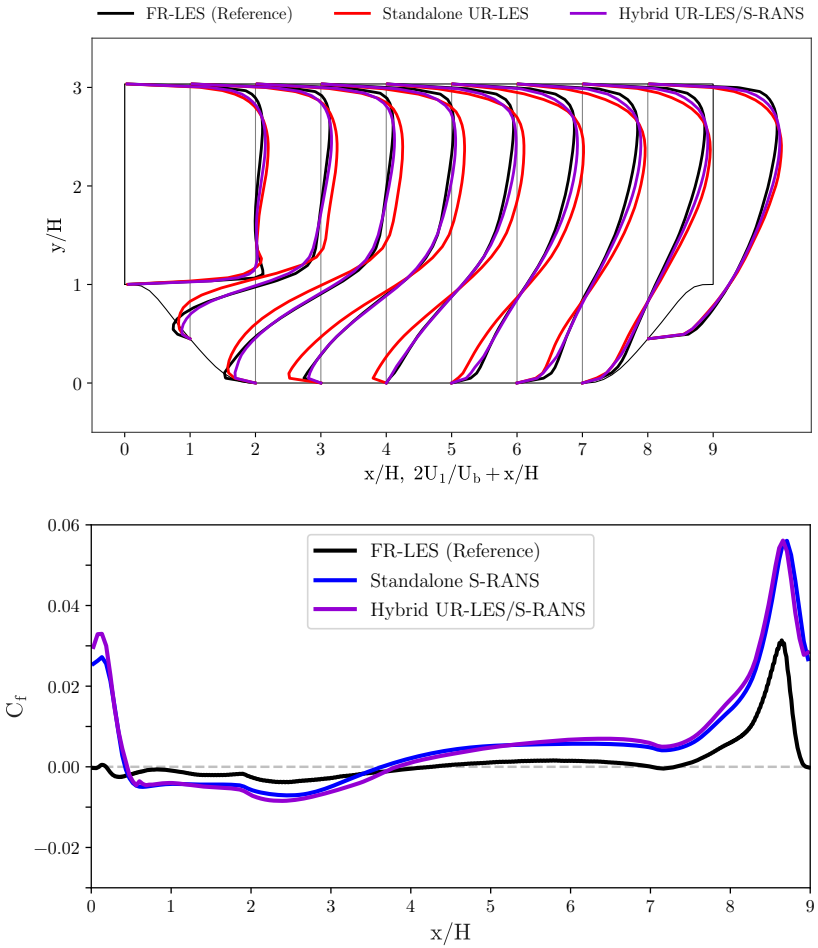


FIGURE 4.5: Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for the loosely coupled LES/RANS model. Note that the friction coefficient of the loosely coupled model is derived from its S-RANS solution.

the inverse problem becomes very ill-conditioned, thus resulting in very inaccurate velocity profiles, even though the wall shear stresses are rather accurate at most parts. This shows that data assimilation of wall shear

stresses aimed at finding the optimal eddy viscosity field is not able to infer correct velocities in most of the regions.

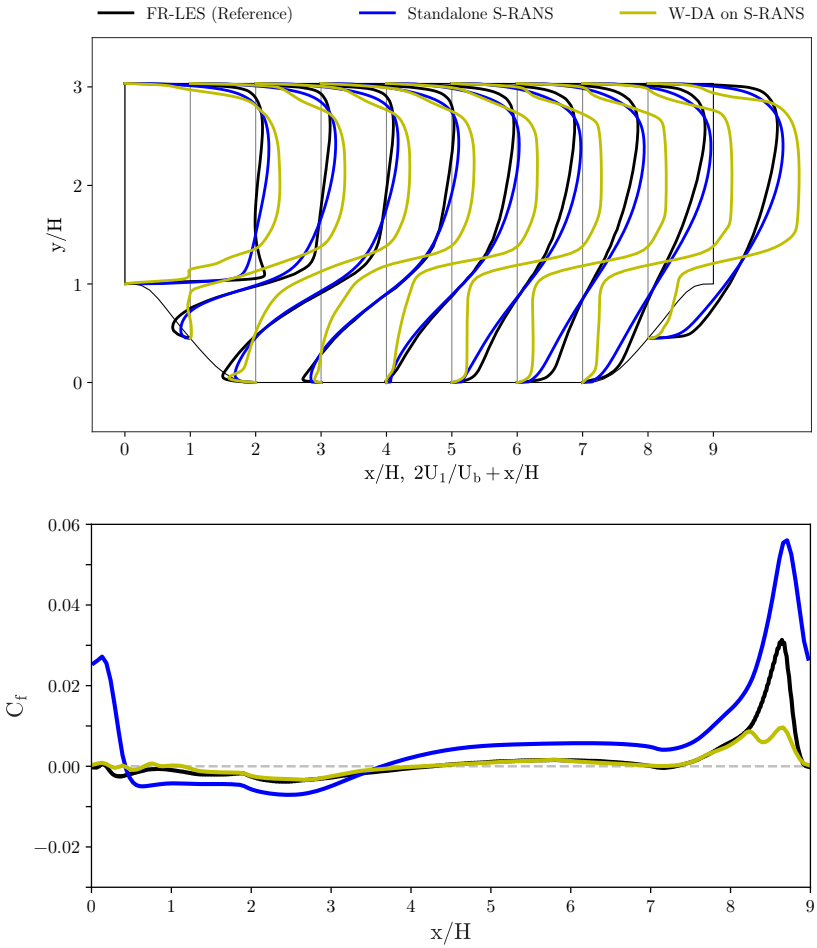


FIGURE 4.6: Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for data assimilation of sparse wall shear stress data on S-RANS.

