

Enhancement of data-driven turbulence models for wind turbine wake applications

Master of Science Thesis
M.W. Goderie

Enhancement of data-driven turbulence models for wind turbine wake applications

Master of Science Thesis

by

M.W. Goderie

to obtain the degree of Master of Science in Aerospace Engineering at the Delft University of Technology and
Master of Science in Engineering (European Wind Energy) at Technical University of Denmark,
to be defended publicly on Thursday August 20, 2020 at 2:00 PM.

Student number: 4363337
Project duration: November, 2019 – July, 2020
Supervisors: Dr. R.P. Dwight, TU Delft
Dr. A. Vire, TU Delft
Dr. M.P. van der Laan, DTU
Ir. J. Steiner, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Wind turbine wakes cause significant reductions in power production and increased fatigue damage for downwind turbines. Thus, they affect the wind levelized cost of energy. Computational Fluid Dynamics (CFD) can be used to quantify the wake characteristics, whereby Reynolds-averaged Navier-Stokes (RANS) has the most potential for industrial applications due to the relatively low computational costs. However, RANS models all turbulence scales, usually done by the linear $k-\epsilon$ turbulence model, which has significant shortcomings in accurately representing the turbulence characteristics in wind turbine wake applications. This results in an underprediction of the wake deficit. Key reasons for these shortcomings are that the eddy viscosity assumption is not valid in the near wake and that the anisotropic Reynolds stresses are not properly modeled. Also, the direct effects of the turbine forcing is not incorporated in the transport equations.

To address for these shortcomings, machine learning can be used to enhance the turbulence model with data-driven corrections. Recent developments showed for fundamental 2D flow cases that a novel algorithm referred to as SpaRTA (Sparse Regression of Turbulent Stress Anisotropy) can be used to discover sparse algebraic turbulence model corrections. These corrections could lead to improved mean-flow fields when trained on high-fidelity data. Disadvantages of SpaRTA are however that it can only cope with a limited input feature set and that the models have difficulty generalizing towards multiple flow regions simultaneously (e.g. free-stream and wake region).

To help resolve these disadvantages, mutual information, which is a measure from information theory that quantifies the general dependency between variables, is used to a priori measure the importance of a large number of features to the turbulence model corrections. As a result, the most important features can be used for correction model construction. In addition to this, to improve the model predictions in the turbine's wake, only the data samples located in the wake regions are used for training, discarding the free-stream data. Given that these data are discarded, it cannot be guaranteed that the correction models fit the trends in the free-stream. The correction models must therefore be neutralized by a newly constructed sparse algebraic logistic regression model, which distinguishes the wake from the free-stream region. The data used in this research consists of three time-averaged LES (Large Eddy Simulation) cases with multiple turbines on wind tunnel scale, under neutral conditions.

This thesis shows that mutual information can detect most of the essential features, which leads to a good match between the model predictions and the corrections derived from high-fidelity data. Discarding the free-stream samples during model training leads to a further reduction in error in the wake region, both in mean-squared as maximum-squared error of the correction terms. By implementing the constructed algebraic models into CFD, significant improvements in mean-flow fields are obtained compared to the linear $k-\epsilon$ turbulence model. Nevertheless, there remains room for improvement as well as further research. Although the mean-flow fields match the high-fidelity data in the near wake closely, a discrepancy remains in the far wake.

Preface

This thesis is the final deliverable to finish my master and obtain the degree in Aerospace Engineering at the Delft University of Technology (TU Delft) and the degree in European Wind Energy at the Technical University of Denmark (DTU) as part of the European Wind Energy Master (EWEM). It has been a tremendous learning experience to conduct research myself, which has definitely not always been easy.

There are a number of people I would like to thank for either their contribution to this work or their personal support. My supervisor Dr. Richard Dwight for giving me the freedom I received throughout the project and for the always critical feedback. My supervisor Julia Steiner for all the support and time, thank you for always being available for input. My DTU supervisor Paul van der Laan for the very informative discussions, thank you for sharing your extensive knowledge about CFD and wind energy.

The last two years have been wonderful, I feel very grateful to be able to have spend two semesters in Denmark where I have met a lot of people. I want to thank my fellow EWEM students for the friendships and all the interesting discussions. I want to thank all my friends in Delft for all the great years. Finally, the ones close to me have been very supportive during the last few years and especially the last months, which I am very grateful for.

*M.W. Goderie
Delft, July 2020*

Nomenclature

Acronyms

k NN	k -Nearest Neighbor
ABL	Atmospheric Boundary Layer
ANN	Artificial Neural Network
CFD	Computational Fluid Dynamics
FN	False Negative
FP	False Positive
GEP	Gene Expression Programming
GP	Gaussian Process
KDE	Kernel Density Estimation
LES	Large Eddy Simulation
LEVM	Linear Eddy Viscosity Model
MGEP	Multidimensional Gene Expression Programming
MI	Mutual Information
NLEVM	Non-Linear Eddy Viscosity Model
NS	Navier-Stokes
QEVN	Quadratic Eddy-Viscosity Model
RANS	Reynolds Averaged Navier Stokes
SOWFA	Simulator for Offshore Wind Farm Applications
SpaRTA	Sparse Regression of Turbulent Stress Anisotropy
TBDT	Tensor Based Decision Tree
TBNN	Tensor-based Neural Network
TBRF	Tensor Based Random Forest
TN	True Negative
TP	True Positive
WALE	Wall-Adapted Local Eddy-Viscosity

Greek Symbols

α	Basis tensor coefficient	–
Ω	Mean rate of rotation	–
τ	Reynolds stress	m^2/s^2

δ	Dirac delta	–
η	Kolmogorov length scale	m
γ	Eigenvalues anisotropy correction	–
κ	von Karman constant	–
λ	Invariant	–
λ	Regularization weight	–
λ	Reynolds stress anisotropy eigenvalue	–
\mathcal{B}	Feature set	–
\mathcal{M}	Model	–
ν	Kinematic viscosity	m ³ /s
ν_t	Eddy viscosity	m ³ /s
Ω	Rotational velocity	rad/s
ω	Specific turbulence dissipation rate	1/s
ϕ	Reynolds stress eigenvalue	m ² /s ²
Π	Second invariant	–
ψ	digamma function	–
ρ	Density	kg/m ³
ρ	Regularization mixing parameter	–
σ	Shear parameter	–
σ	Sigmoid function	–
σ_k	Schmidt number for k	–
σ_u	Stream-wise velocity standard deviation	m/s
σ_ε	Schmidt number for ε	–
τ_{ij}	Reynolds stress	m ² /s ²
τ_{ij}	Subgrid-scale stress	m ² /s ²
$\tilde{\sigma}$	Shear parameter in calibration flow	–
ν	Kolmogorov velocity scale	m/s
ε	Dissipation rate	m ² /s ³
ε	Relative error	–
ε_{rel}	Relative error	–
Roman Symbols		
\mathbf{b}	Anisotropic Reynolds stress	–
\mathbf{b}^Δ	Anisotropic Reynolds stress correction	–
\mathbf{w}	Node weights	–

\boldsymbol{w}	Regression coefficients	–
\boldsymbol{x}	Library of candidate functions	–
$\overline{u'_i u'_j}$	Reynolds stress (turbulent momentum flux)	m^2/s^2
\mathbf{S}	Mean rate of strain	–
\mathbf{T}	Basis tensor	–
A	Additional invariant	–
a	Induction factor	–
b	Node bias	–
C	Inverse regularization strength	–
c	constant	–
C_μ	Eddy viscosity coefficient	–
C_P	Power coefficient	–
C_R	Rotta constant	–
C_T	Thrust coefficient	–
C_μ	Eddy viscosity coefficient	–
$C_{\varepsilon,1}$	Model coefficient	–
$C_{\varepsilon,2}$	Model coefficient	–
C_R	Rotta constant	–
D	Rotor diameter	m
d	Wall distance	m
H	Entropy	nats
h	Bandwidth	–
h	Differential entropy	nats
I	Turbulence Intensity	–
II	Second invariant	–
III	Third invariant	–
K	Kernel	–
K	Number of cross-validation folds	–
k	Number of nearest neighbors	–
k	Turbulent kinetic energy	m^2/s^2
L	Characteristic length scale	m
MI	Mutual Information	–
N	Number of samples	–
P	Mean pressure	N/m^2

p	Classifier prediction	–
p	Marginal probability density	–
p	Pressure	N/m ²
$p(x, y)$	Joint probability density	–
q	Physical flow feature	–
R	Transport equations residual	m ² /s ³
r	covariance	–
R^2	Coefficient of determination	–
Re	Reynolds number	–
t	Time	s
U	Characteristic velocity scale	m/s
U	Velocity	m/s
u'	Fluctuating velocity	m/s
U_d	Velocity deficit	m/s
U_∞	Free-stream velocity	m/s
V_{res}	Relative velocity	m/s

List of Figures

1.1	Master thesis lay-out.	4
2.1	Velocity triangle and force orientation on wind turbine blade cross-section, from Viré and Zaaijer [88].	6
2.2	Schematic overview showing instantaneous and time-averaged flow field around an operating wind turbine, from Porté-Agel et al. [60].	7
2.3	Power deficit in Wieringermeer wind farm for wind direction $275^\circ \pm 3^\circ$, from van der Laan et al. [82]. Comparison of different turbulence models to measurements.	7
2.4	Normalized wind speed for different distances from rotor computed by standard $k-\varepsilon$, $k-\varepsilon-f_p$ and LES compared to measurements, from van der Laan et al. [83].	9
2.5	Non-linear invariant maps. The limiting turbulence states are included where the grey line indicates the plain-strain state.	14
2.6	Linear invariant maps. The limiting turbulence states are included where the grey line indicates the plain-strain state.	15
2.7	Barycentric maps for different CFD data. Left: Channel flow data for a one-dimension vertical profile where the data points are colored according to the wall distance y_+ . Right: Data points of the entire domain from a nozzle flow. Taken from Emory and Iaccarino [21].	15
2.8	Barycentric map including turbulence anisotropy componentality contours.	16
2.9	Turbulence states (stress type) depicted by RBG color system. Data is from LES of a 2D period hill at $Re = 10,595$. Taken from Kaandrop [38].	16
3.1	Neural network with three input units, a single hidden layer containing four hidden units and two output units, from Herlau et al. [33].	18
3.2	Flow chart of GEP algorithm, from Weatheritt and Sandberg [91].	19
3.3	Classifier predictions by (a) SVM, (b) decision tree and (c) random forest on an inclined jet in crossflow. Taken from Ling and Templeton [48].	22
4.1	Mean stream-wise velocity and turbulent kinetic energy at hub-height normalized with free-stream values as function of normalized span-wise position $(y - y_{\text{hub}})/D$. Location in x denoted in title.	24
4.2	Relative error $\varepsilon = MI_{\text{true}}(X; Y) - MI_{\text{KDE}}(X; Y) /MI_{\text{true}}(X; Y)$ for sample size N . Vertical lines indicate standard deviation in computed ε	27
4.3	Nearest data points around point x_i . Maximum norm distance to first neighbor equals $\varepsilon(i)/2$. Number of data points between the two vertical dashed lines equals n_x while number of points between the two horizontal dashed lines equals n_y . From Kraskov et al. [44].	28
4.4	Relative error $\varepsilon = MI_{\text{true}}(X; Y) - MI_{\text{kNN}}(X; Y) /MI_{\text{true}}(X; Y)$ for sample size N . Vertical lines indicate standard deviation in computed ε	29
4.5	Relative error between mutual information between first invariant λ_1 and correction term R for increasing number of samples used in the estimator. Exact mutual information is computed by using the entire data set.	29
4.6	Mutual information between first invariant λ_1 and correction term R for increasing number of nearest neighbors k used in the estimator.	30
4.7	Four functional relations between variables X and Y , the functional relation is denoted in the plot title and the estimated mutual information is depicted in the plot label.	30
4.8	Mutual information between variables X and Y through the relation: $Y = f(X) + \eta$ for increasing noise level η (on the horizontal axis).	31
4.9	Box-plots of $(\frac{\partial U}{\partial y} + \frac{\partial U}{\partial z})/(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial V}{\partial z} + \frac{\partial W}{\partial x} + \frac{\partial W}{\partial y} + \frac{\partial W}{\partial z})$ for data partitioned in wake region and free-stream region.	32

4.10	3-Fold Cross-Validation based on the three wind turbine LES cases. Red indicating test sets while blue indicates training sets.	34
4.11	Sigmoid function.	35
5.1	Wind turbine positioning for each LES case, turbine diameter is on scale. From Steiner et al. [74].	38
5.2	Contour plot of R (top figure) and $\Pi_{\mathbf{b}^\Delta}$ (bottom figure) at turbine location $((y - y_{hub})/D = 0)$ for case A. Turbines indicated by black vertical lines.	39
6.1	Mutual information between invariants and R	43
6.2	Mutual information between invariants and $\Pi_{\mathbf{b}^\Delta}$	43
6.3	Mutual information between individual invariants	44
6.4	Linear fit between invariant $(\lambda_1, -\lambda_2)$ for data partitioned in wake and free-stream, left: wake, right: free-stream. Samples are indicated in blue.	45
6.5	Contours of λ_{1+2} for vertical slice at turbine location.	45
6.6	Mutual information between individual invariants in wake region.	46
6.7	Mutual information between individual invariants conditioned on $\lambda_{1+2} > 75$	46
6.8	Mutual information between invariants and R	46
6.9	Mutual information between invariants and $\Pi_{\mathbf{b}^\Delta}$	46
6.10	Correction terms R and $\Pi_{\mathbf{b}^\Delta}$ as function of invariant λ_1 . Data is partitioned in wake and free-stream region.	46
6.11	Mutual information heat map for basis tensor components with \mathbf{b}^Δ components.	47
6.12	Mutual information between R and double inner dot product basis tensor and velocity gradients for different regions of the domain.	48
6.13	Heat map of mutual dependency between double inner product between basis tensors and velocity gradient for wake samples. Measured by mutual information.	48
6.14	Mutual information between invariants and R for three best scoring invariants when considering all samples.	49
6.15	Mutual information between invariants and R for three best scoring invariants when considering wake samples.	49
6.16	Mutual information between invariants and $\Pi_{\mathbf{b}^\Delta}$ for three best scoring invariants when considering all samples.	49
6.17	Mutual information between invariants and $\Pi_{\mathbf{b}^\Delta}$ for three best scoring invariants when considering wake samples.	49
6.18	Mutual information between individual invariants for wake samples. Invariants are the traces of the listed expressions.	49
6.19	Mutual information between physical flow features and R for all samples, wake samples and free-stream samples. Features are ordered by mutual information in wake region.	50
6.20	Mutual information between physical flow features and $\Pi_{\mathbf{b}^\Delta}$ for all samples, wake samples and free-stream samples. Features are ordered by mutual information in wake region.	50
6.21	Mutual information between selected physical flow features for wake region.	51
6.22	Correction R as function of force coefficient q_6 . Blue indicates wake samples where orange is wake samples including specific condition in legend.	51
6.23	Correction term R as function of A_k^2 for two flow regions (wake and near-wake).	52
7.1	Mutual information between physical flow features and wake classes.	56
7.2	Occurrence of candidate functions in model discovery phase expressed in percentages.	57
7.3	Log loss (top), accuracy (middle) and recall (bottom) as function of complexity n_c for all models analysed in the inference phase. Final chosen model high-lighted with green star.	58
7.4	Wake classifier predictions compared to true label for case A on $x - z$ plane at turbine location $(y - y_{hub})/D = 0$. Upper figure: classifier predictions, lower figure: true label. Turbines indicated in black.	59
7.5	Wake classifier discrete predictions compared to true discrete labels for case A. Upper figure: $x - z$ plane at $(y - y_{hub})/D = 0$, lower figure: $x - y$ plane at $(z - z_{hub})/D = 0$. Turbines indicated in black.	59
7.6	Comparison of classifier predictions to contours of R and $\Pi_{\mathbf{b}^\Delta}$ for case A at different slices in the $y - z$ plane, x -location depicted in title upper row. Upper row: classifier predictions, middle row: R and lower row: $\Pi_{\mathbf{b}^\Delta}$	60

8.1	Mean-squared error in R as function of model complexity n_C for three different feature sets described in Table 8.2.	63
8.2	Mean-squared error R as function of model complexity n_C for feature sets described in Table 8.3.	64
8.3	Mean-squared error R as function of model complexity n_C for feature sets described in Table 8.4.	65
8.4	Mean-squared error in R as function of model complexity for four different feature sets listed in Table 8.5.	66
8.5	Mean-squared error R as function of model complexity n_C for feature sets described in Table 8.6.	67
8.6	Mean squared error as function of model complexity for models trained on predicted wake samples and models trained on all samples.	67
8.7	Maximum squared error as function of model complexity for models trained on predicted wake samples and models trained on all samples.	67
8.8	Predictions in R (no C) and predictions in R multiplied with classification model (C) compared to true values as function of domain height in the free-stream. Location in y : $(y - y_{\text{hub}})/D = -3$, location in x : $(x - x_{T2})/D = 10$	68
8.9	Mean squared error R as function of model complexity. Model for further inspection indicated by orange star.	68
8.10	Maximum squared error R as function of model complexity. Model for further inspection indicated by orange star.	68
8.11	Case A. True (blue) and prediction (orange) of R as function of domain width (top row for $(z - z_{\text{hub}})/D = 0$) and height (bottom row for $(y - y_{\text{hub}})/D = 0$). Relative locations in x with respect to turbines are denoted in title.	69
8.12	True (top row) and prediction (bottom row) of R for $y - z$ planes. Relative locations in x with respect to turbines are denoted in title.	69
9.1	Mean-squared error of components \mathbf{b}^Δ as function of model complexity n_C for feature sets described in Table 9.2.	72
9.2	Component $b_{1,2}^\Delta$ as function of $T_{1,2}^{(1)}$	72
9.3	Component $b_{1,2}^\Delta$ as function of $T_{1,2}^{(6)}$	72
9.4	Mean-squared error of \mathbf{b}^Δ components as function of model complexity for feature sets described in Table 9.3.	73
9.5	Mean squared error in \mathbf{b}^Δ as function of model complexity.	74
9.6	Maximum squared error in \mathbf{b}^Δ as function of model complexity.	74
9.7	Mean-squared error \mathbf{b}^Δ as function of complexity (number of non-zero components) for models trained on all samples and models trained on wake samples.	74
9.8	Maximum-squared error in \mathbf{b}^Δ as function of complexity (number of non-zero components) for models trained on all samples and models trained on wake samples.	74
9.9	Mean-squared error \mathcal{P}_k^Δ for models trained on all samples and models trained on predicted wake samples.	75
9.10	Maximum-squared error \mathcal{P}_k^Δ for models trained on all samples and models trained on predicted wake samples.	75
9.11	Mean squared error \mathbf{b}^Δ as function of complexity.	75
9.12	Max squared error \mathbf{b}^Δ as function of complexity.	75
9.13	Case A. True (blue) and prediction (orange) of \mathcal{P}_k^Δ as function of height z (top row for $(y - y_{\text{hub}})/D = 0$) and y (bottom row for $(z - z_{\text{hub}})/D = 0$). Relative locations in x with respect to most upstream turbine are denoted in title.	76
9.14	True (top row) and prediction (bottom row) of \mathcal{P}_k^Δ for $y - z$ plane. Relative locations in x with respect to turbines are denoted in title.	77
9.15	Predictions in \mathcal{P}_k^Δ (no C) and predictions in \mathcal{P}_k^Δ multiplied with classification model (C) compared to true values as function of domain width. Location in y : $(y - y_{\text{hub}})/D = -3$, location in x : $(x - x_{T2})/D = 10$	77
10.1	Lay out case A. Turbine's are depicted in red. The boundaries are denoted by their cardinal direction. Inflow is from west.	79
10.2	Comparison of baseline $k-\varepsilon$ RANS, LES and static correction RANS via normalized streamwise velocity as function of domain width and height for different downstream locations with respect to turbines.	83

10.3	Comparison of baseline $k-\varepsilon$ RANS, LES and static correction RANS via normalized turbulent kinetic energy as function of domain width and height for different downstream locations with respect to turbines.	83
10.4	Comparison of baseline $k-\varepsilon$ RANS, LES and two coupled RANS corrections via normalized streamwise velocity as function of domain width and height for different downstream locations with respect to turbines.	84
10.5	Comparison of baseline $k-\varepsilon$ RANS, LES and two coupled RANS corrections via normalized turbulent kinetic energy as function of domain width and height for different downstream locations with respect to turbines.	85
10.6	Comparison of true, fixed predictions (subjected to LES variables) and RANS predictions of R as function of domain width and height for different downstream locations with respect to turbines.	86
10.7	Comparison of true, fixed predictions (subjected to LES variables) and RANS predictions of \mathcal{P}_k^Δ as function of domain width and height for different downstream locations with respect to turbines.	86
10.8	Residuals as function of iteration for simulation with classification model (C) and simulation without classification model (no C).	87
10.9	Correction \mathcal{P}_k^Δ in free-stream ($(x - x_{T2})/D = 10$ and $(y - y_{hub})/D = -3$) with classification (C) and without classification (no C) compared to true values.	87
10.10	Turbulent kinetic energy in free-stream ($(x - x_{T2})/D = 10$ and $(y - y_{hub})/D = -3$) with classification (C) and without classification (no C) compared to baseline RANS and LES.	88
10.11	Turbulent kinetic energy at horizontal slice of the domain ($(x - x_{T2})/D = 10$ and $(z - z_{hub})/D = 0$) with classification (C) and without classification (no C) compared to baseline RANS and LES.	88
10.12	Comparison of normalized streamwise velocity of LES, baseline $k-\varepsilon$ model and coupled correction at hub-height.	89
10.13	Comparison of normalized turbulent kinetic energy of LES, baseline $k-\varepsilon$ model and coupled correction at hub-height.	90
10.14	Turbulence state depicted by RGB color system at hub height. Top: LES, middle: baseline $k-\varepsilon$ turbulence model, bottom: $\mathcal{M}^{(2)}$	90
B.1	Two discrete data sets depicted in orange and blue.	106
B.2	Mutual information heat map for basis tensor components with \mathbf{b}^Δ components for entire domain.	106
B.3	Mutual information heat map for basis tensor components with \mathbf{b}^Δ components for free-stream region.	106
C.1	Comparison of classifier predictions to contours of R and $\Pi_{\mathbf{b}^\Delta}$ for case A at different slices in the $y-z$ plane, x -location depicted in title upper row. Upper row: classifier predictions, middle row: R and lower row: $\Pi_{\mathbf{b}^\Delta}$	107
C.2	Case B: log loss (top), accuracy (middle) and recall (bottom) as function of complexity n_c for all models analysed in the inference phase. Final chosen model high-lighted by green star.	108
C.3	Comparison of classifier predictions to contours of R and $\Pi_{\mathbf{b}^\Delta}$ for case B at different slices in the $y-z$ plane, x -location depicted in title upper row. Upper row: classifier predictions, middle row: R and lower row: $\Pi_{\mathbf{b}^\Delta}$	109
C.4	Wake classifier predictions compared to true label for case B on $x-z$ plane at turbine location $(y - y_{hub})/D = 0$. Upper figure: classifier predictions, lower figure: true label.	109
C.5	Wake classifier discrete predictions compared to true discrete labels for case B. Upper figure: $x-z$ plane at $(y - y_{hub})/D = 0$, lower figure: $x-y$ plane at $(z - z_{hub})/D = 0$	110
C.6	Case C: log loss (top), accuracy (middle) and recall (bottom) as function of complexity n_c for all models analysed in the inference phase. Final chosen model high-lighted with green star.	110
C.7	Comparison of classifier predictions to contours of R and $\Pi_{\mathbf{b}^\Delta}$ for case C at different slices in the $y-z$ plane, x -location depicted in title upper row. Upper row: classifier predictions, middle row: R and lower row: $\Pi_{\mathbf{b}^\Delta}$	111
C.8	Wake classifier predictions compared to true label for case C on $x-z$ plane at turbine location $(y - y_{hub})/D = 1.25$. Upper figure: classifier predictions, lower figure: true label.	111
C.9	Wake classifier discrete predictions compared to true discrete labels for case C. Upper figure: $x-z$ plane at $(y - y_{hub})/D = 1.25$, lower figure: $x-y$ plane at $(z - z_{hub})/D = 0$	112

D.1	Comparison of baseline k - ϵ RANS, LES and three coupled RANS corrections via normalized stream-wise velocity as function of domain width and height for different downstream locations with respect to turbines.	113
D.2	Comparison of baseline k - ϵ RANS, LES and three coupled RANS corrections via normalized turbulent kinetic energy as function of domain width and height for different downstream locations with respect to turbines.	114
D.3	Comparison of baseline k - ϵ RANS, LES and three coupled RANS corrections via normalized turbulent kinetic energy as function of domain width and height for different downstream locations with respect to turbines.	115

List of Tables

5.1	LES case [74] parameters	37
5.2	Invariant bases, number of symmetric and antisymmetric tensors for each invariant are indicated by n_s and n_A , respectively. The invariant bases are the trace of the tensors listed. The asterisk on an invariant bases indicates that also the cyclic permutation of the antisymmetric tensors are included.	40
5.3	Physical flow features. For each feature q_i the physical description is denoted including the raw feature with its <u>normalization</u> . The features that are not Galilean invariant are marked with †. Reynolds stress $\overline{u'_i u'_j}$ is based only on Boussinesq eddy viscosity assumption.	41
6.1	Features used for further investigation with respect to the correction terms R and \mathbf{b}^Δ	53
7.1	Performance classifier when tested on specific cases	58
8.1	Features used for modeling correction term R	61
8.2	Features used for modeling correction term $R_{\mathcal{P}}$	63
8.3	Features used for modeling correction term $R_{\mathcal{P}}$	64
8.4	Features used for modeling correction term $R_{\mathcal{P}}$	65
8.5	Features used for modeling correction term $R_{\mathcal{P}}$	66
8.6	Features used for modeling correction term R	66
9.1	Features used for modeling correction term \mathbf{b}^Δ	71
9.2	Features used for modeling correction term \mathbf{b}^Δ	72
9.3	Features used for modeling correction term \mathbf{b}^Δ	73
9.4	Features used for modeling correction term \mathbf{b}^Δ	74
10.1	Boundary conditions RANS solver.	80
10.2	Numerical schemes RANS solver.	80
10.3	Performance coupled algebraic models on case A using normalized error for velocity and turbulent kinetic energy.	84
10.4	Performance coupled algebraic models on case A using normalized error for velocity and turbulent kinetic energy.	87

Contents

List of Figures	xi
List of Tables	xvii
1 Introduction	1
1.1 Background	1
1.2 Research Objective and Questions	2
1.3 Thesis Structure.	3
2 Wind Turbines and Turbulence	5
2.1 Wind Turbines and wakes	5
2.1.1 Wake Modeling	8
2.2 Turbulence	9
2.2.1 Large Eddy Simulation	10
2.2.2 Reynolds-Averaged Navier-Stokes	10
2.2.3 k - ϵ turbulence model	11
2.2.4 Non-linear eddy viscosity models	12
2.3 Visualization of Reynolds stress anisotropy	13
3 Machine Learning in Turbulence Modeling	17
3.1 Machine learning in turbulence modeling	17
3.2 Feature Selection in Machine Learning	20
3.3 Predictions of regions of high uncertainty.	22
3.4 Machine learning in wind energy	22
4 Methodology	23
4.1 Model-form error k - ϵ turbulence model	23
4.2 Mutual Information.	24
4.2.1 Mutual Information Estimation	25
4.2.2 Parameter dependency	29
4.2.3 Measure of General Dependence	29
4.2.4 Physics-informed feature selection	30
4.3 Sparse Regression of Turbulent Stress Anisotropy	31
4.3.1 Physical form \mathbf{b}^Δ and R	31
4.3.2 Constructing library of candidate functions	33
4.3.3 Model selection using elastic net regression	33
4.3.4 Model inference	34
4.4 Sparse logistic regression	34
4.4.1 Performance Evaluation	36
5 Data Cases and Features	37
5.1 Data Cases	37
5.2 Features.	38
5.2.1 Feature requirements	39
5.2.2 Invariants	39
5.2.3 Basis Tensors.	40
5.2.4 Additional Invariants.	40
5.2.5 Physical Flow Features	40

6	Feature Selection	43
6.1	Invariants	43
6.2	Basis tensors	47
6.3	Additional invariants	48
6.4	Physical flow features	49
6.5	Domain specific feature importance	51
6.6	Conclusion	52
7	Wake Classification	55
7.1	Wake Dependency	56
7.2	Model Discovery	56
7.3	Model Inference using cross-validation	57
7.4	Performance visualization	59
7.5	Conclusion	60
8	Transport equation deficit R	61
8.1	Features.	61
8.2	Classification and regression coupling	62
8.3	Sparse regression	62
8.4	Results	63
8.4.1	Basis tensors	63
8.4.2	Additional invariants.	64
8.4.3	Physical flow features	64
8.4.4	Source/sink term	66
8.5	Comparison training on predicted wake/all samples	67
8.6	Cross-validation	68
8.7	Conclusion	70
9	Anisotropy correction b^Δ	71
9.1	Features.	71
9.2	Results	71
9.2.1	Basis tensors	72
9.2.2	Additional invariants.	73
9.2.3	Physical flow features	73
9.3	Comparison training on predicted wake/all samples	73
9.4	Cross validation.	75
9.5	Conclusion	76
10	CFD Implementation	79
10.1	CFD setup.	79
10.2	Algebraic models	81
10.3	Static correction	82
10.4	Coupled correction	82
10.4.1	Performance	82
10.4.2	Turbulent stress anisotropy	89
10.5	Conclusion	90
11	Conclusion and Recommendations	93
11.1	Conclusions.	93
11.2	Recommendations	95
	Bibliography	97
A	SpaRTA	103
B	Mutual Information	105
B.1	Theoretical Example	105
B.2	Tensor basis.	105

C	Wake Classification	107
C.1	Case A.	107
C.2	Case B.	108
C.3	Case C.	108
D	CFD implementations	113
D.1	Coupled corrections	113
D.2	Physical-form R	113



Introduction

1.1. Background

The interaction between wind turbines in a wind farm appears crucial for the overall output of the farm. Overall, this interaction can negatively affect the annual energy output to up to 20% [79]. Energy loss occurs due to reduced incoming wind speed when turbines operate in the wake of another turbine. On top of that, increased turbulence in the turbine wake results in increased fatigue damage, which eventually leads to a decreased turbine lifetime. Hence, to realize the optimal performance of wind farms, it is crucial to estimate the wake effects as accurately as possible. To do so, Computational Fluid Dynamics (CFD) appears an important approach. It can be applied to obtain insights on the physical characteristics of wind turbine wakes [8]. It is superior to analytical methods often used in industry because it relies on fewer assumptions and can additionally give information about wind turbine loads.

CFD can be divided into three disciplines listed in order of fidelity: Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) and Reynolds-Averaged Navier-Stokes (RANS). DNS resolves all turbulent length scales and is currently computationally infeasible for wind turbine applications due to the large range of turbulent scales. LES resolves only the largest, most influential turbulent scales and is therefore a reasonable trade-off between computational requirements and accuracy compared to DNS. However, the computational costs of LES remain today large and this limits its widespread use for wind farm applications. Finally, RANS models all turbulent scales, making it a steady solver which also allows coarse grids, and is therefore roughly two to three orders of magnitude cheaper computationally than LES [65]. This makes RANS a more attractive option for industrial applications. The RANS equations are obtained by applying Reynolds decomposition to the conventional Navier-Stokes equations, resulting in an unclosed system of equations. The system is closed by introducing a turbulence model, specifically the k - ϵ turbulence model for wind turbine wake applications. The k - ϵ model is a linear eddy viscosity model. It models the deviatoric Reynolds stress, the difference between the Reynolds stress $\overline{u'_i u'_j}$ and its isotropic part $\frac{2}{3}k\delta_{ij}$, by a mean strain-rate tensor and an eddy-viscosity ν_t .

Research has shown [64] that numerous k - ϵ model assumptions are breached in wind turbine wake applications, which result in wake deficits that are significantly underestimated and recover much faster than in measurements and high-fidelity LES. Extensions to the standard k - ϵ turbulence model have been proposed [80] that either limit the eddy viscosity in the wake or add source terms to the transport equations [20]. Although the predictions of the wake deficit are improved, the turbulent kinetic energy and anisotropic Reynolds stress remain inaccurate.

This thesis aims to contribute to the research on how to overcome the shortcomings of the standard turbulence models. A means is to implement additional data-driven terms in the turbulence model representing a turbulence anisotropy correction \mathbf{b}^Δ to the stress-strain relationship and a correction R to the turbulence transport equations [69]. These model-form corrections can be obtained from high-fidelity LES data using the k -corrective-frozen-RANS approach in which the turbulence model equations are solved passively with frozen time-averaged LES variables. Subsequently, algebraic models can be learned for \mathbf{b}^Δ and R using the novel machine learning algorithm SpaRTA (Sparse Regression of Turbulent Stress Anisotropy [69]) and implemented in a RANS CFD solver to improve mean flow predictions. SpaRTA builds a library of non-linear functions, constructed from pre-defined features, which is subjected to sparse regression to select the most

influential functions. In this way, a simple, easy to interpret algebraic model is obtained. The disadvantage of SpaRTA is that, for large data sets, it can only cope with a limited set of features as the number of non-linear functions increases exponentially with the number of features. Also, the amount of non-linearity captured by the model is limited, meaning it is not able to capture as many flow details as more complex machine learning algorithms, such as neural networks [49] or random forests [37].

In trying to add further insights to the current body of literature and knowledge, this work starts with defining the most important features, obtained from literature, to the correction terms using mutual information [17]. Mutual information is a measure from information theory that quantifies the mutual dependence between individual features and the correction terms based on the underlying probability density functions.

Earlier work focuses on constructing data-driven corrections trained on all domain samples, including regions where it is assumed that the baseline turbulence model is sufficient. In this thesis, the correction terms \mathbf{b}^Δ and R are solely trained on samples in regions where the baseline turbulence model is expected to have a high uncertainty, which corresponds to the wake regions in wind turbine applications. This implies that the learning algorithm only has to fit the data in the wake region. This can potentially lead to better predictions in the wake region compared to when it is fitted to all domain samples. Although the predictions inside the turbine wake will potentially improve, the learning algorithm will not fit the data outside the wake because it has not been trained upon it. Therefore, an algebraic classification model will be constructed that turns *on* the predictions inside the wake and *off* outside the wake by extending the SpaRTA algorithm such that it is suitable for classification problems. The model predictions are further improved by proposing an extension to the physical-form of the transport equation correction R . Schmelzer et al. [69] modeled R as a turbulent production term. However, it is not guaranteed that the deficit in the transport equations is a result of either over- or under-predictions of the turbulent production. To determine the outcome of the analysis performed in this thesis, the improvements over the baseline $k-\varepsilon$ turbulence model are assessed by implementing the obtained algebraic corrections into the OpenFOAM CFD solver. The data used in this research consists of three time-averaged LES cases with either two or three turbines on wind-tunnel scale, under neutral conditions.

1.2. Research Objective and Questions

The main research objective of this thesis is:

"To reduce the $k-\varepsilon$ turbulence model uncertainty in wind turbine wake applications by means of algebraic data-driven corrections to the model equations that: 1) are specifically constructed for the wind turbine wake region by coupling to an algebraic classification model and 2) are enhanced using features that have the largest dependency with the correction terms, detected with mutual information."

Based on the main research objective, sub-objectives (SO) can be formed:

SO1: Develop a framework based of mutual information to detect *a priori* the most relevant features for the correction terms of the turbulence model equations.

SO2: Assess the potential improvements in data-driven corrections when training models only on samples located in wind turbine wakes.

SO3: Develop a sparse logistic regression framework to construct algebraic wake classification models that are able to turn the data-driven correction *on* inside the turbine wake and *off* in the free-stream region.

SO4: Expand physical-form of the correction term R .

SO5: Learn algebraic models using SpaRTA for the correction terms \mathbf{b}^Δ and R inside the wake region using the most relevant features and assess performance.

SO6: Implement algebraic models in OpenFOAM and assess improvements in mean-flow conditions over the baseline $k-\varepsilon$ turbulence model.

Based on the thesis objectives, research questions (RQ) and their corresponding sub-questions can be formalized. The answers of the sub-questions automatically lead to the answer of the main research question.

RQ1: "In the context of data-driven turbulence modeling, what are the advantages and disadvantages of using mutual information for feature selection?"

RQ1.1: What method is best suited to estimate mutual information when considering accuracy and computational costs?

RQ1.2: What features are identified as having the largest dependency with the turbulence model correction terms?

RQ1.3: What are the limitations of using mutual information for feature selection?

RQ2: "Can an algebraic wake classification model be constructed such that the algebraic models for b^Δ and R are accurately turned off in regions where they were not trained on?"

RQ2.1: What features, using mutual information, are considered to have the largest dependency with the classification label (wake or free-stream)?

RQ2.2: Using sparse logistic regression to construct algebraic classification models, what performance can be reached in terms of accuracy when also considering model complexity?

RQ3: "By training algebraic models for the correction terms b^Δ and R specifically for the wind turbine wake region using the most relevant features and combining it with the algebraic wake classification model, what improvements in mean-flow fields can be achieved when compared to the standard $k-\varepsilon$ turbulence model?"

RQ3.1: Can the physical-form of the correction term R be extended such that more physics is incorporated and predictions are improved?

RQ3.2: What improvements are made in predicting R and b^Δ when trained only on wake samples compared to when trained on all domain samples?

RQ3.3: When implementing the algebraic models for R and b^Δ in OpenFOAM, what improvements in mean-flow fields are obtained when compared to the standard $k-\varepsilon$ turbulence model?

1.3. Thesis Structure

The thesis is divided into three parts. The first part provides the required background information such as relevant literature and the current state-of-the-art in wind energy and data-driven turbulence modeling. A general introduction to wind energy and turbulence is provided in Chapter 2. Relevant information regarding machine learning in turbulence modeling is subsequently detailed in Chapter 3.

The second part describes the applied methodology, data cases and features used. The methodology is explained in Chapter 4. First, the model-form of the $k-\varepsilon$ turbulence model is briefly described, obtained with the k -corrective-frozen-RANS approach. This approach was already available and executed prior to this research. Second, the concept of mutual information is explained including the used estimator. Third, the sparse regression model SpARTA, used to learn algebraic models for both correction terms b^Δ and R is described. Finally, SpARTA is extended and made suitable for classification problems. The data cases, which are on wind-tunnel scale, and features used in this thesis are discussed in Chapter 5.

The third part consists of the research results obtained by utilizing the described methodology to the given data. The dependency analysis for the different features is discussed in Chapter 6 and the construction of algebraic wake classification models is described in Chapter 7. In using the features with the highest dependency, algebraic corrections for b^Δ and R are learned by applying SpARTA. The results for R and b^Δ are discussed in Chapter 8 and Chapter 9, respectively. The algebraic corrections are implemented in OpenFOAM in Chapter 10. Finally, conclusions are drawn and recommendations for future research are presented in Chapter 11.

The thesis lay-out is schematically visualized in Figure 1.1.

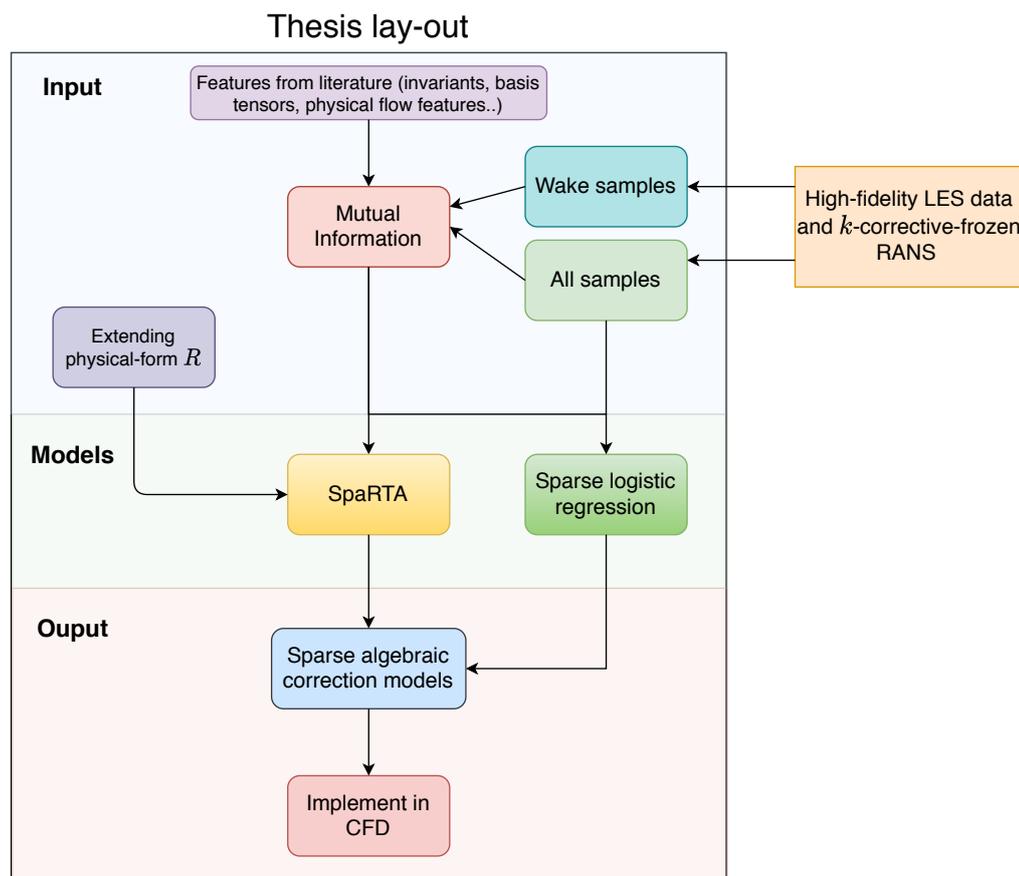


Figure 1.1: Master thesis lay-out.

2

Wind Turbines and Turbulence

Wind energy is seen as one of future's most dominant renewable energy sources [9]. To increase the energy production and lower the costs, wind turbines have grown bigger and bigger and wind farms have been constructed consisting of more than hundred turbines. Not only the physics of the turbine itself is important, also the interaction between different turbines plays a significant role. When turbines are aligned with the direction of the wind, upstream turbines negatively affect the incoming flow conditions of downstream turbines. Due to the momentum extraction by the turbines, downstream turbines experience a lower incoming wind speed, resulting in lower power production. On top of that, operating turbines increase the turbulent kinetic energy in the flow, meaning downstream turbines experience more fluctuating wind conditions which then causes more fatigue damage.

Research has been active for decades to get a better understanding of the physics of turbines and turbines mutual interaction. A comprehensive summary is given to highlight the most important contributions. The focus is towards the applications of Computational Fluid Dynamics, as this research area is most relevant to this thesis.

2.1. Wind Turbines and wakes

As the wind in the atmosphere can be seen as a source of (kinetic) energy, wind turbines have been used for many decades to generate electricity. Betz [7], Joukowsky [36] and Glauert [26] already established in the first half of the twentieth century the basic principles of wind-turbine aerodynamics which are still used today in rotor design optimization.

Two commonly used rotor characteristics are the non-dimensionalised power and thrust, the power and thrust coefficient C_P and C_T , respectively. C_P is proportional to the blade forces tangential to the rotor plane and a measure of the turbine efficiency, which is limited by the Betz limit of $\frac{16}{27}$. C_T is proportional to the forces normal to the rotor plane and defines the change in momentum of the incoming wind.