Finally, the results of assimilating the solution of the loosely coupled hybrid approach augmented by sparse wall data are shown in Figure 4.7. It can be seen that the streamwise velocity profiles have improved in most of

the regions since the velocities in those regions were forced to match the loosely coupled solution by optimizing the eddy viscosity. In addition, we can see the wall shear stress is significantly more accurate than that obtained with the standalone RANS solver without compromising the velocities in the other regions. This shows that the idea of data augmentation of the DA procedure works, or from another perspective, we can say that we could improve the wall shear stress predictions of the loosely coupled approach by assimilating sparse wall shear stress data.

4.5 CONCLUSIONS AND OUTLOOK

In this work we propose a modification of the basic dual-mesh hybrid LES/RANS model in that the under-resolved LES model is coupled with a steady instead of an unsteady RANS simulation resulting in less frequent coupling and thus lower computational cost and simpler implementation. We then showed that wall shear stresses do not substantially improve with either approach and that if sparse wall shear stress measurements are available these can be used to improve the performance of such models via data assimilation. For example, one can consider a case where only sparse wall shear stress measurements are available (e. g. on an airfoil), while accurate velocities and wall shear stresses are sought. Here we propose to augment the data with the results of rather cheap loosely coupled hybrid LES/RANS simulations in order to reduce ill-conditioning and non-uniqueness in solving this inverse problem. The framework was tested for the case of flow over periodic hills for $Re = 10595$. The results confirm both the capability of the loosely coupled approach and the importance of augmenting the sparse wall data assimilation.

Future work should further verify the validity of the assumption of steady RANS in the wall regions for other complex cases. In addition, the loose coupling can further be improved by performing shorter LES after each RANS simulation. Since the communication between the S-RANS and UR-LES solvers occurs sequentially and infrequently (in contrast to the tightly coupled hybrid method) it is efficient enough to use separate codes and to simplify the information transfer via files.

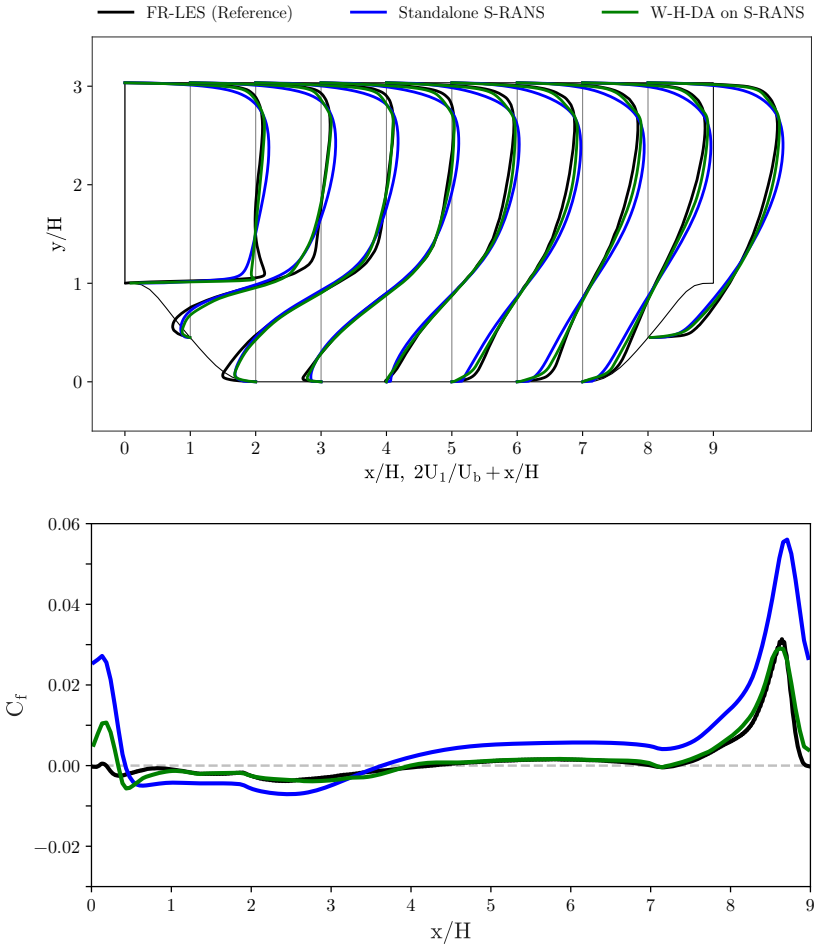


FIGURE 4.7: Horizontal velocity component u_1 (top plot) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot) for data assimilation of sparse wall shear stress and the solution of the loosely coupled model on S-RANS.