A more detailed illustration of the force distribution on a wind turbine blade is depicted in Figure 2.1. The rotational motion of the blade causes a velocity component Ωr tangential to the rotating plane. The rotational velocity Ω is in rad/s and r is the local distance to the rotor center. Together with the velocity component normal to the plane $U(1 - a)$, the relative velocity V_{res} is defined. The induction factor a is a measure to what extent the incoming wind speed decreases with respect to the free-stream velocity. The incoming air causes a lift and drag force on the blade characterized by the airfoil shape of the rotor cross-section. These forces decomposed tangentially and normally to the rotor plane then define the C_P and C_T when non-dimensionalized.

To investigate the operations of wind turbines and get a physical understanding of the interaction of the air flow with the turbine, specifically when the incoming conditions are influenced by other turbines, Computational Fluid Dynamics (CFD) is often applied. CFD numerically solves the Navier-Stokes (NS) equations, these equations describe fluid motions and are based on conservation of mass and momentum. The fundamentals of CFD will be elaborated in a later stage.

The presence of a wind turbine in CFD can be modeled in two ways, representing the blades as body forces or discretizing the actual blades [65]. The latter is the most accurate and physically most sound method but is computationally very expensive because it is required to simulate the boundary layer on the blades.

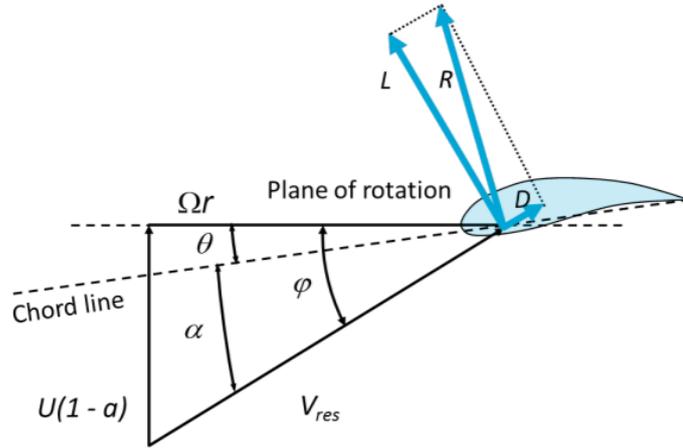


Figure 2.1: Velocity triangle and force orientation on wind turbine blade cross-section, from Viré and Zaaijer [88].

Also compressibility effects have to be included at the blade tips while the rest of the domain is essentially incompressible.

The rotor can also be described using body forces by an actuator disk, line or surface to reduce computational costs. In the case of an actuator disk, a uniform loading determined by only the C_T can be used or a non-uniform loading which also requires lift and drag coefficients as function of radial position. The actuator line model is applicable under unsteady applications in which the line forces are time dependent. It can take into account tip vortices [75] but is computationally more expensive. This can be extended to representing the blade by a surface instead of a line, requiring more accurate airfoil data such as pressure and skin-friction distribution [72]. Actuator surface models are mainly applied to vertical axis wind turbines. For steady RANS, actuator disk is the only suitable method as it is the only method applicable to steady simulations.

The forces exerted by the wind turbine blades affect the airflow both upwind as downwind of the rotor. Due to conservation of mass, the wind speed will slow down in the region before the turbine, referred to as the induction region, as depicted by Figure 2.2. After the wind has passed the rotor, it enters the rotor's *wake* where the flow properties are directly affected by the presence of the rotor. The wake is often subdivided in the near-wake and far-wake [87]. The near-wake is directly influenced by the characteristics of the turbine, such as blade profile, nacelle and hub geometry. Therefore, the near-wake is highly complex and heterogeneous.

In the near-wake, which has a length of around 2-4 rotor diameters, vortices are present from both the blade tip and root, as from the turbine hub. The tip and root vortices are produced by pressure differences between the suction and pressure side of the blades [2] and can reduce flow entrainment because they separate the wake from the free-stream flow [47].

The far-wake is less influenced by the characteristics of the turbine but more by its global parameters such as the power and thrust coefficients and inflow conditions. In the far-wake, the flow properties slowly converge back to free-stream conditions. As most of the wind turbines are not located in the near, but in the far-wake of other turbines, it is essential to understand the far-wake characteristics. The far-wake tends to grow when moving further downstream as it entrains with the outer flow. The streamwise velocity increases gradually until the wake is fully recovered far downstream. With uniform inflow conditions, the streamwise wake velocity profiles have an axisymmetric Gaussian shape. However, due to the incoming shear and the presence of the surface, the streamwise velocity profile loses its Gaussian shape. The velocity deficit, the difference between free-stream and wake velocity, still retains the Gaussian distribution [11, 60].

In the far-wake, the streamwise turbulence intensity is high compared to the incoming flow. Turbulence intensity I is defined as $I = \sigma_u / U$ where σ_u is, in this case, the streamwise velocity standard deviation. Particularly, the upper part of the wake has an increased I while near the ground the turbulence is suppressed by the turbine. The maximum of I occurs around two to four rotor diameters downstream and subsequently converges back to free-stream values [60].

The turbulent momentum fluxes, $\overline{u'_i u'_j}$ (u'_i is the fluctuating velocity in i -direction, overline represents the time-averaged mean) in turbulent wakes reflect the entrainment of fluid from the free-stream flow towards the wake centre. Its magnitude is largest around the wake's edges, especially near the upper edge.

When wind turbines are placed in series, they operate in the wake of the upstream turbines and only the

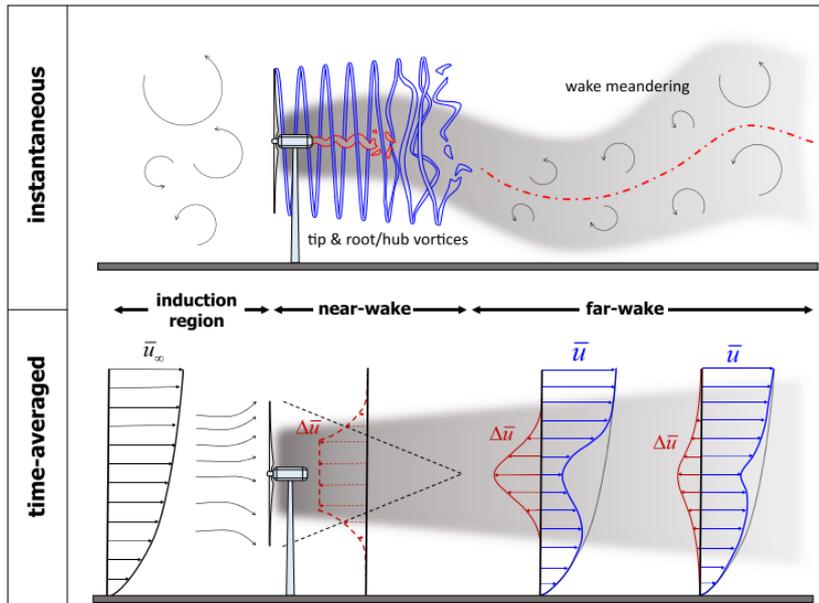


Figure 2.2: Schematic overview showing instantaneous and time-averaged flow field around an operating wind turbine, from Porté-Agel et al. [60].

most upstream wind turbine is unaffected. From measurements of a wind farm with small turbine spacing [82], it was observed for the first wind turbine, because it operates in unaffected flow conditions, extracts a relatively large amount of energy from the air compared to the other turbines. Because of this, the second turbine is barely extracting power and the wake has a longer length to recover. Therefore, the third wind turbine extracts more power than the second turbine. From the third wind turbine on wards, the extracted power is approximately constant. This effect is depicted in Figure 2.3 in which measurement data is compared to CFD simulations. The effects are largest for low I and occur only when the spacing between the turbines is small.

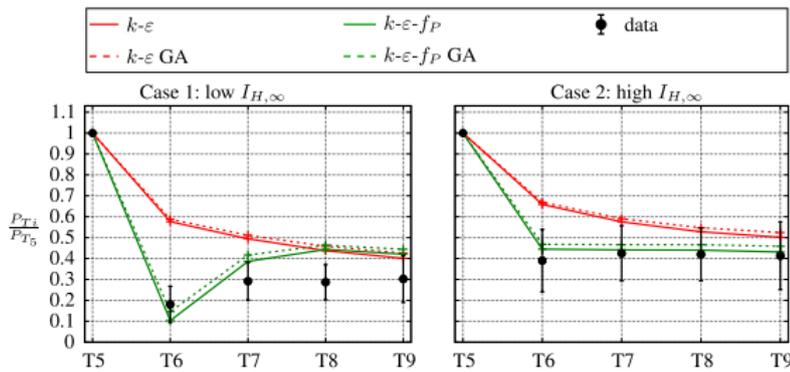


Figure 2.3: Power deficit in Wieringermeer wind farm for wind direction $275^\circ \pm 3^\circ$, from van der Laan et al. [82]. Comparison of different turbulence models to measurements.

Wind turbine wakes were investigated in wind-tunnels by Hancock and Zhang [31] for a large wind turbine in a weakly-convective atmospheric boundary layer (ABL) and stable ABL by Hancock and Pascheke [32]. Convective conditions mean that the vertical potential temperature gradient is negative, while for stable conditions it is positive. They found that the height and width of the wake grow more rapidly in convective conditions than in stable conditions. Vertical heat flux measurements in convective conditions showed first a significant reduction when the flow passed through the rotor, which was followed by an increase first in the wake's lower part and then in the upper part, before it recovered to the undisturbed level at $x/D = 10$. The vertical heat flux in stable stratification behaved differently, where there was only a small effect when the

flow passed through the rotor and it first increased in the upper part to a magnitude significant above the undisturbed value.

Hancock and Farr [30] also performed wind-tunnel simulations of arrays of turbines in both neutral and non-neutral conditions. They came to the same conclusion regarding the wake turbulent heat flux, namely that it is initially small in the near-wake and large in far-wake for stable conditions and opposite for unstable. They proposed that there are two effects of stratification, indirect and direct. Indirect effects are influenced by the turbulence intensity in the ABL and are observed in the earlier part of the wake while direct effects, introduced by buoyancy forces, are seen mainly more downstream.

2.1.1. Wake Modeling

For wind farm lay-out optimization, simple analytical models are deployed as their computational costs are very low, making them suitable for running thousands of cases [60]. Jensen [35] and Frandsen et al. [24] developed such models which show some fundamental insight in the physics of wakes, as their formulation is based on conservation equations combined with empirical observations. These models however have a high uncertainty and low accuracy due to their simplicity.

As mentioned earlier, to increase the reliability and accuracy, one can use CFD [65] by either time averaged or spatial averaged Navier-Stokes equations. The first referred to as Reynolds-averaged Navier Stokes (RANS) and the second Large Eddy Simulations (LES).

In RANS, the statistical properties of the flow are solved, such as the mean velocity, mean pressure gradient, turbulent kinetic energy and so on. Due to the closure problem that arises from Reynolds averaging, turbulence models are required to quantify the Reynolds stresses. The k - ϵ turbulence model is one of the most used turbulence models in wind applications [28]. It is defined by the Boussinesq approximation and is a linear eddy-viscosity model (LEVM). This model however, has some serious shortcomings [80] as it significantly underpredicts the wake velocity deficit and overpredicts the turbulence intensity due to an overprediction of the eddy viscosity. In the region surrounding the rotor, Réthoré [64] observed that the eddy viscosity increases significantly, while it actually should be smaller than the atmospheric values. It was found that the Boussinesq approximation, which relates the Reynolds stresses to the eddy viscosity and the local velocity gradients, is not valid in wake regions where the velocity distribution is highly nonlinear and in regions where turbines apply a pressure gradient to the flow. He also found that neglecting the pressure-velocity correlation in the transport equations introduces significant errors. Due to this assumption, the modeled turbine was not able to extract energy from the turbulence. This results in an overestimation of the wake's turbulence.

These turbulence modeling problems have barely effect in LES, as in there the eddy viscosity is solely used to model small-scale turbulence, which does not effect the results significantly.

Extensions to the standard k - ϵ model have been proposed, such as the k - ϵ - f_p model by van der Laan [80], which introduces a scalar function that limits the eddy viscosity in regions with high velocity gradients. This scalar function is used to have a variable eddy viscosity coefficient C_μ to limit the eddy viscosity and turbulence length scale in regions with large velocity gradients. This scalar function is described as

$$f_p(\sigma/\bar{\sigma}) = \frac{2f_0}{1 + \sqrt{1 + 4f_0(f_0 - 1)(\frac{\sigma}{\bar{\sigma}})^2}}, f_0 = \frac{C_R}{C_R - 1} \quad (2.1)$$

to model non-equilibrium effects. C_R is the Rotta constant, $\frac{\sigma}{\bar{\sigma}}$ is the ratio of the shear parameter ($\frac{k}{\epsilon} \sqrt{(U_{i,j})^2}$) to the shear parameter in the calibration flow ($\frac{k}{\epsilon} |\frac{\partial U}{\partial z}|$), that quantifies how far the local flow deviates from the logarithmic law in a simple shear flow. The prediction of the wake deficit is significantly improved as shown in Figure 2.4, while the turbulence intensity remains inaccurate in the far wake. The k - ϵ - f_p model is compared to LES results and other extended k - ϵ models by Shih et al. [73] and Durbin [19] for wake recovery, turbulence intensity and turbulent time and length scales [80]. For all models, the turbulent time scales are comparable to LES while the turbulent length scales and intensity deviate significantly from LES because all models recover towards the standard k - ϵ turbulence model. It was recommended to use either the k - ϵ - f_p model or the k - ϵ model of Shih as the k - ϵ model of Durbin is very sensitive to the atmospheric turbulence levels. Another extension to the k - ϵ turbulence model was proposed by El Kasmi and Masson [20] in which additional source terms were introduced to the transport equations. These source terms represent transfer of energy from large-scale turbulence to small-scale turbulence. When compared to measurements, improvements were made compared to the standard k - ϵ turbulence model, although the predictions in the far wake remained inaccurate.

All considered extensions are linear eddy viscosity models, as they are based on the Boussinesq approximation. This means that the anisotropic Reynolds stresses remain predicted inaccurately.

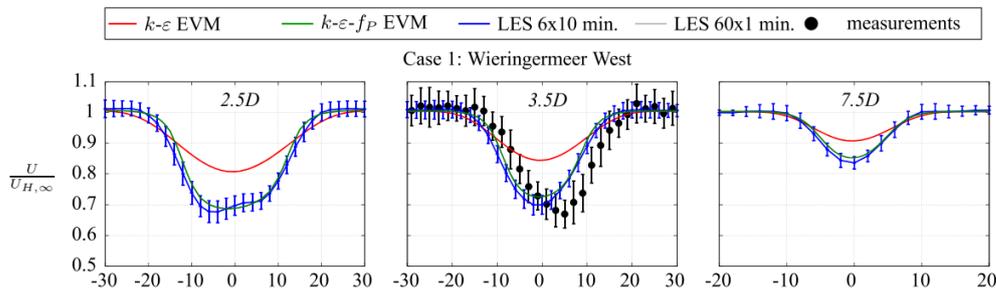


Figure 2.4: Normalized wind speed for different distances from rotor computed by standard $k-\varepsilon$, $k-\varepsilon-f_P$ and LES compared to measurements, from van der Laan et al. [83].

Instead of modeling all turbulent scales in RANS, one could use LES to resolve parts of these scales, and use only a sub-grid model to represent the smallest scales. LES therefore relies on spatial averaging (or filtering) instead of time-averaging, meaning it is by definition unsteady. As advantage, it is able to resolve the dominant anisotropic, unsteady turbulent structures and turbulent mixing but comes with significant increases in computational costs [65]. Chivaee [15] investigated numerous sub-grid stress (SGS) models for wind turbine wakes and found that the flow structures and loadings were not very sensitive to the choice of SGS model. Réthoré [64] compared LES to RANS and found that the LES results are superior as they have a more accurate prediction of mean velocity profiles and turbulence quantities. However, the computational costs increased from hours for RANS to days for LES [65].

Lu and Porté-Agel [50] concluded from their work that LES results can give valuable insights in the high-resolution three-dimensional velocity and temperature fields required for quantitative descriptions of wind turbine wakes and the effects on turbulent heat and momentum fluxes. They found from simulations of a very large wind farm that, due to the presence of the wind farm, the surface boundary layer height was increased and that the surface moment and buoyancy flux decreased with more than 30% and 15%, respectively. Also the wind farm had strong effects on vertical momentum and heat fluxes.

As the turbulent characteristics depend on its length and time scales, it was found that stable conditions require more resolution compared to convective conditions to get reliable results. This is because the characteristic eddies are much smaller in stable flows [6]. Tian et al. [77] compared LES in neutral and convective conditions to experimental results and found good agreement. The results showed that stratification has a significant effect on the wake shape, growth and the recovery. Chaudhari et al. [14] performed a numerical study on the effects of atmospheric stratification on wind turbine performance, they found that for an infinitely long wind farm, the produced power by a wind turbine in stable stratification is reduced by 81% compared to neutral conditions.

Although the effects of stratification on wind turbine wakes have been investigated using LES many times, this is not the case for RANS. Stratified atmospheric boundary layers have been investigated using RANS but did not include wind turbines [1, 8, 40, 41]. Prospathopoulos et al. [61] simulated wake effects of a single turbine with $k-\varepsilon$ and $k-\omega$ turbulence models. Predictions were improved by increasing the turbulent dissipation rate near the turbine, but only the wake deficit and turbulence intensity were investigated in the near-wake. Results were only compared to measurements of a test wind farm, not to LES.

2.2. Turbulence

A flow of a fluid can be regarded as either *laminar* or *turbulent*. Laminar flows are characterized by particles moving in parallel layers without any or very little intermediate mixing. Laminar flows are dominated by momentum diffusion with convection being less influential, and therefore occur at low Reynolds numbers. The Reynolds number is defined by

$$Re = \frac{UL}{\nu} \quad (2.2)$$

and is the ratio between inertial forces and viscous forces in the fluid. Parameters U and L are defined as the characteristic velocity and length scales for the fluid and ν is the kinematic viscosity.

If the Reynolds number exceeds a critical value ($Re \approx 4000$ for pipe-flows [59]), the flow transitions to turbulent. Turbulence is characterized by chaotic changes in fluid velocity and pressure and behaves unsteady. One of the most important characteristics of turbulence is the ability to mix and transport fluid much more effectively than a laminar flow. In the atmospheric boundary layer, turbulence is very effective in mixing momentum and scalars such as temperature.

Turbulent motions extract their energy from the mean flow and are full of unsteady vortices, often referred to as eddies, of different sizes that are characterized by their own velocity and length scales. When turbulence is created, it consists of large eddies produced by shear, rough walls and flow separation. These large eddies are broken down into smaller and smaller eddies (from low to high wavenumbers). After a certain point, the turbulence *forgets* the influence of the initial large-scale motions [55] and the turbulence behaves universal. This range is called the inertial sub-range in which the turbulent kinetic energy is transferred from lower to higher wavenumbers without any significant dissipation. At moderately high wavenumbers the inertial sub-range transitions into the viscous dissipation range, in which the small eddies are dominated by viscous dissipation and its energy is converted into heat.

Kolmogorov [42] hypothesized that the scales of the smallest, dissipative eddies only depend on ν , the kinematic viscosity, and ε , the dissipation rate. The corresponding velocity v and length scale η are defined as

$$v = (\nu\varepsilon)^{1/4} \quad \eta = \left(\frac{\nu^3}{\varepsilon}\right)^{1/4} \quad (2.3)$$

often mentioned as the Kolmogorov scales. In the atmospheric boundary layer, the eddies range from roughly 1000 m to 1 mm [95].

2.2.1. Large Eddy Simulation

To obtain a numerical description of a turbulent fluid, all scales up to the Kolmogorov scales have to be resolved. This direct approach is called Direct Numerical Simulation (DNS), in which the Navier-Stokes equations are solved, numerically. The Navier-Stokes equations are depicted as

$$\frac{\partial u}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \quad (2.4)$$

from which a numerical solution can be obtained by including mass continuity. Numerically solving the Navier-Stokes equations directly is only possible for low Reynolds number flows. For large Reynolds numbers, the dissipative eddies are so small it is impractical to resolve them.

Instead of resolving all turbulent scales with DNS, one could choose to resolve only the largest, most dominant eddies. This method is called Large Eddy Simulation (LES) which space-averages (or spatial filters) (2.4). By splitting up a variable into a resolvable and subgrid-scale part, as in

$$f = f^r + f^s. \quad (2.5)$$

By applying this filtering procedure to (2.4), the spatial-filtered Navier-Stokes equations are obtained

$$\frac{\partial u_i^r}{\partial t} + \frac{\partial}{\partial x_j} \left[u_i^r u_j^r - \tau_{ij} - \nu \left(\frac{\partial u_i^r}{\partial x_j} + \frac{\partial u_j^r}{\partial x_i} \right) \right] = -\frac{1}{\rho} \frac{\partial p^r}{\partial x_i} \quad (2.6)$$

where τ_{ij} defines the subgrid-scale stresses $\tau_{ij} = u_i^r u_j^r - (u_i u_j)^r$.

Because LES resolves the most dominant eddies, it still gives a good representation of the instantaneous turbulent fields. Although the subgrid-scale stress is much smaller than the advection term [95], it has to be included because it is responsible for the extraction of turbulent kinetic energy from the resolved scales in the energy cascade. An expression for the unresolved contribution can be found by, for example, using a subgrid-scale model.

2.2.2. Reynolds-Averaged Navier-Stokes

To reduce the computational costs even further, one could model all the turbulent scales and resolve none. By splitting up the Navier-Stokes equations into a mean and fluctuating part (Reynolds decomposition), and taking the time-average, the Reynolds-Averaged Navier-Stokes (RANS) equations are obtained as

$$\frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j} \left[U_i U_j + \overline{u'_i u'_j} - \nu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right] = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} \quad (2.7)$$

where the mean and fluctuating velocity are depicted as U and u' , respectively.