5

SUMMARY AND OUTLOOK

In this project, we investigated different approaches to efficiently incorporate sparse measurements into the turbulence models to obtain as accurate and physical results as possible. We defined a data assimilation problem in which the eddy viscosity of an LEV RANS model is optimized such that the discrepancy between the sparse measurement data and the solution, measured by a cost function, is minimized. We used the discrete adjoint method to compute the gradients of this cost function with respect to the parameter field. We observed that the assimilation of sparse data into the model is severely under-determined and ill-posed resulting in nonphysical jagged velocity profiles that are accurate at the measurement locations but deviate largely in other regions. This motivated the use of different regularization schemes. The L_2, total variation, and Sobolev gradient regularization methods were used, which, to some extent, could avoid unphysical solutions. The quality of the DA approach and its generalizability was tested not only based on the discrepancies at the measurement locations, but also at other locations called testing points. Applying the regularization methods, we could obtain accurate velocity profiles. However, we observed that the reconstructed wall shear stresses were not satisfactory for all the mentioned regularization methods, even if wall shear stress measurements were available. To circumvent this issue, we proposed a novel approach that reduces the dimension of the problem, such that the corrective field is forced to behave piecewise linearly thus ensuring noise-free parameter fields. The method is called piecewise linear dimension reduction (PLDR). It was shown that PLDR can provide accurate and smooth velocity profiles as well as wall shear stresses. Next, we investigated a scenario where only sparse wall shear measurements are available but accurate wall shear stress and velocity profiles are sought. We showed that with basic DA we get a very inaccurate solution in all regions. Therefore, we attempted to generate additional data by running a high-fidelity, but computationally efficient and easy-to-implement simulation. An existing dual-mesh hybrid LES/RANS method was modified such that the under-resolved LES is loosely coupled with a steady RANS. We showed that the loose coupling approach provides accurate results with ease of implementation. However, we observed that the wall shear stress did not improve, compared to that obtained with the

tightly coupled approach. It thus motivated us to incorporate sparse wall shear stress measurements. Both the DA process and the hybrid model benefited from their combination. Using the PLDR approach for DA, we were able to provide accurate velocity profiles and wall shear stresses based on a computationally efficient simulation and only a few wall shear stress data.

The first conclusion is that the choice of linear eddy viscosity model was an appropriate modeling level. It provided enough flexibility to search the solution space and find the optimal one. On the other hand, with help of regularization, it provided enough rigidity to reduce the degrees of freedom avoiding many nonphysical solutions that we may have encountered if we have chosen the Reynolds stress tensor or Reynolds forcing vector as parameters. For the separated flow over periodic hills, the data-driven LEV model was able to reproduce accurate velocity profiles and wall shear stresses which is a significant result and shows that the inaccuracies of LEV models for many problems may be due to the inaccurate calculation of eddy viscosity itself rather than the LEV assumption.

The second conclusion is that regularization is a necessity for the sparse data-driven LEV models. We showed that without any form of regularization the reconstructed profiles are overfitted, noisy, and nonphysical. Among the regularization schemes TV provided the best velocity profiles, if the weighting factor is optimally chosen. Sobolev gradient, on the other hand, had the advantage of having a hyperparameter, filtering length, that has physical meaning whose optimal value is easy to guess. L_2, TV, and Sobolev gradient failed to reconstruct correct wall shear stresses. PLDR method was motivated by this observation and we showed that is able to reconstruct both velocity profiles and wall shear stresses. We, however, showed that PLDR also needs some form of regularization on its parameters to reach optimal results.

The third conclusion is that the proposed DA works only if data is distributed in the whole domain. We, specifically, showed that if data is only available near the wall the reconstructed profiles are very inaccurate since the optimizer has little information in the free shear flow region to infer correct parameters. We thus suggested the data augmentation using a higher fidelity method namely the loosely coupled dual-mesh hybrid LES/RANS method.

The fourth conclusion is that both tightly coupled and loosely coupled dual-mesh hybrid methods are able to provide accurate velocity profiles but are unable to improve wall shear stresses compared to standalone RANS

simulations. The loose coupling has the advantage of easier implementation and the RANS part, due to steady state assumption, is able to assimilate measurement data.

And finally, we showed that our framework is able to provide accurate velocity profiles and wall shear stresses using only a few near-wall measurements. The computational cost of the approach is low compared to conventional hybrid LES/RANS, LES, and DNS methods.

The following limitations of the project can be stated:

- The approximations in the calculation of gradients introduced inaccuracies, which could affect the efficiency of the optimization process.
- DA was performed only for LEV models. Therefore, the improvement could be limited in some cases.
- Inaccuracies in the gradient computation at inflow and outflow boundaries can be attributed to the limitations in the OpenFOAM implementation.
- The initial eddy viscosity was computed based on the $k - \varepsilon$ model. The effect of the initialization with different LEV models was not further studied.
- All the proposed methods were tested for one particular Re number, geometry, and measurement configurations.
- Due to implementation issues, quasi-Newton optimization methods such as the BFGS method were not tested. They may accelerate the process.
- The performance of the loose coupling method was only tested for the 2-dimensional case of flow over periodic hills.

The following suggestions and outlooks can be drawn for future works:

- DA can be incorporated to correct the LEV model by including anisotropy effects (For example the parameters would be a vector instead of a scalar field). This may increase the degrees of freedom of the optimization problem and thus needs further regularization.
- The calculation of the gradients can be improved using Automatic Differentiation.

- A data-driven ML model can be developed based on the DA process for different configurations such that the corrective field can automatically be constructed based on local flow features without the need for sparse measurements and DA.
- To further verify the capability of the proposed methods, they can be tested for more complex geometries such as 3D geometries with various flow conditions.
- Different LEV models can be tested to see the effect of the initialization of the eddy viscosity in the optimization process.
- The PLDR method was only developed for two-dimensional cases. The code can be extended for 3D cases.
- The implementation of the loose coupling can further be improved. Since the forcing applied to the LES model, which depends on the RANS simulation, changes slightly from one exchange iteration to the next, averaging the LES solution may be done over a much shorter period.

BIBLIOGRAPHY

1. Kumar, K. *Engineering fluid mechanics* (S. Chand Publishing, 2008).
2. contributors, W. *Lagrangian and Eulerian specification of the flow field — Wikipedia, The Free Encyclopedia* [Online; accessed 12-July-2022]. 2022.
3. Batchelor, C. K. & Batchelor, G. *An introduction to fluid dynamics* (Cambridge university press, 2000).
4. Bouremel, Y., Yianneskis, M. & Ducci, A. On the utilisation of vorticity and strain dynamics for improved analysis of stirred processes. *Chemical Engineering Research and Design* **87**, 377 (2009).
5. Pukhnachev, V. Mathematical model of an incompressible viscoelastic Maxwell medium. *Journal of applied mechanics and technical physics* **51**, 546 (2010).
6. Pope, S. B. & Pope, S. B. *Turbulent flows* (Cambridge university press, 2000).
7. Markatos, N. The mathematical modelling of turbulent flows. *Applied Mathematical Modelling* **10**, 190 (1986).
8. Richardson, L. F. & Lynch, P. *Weather Prediction by Numerical Process* 2nd ed. (Cambridge University Press, 2007).
9. Kolmogorov, A. N. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Cr Acad. Sci. URSS* **30**, 301 (1941).
10. Box, G. All models are wrong, but some are useful. *Robustness in Statistics* **202**, 549 (1979).
11. *What is Turbulent Flow?* 2021.
12. Ziaei, D., Hekmati Athar, S. P. & Goudarzi, N. *Assessment of a CFD-based machine learning approach on turbulent flow approximation* in *Energy Sustainability* **59094** (2019), V001T14A003.
13. Basu, S. in *Handbook of Environmental Fluid Dynamics, Volume Two: Systems, Pollution, Modeling, and Measurements* (ed Fernando, H. J.) chap. 20 (CRC press, 2012).

14. Xiao, H. & Jenny, P. A consistent dual-mesh framework for hybrid LES/RANS modeling. *Journal of Computational Physics* **231**, 1848 (2012).

15. Duraisamy, K., Iaccarino, G. & Xiao, H. Turbulence modeling in the age of data. *arXiv preprint arXiv:1804.00183* (2018).

16. Spalart, P. R. Philosophies and fallacies in turbulence modeling. *Progress in Aerospace Sciences* **74**, 1 (2015).

17. Banerjee, S., Krahl, R., Durst, F. & Zenger, C. Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches. *Journal of Turbulence*, N32 (2007).

18. Apsley, D. *Lecture notes in Computational Hydraulics* 2022.

19. Prandtl, L. 7. Bericht über Untersuchungen zur ausgebildeten Turbulenz. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* **5**, 136 (1925).

20. McDonough, J. M. Introductory lectures on turbulence: physics, mathematics and modeling (2007).

21. Wilcox, D. C. *et al. Turbulence modeling for CFD* (DCW industries La Canada, CA, 1998).

22. Johnson, D. & King, L. *A new turbulence closure model for boundary layer flows with strong adverse pressure gradients and separation* in *22nd Aerospace Sciences Meeting* (1984), 175.

23. Baldwin, B. & Barth, T. *A one-equation turbulence transport model for high Reynolds number wall-bounded flows* in *29th aerospace sciences meeting* (1991), 610.

24. Spalart, P. & Allmaras, S. *A one-equation turbulence model for aerodynamic flows* in *30th aerospace sciences meeting and exhibit* (American Institute of Aeronautics and Astronautics, 1992), 439.

25. Launder, B. E. & Spalding, D. B. in *Numerical prediction of flow, heat transfer, turbulence and combustion* 96 (Elsevier, 1983).

26. Tsukahara, T. *Analytical derivation* Accessed: 20-7-2022.

27. Spalding, B. in *Advances in heat transfer* 1 (Elsevier, 2013).

28. Wilcox, D. C. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA journal* **26**, 1299 (1988).

29. Menter, F. R. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal* **32**, 1598 (1994).

30. Xiao, H., Wu, J.-L., Laizet, S. & Duan, L. Flows over periodic hills of parameterized geometries: a dataset for data-driven turbulence modeling from direct simulations. *Computers & Fluids* **200**, 104431 (2020).

31. Smagorinsky, J. General circulation experiments with the primitive equations: I. The basic experiment. *Monthly weather review* **91**, 99 (1963).

32. Lilly, K. E. *On the application of the eddy viscosity concept in the Inertial sub-range of turbulence* in (1966).

33. Deardorff, J. The use of subgrid transport equations in a three-dimensional model of atmospheric turbulence (1973).

34. Kosović, B. Subgrid-scale modelling for the large-eddy simulation of high-Reynolds-number boundary layers. *Journal of Fluid Mechanics* **336**, 151 (1997).

35. Germano, M., Piomelli, U., Moin, P. & Cabot, W. H. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics* **3**, 1760 (1991).

36. Yoshizawa, A. & Horiuti, K. A statistically-derived subgrid-scale kinetic energy model for the large-eddy simulation of turbulent flows. *Journal of the Physical Society of Japan* **54**, 2834 (1985).

37. Deardorff, J. W. Three-dimensional numerical study of the height and mean structure of a heated planetary boundary layer. *Boundary-Layer Meteorology* **7**, 81 (1974).

38. Kundu, P. K., Cohen, I. M. & Dowling, D. R. *Fluid mechanics* (Academic press, 2015).

39. Avraham, T. *Law of the Wall* [Online; accessed 25-July-2022]. 2018.

40. White, F. M. *Fluid mechanics* (Tata McGraw-Hill Education, 1979).

41. Richards, P. & Norris, S. Appropriate boundary conditions for computational wind engineering models revisited. *Journal of Wind Engineering and Industrial Aerodynamics* **99**, 257 (2011).