In RANS, temporal statistics like U_i , mean pressure (P) and covariances ($\overline{u'_i u'_j}$) are computed, instead of instantaneous fields like in DNS or LES. The effect of the turbulent fluctuations are now represented by the Reynolds stress term $\overline{u'_i u'_j}$, this however makes it impossible to solve the equations because there are ten unknowns and only four equations, and therefore requires a model just as in LES. This is referred to as the closure problem [55] and can be solved by introducing turbulence models.

2.2.3. k - ε turbulence model

The most commonly used turbulence model in the atmospheric boundary layer and in wind turbine applications is the k - ε turbulence model [65]. The k - ε model is a linear eddy viscosity model, which models the deviatoric Reynolds stress, the difference between the Reynolds stress $\overline{u'_i u'_j}$ and its isotropic part $\frac{2}{3} k \delta_{ij}$, by a mean strain-rate tensor and an eddy-viscosity ν_t as

$$\overline{u'_i u'_j} - \frac{2}{3} k \delta_{ij} = -\nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right). \quad (2.8)$$

The eddy viscosity ν_t is defined by a constant eddy viscosity coefficient C_μ as

$$\nu_t = C_\mu \frac{k^2}{\varepsilon}, \quad (2.9)$$

and is therefore assumed to be isotropic, meaning the ratio of Reynolds stresses and strain rate is the same in all flow directions.

The transport equations for turbulent kinetic energy k and dissipation rate ε are two differential equations described as

$$\frac{\partial k}{\partial t} + U_i \frac{\partial k}{\partial x_i} = \mathcal{P}_k - \varepsilon + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \quad (2.10)$$

$$\frac{\partial \varepsilon}{\partial t} + U_i \frac{\partial \varepsilon}{\partial x_i} = [C_{\varepsilon 1} (\mathcal{P}_k) - C_{\varepsilon 2} \varepsilon] \cdot \frac{\varepsilon}{k} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] \quad (2.11)$$

where \mathcal{P}_k represents the turbulent kinetic energy production and is defined as

$$\mathcal{P}_k = -\overline{u'_i u'_j} \frac{\partial U_i}{\partial x_j}. \quad (2.12)$$

The coefficients σ_k and σ_ε are the Schmidt numbers for k and ε , respectively. $C_{\varepsilon,1}$ and $C_{\varepsilon,2}$ are model coefficients. The k -equation originates from the trace of the Reynolds-stress transport equation. The Reynolds-stress equation is described verbally [79] as

$$\frac{D \overline{u'_i u'_j}}{Dt} + \text{transport} = \text{production} + \text{pressure redistribution} - \text{dissipation}. \quad (2.13)$$

An approximation of the Reynolds-stress equation can be formed by modeling the transport and pressure redistribution and assuming isotropic dissipation. Then the k -equation is obtained by taking the trace of the Reynolds-stress equation approximation. The pressure redistribution term does no longer appear because its trace is zero (divergence is zero). All external forcing terms are neglected. The k -equation is then

$$\frac{\partial k}{\partial t} + U_i \frac{\partial k}{\partial x_i} = \mathcal{P}_k - \varepsilon + \frac{\partial}{\partial x_j} \left[\nu \frac{\partial k}{\partial x_j} - \frac{1}{2} \overline{u'_i u'_i u'_j} - \frac{1}{\rho} \overline{p' u'_j} \right] \quad (2.14)$$

from which the form in (2.10) is obtained by approximating the turbulent transport and pressure diffusion by

$$-\frac{1}{2} \overline{u'_i u'_i u'_j} - \frac{1}{\rho} \overline{p' u'_j} \approx \frac{\nu_t}{\sigma_k} \frac{\partial k}{\partial x_j}. \quad (2.15)$$

The k - ε turbulence model is due its simplicity and robustness applicable to a diverse range of applications. While it can provide reasonable accuracy to simple cases, it gives quite inaccurate results for complex flows. These inaccuracies originate from the Boussinesq hypothesis, in which the Reynolds-stress is not represented directly but modeled as function of k , ε , the strain-rate and the dissipation equation [59]. Besides that, the k and ε equations are a simplified version of the actual transport equation, resulting in errors in the modeled terms.

In wind turbine wake modeling using the k - ε turbulence model, the wake deficit is significantly underestimated and recovers faster compared to measurements. Also upstream of the turbine, k and the eddy viscosity ν_t build up unrealistically. Due to the underestimation of the wake deficit, the turbulent diffusion and so the Reynolds stresses are overestimated. This implies an overprediction of the eddy viscosity ν_t . Réthoré [64] found that the general eddy viscosity concept is not valid in the region surrounding the turbine which can be explained by analyzing LES data just upstream of a rotor. It is observed that the axial velocity gradient undergoes a steep increase but the normal Reynolds stress $\overline{u'_1 u'_1}$ decreases.

The k - ε model also breaks down when, over distances smaller than the turbulence length scale, the mean-velocity field changes [64]. This issue occurs often in wind turbine applications as the turbulence length scales in the atmospheric boundary layer are relatively large in size. Schmitt [70] analysed the correlation between the strain rate tensor and the Reynolds stress using LES and DNS and found that, in practise, the Boussinesq approximation almost never holds. Yet it is universally used because of its simplicity and robustness.

The Boussinesq approximation also assumes particles conserve their momentum between t_1 and t_2 where $t_2 - t_1$ is long enough such that the fluctuation components between the two time stamps are uncorrelated. However, in volumes with adverse pressure gradients, for example upstream and downstream of a turbine, the momentum of particles is systematically reduced. Therefore this approximation is not valid in wind turbine applications.

In wind turbine applications, the mean momentum equations contain body force and pressure-strain terms to take into account the rotor forcing. The k - and ε -equations can be derived from the dot product of the mean velocity field and mean momentum equation, in both the k - and ε -equations the force and pressure-strain terms are neglected. Réthoré [64] found that the two terms in the k -equations can be up to two orders of magnitude larger than the local dissipation rate, indicating they cannot be neglected. The terms in the ε -equation are small compared to the ratio ε^2/k , making the neglect in this equation acceptable.

2.2.4. Non-linear eddy viscosity models

To avoid some of the fundamental shortcomings of the linear eddy viscosity models, such as assuming the eddy viscosity is isotropic, more complex models could be chosen by including non-linear terms. These models are the so-called non-linear eddy viscosity models (NLEVMs). NLEVMs are also known as explicit algebraic Reynolds stress models, which rely on the assumptions that the Reynolds stress transport can be discarded, dissipation is isotropic and that the pressure-rate-of-strain relation is linear [79]. This makes the Reynolds stress not only a function of the strain-rate but also the vorticity. For the derivation of NLEVMs, it is convenient to work with normalized parameters, the anisotropic Reynolds stress, the strain-rate tensor and vorticity tensor, respectively

$$\mathbf{b} = b_{ij} = \frac{\overline{u'_i u'_j}}{k} - \frac{2}{3} \delta_{ij}, \quad \mathbf{S} = S_{ij} = \frac{1}{2} \frac{k}{\varepsilon} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right), \quad \mathbf{\Omega} = \Omega_{ij} = \frac{1}{2} \frac{k}{\varepsilon} \left(\frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i} \right). \quad (2.16)$$

For 3D flows, Pope [58] derived the explicit solution for the anisotropic Reynolds stress \mathbf{b} as a relation between ten tensorially independent basis tensors $\mathbf{T}^{(n)}$ and scalar functions $\alpha^{(n)}(\lambda_i)$ of the invariants λ_i . The basis tensors are defined as

$$\begin{aligned} \mathbf{T}^{(1)} &= \mathbf{S} & \mathbf{T}^{(6)} &= \mathbf{\Omega}^2 \mathbf{S} + \mathbf{S} \mathbf{\Omega}^2 - \frac{2}{3} \{ \mathbf{S} \mathbf{\Omega}^2 \} \mathbf{I} \\ \mathbf{T}^{(2)} &= \mathbf{S} \mathbf{\Omega} - \mathbf{\Omega} \mathbf{S} & \mathbf{T}^{(7)} &= \mathbf{\Omega} \mathbf{S} \mathbf{\Omega}^2 - \mathbf{\Omega}^2 \mathbf{S} \mathbf{\Omega} \\ \mathbf{T}^{(3)} &= \mathbf{S}^2 - \frac{1}{3} \{ \mathbf{S}^2 \} \mathbf{I} & \mathbf{T}^{(8)} &= \mathbf{S} \mathbf{\Omega} \mathbf{S}^2 - \mathbf{S}^2 \mathbf{\Omega} \mathbf{S} \\ \mathbf{T}^{(4)} &= \mathbf{\Omega}^2 - \frac{1}{3} \{ \mathbf{\Omega}^2 \} \mathbf{I} & \mathbf{T}^{(9)} &= \mathbf{\Omega}^2 \mathbf{S}^2 + \mathbf{S}^2 \mathbf{\Omega}^2 - \frac{2}{3} \{ \mathbf{\Omega}^2 \mathbf{S}^2 \} \mathbf{I} \\ \mathbf{T}^{(5)} &= \mathbf{\Omega} \mathbf{S}^2 - \mathbf{S}^2 \mathbf{\Omega} & \mathbf{T}^{(10)} &= \mathbf{\Omega} \mathbf{S}^2 \mathbf{\Omega}^2 - \mathbf{\Omega}^2 \mathbf{S}^2 \mathbf{\Omega}. \end{aligned} \quad (2.17)$$

This relation is valid for high Reynolds numbers and nearly homogeneous flows, where local effects dominate the transport effects. The formulation for \mathbf{b} is then given as

$$\mathbf{b} = \sum_{n=1}^{10} \alpha^{(n)}(\lambda_i) \mathbf{T}^{(n)}(\mathbf{S}, \mathbf{\Omega}). \quad (2.18)$$

There is only a finite number of linear independent basis tensors because other higher order tensors can be formulated as a linear combination of $\mathbf{T}^{(n)}$, employing the Cayley-Hamilton theorem [59, 79]. Therefore, the ten basis tensors are referred to as an integrity basis. There exist five linear independent invariants, the braces indicate the trace

$$\lambda_1 = \{\mathbf{S}^2\} \quad \lambda_2 = \{\mathbf{\Omega}^2\} \quad \lambda_3 = \{\mathbf{S}^3\} \quad \lambda_4 = \{\mathbf{S}\mathbf{\Omega}^2\} \quad \lambda_5 = \{\mathbf{S}^2\mathbf{\Omega}^2\}. \quad (2.19)$$

Van der Laan et al. [81] used two NLEVMs to predict the wake of a single wind turbine and compared it to field measurements. The predictions were improved compared to the standard k - ε turbulence model but behaved numerically unstable for high turbulence levels.

2.3. Visualization of Reynolds stress anisotropy

To visualize quantitative numerical CFD results, profiles of velocity and turbulent kinetic energy are often plotted at different positions in the domain. In many applications, it is crucial to also accurately predict the type and amount of turbulence anisotropy. This can be quantified by different anisotropy invariant maps [51]. These maps make use of the eigenvalues of the Reynolds stress anisotropy \mathbf{b} , which quantify the magnitude of turbulence anisotropy [21].

Three eigenvalues can be derived from \mathbf{b} , defined as $\lambda_1 \geq \lambda_2 \geq \lambda_3$. The eigenvalues λ_i are related to the eigenvalues of the Reynolds stress ϕ_i [37] through

$$\lambda_i = \frac{\phi_i}{2k} - \frac{1}{3}. \quad (2.20)$$

The Reynolds stress $\boldsymbol{\tau}$ is formed by the temporal average over n samples of the outer product of the turbulent fluctuations \mathbf{u}'

$$\boldsymbol{\tau} = \overline{u'_i u'_j} = \frac{1}{n} \sum_n u'_i u'_j = \frac{1}{n} \sum_n \mathbf{u}' \otimes \mathbf{u}'. \quad (2.21)$$

The tensor obtained by taking the outer product with itself is positive semi-definite, which is when $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$, where \mathbf{x} is a column vector with non-zero real numbers. Because the Reynolds stress tensor is positive semi-definite, its diagonal components and determinant are non-negative

$$\tau_{\alpha\alpha} \geq 0 \quad \forall \alpha \in 1, 2, 3, \quad \det(\boldsymbol{\tau}) \geq 0, \quad (2.22)$$

and the Cauchy-Schwarz inequality holds

$$\tau_{\alpha\beta}^2 \leq \tau_{\alpha\alpha} \tau_{\beta\beta} \quad \forall \alpha \neq \beta. \quad (2.23)$$

The eigenvalues of a positive semi-definite tensor are non-negative, meaning the lower limit of ϕ_i is zero. Given that the trace of $\boldsymbol{\tau}$ equals $2k$, this acts as an upper limit for the eigenvalues ϕ_i .

Because the Reynolds stress eigenvalues ϕ_i are bounded to $[0, 2k]$, the eigenvalues of the Reynolds stress anisotropy λ_i are subsequently bounded to $[-\frac{1}{3}, \frac{2}{3}]$. Given this, all realizable states of turbulence can be visualized on a triangular eigenvalue map. Given that the eigenvalues λ_i are bounded, the componentality (the relative strengths of the fluctuating components) is also bounded by three limiting states. These limiting states can be described as [21]

- one-component: when $\lambda_i = [\frac{2}{3}, -\frac{1}{3}, -\frac{1}{3}]^T$, is characterized by turbulent fluctuations only along one direction. One-component turbulence is often referred to as cigar-shaped or rod-like turbulence.
- two-component: when $\lambda_i = [\frac{1}{6}, \frac{1}{6}, -\frac{1}{3}]^T$, turbulent fluctuations occur along two directions with equal magnitude. Two-component turbulence is often specified as pancake-like turbulence.
- isotropic: when $\lambda_i = [0, 0, 0]^T$, often denoted as spherical turbulence.

When connecting the three components, boundaries are formed of the invariant maps. These boundaries also supplement further physical insight:

- axisymmetric expansion: by connecting one- and three-component turbulence. This occurs when $0 < \lambda_1 < \frac{1}{3}$ and $-\frac{1}{6} < \lambda_2 = \lambda_3 < 0$;
- axisymmetric contraction: by connecting two- and three-component turbulence. This occurs when $-\frac{1}{3} < \lambda_1 < 0$ and $0 < \lambda_2 = \lambda_3 < \frac{1}{6}$;
- two-component: by connecting one- and two-component turbulence. This occurs when $\lambda_1 + \lambda_3 = \frac{1}{3}$ and $\lambda_2 = -\frac{1}{3}$.

When at least one $\lambda_i = 0$, the turbulence reaches the state of plain-strain.

There are four commonly used anisotropy invariant maps, which are either constructed by linear or non-linear combinations of λ_i . The Lumley triangle, proposed by Lumley and Newman [51], is a non-linear invariant map defined by the second and third invariant of \mathbf{b} , defined as

$$II = b_{ij}b_{ji}/2 = \lambda_1^2 + \lambda_1\lambda_2 + \lambda_2^2, \quad (2.24)$$

$$III = b_{ij}b_{jn}b_{ni}/3 = -\lambda_1\lambda_2(\lambda_1 + \lambda_2) \quad (2.25)$$

and depicted in Figure 2.5. One-, two- and isotropic component turbulence are depicted by \mathbf{x}_{1c} , \mathbf{x}_{2c} and \mathbf{x}_{3c} , respectively. Axisymmetric expansion is denoted by the black line, axisymmetric contraction by the blue line and two-component turbulence by the red line.

To lay more emphasis on the region near the isotropic corner (\mathbf{x}_{3c}) [21], the turbulence triangle can be constructed as shown in Figure 2.5, in which the lower left quadrant is more stretched. The coordinates of the turbulence triangle (ξ , η) are constructed as

$$\xi^3 = III/2, \quad \eta^2 = II/3. \quad (2.26)$$

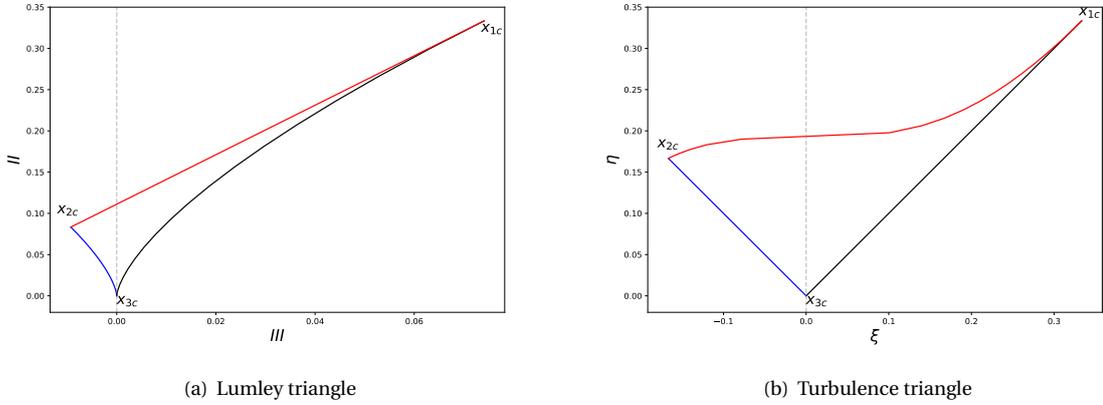


Figure 2.5: Non-linear invariant maps. The limiting turbulence states are included where the grey line indicates the plain-strain state.

Besides the non-linear invariant maps based on the second and third invariant, linear invariant maps can be constructed based on the eigenvalues of the Reynolds stress anisotropy. The eigenvalue map in Figure 2.6 uses the first two eigenvalues (λ_2 , λ_1) to determine the coordinate system.

The final invariant map discussed is the barycentric map. All realizable states of turbulence are located within an equilateral triangle such that all limiting states are weighted equally. The coordinates within the barycentric map (x_B, y_B) are defined as

$$x_B = C_{1c}x_{1c} + C_{2c}x_{2c} + C_{3c}x_{3c} = C_{1c} + C_{3c}\frac{1}{2}, \quad (2.27)$$

$$y_B = C_{1c}y_{1c} + C_{2c}y_{2c} + C_{3c}y_{3c} = C_{3c}\frac{\sqrt{3}}{2}, \quad (2.28)$$

with the weights as

$$C_{1c} = \lambda_1 - \lambda_2, \quad (2.29)$$

$$C_{2c} = 2(\lambda_2 - \lambda_3), \quad (2.30)$$

$$C_{3c} = 3\lambda_3 + 1. \quad (2.31)$$

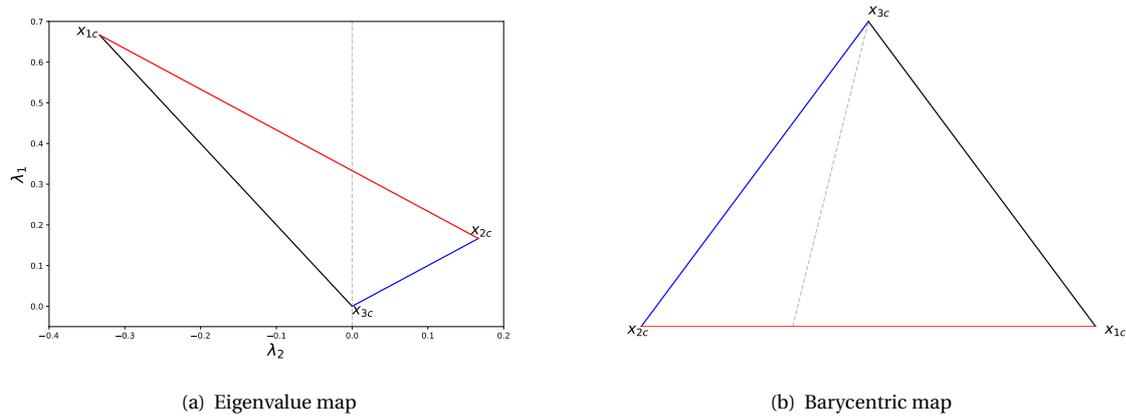


Figure 2.6: Linear invariant maps. The limiting turbulence states are included where the grey line indicates the plain-strain state.

The sum of the weights $C_{ic} = 1$, meaning uniqueness is enforced within the barycentric map. The difference between the discussed invariant maps is that non-linear maps have a bias towards the axisymmetric expansion limit, when compared to the linear maps. Also the linear and non-linear maps approach the isotropic corner differently. The barycentric triangle is the only map that provides equal weights to all limiting states, meaning the results are more easily interpretable.

There are two significant disadvantages of invariant maps [21]:

- (1) loss of physical context, because the maps are solely a function of the turbulence anisotropy eigenvalues, no information regarding the spatial domain is included. This problem could partially be solved by coloring the samples according to, for example, their wall distance as done in Figure 2.7.
- (2) difficult to properly represent large amounts of data, plotting all domain samples in an single invariant map can result in loss of interpretability because nearly identical \mathbf{b} will overlap in the invariant map, as can be seen in Figure 2.7. To avoid this, usually only small subsets of the domain are plotted, typically one-dimensional profiles.

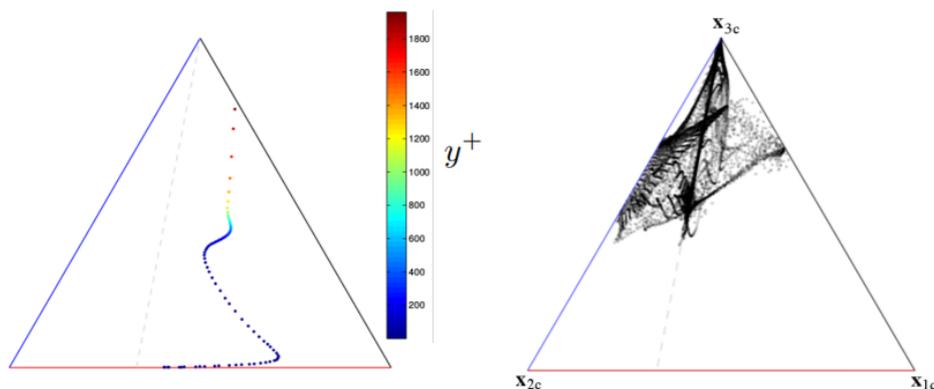


Figure 2.7: Barycentric maps for different CFD data. Left: Channel flow data for a one-dimension vertical profile where the data points are colored according to the wall distance y_+ . Right: Data points of the entire domain from a nozzle flow. Taken from Emory and Iaccarino [21].

To properly represent the information about turbulence anisotropy componentality in the physical domain, Emory and Iaccarino [21] proposed a color map constructed on the coordinates in the barycentric map, as depicted in Figure 2.8. This map can act as a colormap when constructing contour plots of the turbulence componentality in the physical domain. Using the red-green-blue (RGB) color system, the colors are defined by

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = C_{1c} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + C_{2c} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + C_{3c} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.32)$$

meaning one-component turbulence is visualized in red, two-component in green and isotropic turbulence in blue. All other possible states of turbulence within the barycentric map can be depicted as a combination of those three colors. An example is shown in Figure 2.9 for LES data on a 2D periodic hill.

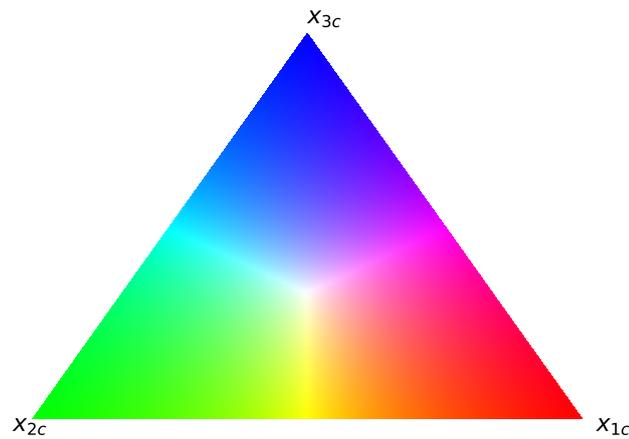


Figure 2.8: Barycentric map including turbulence anisotropy componentality contours.

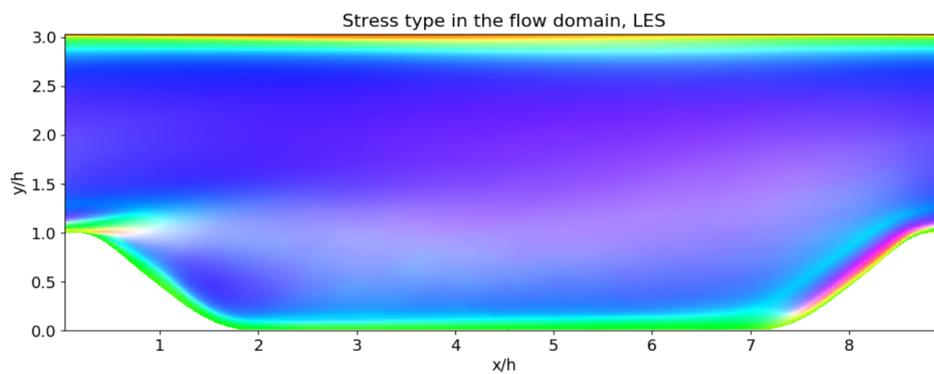


Figure 2.9: Turbulence states (stress type) depicted by RGB color system. Data is from LES of a 2D period hill at $Re = 10,595$. Taken from Kaandrop [38].

3

Machine Learning in Turbulence Modeling

Almost seventy years ago, Alan Turing wrote his famous essay "*Computing machinery and intelligence*" [78] in which he asked how we can construct a machine that can do the same things a human can do. Instead of building a computer program that acts like a human from scratch, Turing proposed to build a machine that, initially, cannot do many things but which is able to learn from past experience.

The implementation of Turing's idea is machine learning, algorithms that are initially of little use but can be tuned using existing observations such that they create a predictive ability for future situations. If one would define the extent that a model, on average, fails to predict future observations as the *generalization error*, then machine learning aims to minimize this generalization error [33]. In other words, machine learning tries to find the optimal model (algorithm) and its corresponding parameters such that future observations are predicted correctly as many times as possible.

Machine learning can be categorized into supervised and unsupervised learning. Supervised learning is the task to predict a specific unseen quantity based on observed quantities. In *regression* this quantity is continuous while in *classification* it is discrete. In unsupervised learning the quantity of interest is unknown. The task is therefore to find previously unknown patterns in the data.

3.1. Machine learning in turbulence modeling

In recent years the availability of large and diverse numerical data sets has increased significantly. Researchers in the field of turbulence started to explore methods that can *learn* from the available data in order to reduce turbulence model uncertainties. Turbulence model uncertainties are caused by the introduction of many model assumptions and simplifications by going from the Navier-Stokes equations to the RANS framework with model closure.

Ling and Templeton [48] used different machine learning algorithms to predict regions of high uncertainty in RANS. This uncertainty was based on violation of different RANS assumptions. Duraisamy et al. [18] listed four stages of simplifications to construct a Reynolds-averaged turbulence model, described below.

- L1: The application of ensemble- or time-averaging operators which lead to a system of equations that is undetermined. In the averaging process, information of the instantaneous realizations is lost which is fundamentally irrecoverable.
- L2: Representing the Reynolds stress by a model to close the undetermined system of equations. Examples of such models are linear eddy viscosity models and algebraic stress models.
- L3: The specific functional form within a model. Physical processes or assumptions can be represented by either algebraic or differential equations.
- L4: Specifications of model coefficients often obtained through empirical evidence.

When enhancing a turbulence model with a machine learning algorithm, the starting point are the L2 assumptions. This means the uncertainties introduced by time-averaging and model closure are still present. Machine learning models can, however, be used to lower the L2 uncertainty by introducing non-linear terms in a LEVM. A popular strategy is to include the generalized expansion of the Reynolds stress tensor [58] in

the form of (2.18) assuming that the Reynolds stresses only depend on mean velocity gradients. The machine learning algorithm is then used to learn the relation and values between the basis tensors and invariants.

Ling et al. [49] presented a method to train a model for the anisotropic part of the Reynolds stress tensor from high-fidelity (DNS and well-resolved LES) data using deep neural networks. Neural networks consist of connecting nodes (often called *neurons*), each node has an input \mathbf{x} which is transformed to an output using an activation function f . These activation functions also have node weights \mathbf{w} and a bias b (often defined as node weight w_0) as inputs, which are tuned during the learning process of the algorithm. The activation function is formed as

$$y = f(\mathbf{w} \cdot \mathbf{x} + b) \quad (3.1)$$

where f can be, for example, a sigmoid function.

In a neural network, hidden layers contain one or more nodes which are connected to all the nodes in the next and previous hidden layer. The first hidden layer has as input the pre-defined input parameters which are in the case of Ling et al. [49] the five invariants and ten basis tensors listed in (2.19) and (2.17), respectively. The last hidden layer has as output the estimated parameter, in this case the normalized Reynolds stress anisotropy tensor \mathbf{b} . A conventional neural network consists of a single hidden layer while deep neural networks are defined as having at least two hidden layers. A simple neural network is depicted in Figure 3.1 which consists of three input units, a single hidden layer with four hidden units and two output units.

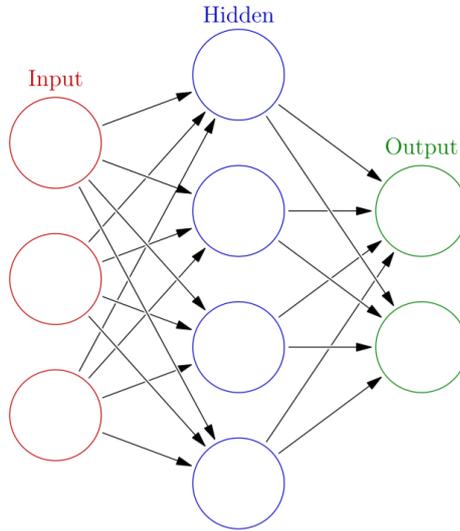


Figure 3.1: Neural network with three input units, a single hidden layer containing four hidden units and two output units, from Herlau et al. [33].

The numerous neural networks weights are optimized for the specific problem by *training* on available data sets using back propagation with gradient descent. By iteratively adjusting the weights, the lowest mean squared error can be obtained by following the direction of steepest slope in the cost function (a measure of the difference between the output parameters and the true values). Ling et al. [49] used a tensor basis neural network (TBNN) which, by enforcing the prediction to lie on a basis of isotropic tensors, embeds rotational invariance. Rotational invariance is a physical principle that every tensor should obey to, as it means the fluid physics are independent on the orientation of the coordinate frame. The TBNN has two input layers, one containing the five invariants and the other the ten basis tensors. This ensures the predicted anisotropy tensor has the form of (2.18). The TBNN showed to have significant improvements over the default LEVM and quadratic eddy viscosity model (QEVN) while predicting the Reynolds stress anisotropy, *a priori*. Subsequently, the predicted anisotropy tensor was implemented into a RANS solver in the momentum equation and turbulent kinetic energy production term. The RANS solver first converged using the default LEVM before prescribing the TBNN predictions and to re-converge. The TBNN showed to be able to improve the predictions of flow characteristics, such as corner vortices and flow separation when compared to the LEVM model. However, there still remained a large discrepancy with the high-fidelity DNS data. A major drawback of neural networks is the computational costs of the training phase. It took Ling et al. [49] 4000 CPU hours to train their ensemble of 20 neural networks.

Another promising framework is the novel expansion of Gene Expression Programming (GEP) to model tensors, proposed by Weatheritt and Sandberg [91, 92]. This framework gives the algorithm the freedom to produce a constraint-free model and therefore to come up with its own functional form. This results in an algebraic equation than can easily be implemented into a CFD solver. For this, symbolic regression is required, which searches the space of mathematical expressions with the goal to find a model that best fits the given dataset. GEP [23] revolves around the philosophy of iteratively improving a population of candidate solutions according to the principle of survival of the fittest. It relies, just as in biology, on random changes and is therefore non-deterministic. The objective of GEP is to search for the function, from the space of all functions, that best fits the data. This is done according to the *fitness* of the candidate function that defines the relative distance of the candidate function outcome to the true value. To search the space of all functions, functions are expressed in a compact notation related to the coding sequence of genes. The GEP algorithm is depicted in Figure 3.2 which has analogies to an evolutionary process. At $i = 0$, a population P^i of N candidate solutions is created randomly from the allowed set of mathematical symbols. In the selection phase, P^i is filtered in the direction of more fit solutions via tournament selection. This results in the individual candidate solutions with the highest fitness. In the reproduction and genetic operators phase, genetic diversification is created by introducing modifications to allow for evolution in the long run. This is done according to, for example, replication, recombination and mutation. In the next selection step, poor variants will subsequently be filtered out. The algorithm is made suitable for tensor expressions by introducing multidimensional gene expression programming (MGEP) which allows for co-evolution of two populations with one solely consisting of tensor symbols and the other out of scalar symbols.

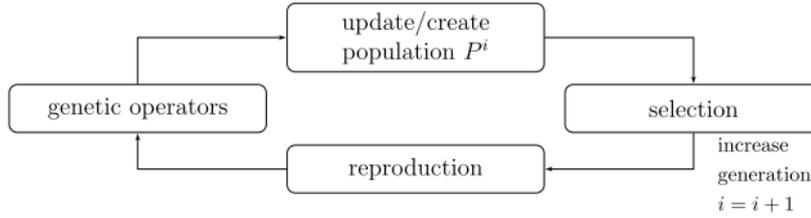


Figure 3.2: Flow chart of GEP algorithm, from Weatheritt and Sandberg [91].

Using the MGEP algorithm, a model for \mathbf{b}^x , the extra anisotropy with respect to the linear Boussinesq approximation, is formed. Although the approach to symbolically regress turbulence model corrections is still novel, good predictions of the streamwise velocity and shear stress on a backward facing step and periodic hills flow were found. Weatheritt and Sandberg [92] noted that the test cases do not represent the same amount of complexity as flows in industrial applications. Therefore, to test the potential of MGEP, further research on more complex flow cases is needed.

In the GEP algorithm, each run discovers different models due to its non-deterministic nature. Schmelzer et al. [67, 69] proposed a deterministic symbolic regression method which searches for sparse algebraic models using regression techniques that promote sparsity. The method is referred to as SpaRTA (Sparse Regression of Turbulent Stress Anisotropy). From pre-defined features, a library of candidate functions is created based on non-linear combinations of the features. Sparsity is promoted by using elastic net regularization with the goal to improve the predictive performance and interpretability of the model. Elastic net regularization is an extension to ordinary least squares regression to reduce the model variance while increasing the model bias. This is enforced by reducing the magnitude of large coefficients (l_2 regularization) and by setting lesser important coefficients to zero (l_1 regularization). This results in the following optimization

$$\mathbf{w} = \underset{\hat{\mathbf{w}}}{\operatorname{argmin}} \left[\left\| \hat{\mathbf{w}}^T \mathbf{x} - \mathbf{y} \right\|_2^2 + \lambda \rho \left\| \hat{\mathbf{w}} \right\|_1 + 0.5 \lambda (1 - \rho) \left\| \hat{\mathbf{w}} \right\|_2^2 \right] \quad (3.2)$$

in which \mathbf{w} are the model coefficients, \mathbf{x} the library of candidate functions and \mathbf{y} the true values. The regularization strength and mixing are defined by λ and ρ , respectively.

To construct such a deterministic model, the model-form error of the standard baseline turbulence model (LEVM) is first determined. This error is represented by a turbulence anisotropy correction \mathbf{b}^Δ and a transport equation deficit R in the transport equations. To obtain the model-form errors \mathbf{b}^Δ and R , the turbulence equations are solved using the time-averaged velocity field, turbulent kinetic energy and Reynolds stress, obtained from LES or DNS simulations. Because the turbulence equations consist of three equations (two transport equations and the Boussinesq hypothesis) and three unknowns (two correction terms and the dissipation

rate ε or specific rate of dissipation ω), the system is closed. This procedure is referred to as k -corrective-frozen RANS. Subsequently, SpaRTA is used to obtain algebraic expressions for both correction terms. It was found that by using only the correction for R already improvements for the velocity field were obtained. The advantage of symbolic regression is that simple explicit algebraic equations are obtained that can easily be interpreted and implemented into a solver. The disadvantage is that, due to its simplicity, it is in a lesser extent able to discover all flow details, when compared to more complex machine learning algorithms [69]. Similarly as to the work of Weatheritt and Sandberg [91, 92], a limited set of test cases was used. To show the potential of SpaRTA for industrial purposes, a variety of (more complex) flow cases have to be tested.

Kaandorp and Dwight [37] presented a stochastic Tensor Based Random Forest (TBRF) algorithm for turbulence modeling. A random forest is an ensemble of decision trees where each individual decision tree is trained on a randomly selected subset of the data. A common decision tree algorithm is the Classification and Regression Tree (CART) algorithm which binary splits the data at each of its nodes by making use of a *if-then* paradigm [48]. This split is based on a threshold value coupled to a specific feature and are selected to minimize the difference between the output and a constant approximation of the output in both splits [37]. Based on the CART algorithm, a Tensor Based Decision Tree (TBDT) algorithm is constructed which, instead of defining a constant approximation of the output in both splits, approximates a constant value for every basis tensor coefficient $\alpha^{(n)}$ in (2.18). This is done to minimize the difference between the high-fidelity anisotropy tensor and the expression in (2.18). Specifically, the following is solved

$$\min_{j,s} \left[\min_{\alpha_L^{(n)} \in \mathbb{R}^{10}} \sum_{x_i \in R_L(j,s)} \left\| \sum_{n=1}^{10} \mathbf{T}_i^{(n)} \alpha_L^{(n)} - \mathbf{b}_i \right\|_F^2 + \min_{\alpha_R^{(n)} \in \mathbb{R}^{10}} \sum_{x_i \in R_R(j,s)} \left\| \sum_{n=1}^{10} \mathbf{T}_i^{(n)} \alpha_R^{(n)} - \mathbf{b}_i \right\|_F^2 \right] \quad (3.3)$$

where \mathbf{b}_i is the index i of the high-fidelity anisotropy tensor. Subscript L and R denote the left and right split, respectively. The subscript F resembles the Frobenius norm. The split is specified by the feature index j and value s . The tensor basis coefficients can be found by solving two least-squares problems.

The TBRF is a collection of multiple TBDTs in which the anisotropy tensor \mathbf{b} is predicted using baseline RANS mean-fields as input. Five two-dimensional flow cases were used for training and testing. After the baseline RANS simulation has converged, the predicted anisotropy tensor is introduced in the momentum equation. To improve the stability, a relaxation parameter is implemented and a modified k -equation is solved simultaneously. The improvements over the k - ω turbulence model were similar as those of the TBNN from Ling et al. [49]. However, the TBRF has several advantages over the TBNN as it provides prediction variance, is less sensitive to its hyperparameters and cheaply trained.

3.2. Feature Selection in Machine Learning

Feature selection (or reduction) is commonly applied in the field of machine learning to improve the predictive performance of an algorithm. Feature selection can be applied for the following reasons

- Model simplification and interpretation
- Reduce training costs (time and storage requirements)
- Improve model accuracy
- Reduce overfitting.

By reducing the dimensionality of the feature set, features are eliminated that have little or no information towards the target variable. Features can either be relevant, irrelevant or redundant [46]. Relevant features have an influence on the target variable and should be kept in the set of features. Irrelevant features do not have (or very little) influence on the target variable, meaning they mainly contribute as noise sources and should be removed from the set of features. A redundant feature shares the same influence towards the target variable as another feature (high dependency between features), indicating that it is unnecessary to include the feature in the feature set.

Methods for feature selection can be decomposed into three classes

- Filter methods
- Wrapper methods

- Embedded methods.

Filter methods are based on measures obtained from statistical tests or information theory. Examples are the correlation coefficient to detect linear, monotonic relations between features and the target variable, and mutual information to measure a more general dependency. Mutual Information (MI) measures the *shared information* between two variables [17]. MI can be expressed as

$$MI(X, Y) = H(X) - H(X|Y) \quad (3.4)$$

where $H(X)$ is the marginal entropy and $H(X|Y)$ the conditional entropy. The marginal entropy $H(X)$ defines the uncertainty in variable X , while the conditional entropy $H(X|Y)$ is a measure of the uncertainty remaining when a different variable Y is known. Then, the mutual information $MI(X, Y)$ is equivalent to the amount of uncertainty in variable X which is removed by knowing variable Y . Mutual information can also be expressed as

$$MI(X; Y) = \int_y \int_x p_{(X,Y)}(x, y) \log \left(\frac{p_{(X,Y)}(x, y)}{p_X(x)p_Y(y)} \right) dx dy \quad (3.5)$$

and is defined by the Kullback-Leibler divergence, which measures how different one probability distribution is from another. If variables X and Y are independent, then $p_{(X,Y)}(x, y) = p_X(x)p_Y(y)$ and the divergence is zero (so mutual information is zero when two variables are independent). The advantage of filter methods is that the computational costs are low and therefore a large number of features can be evaluated in a relatively short period of time.

Wrapper methods make use of a machine learning algorithm to score the considered features. They are computationally more expensive than the filter methods (filter methods do not require a learning algorithm to quantify the importance of features). Examples of wrapper methods are forward feature selection, backward feature reduction and recursive feature elimination. The advantage of wrapper methods is that the feature importance is directly derived from the algorithm performance.

In forward feature selection, from an empty feature set, the feature (one from all available features) is added that gives the lowest test error (when tested using an external estimator e.g. learning algorithm) compared to all other features. This means that if there are N features, N models are constructed that are all based on a different feature. The feature that corresponds to the best performing model is added to the empty feature set. Subsequently, an additional feature (out of all remaining features) is added that decreases the error as much as possible. This procedure goes on up to the point that the error stops diminishing or when a pre-defined maximum number of features is reached. The main disadvantages are that it is not able to remove any features that may have become obsolete after another feature has been added and that the computational costs are significant as $\sum_{n=0}^k (N - n)$ models have to be trained. The total number of features is N and the number of steps in the selection process is k .

Backward elimination works in the opposite direction as forward selection. The algorithm starts with the full set of features and sequentially removes the specific feature that results in largest decrease of the error. This procedure is continued until the error stops decreasing. The main disadvantages of backward elimination is that it is unable to add features that would have been of use after the removal of another feature. Similarly to forward selection, the computational costs are high.

Recursive feature elimination also performs backward elimination. The algorithm first fits an external estimator on all features, this external estimator provides a measure of feature importance either by coefficients (e.g. from a linear fit) or a feature ranking (e.g. random forest). Based on this feature ranking, the top n ranked features are used to refit the model. The procedure is repeated recursively until a pre-defined number of remaining features is obtained. The disadvantage of recursive feature elimination is that the procedure starts with the entire dataset, which can be infeasible in particular applications where the number of initial features is large. Weatheritt et al. [93] used this method to reduce the dimensionality and to determine the most important features for their scalar flux model. They used a linear fit as the external estimator.

In embedded methods, the feature selection is incorporated in the learning algorithm. For example, regression with l_1 regularization forces coefficients of irrelevant features to go to zero. This results in that only the important features remain to have non-zero coefficients. For tree algorithms, features are ranked on how well they can split the data at each node. Ling and Templeton [48] used feature ranking based on the Gini information of the Adaboost decision tree (DT) to rank the features. They also applied forward feature selection, which required training the algorithms hundreds of times, which is significantly slower than the feature

ranking method. It was also found that the forward selection results were sensitive to the randomly sampled training and test data, meaning the results were inconsistent.

3.3. Predictions of regions of high uncertainty

Machine learning can be utilized to identify regions of high uncertainty in RANS. This is of significant value because RANS often produces inaccurate flow predictions. Ling and Templeton [48] evaluated three different machine learning classifier algorithms to predict regions of high RANS uncertainty: support vector machines (SVM), Adaboost decision trees, and random forests. The classifiers were trained to evaluate three different basic RANS eddy viscosity assumptions: eddy viscosity isotropy, linearity of the Boussinesq hypothesis, and the non-negativity of the eddy viscosity. The classifiers functioned as markers by either being switched *on*, predicting high uncertainty or *off*, indicating low uncertainty. The random forest came forward as the best performing classifier, this could be due to its resilience to noisy training data (statistical noise in high-fidelity data). Part of the results are shown in Figure 3.3 for an inclined jet in crossflow. The regions with true positive and false negative indicate where the RANS model assumption is violated, in this case the linearity assumption. The regions with true negative and false positive is where the model assumption is valid.