42. Yang, X., Sadique, J., Mittal, R. & Meneveau, C. Integral wall model for large eddy simulations of wall-bounded turbulent flows. *Physics of Fluids* **27**, 025112 (2015).

43. Suga, K. Analytical wall-functions of turbulence for complex surface flow phenomena. *Computational Fluid Dynamics and Heat Transfer Emerging Topics*, 331 (2010).

44. Craft, T., Gerasimov, A., Iacovides, H. & Launder, B. Progress in the generalization of wall-function treatments. *International Journal of Heat and Fluid Flow* **23**, 148 (2002).
45. Hanjalić, K. & Hrebtov, M. Ground boundary conditions for thermal convection over horizontal surfaces at high rayleigh numbers. *Boundary-layer meteorology* **160**, 41 (2016).
46. Kawai, S. & Larsson, J. Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy. *Physics of Fluids* **24**, 015105 (2012).
47. Balaras, E., Benocci, C. & Piomelli, U. Two-layer approximate boundary conditions for large-eddy simulations. *AIAA journal* **34**, 1111 (1996).
48. Fröhlich, J. & Von Terzi, D. Hybrid LES/RANS methods for the simulation of turbulent flows. *Progress in Aerospace Sciences* **44**, 349 (2008).
49. Spalart, P. R. Detached-eddy simulation. *Annual review of fluid mechanics* **41**, 181 (2009).
50. Durbin, P. A. Some recent developments in turbulence closure modeling. *Annual Review of Fluid Mechanics* **50**, 77 (2018).
51. Durbin, P. & Shih, T. An overview of turbulence modeling. *Modelling and Simulation of Turbulent Heat Transfer* **16**, 3 (2005).
52. Li, Y., Perlman, E., Wan, M., Yang, Y., Meneveau, C., Burns, R., Chen, S., Szalay, A. & Eyink, G. A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. *Journal of Turbulence*, N31 (2008).
53. Rumsey, C. L. *Recent developments on the turbulence modeling resource website* in *22nd AIAA Computational Fluid Dynamics Conference* (2015), 2927.
54. Margheri, L., Meldi, M., Salvetti, M. V. & Sagaut, P. Epistemic uncertainties in RANS model free coefficients. *Computers & Fluids* **102**, 315 (2014).
55. Schaefer, J., Hosder, S., West, T., Rumsey, C., Carlson, J.-R. & Kleb, W. Uncertainty quantification of turbulence model closure coefficients for transonic wall-bounded flows. *AIAA Journal* **55**, 195 (2017).
56. Edeling, W., Cinnella, P. & Dwight, R. P. Predictive RANS simulations via Bayesian model-scenario averaging. *Journal of Computational Physics* **275**, 65 (2014).

57. Edeling, W. N., Cinnella, P., Dwight, R. P. & Bijl, H. Bayesian estimates of parameter variability in the k–ε turbulence model. *Journal of Computational Physics* **258**, 73 (2014).

58. Edeling, W. N., Iaccarino, G. & Cinnella, P. Data-free and data-driven RANS predictions with quantified uncertainty. *Flow, Turbulence and Combustion* **100**, 593 (2018).

59. Vollant, A., Balarac, G., Geraci, G. & Corre, C. E. *Optimal estimator and artificial neural network as efficient tools for the subgrid-scale scalar flux modeling* in *Proceedings of the Summer Program* (2014), 435.

60. Tracey, B., Duraisamy, K. & Alonso, J. *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* (2013), 259.

61. Wu, J., Xiao, H., Sun, R. & Wang, Q. Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned. *Journal of Fluid Mechanics* **869**, 553 (2019).

62. Dow, E. & Wang, Q. *Uncertainty quantification of structural uncertainties in RANS simulations of complex flows* in *20th AIAA Computational Fluid Dynamics Conference* (2011), 3865.

63. Singh, A. P., Medida, S. & Duraisamy, K. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA journal* **55**, 2215 (2017).

64. Ling, J., Kurzawski, A. & Templeton, J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics* **807**, 155 (2016).

65. Pope, S. A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics* **72**, 331 (1975).

66. Wu, J.-L., Xiao, H. & Paterson, E. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids* **3**, 074602 (2018).

67. Lumley, J. L. & Newman, G. R. The return to isotropy of homogeneous turbulence. *Journal of Fluid Mechanics* **82**, 161 (1977).

68. Parish, E. J. & Duraisamy, K. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of computational physics* **305**, 758 (2016).

69. Ho, J. & West, A. *Field Inversion and Machine Learning for turbulence modelling applied to three-dimensional separated flows* in *AIAA Aviation 2021 Forum* (2021), 2903.

70. Li, Z., Hoagg, J. B., Martin, A. & Bailey, S. C. Retrospective cost adaptive Reynolds-averaged Navier–Stokes k–ω model for data-driven unsteady turbulent simulations. *Journal of Computational Physics* **357**, 353 (2018).

71. Ali, M. Y. B., Tissot, G., Heitz, D., Aguinaga, S. & Mémin, E. *An adjoint approach for the analysis of RANS closure using pressure measurements on a high-rise building* Preprint. 2020.

72. Foures, D. P., Dovetta, N., Sipp, D. & Schmid, P. J. A data-assimilation method for Reynolds-averaged Navier–Stokes-driven mean flow reconstruction. *Journal of fluid mechanics* **759**, 404 (2014).

73. Liu, Y. & Zhang, W. A new data assimilation method of recovering turbulent flow field at high-Reynolds numbers for turbulence machine learning. *arXiv preprint arXiv:2007.05671* (2020).