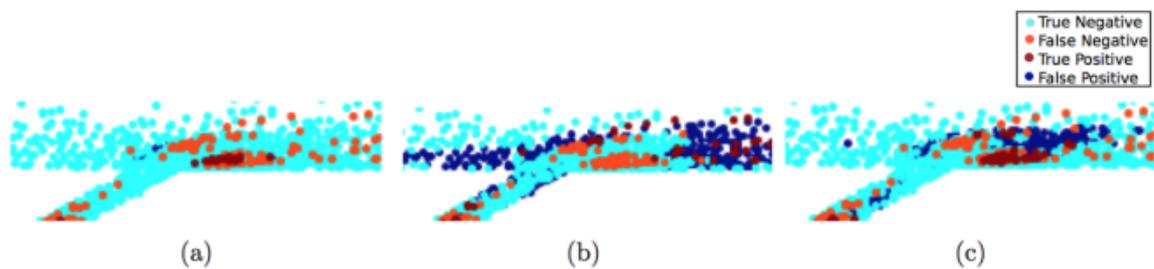


Figure 3.3: Classifier predictions by (a) SVM, (b) decision tree and (c) random forest on an inclined jet in crossflow. Taken from Ling and Templeton [48].

Gorlé et al. [27] argued that linear eddy viscosity models are only applicable when there is an alignment between the anisotropy and Reynolds stress tensor. For example, this is valid in simple parallel shear flows but not in complex three-dimensional flows. RANS models are generally calibrated on nearly parallel shear flows and validated to be reasonable accurate for such flows as wakes, jets, boundary layers and mixing layers. Based on this, they formulated a marker function which measures the deviation locally from a parallel shear flow. This can be accomplished by examining the velocity gradient and the streamline, locally. By comparing RANS with high-fidelity LES/DNS for varying cases, it was found that the marker predictions largely coincided with regions where the divergence of the Reynolds stress tensor is inaccurate.

3.4. Machine learning in wind energy

Machine learning is also applied to enhance the modeling of wind turbine wakes. Iungo et al. [34] calibrated a RANS mixing length turbulence model with high-fidelity LES data for different tip speed ratios. However, they only compared the results between downstream locations $x/D = 2$ and $x/D = 5$ and did not compare to any baseline turbulence model. King et al. [39] used a Gaussian process algorithm to predict the RANS eddy viscosity based on mean flow field variables. By comparing the results to the baseline mixing length turbulence model, significant improvements were made in modeling the velocity in a wind farm. However, there also remained a discrepancy with the high-fidelity LES data. The RANS simulations were performed in 2D, meaning only the velocity fields at hub height are obtained.

Using machine learning, Ti et al. [76] developed a new wake model to improve the turbine wake predictions. They constructed an artificial neural network (ANN) with turbulence intensity and wind speed obtained from RANS (using the $k-\epsilon-f_p$ turbulence model) as input. The output variables of the neural network are the wake velocity deficit and added turbulent kinetic energy. For a standalone turbine case, the results were compared to the analytical Jensen [35] and Gauss [4] models and showed great improvements with errors less than 5% with respect to the RANS simulation.

4

Methodology

This chapter discusses the different methodologies used and developed in this thesis. First the model-form errors of the k - ε turbulence model are discussed in Section 4.1, which were obtained through the k -corrective-frozen-RANS approach prior to this thesis. Secondly, the mutual information metric and its estimator are explained comprehensively in Section 4.2. Thirdly, the SpARTA algorithm developed by Schmelzer et al. [69] is described in Section 4.3 and finally the sparse logistic regression framework in Section 4.4.

4.1. Model-form error k - ε turbulence model

By introducing correction terms to the turbulence model, the goal is to improve the RANS predictions and specifically mean flow fields such as velocity and turbulence kinetic energy. Schmelzer et al. [69] proposed to add a correction to the Reynolds stress anisotropy, \mathbf{b}^Δ , and a residual to the transport equations, R . The corrections are incorporated in the k - ε turbulence model (and indicated in red) as

$$\frac{Dk}{Dt} = \mathcal{P}_k + R - \varepsilon + \frac{\partial}{\partial x_j} \left[\left(v + \frac{v_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \quad (4.1)$$

$$\frac{D\varepsilon}{Dt} = [C_{\varepsilon 1} (\mathcal{P}_k + R) - C_{\varepsilon 2} \varepsilon] \cdot \frac{\varepsilon}{k} + \frac{\partial}{\partial x_j} \left[\left(v + \frac{v_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] \quad (4.2)$$

$$\mathcal{P}_k = 2k b_{ij} \frac{\partial U_i}{\partial x_j}, \quad b_{ij} = -\frac{v_t}{k} S_{ij} + b_{ij}^\Delta. \quad (4.3)$$

To extract both correction terms from LES data, the RANS turbulence equations are solved passively using time-averaged LES fields for velocity, turbulent kinetic energy and Reynolds stresses, as described in [69]. These fields remain frozen during the procedure. The turbulence equations contain three unknowns, ε , R and b_{ij}^Δ (in this work denoted as \mathbf{b}^Δ) and three equations, the two transport equations and the eddy viscosity formulation. The unknown quantities can therefore be obtained through an iterative process. This procedure is referred to as the k -corrective-frozen-RANS approach.

When the obtained values for \mathbf{b}^Δ and R are injected as a static/constant correction to the standard k - ε turbulence model equations, the mean-flow conditions are significantly improved. The predictions of the mean stream-wise velocity component and turbulent kinetic energy are depicted in Figure 4.1 for several horizontal profiles in case A (cases described in Chapter 5). Case A consists of two turbines positioned in series, where the second turbine is located $5D$ downstream of the first. The LES profiles are compared to the profiles obtained by the baseline k - ε turbulence model and the k - ε turbulence model where the frozen corrections are injected. Clearly, the baseline turbulence model underpredicts the velocity deficit when compared to LES. When the k - ε turbulence model is injected with the frozen corrections, the velocity profiles are significantly improved, matching LES accurately. Similarly, for the profiles of k , the baseline k - ε turbulence model is inaccurate and overpredicts the amount of turbulent kinetic energy in the turbine wake while the frozen correction results in a good match with LES.

It should be noted that to both the baseline turbulence model and the turbulence model with the frozen corrections, free-stream corrections in the form of R_{FS} and \mathbf{b}_{FS}^Δ are added such that the free-stream profiles in RANS match the profiles in LES.

Given that the frozen corrections result in a good match with the LES data, a machine learning model that can accurately estimate the frozen corrections would then potentially also lead to a good match with LES.

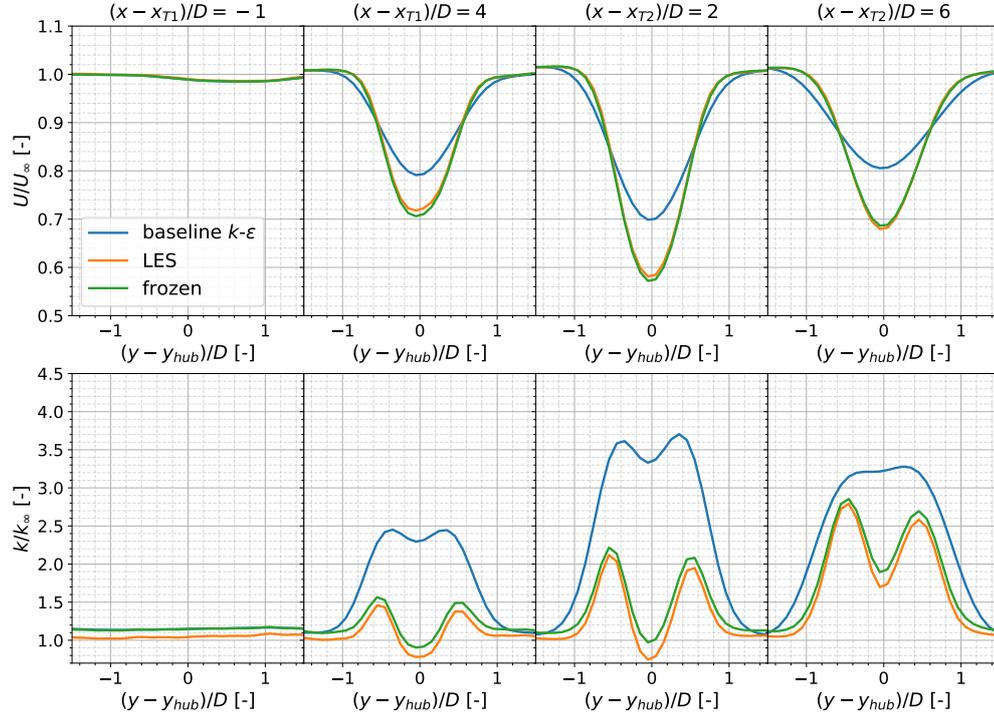


Figure 4.1: Mean stream-wise velocity and turbulent kinetic energy at hub-height normalized with free-stream values as function of normalized span-wise position $(y - y_{hub})/D$. Location in x denoted in title.

4.2. Mutual Information

Mutual information is a tool for feature selection in machine learning applications. Feature selection is often utilized to reduce computational resources, improve prediction performance and increase the interpretability of the model [13]. A common approach for feature selection is to define the dependency between variables. The most important features are then considered to have a high dependency with the output variable and low dependency with other chosen features. The correlation coefficient measures the statistical relation between variables but is limited to linear dependencies and monotonic functions [5]. A more general measure of dependency, not limited to linearity and monotonicity, is mutual information (MI). Mutual information quantifies the amount of *information* of a variable obtained through observing a different variable. Mutual information is widely used for feature selection for both categorical and continuous variables ranging in disciplines from statistics to machine learning and computational neuroscience [25].

For continuous random variables, the mutual information between variable X and Y can be computed by

$$MI(X; Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x, y) \log \frac{p_{(X,Y)}(x, y)}{p_X(x) p_Y(y)} dx dy, \quad (4.4)$$

in which $p_{(X,Y)}(x, y)$ is the joint probability density between the variables and $p_X(x)$ and $p_Y(y)$ are marginal distributions. When X and Y are independent, the joint distribution is the product of the marginal distributions and $MI(X; Y)$ equals zero. Mutual information measures the difference between the joint distribution compared to its marginal distributions.

The definition of mutual information is often denoted as the Kullback-Leibler divergence that measures the distance between probability densities [17]. Some useful properties are that the divergence is always

positive and invariant to parameter transformations.

In information theory, the concept of mutual information is closely related to that of entropy. Entropy can be interpreted as the average level of uncertainty in the outcome of a random variable. Uncertainty is equivalent to the average amount of information required to describe a random variable's outcome. The information is quantified in number of bits (when using log-base of 2) or nats (when using natural log-base) to store the outcome's information [57]. For discrete random variables it is defined as

$$H(X) = - \sum_{i=1}^n p_X(x_i) \log p_X(x_i). \quad (4.5)$$

When computing the entropy for continuous variables, the differential entropy has to be used instead

$$h(X) = - \int_{\mathcal{X}} p_X(x) \log p_X(x) dx \quad (4.6)$$

which is defined by an integral instead of a summation. The main difference between entropy $H(X)$ and differential entropy $h(x)$ is that $h(x)$ can be negative [17]. (Differential) entropy can be interpreted as when all possible events of X have a low probability, it is difficult to guess the outcome, resulting in a high entropy. Contrary, if X consists of events with high probability, it is more easy to guess the outcome, leading to a small entropy.

The concept of differential entropy can be extended to joint differential entropy of two variables X and Y as $h(X, Y)$, which measures the uncertainty associated with multiple variables. Similarly, conditional differential entropy $h(X|Y)$ defines the uncertainty in a variable X , given Y is known. From these definitions, mutual information can also be expressed as

$$MI(X; Y) = h(X) - h(X|Y) = h(X) + h(Y) - h(X, Y). \quad (4.7)$$

Thus, mutual information equals the difference between the differential entropy of random variable X and the conditional differential entropy of X with respect to Y . Therefore, mutual information can be interpreted as the amount of uncertainty in X which is removed by knowing Y . Hence, when Y is independent from X , then $h(X|Y) = h(X)$ and $MI(X; Y) = 0$.

4.2.1. Mutual Information Estimation

For most continuous data sets, the underlying probability density functions are unknown, which obstructs direct computations of mutual information. Different techniques exist to estimate either the probability densities or the entropy of continuous random variables from samples. Entropy estimations can be made from k -nearest neighbor distances [44, 63, 89] while kernel density estimators [45, 54, 89] or histogram-based methods [84, 89] are often used to estimate the data's underlying probability density functions.

Histogram-based methods discretize the continuous data into bins which, for example as used in [63], all contain an equal amount of points n . It was shown that for small values of n , MI was overestimated while when n was large, MI was underestimated as MI goes to zero when n goes to infinity. There is no simple way to determine the best value of n to minimize this bias.

Kernel density estimation (KDE) is a non-parametric method that estimates the probability density function of a data set. Non-parametric means no assumption is made on the underlying distribution. In this work, the focus is on the KDE and the k -nearest neighbor methods as these are frequently applied in literature and give the most reliable results.

Kernel Density Estimation

Kernel density estimation (KDE) is a well-known method to estimate probability density functions from samples. In the context of mutual information, it is often used to define the joint and marginal density functions in (4.4). KDE is superior to the histogram-based methods because it has a better rate of convergence to the underlying density [54].

KDE computes an estimator $\hat{p}(\mathbf{x})$ based on N samples \mathbf{x}_i from an unknown density, by constructing a kernel K at each data point together with a smoothing parameter (often referred to as the bandwidth parameter) h as follows

$$\hat{p}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N K(\mathbf{x} - \mathbf{x}_i, h). \quad (4.8)$$

A kernel is a non-negative function, often a uniform or Gaussian probability density function. The bandwidth h is a parameter that affects the smoothness of the resulting distribution. This parameter strongly influences the KDE estimations, and has to be chosen carefully. Small h generally results in low bias and high variance (overfitting), while large h leads to the opposite: high bias and low variance (underfitting). The bandwidth value can quickly be estimated using assumptions about the data, for example using "rules of thumbs" [71] based on data characteristics, such as number of samples and the scale of the data. However because these are crude estimations, these methods give in practise not reliable results.

There exist automatic, empirical bandwidth selection techniques [22] that fit the KDE to part of the data and then determines how well the KDE agrees with the remaining samples by computing the data log-likelihood. Although this leads to a more trustworthy bandwidth parameter, the procedure is computationally inefficient because a KDE has to be constructed for each potential bandwidth value.

From the constructed density functions, the integral in (4.4) is efficiently solved by assuming the locations of the data are convenient samples from the underlying density function

$$MI(X; Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x, y) \log \frac{p_{(X,Y)}(x, y)}{p_X(x)p_Y(y)} dx dy \approx \frac{1}{N} \sum_{i=1}^N \log \frac{p_{(X,Y)}(x_i, y_i)}{p_X(x_i)p_Y(y_i)}. \quad (4.9)$$

The performance of the estimator is examined for a test case for which the exact mutual information is known. Given two variables X and Y , both Gaussian variables with zero mean, variance equal to unity and covariance r , $MI_{\text{true}}(X; Y)$ is exact and defined as

$$MI_{\text{true}}(X; Y) = -\frac{1}{2} \log(1 - r^2). \quad (4.10)$$

If $r = 0$, the variables are uncorrelated and the mutual information is 0, when the two variables are perfectly correlated, $r = \pm 1$, the mutual information is infinite. For $r = 0.6$, the exact solution for the mutual information between two Gaussian variables $MI_{\text{true}}(X; Y) = 0.223$. By constructing a KDE using a Gaussian kernel and an empirically optimized bandwidth parameter through cross-validation, the error $\varepsilon = |MI_{\text{true}}(X; Y) - MI_{\text{KDE}}(X; Y)| / MI_{\text{true}}(X; Y)$ is obtained and depicted in Figure 4.2 where the vertical lines indicate the standard deviation of the error. The computation of $MI_{\text{KDE}}(X; Y)$ is repeated 50 times for each N to investigate the variance in the estimation, each time a new set of samples is drawn from the Gaussian distribution. It is chosen to only use 50 repetitions because for $N = 2 \cdot 10^4$, each run takes around 15 seconds, and therefore it is computationally not feasible to do much more repetitions.

The results are remarkable, because it was expected that the error would converge for increasing sample size. It is observed that for a sample size of $N = 100$, the mean error is relatively small (6%) while for $N = 500$ it is large (30%). For $N > 5000$, the mean error remains roughly constant. The standard deviation does have a downward trend for increasing sample size. Hanchuan et al. [29] stated that with a properly chosen bandwidth and kernel, the estimate $\hat{p}(\mathbf{x})$ converges to the true underlying density for increasing N . This means that either the kernel or bandwidth is causing the error to not converge. For each N , a new bandwidth is chosen by means of cross-validation from a limited set of possible values. Because the kernel is kept constant, it is expected that the difference between the chosen bandwidth and the optimal bandwidth differs for each N . This explains why the error can be larger for larger N . This shows that the performance of the KDE method depends significantly on its parameters and that increasing the sample size does not necessarily lead to more accurate predictions if it is not guaranteed that the chosen parameter values are close enough to the optimal values.

Besides that the results depend significantly on the chosen parameters, another drawback is the computational costs. It takes around 15 seconds for each estimation for $N = 2 \cdot 10^4$, which excludes the computations required for selecting the bandwidth. The costs scale with $\mathcal{O}(N^2)$. For each sample, the effects of N kernels have to be included. Because the mutual information is estimated by the average over N samples, the costs scale with N^2 . The cost can be reduced by increasing the tolerances such that not all constructed kernels are used while making estimations, however this does not result in major cost reductions and also makes the estimation more inaccurate.

***k* Nearest Neighbor**

Contrary to the KDE method, the k -nearest neighbor (k NN) approach is computationally more effective because, for every data point, it locally estimates the differential entropy. The performance of histogram-based,

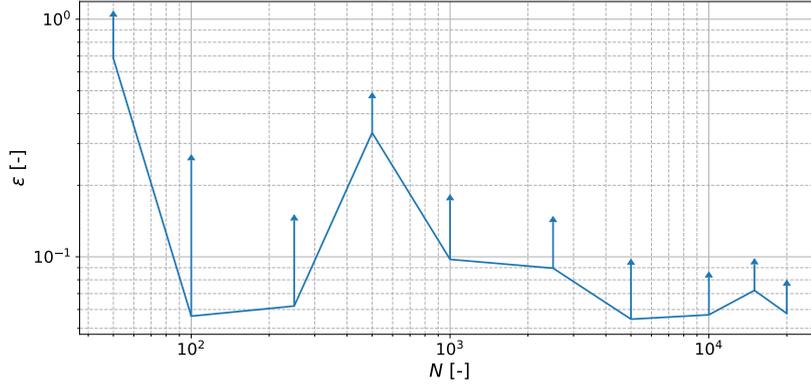


Figure 4.2: Relative error $\varepsilon = |MI_{true}(X; Y) - MI_{KDE}(X; Y)| / MI_{true}(X; Y)$ for sample size N . Vertical lines indicate standard deviation in computed ε .

KDE and k NN methods were compared for time series from nonlinear dynamical systems, and k NN outperformed the other two [57] because it behaved most stable and was less affected by method-specific parameters.

The methodology relies on non-parametric differential entropy estimations from k -nearest neighbor distances assuming the probability density within the distance to the k th neighbor remains constant [44]. As mentioned earlier, mutual information is composed from differential entropy terms as

$$MI(X; Y) = h(X) + h(Y) - h(X, Y). \quad (4.11)$$

The differential entropy can be estimated by averaging local contributions to the differential entropy in the neighborhood of each data point. The differential entropy estimator is

$$\hat{h}(X) = -\frac{1}{N} \sum_{i=1}^N \log p(x_i). \quad (4.12)$$

Consider $\mu_i(\varepsilon)$ as the probability mass of a ball with radius $\varepsilon(i)/2$ (to the k th nearest neighbor) centered at x_i , then by assuming that the probability density $p(x_i)$ inside the ball is constant, the $p(x_i)$ can be estimated by

$$p_i(x_i) \approx \mu_i(\varepsilon) / (c_d \varepsilon^d) \quad (4.13)$$

where d is the dimension of x and c_d the volume of the d -dimensional unit ball. Kozachenko and Leonenko [43] showed that the expectation of $\log \mu_i(\varepsilon)$ can be computed as

$$E[\log \mu_i] = \psi(k) - \psi(N) \quad (4.14)$$

where ψ is the digamma function, the logarithmic derivative of the gamma function $\psi(x) = \frac{d}{dx} \log((n-1)!)$. By combining (4.13) and (4.14), one obtains an estimate for the logarithmic of the local density as

$$\log p(x_i) \approx \psi(k) - \psi(N) - dE[\log \varepsilon] - \log c_d \quad (4.15)$$

which then leads to the estimate of the differential entropy

$$\hat{h}(X) = -\psi(k) + \psi(N) + \log c_d + \frac{d}{N} \sum_{i=1}^n \log \varepsilon(i) \quad (4.16)$$

and is easily extended to joint differential entropy

$$\hat{h}(X, Y) = -\psi(k) + \psi(N) + \log c_{d_x} c_{d_y} + \frac{d_X + d_Y}{N} \sum_{i=1}^n \log \varepsilon(i). \quad (4.17)$$

The estimate of mutual information can be obtained by inserting (4.16) and (4.17) into (4.11). However, this will result in the use of different distance scales because the distance to the k th nearest neighbor in joint

space is larger than in marginal space. To avoid this, k in (4.16) is replaced by $n_x(i)$, the number of points in X within $x_i \pm \varepsilon(i)/2$, then $\varepsilon(i)/2$ is the distance to the $[n_x(i) + 1]$ st neighbor of x_i .

Consider the points depicted in Figure 4.3 where $k = 1$ and where the maximum norm is used, $\varepsilon(i)/2$ is the distance to the nearest neighbor. There are five points within distance $x = x_i \pm \varepsilon/2$ and three within $y = y_i \pm \varepsilon/2$, resulting in $n_x(i) = 5$ and $n_y(i) = 3$.

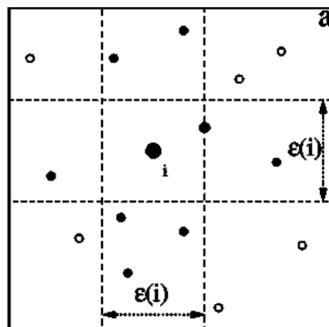


Figure 4.3: Nearest data points around point x_i . Maximum norm distance to first neighbor equals $\varepsilon(i)/2$. Number of data points between the two vertical dashed lines equals n_x while number of points between the two horizontal dashed lines equals n_y . From Kraskov et al. [44].

The mutual information can then be estimated by [85]

$$MI(X; Y) = \psi(k) + \psi(N) - \frac{1}{N} \sum_{i=1}^N (\psi(n_x(i) + 1) + \psi(n_y + 1)). \quad (4.18)$$

The two assumptions made in this approach are that the entropy is estimated by N realizations through (4.12) and that within distance $\varepsilon(i)/2$ around x_i the probability density remains constant. By increasing the number k , the bias introduced by the second assumption becomes more extant, although the estimator variance is reduced.

Just as for the KDE described in Section 4.2.1, the error with respect to the exact mutual information can be computed for two Gaussian distributed variables with zero mean, unit variance and covariance $r = 0.6$. The estimator consists of a single parameter namely the number of nearest neighbors considered, k . By setting k to a large value, the variance in the estimate is reduced while a bias is introduced, small values of k lead to high variance but low bias. Kraskov et al. [44] proposed to set k between 2-4, therefore k is set to 3. The average relative error $\varepsilon = |MI_{true}(X; Y) - MI_{kNN}(X; Y)| / MI_{true}(X; Y)$ is depicted in Figure 4.4 for $k = 3$ and $k = 5$, where the vertical lines indicate the standard deviation based on 250 repeated runs for each N . The difference in error for the two k values is that $k = 3$ results in (on average) more accurate results while it has a slightly higher variance compared to $k = 5$. This matches the statements from Ver Steeg and Galstyan [86]. Overall, there is a clear downward trend in the averaged error and standard deviation for increasing N . For large N , the average error is around a few tenth of a percentage. The standard deviation in the error is around 1%.

Estimating mutual information for $N = 2 \cdot 10^4$ takes around 0.17 seconds, which is roughly two orders of magnitude quicker than the KDE approach. However, the computational costs also scale with $\mathcal{O}(N^2)$ because for each sample the distance to all other samples has to be computed. Because the estimated mutual information is the average over N samples, the computational costs scale with N^2 . Besides the relatively low computational costs, the kNN estimates are more accurate and show less variation compared to the KDE approach. Finally, setting the k parameter is more straightforward than defining the optimal bandwidth for the KDE approach.

All together, the kNN method is found to be more suitable for further applications compared to the KDE approach. In the remaining of this research, the features and correction terms are first standardized before the mutual information is estimated. Standardization is applied because the estimator is sensitive to coordinate transformations due to the probability density uniformity assumption [10]. Theoretically, mutual information is invariant to coordinate transformations, since

$$p(x, y) dx dy = p(x', y') dx' dy', \quad (4.19)$$

and then $MI(X, Y) = MI(X', Y')$. However due to the uniformity assumption, this invariance does not hold for mutual information estimators.

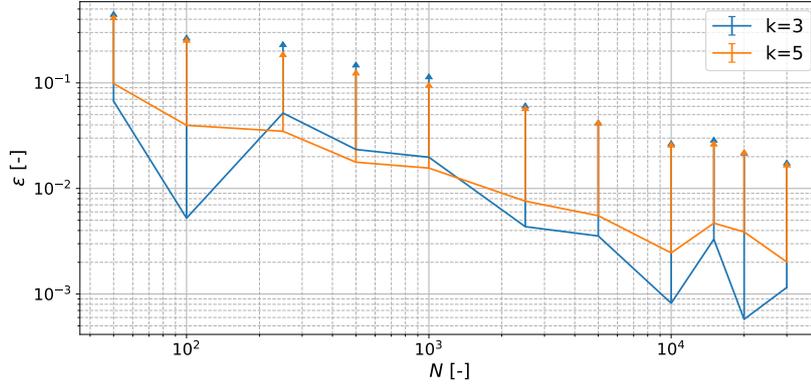


Figure 4.4: Relative error $\varepsilon = |MI_{true}(X; Y) - MI_{kNN}(X; Y)| / MI_{true}(X; Y)$ for sample size N . Vertical lines indicate standard deviation in computed ε .

4.2.2. Parameter dependency

The estimate of mutual information depends on the sample size and the number of nearest neighbors k used in the estimator. To test the convergence of the mutual information estimation for increasing number of samples, the mutual information between the first invariant λ_1 and correction term R ($MI(\lambda_1, R)$) is computed for increasing sample sizes which are drawn randomly from the complete data set. For each N , ten repeated runs are performed using randomly picked samples. The estimates are shown in Figure 4.5 and after roughly $2 \cdot 10^5$ samples, the mean error has decreased to 0.1% and the standard deviation of the error to 1.0%.

The error depicted is the relative difference to the mutual information computed over the entire data set.

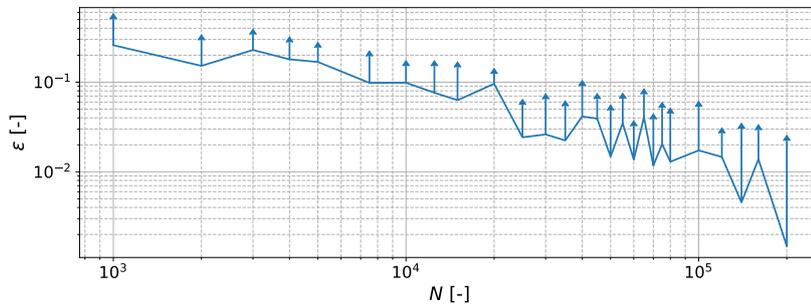


Figure 4.5: Relative error between mutual information between first invariant λ_1 and correction term R for increasing number of samples used in the estimator. Exact mutual information is computed by using the entire data set.

To define the dependence of the estimator on the hyperparameter k , the estimate $MI(\lambda_1, R)$ is plotted for a varying k in Figure 4.6. Clearly, the value of the amount of chosen nearest neighbors influences the mutual information estimate. It indeed seems that for increasing k the bias increases. Kraskov et al. [44] advised to use k between 2-4 to minimize the bias, therefore k is set to 3 in the remaining of this work.

4.2.3. Measure of General Dependence

The advantage of mutual information is that it does not rely on specific assumptions such as linearity or monotonicity and can therefore detect general dependencies between variables. To investigate its abilities, the mutual information is estimated between different functional relations and for different noise levels. In Figure 4.7, four functional relations are depicted (linear, exponential, logarithmic and sine wave) with the corresponding estimated mutual information. Clearly, this metric can observe the dependencies between variable X and Y . It is not dependent on linearity as it can detect the logarithmic relation. On top of that,

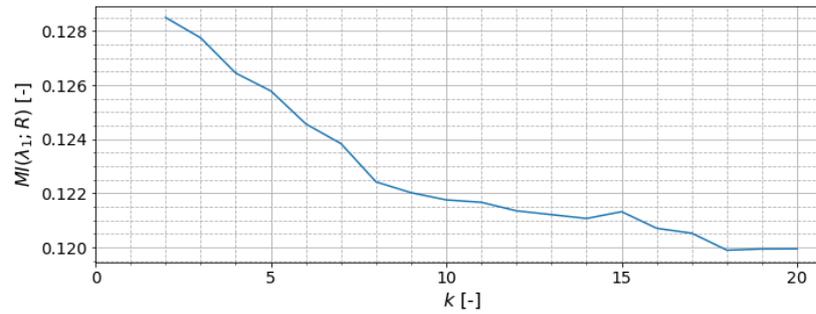


Figure 4.6: Mutual information between first invariant λ_1 and correction term R for increasing number of nearest neighbors k used in the estimator.

it is not dependent on monotonicity as it identifies the dependency between variables X and Y when the functional relation is a sine wave.

Theoretically, mutual information goes to infinity for perfectly correlated variables. However the mutual information estimate is based on two assumptions (entropy is estimated by N samples and within distance $\varepsilon(i)/2$ around x_i the local probability density remains uniform), making the estimate finite for $Y = X$. Gao et al. [25] stated that very strong relationships between variables are more difficult to measure accurately, as the local uniformity assumption is less valid. To compensate this, more samples are required to get an accurate estimate. However, for perfectly correlated variables this means that $N \rightarrow \infty$.

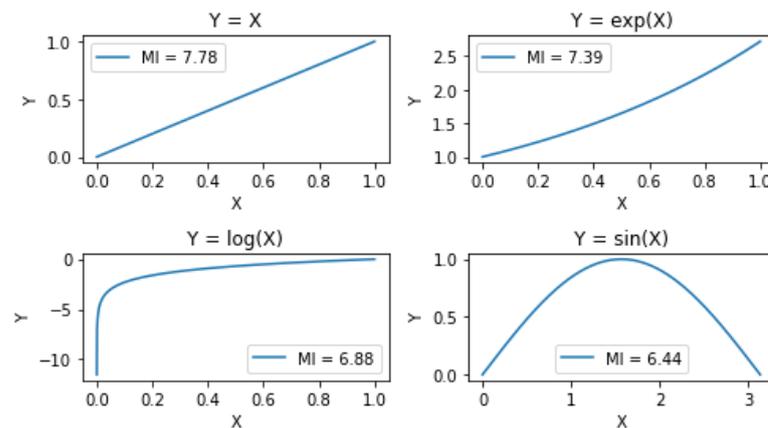


Figure 4.7: Four functional relations between variables X and Y , the functional relation is denoted in the plot title and the estimated mutual information is depicted in the plot label.

In Figure 4.8, the effects of a noise factor η on the mutual information estimate between two variables is denoted, both variables are related through $Y = f(X) + \eta$. For an increasing noise level, the *shared information* between the variables decreases and so also the mutual information.

4.2.4. Physics-informed feature selection

Behaviour of features can be quantified based on metrics such as mutual information, but also by gaining knowledge about the physical conditions. For example, in simple shear flow [3] where the only non-zero velocity gradient is $\frac{\partial U}{\partial z}$, the set of ten basis tensors and five invariants, as listed in (2.17) and (2.19), respectively, reduces to eight non-zero basis tensors and three non-zero invariants, which also have a high mutual dependency [79].

By examining the dominant velocity gradients in the cases considered in this work, similar reasoning can be applied. Outside of the turbine wake, the flow is dominated by the vertical shear $\frac{\partial U}{\partial z}$ in the boundary layer. Inside the turbine's wake, the $\frac{\partial U}{\partial z}$ remains the largest gradient, however also the stream-wise and transversely gradients of the U -velocity component grow. Assuming the only non-zero velocity gradients are $\frac{\partial U}{\partial z}$ and $\frac{\partial U}{\partial y}$, the definitions of the invariants and basis tensors can be simplified significantly. Although this assumption

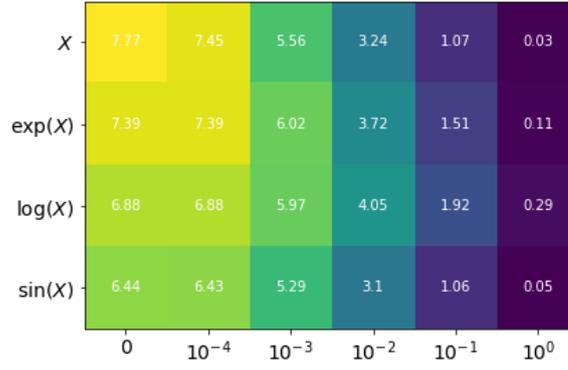


Figure 4.8: Mutual information between variables X and Y through the relation: $Y = f(X) + \eta$ for increasing noise level η (on the horizontal axis).

holds for the the larger part of the domain, including large portion of the turbine's wake, there are regions where it fails. These regions have to be localised in order to get an understanding whether the assumption is applicable or not. Reasons for this assumption to fail are

- Shear gradient $\frac{\partial U}{\partial z}$ reduces due to turbulent mixing in the wake
- Gradient $\frac{\partial U}{\partial x}$ becomes large in near wake region
- Near turbine rotors, where gradients of V and W become large.

The identification of the regions where the assumption is invalid are discussed in Section 6.1. The assumption is referred to as the velocity-gradient assumption.

In Figure 4.9, two box-plots are depicted that quantify the spread of the ratio of $\frac{\partial U}{\partial z}$ and $\frac{\partial U}{\partial y}$ to the other seven gradients. The data is split in two partitions based on the velocity deficit U_d , the local streamwise velocity minus the free-stream velocity

$$U_d(x, y, z) = U_{loc}(x, y, z) - U_\infty(z). \quad (4.20)$$

If U_d is smaller than -0.05 or when $\Pi_{b^\Delta} > 0.025$, it is part of the turbine wake. Π_{b^Δ} is the second invariant of b^Δ and is a quantitative measure of the turbulent anisotropy in the flow, elaborated in Chapter 5. The box-plots represent the spread of the data, the median and mean are denoted by the orange and green line, respectively. The first and third quartile are the upper and lower edges of the box and data that is outside the percentile range of $[5, 95]$ is explicitly shown. The two partitions represent data located in the turbine wake and data located outside the turbine wake (referred to as free-stream).

From analyzing the box-plots, it is clear that the velocity-gradient assumption is valid in large regions of both the wake and free-stream as the boxes are located well above 1. It should be noted that not only for the wake, but also for the free-stream the assumption tends to fail for a portion of the region as the ratio is less than 4 for 5% of data. Further details will be given in Section 6.1.

4.3. Sparse Regression of Turbulent Stress Anisotropy

To enhance the turbulence models, data-driven expressions for R and b^Δ are learned through the SpARTa algorithm, which was developed by Schmelzer et al. [67]. The main advantages of this approach are that a sparse algebraic model representation is obtained that can easily be coupled to a CFD solver, which is normally not straightforward for other black-box algorithms. Besides that, because it is an algebraic expression, the model is easily interpretable which can help in understanding why the model is formed the way it is. Downsides are that it can incorporate only a limited set of features and pre-defined non-linear effects and that due to its simplicity it is not able to capture all complex flow details [69].

4.3.1. Physical form b^Δ and R

By applying the effective-viscosity hypothesis by Pope [58], the anisotropy correction b^Δ can be represented as

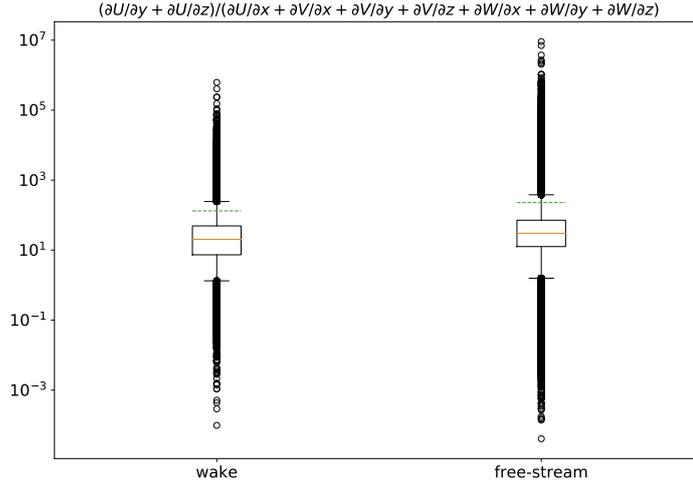


Figure 4.9: Box-plots of $(\frac{\partial U}{\partial y} + \frac{\partial U}{\partial z})/(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial V}{\partial z} + \frac{\partial W}{\partial x} + \frac{\partial W}{\partial y} + \frac{\partial W}{\partial z})$ for data partitioned in wake region and free-stream region.

$$\mathbf{b}^\Delta = \sum_{n=1}^{10} \alpha(\lambda_1, \dots, \lambda_5) \mathbf{T}^{(n)}. \quad (4.21)$$

This model-form is slightly extended such that it also incorporates other scalar features besides the invariants λ_i

$$\mathbf{b}^\Delta = \sum_{n=1}^{10} \alpha(\mathbf{f}) \mathbf{T}^{(n)} \quad (4.22)$$

where \mathbf{f} consists of all scalar features discussed in this work. The correction to the transport equations R is then formed as a turbulent production term

$$R = 2k \sum_{n=1}^{10} \alpha(\mathbf{f}) \mathbf{T}^{(n)} \partial_j U_i, \quad (4.23)$$

as was proposed by Schmelzer et al. [69]. Although this is a valid model-form for the discrepancy of turbulent production, it is not legitimate for turbulent dissipation.

In wind turbine applications, the discrepancy in the transport equations is mainly caused by the turbine-flow interaction because the turbine interaction terms are neglected. Réthoré [64] stated the complete form of the k -equation as

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = \mathcal{P}_k - \varepsilon + \overline{F}_k + \overline{\Pi}_k + \nu \nabla^2 k - \frac{1}{2} \frac{\partial \overline{u'_j u'_i u'_i}}{\partial x_j} \quad (4.24)$$

in which \overline{F}_k is the force-velocity term and $\overline{\Pi}_k$ the pressure-strain term, which are affected by the turbine-flow interaction. These two terms are neglected in the transport equations in the k - ε turbulence model although wind turbines do have an effect on the turbulent kinetic energy in the flow. Réthoré [64] found that the two terms combined represent a turbulence source/sink term and can be up to two orders of magnitude larger than the local dissipation rate ε . This means the k -equation in the k - ε model lacks a source/sink term caused by the presence of the turbine because the pressure-strain and force-velocity terms are neglected. Sanz [66] proposed to add a source/sink term to the transport equations proportional to k in the k -equation and ε in the ε -equation to take into account the effect of wakes due to vegetation.

In this work it is proposed to represent the R correction as a combination of turbulent production $R_{\mathcal{P}}$ and an additional source/sink term R_ε (modeled proportional to the local dissipation rate ε) which is incorporated in the model equations as

$$\frac{Dk}{Dt} = \mathcal{P}_k + R_{\mathcal{P}} + R_\varepsilon - \varepsilon + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \quad (4.25)$$

$$\frac{D\varepsilon}{Dt} = [C_{\varepsilon 1} (\mathcal{P}_k + R_{\mathcal{P}} + R_{\varepsilon}) - C_{\varepsilon 2} \varepsilon] \cdot \frac{\varepsilon}{k} + \frac{\partial}{\partial x_j} \left[\left(v + \frac{v_t}{\sigma_{\varepsilon}} \right) \frac{\partial \varepsilon}{\partial x_j} \right], \quad (4.26)$$

where the R correction is modeled as

$$R = R_{\mathcal{P}} + R_{\varepsilon} = 2k \sum_{n=1}^{10} \alpha(\mathbf{f}) \mathbf{T}^{(n)} \partial_j U_i + \varepsilon(\alpha(\mathbf{f})). \quad (4.27)$$

The objective of the data-driven approach is to define the functional form of both \mathbf{b}^{Δ} and R .

4.3.2. Constructing library of candidate functions

SpaRTA first builds a library of candidate functions \mathbf{x} which is subsequently used to regress the data. The candidates are non-linear combinations of pre-defined features and pre-defined exponents and constructed through the following steps

1. Combine all pre-defined features (e.g. λ_1, λ_2) with all pre-defined exponents (e.g. 1.0): \mathcal{B}_1
2. Make interacting features: multiply features from \mathcal{B}_1 with other features $\mathcal{B}_1: \mathcal{B}_2$ (e.g. $\lambda_1 \lambda_2$)
3. Multiply features from \mathcal{B}_1 with features $\mathcal{B}_2: \mathcal{B}_3$ (e.g. $\lambda_1^2 \lambda_2, \lambda_1 \lambda_2^2$)
4. Library of non-linear features $\alpha(\mathbf{f}) = \mathcal{B}_1 + \mathcal{B}_2 + \mathcal{B}_3 + c$ (c is a constant and duplicates are removed)
5. Library of candidate functions \mathbf{x} obtained by inserting $\alpha(\mathbf{f})$ in (4.21) for \mathbf{b}^{Δ} and in (4.27) for R .

It should be noted that in step 2 not all possible interactions between the features from \mathcal{B}_1 are constructed. This is elaborated in Appendix A.

Given the data for \mathbf{b}^{Δ} or R as \mathbf{y} , then the regression coefficients \mathbf{w} can be found through

$$\mathbf{y} = \mathbf{w}^T \mathbf{x}. \quad (4.28)$$

4.3.3. Model selection using elastic net regression

Because including all regression coefficients in the correction models would lead to overly complex models, which are prone to overfitting and increase the numerical stiffness [67], regularization parameters are included to reduce the magnitude and number of non-zero coefficients. The elastic net regularization is used which searches the space of possible \mathbf{w} by minimizing the following argument

$$\mathbf{w} = \underset{\hat{\mathbf{w}}}{\operatorname{argmin}} \left[\left\| \hat{\mathbf{w}}^T \mathbf{x} - \mathbf{y} \right\|_2^2 + \lambda \rho \left\| \hat{\mathbf{w}} \right\|_1 + 0.5 \lambda (1 - \rho) \left\| \hat{\mathbf{w}} \right\|_2^2 \right] \quad (4.29)$$

where λ is the regularization weight and ρ the mixing parameter defining the ratio between l_1 and l_2 regularization. The second term on the right-hand side of (4.29) defines the l_1 regularization, which allows only a limited amount of non-zero coefficients while setting the others to zero. The third term on the right-hand side promotes l_2 regularization which penalizes large coefficients. The first term tries to minimize the distance between the training output and regression fit, however the \mathbf{w} that minimizes the sum of all three terms is considered as most appropriate.