74. Li, S., He, C. & Liu, Y. A data assimilation model for wall pressure-driven mean flow reconstruction. *Physics of Fluids* **34**, 015101 (2022).

75. Mons, V. & Marquet, O. Linear and nonlinear sensor placement strategies for mean-flow reconstruction via data assimilation. *Journal of Fluid Mechanics* **923** (2021).

76. Robinson, C. *Image data assimilation with fluid dynamics models: Application to 3D flow reconstruction* PhD thesis (Rennes 1, 2015).

77. Franceschini, L., Sipp, D. & Marquet, O. Mean-flow data assimilation based on minimal correction of turbulence models: Application to turbulent high Reynolds number backward-facing step. *Physical Review Fluids* **5**, 094603 (2020).

78. Symon, S., Dovetta, N., McKeon, B. J., Sipp, D. & Schmid, P. J. Data assimilation of mean velocity from 2D PIV measurements of flow over an idealized airfoil. *Experiments in fluids* **58**, 1 (2017).

79. Yegavian, R., Leclaire, B., Champagnat, F. & Marquet, O. *Performance assessment of PIV super-resolution with adjoint-based data assimilation* in *11th International symposium on particle image velocimetry, Santa Barbara, USA* (2015), 14.

80. Asch, M., Bocquet, M. & Nodet, M. *Data assimilation: methods, algorithms, and applications* (SIAM, 2016).

81. Fujii, K. Extended kalman filter. *Refernce Manual*, 14 (2013).

82. Andrieu, C. & Doucet, A. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **64**, 827 (2002).

83. Evensen, G. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics* **53**, 343 (2003).

84. Surace, S. C., Kutschireiter, A. & Pfister, J.-P. How to avoid the curse of dimensionality: Scalability of particle filters with and without importance weights. *SIAM review* **61**, 79 (2019).

85. Fang, P., He, C., Wang, P., Xu, S. & Liu, Y. Data assimilation of steam flow through a control valve using ensemble Kalman filter. *Journal of Fluids Engineering* **143** (2021).

86. Introini, C., Lorenzi, S., Cammi, A., Baroli, D., Peters, B. & Bordas, S. A mass conservative Kalman filter algorithm for computational thermo-fluid dynamics. *Materials* **11**, 2222 (2018).

87. Deng, Z., He, C., Wen, X. & Liu, Y. Recovering turbulent flow field from local quantity measurement: turbulence modeling using ensemble-Kalman-filter-based data assimilation. *Journal of Visualization* **21**, 1043 (2018).

88. Reagan, A. J., Dubief, Y., Dodds, P. S. & Danforth, C. M. Predicting flow reversals in a computational fluid dynamics simulated thermosyphon using data assimilation. *PloS one* **11**, e0148134 (2016).

89. Kato, H., Yoshizawa, A., Ueno, G. & Obayashi, S. A data assimilation methodology for reconstructing turbulent flows around aircraft. *Journal of Computational Physics* **283**, 559 (2015).

90. Mons, V., Chassaing, J.-C., Gomez, T. & Sagaut, P. Reconstruction of unsteady viscous flows using data assimilation schemes. *Journal of Computational Physics* **316**, 255 (2016).

91. Xiao, H., Wu, J.-L., Wang, J.-X., Sun, R. & Roy, C. Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier–Stokes simulations: A data-driven, physics-informed Bayesian approach. *Journal of Computational Physics* **324**, 115 (2016).

92. Zhang, X.-L., Xiao, H., He, G.-W. & Wang, S.-Z. Assimilation of disparate data for enhanced reconstruction of turbulent mean flows. *Computers & Fluids* **224**, 104962 (2021).

93. Zhang, X.-L., Xiao, H. & He, G. Assessment of Regularized Ensemble Kalman Method for Inversion of Turbulence Quantity Fields. *AIAA Journal* **60**, 3 (2022).

94. Rabier, F. & Liu, Z. *Variational data assimilation: theory and overview* in *Proc. ECMWF Seminar on Recent Developments in Data Assimilation for Atmosphere and Ocean, Reading, UK, September 8–12* (2003), 29.

95. Frerix, T., Kochkov, D., Smith, J., Cremers, D., Brenner, M. & Hoyer, S. *Variational data assimilation with a learned inverse observation operator* in *International Conference on Machine Learning* (2021), 3449.

96. Martins, J. R. R. A., Sturdza, P. & Alonso, J. J. The complex-step derivative approximation. *ACM Transactions on Mathematical Software (TOMS)* **29**, 245 (2003).

97. Martins, J., Sturdza, P. & Alonso, J. *The connection between the complex-step derivative approximation and algorithmic differentiation* in *39th Aerospace Sciences Meeting and Exhibit* (2001), 921.

98. Giles, M., Ghate, D. & Duta, M. C. Using automatic differentiation for adjoint CFD code development (2005).

99. Mader, C. A., Martins, J. R., Alonso, J. J. & Van Der Weide, E. ADjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA journal* **46**, 863 (2008).

100. Fleischli, B., Mangani, L., Del Rio, A. & Casartelli, E. A discrete adjoint method for pressure-based algorithms. *Computers & Fluids* **227**, 105037 (2021).

101. Roth, R. & Ulbrich, S. A discrete adjoint approach for the optimization of unsteady turbulent flows. *Flow, turbulence and combustion* **90**, 763 (2013).

102. Ströfer, C. A. M. & Xiao, H. End-to-end differentiable learning of turbulence models from indirect observations. *Theoretical and Applied Mechanics Letters* **11**, 100280 (2021).

103. He, P., Mader, C. A., Martins, J. R. & Maki, K. J. An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM. *Computers & Fluids* **168**, 285 (2018).

104. He, C., Liu, Y. & Gan, L. A data assimilation model for turbulent flows using continuous adjoint formulation. *Physics of fluids* **30**, 105108 (2018).

105. Othmer, C., de Villiers, E. & Weller, H. *Implementation of a continuous adjoint for topology optimization of ducted flows* in *18th AIAA Computational Fluid Dynamics Conference* (2007), 3947.

106. Kenway, G. K. W., Mader, C. A., He, P. & Martins, J. R. R. A. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences* **110**, 100542 (2019).

107. Guo, M. & Hesthaven, J. S. Data-driven reduced order modeling for time-dependent problems. *Computer methods in applied mechanics and engineering* **345**, 75 (2019).

108. Brunton, S. L. & Kutz, J. N. *Data-driven science and engineering: Machine learning, dynamical systems, and control* (Cambridge University Press, 2022).

109. Borggaard, J., Duggleby, A., Hay, A., Iliescu, T. & Wang, Z. *Reduced-order modeling of turbulent flows* in *Proceedings of MTNS* **2008** (2008).

110. Hijazi, S., Stabile, G., Mola, A. & Rozza, G. Data-driven POD-Galerkin reduced order model for turbulent flows. *Journal of Computational Physics* **416**, 109513 (2020).

111. Imtiaz, H. & Akhtar, I. Nonlinear closure modeling in reduced order models for turbulent flows: a dynamical system approach. *Nonlinear Dynamics* **99**, 479 (2020).

112. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* **378**, 686 (2019).

113. He, P., Mader, C. A., Martins, J. R. R. A. & Maki, K. *An object-oriented framework for rapid discrete adjoint development using OpenFOAM* in *AIAA Scitech 2019 Forum* (American Institute of Aeronautics and Astronautics, 2019), 1210.

114. Gauger, N. R., Walther, A., Moldenhauer, C. & Widhalm, M. in *New Results in Numerical and Experimental Fluid Mechanics VI* 454 (Springer, 2007).

115. Gomes, P. & Palacios, R. Pitfalls of discrete adjoint fixed-points based on algorithmic differentiation. *AIAA Journal* **60**, 1 (2021).

116. Caretto, L. S., Gosman, A. D., Patankar, S. V. & Spalding, D. B. *Two calculation procedures for steady, three-dimensional flows with recirculation* in *Proceedings of the third international conference on numerical methods in fluid mechanics* (Springer Berlin Heidelberg, 1973), 60.

117. Clifford, I. Block-coupled simulations using openfoam. *Presentation*, 1 (2011).

118. Jareteg, K. Block coupled calculations in OpenFOAM. *Project within course: CFD with OpenSource software. Chalmers University of Technology* (2012).

119. Brenner, O., Piroozmand, P. & Jenny, P. *Efficient assimilation of sparse data into RANS-based turbulent flow simulations using a discrete adjoint method* submitted to JCP. 2022.

120. Liu, D. C. & Nocedal, J. On the limited memory BFGS method for large scale optimization. *Mathematical programming* **45**, 503 (1989).

121. Chen, J. & Kyrillidis, A. *Decaying momentum helps neural network training* 2019.

122. Chen, J. *An updated overview of recent gradient descent algorithms* `https://johnchenresearch.github.io/demon/`. Accessed: 2022-03-01. 2020.

123. Gloerfelt, X. & Cinnella, P. Large eddy simulation requirements for the flow over periodic hills. *Flow, Turbulence and Combustion* **103**, 55 (2019).

124. Huang, Y., Ng, M. K. & Wen, Y.-W. A fast total variation minimization method for image restoration. *Multiscale Modeling & Simulation* **7**, 774 (2008).

125. Liu, W., Fang, J., Rolfo, S., Moulinec, C. & Emerson, D. R. An iterative machine-learning framework for RANS turbulence modeling. *International Journal of Heat and Fluid Flow* **90**, 108822 (2021).

126. Craft, T. J., Launder, B. E. & Suga, K. Development and application of a cubic eddy-viscosity model of turbulence. *International Journal of Heat and Fluid Flow* **17**, 108 (1996).

127. Gatski, T. B. & Speziale, C. G. On explicit algebraic stress models for complex turbulent flows. *Journal of fluid Mechanics* **254**, 59 (1993).

128. Hanjalić, K. Advanced turbulence closure models: a view of current status and future prospects. *International Journal of Heat and Fluid Flow* **15**, 178 (1994).

129. Larsson, J., Kawai, S., Bodart, J. & Bermejo-Moreno, I. Large eddy simulation with modeled wall-stress: recent progress and future directions. *Mechanical Engineering Reviews* **3**, 15 (2016).

130. Bocquet, S., Sagaut, P. & Jouhaud, J. A compressible wall model for large-eddy simulation with application to prediction of aerothermal quantities. *Physics of fluids* **24**, 065103 (2012).

131. Schlüter, J. *Toward the prediction of turbulent boundary layers using a coupled RANS-LES method* in *Proceedings of the Summer Program* (2006), 179.

132. Schlüter, J. U., Wu, X., Kim, S., Shankaran, S., Alonso, J. & Pitsch, H. A framework for coupling Reynolds-averaged with large-eddy simulations for gas turbine applications. *Journal of Fluids Engineering* **127**, 806 (2005).

133. Haering, S. W., Oliver, T. A. & Moser, R. D. Active model split hybrid RANS/LES. *Physical Review Fluids* **7**, 014603 (2022).

134. Ashton, N., West, A. & Mendonça, F. Flow dynamics past a 30P30N three-element airfoil using improved delayed detached-eddy simulation. *AIAA Journal* **54**, 3657 (2016).

135. Gritskevich, M. S., Garbaruk, A. V. & Menter, F. R. Fine-tuning of DDES and IDDES formulations to the k-ω shear stress transport model. *Progress in Flight Physics* **5**, 23 (2013).

136. Spalart, P. R., Deck, S., Shur, M. L., Squires, K. D., Strelets, M. K. & Travin, A. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theoretical and computational fluid dynamics* **20**, 181 (2006).

137. Thé, J. & Yu, H. A critical review on the simulations of wind turbine aerodynamics focusing on hybrid RANS-LES methods. *Energy* **138**, 257 (2017).

138. Panguluri, S., Reasor, D. & LeBeau, R. *Investigation of grey area construction on the performance of detached eddy simulation* in *18th AIAA Computational Fluid Dynamics Conference* (2007), 4095.

139. Piomelli, U., Balaras, E., Pasinato, H., Squires, K. D. & Spalart, P. R. The inner–outer layer interface in large-eddy simulations with wall-layer models. *International Journal of heat and fluid flow* **24**, 538 (2003).

140. Davidson, L. Hybrid LES–RANS: back scatter from a scale-similarity model used as forcing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **367**, 2905 (2009).

141. Xiao, H., Wang, J. & Jenny, P. Dynamic evaluation of mesh resolution and its application in hybrid LES/RANS methods. *Flow, turbulence and combustion* **93**, 141 (2014).

142. Xiao, H., Wang, J.-X. & Jenny, P. An implicitly consistent formulation of a dual-mesh hybrid LES/RANS method. *Communications in Computational Physics* **21**, 570 (2017).

143. Revell, A., Afgan, I., Ali, A., Santasmasas, M., Craft, T., de Rosis, A., Holgate, J., Laurence, D., Iyamabo, B., Mole, A., *et al.* Coupled hybrid RANS-LES research at the university of manchester. *ERCOFTAC Bulletin* **120**, 67 (2020).

144. Ali, A. E., Afgan, I., Laurence, D. & Revell, A. A dual-mesh hybrid Reynolds-averaged Navier-Stokes/Large eddy simulation study of the buoyant flow between coaxial cylinders. *Nuclear Engineering and Design* **393**, 111789 (2022).

145. Davidson, L. Non-zonal detached eddy simulation coupled with a steady RANS solver in the wall region. *International Journal of Heat and Fluid Flow* **92**, 108880 (2021).

146. Jones, W. P. & Launder, B. E. The prediction of laminarization with a two-equation model of turbulence. *International journal of heat and mass transfer* **15**, 301 (1972).

147. Piroozmand, P., Brenner, O. & Jenny, P. *Dimensionality Reduction for Regularization of Sparse Data-Driven RANS Models* submitted to JCP. 2022.

148. Šarić, S., Jakirlić, S., Breuer, M., Jaffrézic, B., Deng, G., Chikhaoui, O., Fröhlich, J., Von Terzi, D., Manhart, M. & Peller, N. *Evaluation of detached eddy simulations for predicting the flow over periodic hills* in *ESAIM: proceedings* **16** (2007), 133.

PUBLICATIONS

Articles in peer-reviewed journals:

1. **Piroozmand, Pasha** & Boroushaki, M. A computational method for optimal design of the multi-tower heliostat field considering heliostats interactions. *Energy* **106**, 240 (2016).
2. **Piroozmand, Pasha**, Mussetti, G., Allegrini, J., Mohammadi, M. H., Akrami, E. & Carmeliet, J. Coupled CFD framework with mesoscale urban climate model: Application to microscale urban flows with weak synoptic forcing. *Journal of Wind Engineering and Industrial Aerodynamics* **197**, 104059 (2020).
3. Brenner, O., **Piroozmand, Pasha** & Jenny, P. Efficient assimilation of sparse data into RANS-based turbulent flow simulations using a discrete adjoint method. *Journal of Computational Physics*, 111667 (2022).

Conference contributions:

4. **Piroozmand, Pasha**, Brenner, O. & Jenny, P. *Piecewise Linear Dimension Reduction as a Regularization Strategy in Data Assimilation for RANS Simulations* in *APS Division of Fluid Dynamics Meeting Abstracts* (2021), Q24.
5. **Piroozmand, Pasha** & Jenny, P. *A data-driven hybrid LES/RANS framework for wall-bounded turbulent flows* in *APS Division of Fluid Dynamics Meeting Abstracts* (2020), R07.
6. Brenner, O., **Piroozmand, Pasha** & Jenny, P. *Variational Data Assimilation for Incompressible RANS Closure Models* in *APS Division of Fluid Dynamics Meeting Abstracts* (2020), R08.
7. Brenner, O., **Piroozmand, Pasha** & Jenny, P. *Variational Data Assimilation for Stationary Euler-Euler Spray Simulation* in *APS Division of Fluid Dynamics Meeting Abstracts* (2021), T01.

8. **Piroozmand, Pasha**, Mussetti, G., Allegrini, J. & Carmeliet, J. *Heat Island study with coupled regional mesoscale (COSMO), Building effect parameterization (DCEP), and CFD microscale (OpenFOAM) models* in *The 7th International Symposium on Computational Wind Engineering* (2018).