Regularization is applied in the context of machine learning to increase the model's predictive performance and to avoid overfitting. Overfitting occurs when a model has a high accuracy on the data upon which it has been trained, while it has a low accuracy on unseen test data. This implies that the model tried to fit the training data better than necessary with the downside that it loses its predictive capabilities. To avoid this, the complexity and flexibility of the model can be reduced by introducing regularization parameters such that the model only distinguishes the most obvious trends in the training data that probably will also occur in unseen test data. Regularization is used to balance the variance and bias of the model. More regularization reduces the model's variance as the regression coefficients are shrunk, while simultaneously a bias is introduced. By using a regularization parameter that is too large, the introduced bias will be so significant that the model will underfit, meaning it is not able to recognize any important relation at all. To produce accurate models, it is therefore important to test for a large range of values for (λ_i, ρ_i) .

To test the predictive performance of the trained models, the cross-validation strategy is applied. In this strategy, the models are trained upon a part of the data, while tested on a different part. In this work, the

procedure of K -fold cross-validation is used, in which the total data set is split into K folds. The models are trained on $K - 1$ folds and tested on the remaining unused fold. Because the database consists of three wind turbine cases (as will be elaborated in Chapter 5), 3-Fold cross-validation is applied where the model is trained on two cases and tested on the third as depicted by Figure 4.10.

A large range of values is used for (λ_i, ρ_i) , resulting in the discovery of many different regression models. Only the models with unique non-zero coefficient entries are forwarded to the model inference phase, as the specific coefficient values will be determined in the next phase.

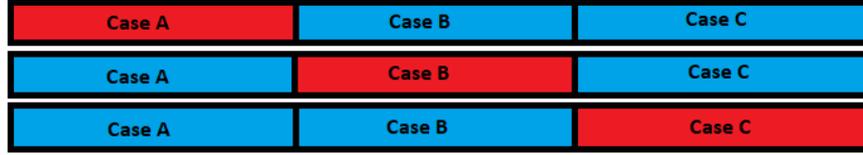


Figure 4.10: 3-Fold Cross-Validation based on the three wind turbine LES cases. Red indicating test sets while blue indicates training sets.

4.3.4. Model inference

The elastic net procedure described above is using standardized candidate functions to ensure that the magnitude of all features is in a similar range. Standardized functions are used because functions with large magnitude will generally attribute more to the model than functions with small magnitude, although this does not mean these functions are more important.

To define models that are not built upon standardized functions but rather on functions with their original magnitudes, an additional regression step is introduced in the model inference phase. For each model, only the subset of non-zero candidate functions \mathbf{x}_u obtained from the previous regression is used, unstandardized. The subscript u denotes that unstandardized functions are used. To improve the possibility of convergence in a CFD solver, l_2 regularization is introduced by the parameter λ_r

$$\mathbf{w}_u = \underset{\hat{\mathbf{w}}_{nz}}{\operatorname{argmin}} \left[\left\| (\hat{\mathbf{w}}_{nz})^T \mathbf{x}_u - \mathbf{y} \right\|_2^2 + \lambda_r \left\| \hat{\mathbf{w}}_{nz} \right\|_2^2 \right]. \quad (4.30)$$

The final model-form is obtained by the dot product between the resulting set of coefficients \mathbf{w}_u and the candidate functions through

$$M = (\mathbf{w}_u)^T \mathbf{x}_u. \quad (4.31)$$

4.4. Sparse logistic regression

The advantage of the SpaRTA algorithm is that it is able to construct simple, algebraic model expressions for the correction terms. Ideally, these expressions generalize to all regions in the domain, e.g. they fit the trends in the correction terms in both the free-stream and wake region. However, in practice it is difficult to have an algebraic model that fulfills these characteristics and likely gives inaccurate predictions inside the wake region. By making the assumption that the baseline turbulence model is able to simulate the free-stream flow accurately, correction models can be trained specifically for the wake region. As a consequence, the models will not fit the trends in the free-stream flow (which are mostly close to zero).

To solve this problem, a classifier can be used that *turns on* the correction models in the turbine's wake. This gives the opportunity to construct correction models trained solely on the data in the wake region. On top of that, the model does not have to fit the data in regions where the baseline is sufficient (free-stream), because there the classifier will turn the correction model *off*, resulting in more flexibility of the model. Another advantage of using a classifier is that the data set for the correction model is significantly reduced. This decreases the training costs of the model, meaning the library of candidate functions can be extended. The classifier will be constructed such that it is turned *on* in the wake of the turbines and *off* in the remaining of the domain. The wake is defined as the region where

$$U_d(x, y, z) = U_{loc}(x, y, z) - U_\infty(z) \quad (4.32)$$

is smaller than -0.05 . Because it is found that there is a small portion of data outside the near-wake where Π_{b^Δ} is large (for example, caused by speed-up effects at the wake edges), samples where $\Pi_{b^\Delta} > 0.025$ are also included in the wake definition.

The wake classifier is constructed by extending SpaRTA to classification problems. Instead of using standard regression, logistic regression is applied. Logistic regression is used to model the probability of a class, either true or false, which in this case means that a sample is either in the turbine wake or in the free-stream. Instead of the regular regression expression in (4.31), the sigmoid function σ is applied

$$\mathbf{y}_{\text{pred}} = \sigma(\mathbf{w}^T \mathbf{x} + c) \quad (4.33)$$

as an activation function, where c is a constant. The sigmoid function has a value between 0 and 1 and an S-shaped transition in between, as depicted in Figure 4.11. This ensures a smooth and not discrete transition between the class labels.

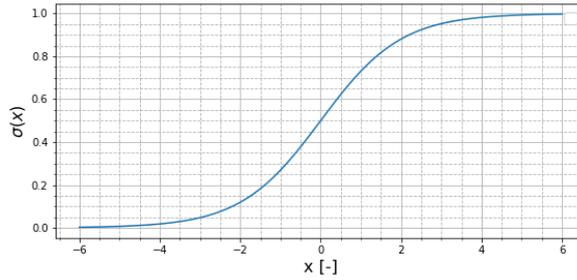


Figure 4.11: Sigmoid function.

The sigmoid function relates the input and output through

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \quad (4.34)$$

The model coefficients are found by maximizing the log-likelihood

$$\log L(\mathbf{x}, \mathbf{w}, c, \mathbf{y}_{\text{true}}) = \sum_{i=1}^N l_i \quad (4.35)$$

where l_i is defined as

$$l_i = y_{i,\text{true}} \cdot \log\left(\frac{1}{1 + \exp(\mathbf{w}^T \mathbf{x}_i + c)}\right) + (1 - y_{i,\text{true}}) \cdot \log\left(\frac{1}{1 + \exp(\mathbf{w}^T \mathbf{x}_i + c)}\right) = -\log\left(\exp(-y_{i,\text{true}}(\mathbf{w}^T \mathbf{x}_i + c)) + 1\right). \quad (4.36)$$

Both l_1 and l_2 regularization can be implemented, l_1 shrinks lesser important coefficients to zero, while l_2 penalizes large coefficients.

The first step is model discovery in which the logistic regression coefficients are found by minimizing the following cost function, which includes the regularization parameters

$$\mathbf{w}, c = \underset{\mathbf{w}, c}{\text{argmin}} \left[\frac{1-\rho}{2} \mathbf{w}^T \mathbf{w} + \rho \|\mathbf{w}\|_1 + C \sum_{i=1}^n \log(\exp(-y_i(\mathbf{x}_i^T \mathbf{w} + c)) + 1) \right]. \quad (4.37)$$

The third term on the right-hand side is the negative log-likelihood (it is a minimization problem) and the first two terms define the regularization. The parameter C is the inverse of the regularization strength, smaller values imply adding more regularization. The mixing parameter ρ defines the ratio of l_2 to l_1 regularization. If $\rho = 1$, only l_1 regularization is applied while only l_2 regularization is applied when $\rho = 0$. For $0 < \rho < 1$, the regularization is a combination of l_1 and l_2 . By applying different values for (C_i, ρ_i) , a large range of models can be discovered.

The \mathbf{w} and c obtained through (4.37) are based on standardized functions. To obtain models with unstandardized functions, the remaining non-zero functions from (4.37) can be exposed to a second regression fit with only l_2 regularization through

$$\mathbf{w}_u, c_u = \underset{\mathbf{w}_{nz}, c}{\operatorname{argmin}} \left[\mathbf{w}_{nz}^T \mathbf{w}_{nz} + C \sum_{i=1}^n \log(\exp(-y_i(\mathbf{x}_{u,i}^T \mathbf{w}_{nz} + c)) + 1) \right] \quad (4.38)$$

where \mathbf{w}_{nz} are the non-zero coefficients from \mathbf{w} in (4.37), \mathbf{w}_u and c_u are the coefficients and intercept based on the unstandardized functions \mathbf{x}_u , respectively. This final step is the model inference step.

4.4.1. Performance Evaluation

The performance of the classification model can be evaluated using different metrics, the ones discussed and used in this research are the log (logarithmic) loss, accuracy, recall and complexity.

The log loss is a loss function based on the probabilistic predictions made by the model. It is defined as

$$L_{\log}(Y, P) = -\frac{1}{N} \sum_{i=1}^N (y_i \log p_i + (1 - y_i) \log(1 - p_i)) \quad (4.39)$$

and grows when the estimates deviate from the true value. Its advantage is that it does not take into account the discrete prediction from the model, but its probabilistic prediction. In this work it is desired that the model returns either values close to 0.0 or 1.0, meaning predictions that deviate from these extremes should be penalized more than usual.

The accuracy metric defines the fraction of correct predictions and can be computed by

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(y_{\text{pred},i} = y_{\text{true},i}) \quad (4.40)$$

where true positive (TP) is the number of correct predictions in the turbine wake, true negative (TN) the number of correct predictions in the free-stream, false positive (FP) the incorrect prediction in the free-stream, false negative (FN) the incorrect predictions in the wake. The accuracy is a concise number that depicts the performance of the model, however can be misleading for imbalanced data. When combining the three data sets, the wake represents 14% of all data points, meaning that a model that always predicts an outcome of 0 will have an accuracy of 86%.

The recall is a convenient metric to evaluate the ability of a model to identify all positive samples e.g. how well it can predict if a sample is in the turbine wake. It is defined as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.41)$$

and returns the fraction of correctly predicted samples in the wake over the total number of samples in the wake. The advantage is that, for imbalanced data, it gives a clear picture of the model's ability in correctly predicting the under-sampled class. The disadvantage is that it does not take into account the predictions in the region outside the wake. This means that when a model always gives 1 as a prediction, the recall also equals 1.

The final metric discussed is the model complexity, defined as the remaining number of non-zero coefficients. The complexity is important and should preferably be minimized to increase the interpretability of the model. On top of that, very complex models tend to increase the numerical stiffness in a CFD solver and impede convergence [68].

5

Data Cases and Features

This chapter starts with describing the different data sets used in this research, in Section 5.1. Subsequently, the investigated features are outlined in Section 5.2.

5.1. Data Cases

In this research, three LES data sets on wind-tunnel scale are used, set up by Steiner et al. [74]. The datasets are based on wind tunnel experiments from Chamorro and Porté-Agel [12] and the results are validated against measurements from this particular experiment. The case parameters are denoted in Table 5.1 and are equal for each dataset. In the simulations, the turbine rotational speed Ω is kept constant, this implies that the thrust coefficient can vary and therefore the listed quantity is the time average of a turbine located in undisturbed conditions. The instantaneous quantities are time-averaged for further processing.

Table 5.1: LES case [74] parameters

Parameter	
Reynolds number	$Re = 63,800$
Diameter	$D = 0.15$ m
Hub-height	$h_{hub} = 0.125$ m
Rotation speed	$\Omega = 1190$ rpm
Thrust coefficient	$C_T = 0.51$
Free-stream velocity	$U_{hub} = 2.2$ m/s
Turbulence intensity	$\sigma_U / U(h_{hub}) = 9\%$
Domain size	$5.4 \times 1.8 \times 0.46$ m ³
Resolution	$360 \times 120 \times 64$

The turbine positioning in each simulation is depicted in Figure 5.1. Case A consists of two turbines in series, while case B and C have two turbines parallel downstream of another turbine. All turbines are aligned with the free-stream velocity, only for case C the upstream turbine has a yaw angle of 15° . The stream-wise spacing between the upstream and downstream turbines is 0.75 m, which equals five rotor diameters.

For the simulations, OpenFOAM-6.0 was used together with the SOWFA-6 toolbox [16]. The solver is based on the incompressible finite-volume formulation. To model the unresolved scales, the WALE (Wall-adapting Local Eddy-viscosity) sub-grid scale model was used [62] to ensure correct wall-asymptotic scaling of the eddy viscosity. To represent the forcing of the turbines, SOWFA's actuator disk model is used.

The simulations mimic wind-tunnel experiments which do not fully represent the physical behaviour of full-scale wind turbine applications. Wind-tunnels operate at a significantly lower Reynolds number and the turbulent length scales are also noteworthy smaller than in the planetary atmospheric boundary layer. This affects the turbulence properties and therefore also the wake characteristics [11]. However, the goal of this

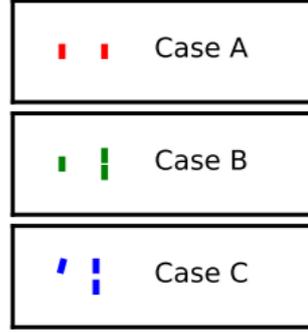


Figure 5.1: Wind turbine positioning for each LES case, turbine diameter is on scale. From Steiner et al. [74].

research is to investigate the ability to construct data-driven correction terms for turbulence models applied to wind turbine applications. The approach of enhancing turbulence models with data-driven corrections is still novel and most of the work performed in this field uses either 2D simulations [49, 67, 94] or low Reynolds number cases [53, 92]. Therefore, starting with more elementary cases at wind-tunnel scale can lead to preliminary conclusions about the potential of such enhanced turbulence models, which can act as a framework for further research focused on more practical full-scale cases.

5.2. Features

The investigated features in this work can be divided into four categories

- Invariants based on \mathbf{S} and $\mathbf{\Omega}$ in (2.19) [59]
- Basis tensors in (2.17) [59]
- Invariants when including ∇k and ∇p [94]
- Physical flow features [48, 83, 90].

Because most features are scalars and \mathbf{b}^Δ is a second-order tensor, the scalar features are compared to the second invariant of \mathbf{b}^Δ : $\Pi_{\mathbf{b}^\Delta}$. The second invariant $\Pi_{\mathbf{b}^\Delta}$ is a key scalar metric of \mathbf{b}^Δ because it quantifies the degree of anisotropy [48]. Because \mathbf{b}^Δ is the anisotropy correction in the stress-strain relation, $\Pi_{\mathbf{b}^\Delta}$ is a convenient scalar target to use for comparison to the evaluated features. The second invariant of the tensor \mathbf{b}^Δ can be computed by [21]

$$\Pi_{\mathbf{b}^\Delta} = b_{ij}^\Delta b_{ji}^\Delta / 2 = \gamma_1^2 + \gamma_1 \gamma_2 + \gamma_2^2 \quad (5.1)$$

where $\boldsymbol{\gamma}$ is the vector of eigenvalues of \mathbf{b}^Δ in which $\gamma_1 \geq \gamma_2 \geq \gamma_3$. While the second invariant of \mathbf{b} , $\Pi_{\mathbf{b}}$, is bounded to $[0, \frac{1}{3}]$ as can be seen in Section 2.3, this does not hold for $\Pi_{\mathbf{b}^\Delta}$ because the eigenvalues of \mathbf{b}^Δ cannot be derived from the Reynolds stress eigenvalues ϕ_i because

$$\gamma_i \neq \frac{\phi_i}{2k} - \frac{1}{3}, \quad (5.2)$$

implying that $\Pi_{\mathbf{b}^\Delta}$, contrary to $\Pi_{\mathbf{b}}$, is not bounded to $[0, \frac{1}{3}]$.

The contours of both correction terms are plotted in Figure 5.2 at a slice at turbine location for case A. There are similarities and differences between the two correction terms. Both are most dominant in the upper part of the wake, while the magnitude of R starts to grow gradually downstream of the first turbine, the magnitude of $\Pi_{\mathbf{b}^\Delta}$ is most dominant around the first turbine. This is because large velocity gradients significantly affect the anisotropy in the flow, which are mainly present in the region near the rotor. The correction R grows for increasing turbulence intensity, which grows in magnitude in the near wake. While \mathbf{b}^Δ is dimensionless, as the turbulent anisotropy is normalized by the turbulent kinetic energy, the correction R has units $[\text{m}^2/\text{s}^3]$. Both correction terms are close to zero outside of the turbine wake region.

It must be noted that for the analysis in this research (feature selection and construction of the correction models) the samples with $\sqrt{\overline{k}d}/(50\nu) < 10$ are discarded. This corresponds to roughly the samples below

$(z - z_{\text{hub}})/D < -0.5$. The samples in this region are discarded because near the wall the magnitude of the correction models grows exponentially, which hinders the construction of accurate correction models for the remaining of the domain. In the CFD solver, a blending factor is used to damp the correction models near the wall. This is further elaborated in Section 10.1.

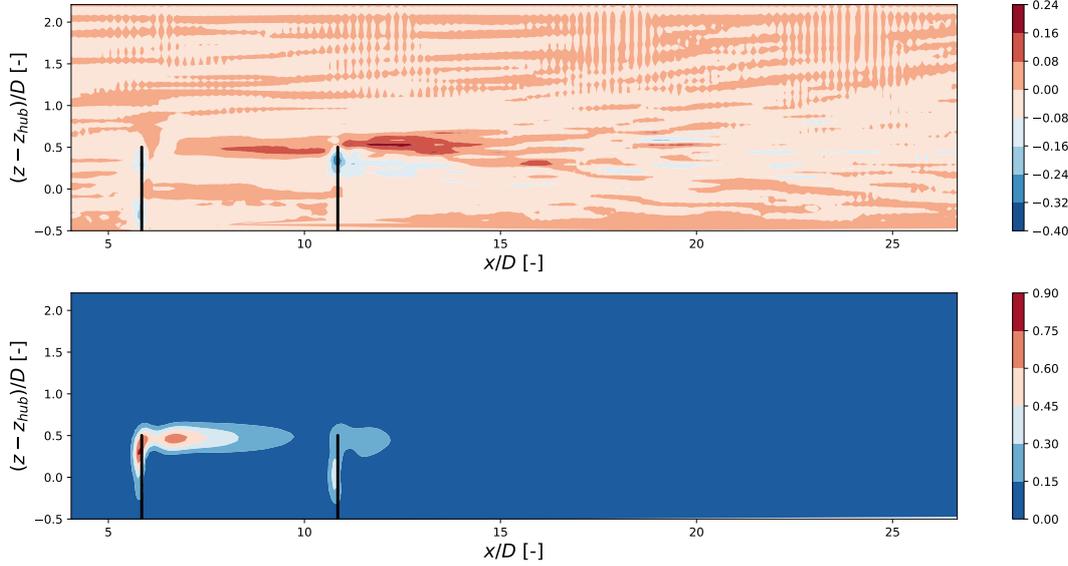


Figure 5.2: Contour plot of R (top figure) and $\Pi_{\mathbf{b}\Delta}$ (bottom figure) at turbine location $((y - y_{\text{hub}})/D = 0)$ for case A. Turbines indicated by black vertical lines.

5.2.1. Feature requirements

The features should obey three requirements, they must be non-dimensional, rotational invariant and Galilean invariant. Non-dimensional implies that the features must be unitless, meaning the units of the obtained algebraic models are consistent with the correction terms. Rotational invariant states that the features are invariant to any arbitrary rotation of the coordinate frame [37, 94], which is hold for scalar features f when

$$f(\mathbf{S}, \mathbf{v}, c) = f(\mathbf{Q}\mathbf{S}\mathbf{Q}^T, \mathbf{Q}\mathbf{v}, c), \quad \forall \mathbf{Q}, \mathbf{S}, \mathbf{v}, c \quad (5.3)$$

and for tensor features \mathbf{h} when

$$\mathbf{Q}\mathbf{h}(\mathbf{S}, \mathbf{v}, c)\mathbf{Q}^T = \mathbf{h}(\mathbf{Q}\mathbf{S}\mathbf{Q}^T, \mathbf{Q}\mathbf{v}, c), \quad \forall \mathbf{Q}, \mathbf{S}, \mathbf{v}, c \quad (5.4)$$

where $\mathbf{S} \in \mathbb{R}^{3 \times 3}$ is a tensor argument, $\mathbf{v} \in \mathbb{R}^3$ a vector argument, $c \in \mathbb{R}$ a scalar argument and $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ an arbitrary orthogonal transformation matrix. Rotational invariance can be enforced by using invariants as features or taking the vector or matrix norm of either vector or tensor variables, respectively.

Galilean invariance implies that, in all frames with constant velocity, the laws of motion are identical. For example, the velocity U_i is not Galilean invariant while its gradient $\partial_j U_i$ is, as it does not depend on a reference value [94]. Although all features considered in this work are non-dimensional and rotational invariant, five physical flow features are not Galilean invariant as they depend on the velocity U_i . It should therefore be taken into account that these features are more prone to a loss of generalizability than the other considered features.

The different features used in this work are described below.

5.2.2. Invariants

The invariants used in the effective-viscosity hypothesis proposed by Pope [59] were already listed in (2.19), but are repeated here for convenience

$$\lambda_1 = \{\mathbf{S}^2\} \quad \lambda_2 = \{\mathbf{\Omega}^2\} \quad \lambda_3 = \{\mathbf{S}^3\} \quad \lambda_4 = \{\mathbf{S}\mathbf{\Omega}^2\} \quad \lambda_5 = \{\mathbf{S}^2\mathbf{\Omega}^2\}. \quad (5.5)$$

5.2.3. Basis Tensors

Similarly the integrity basis, the ten non-linear basis tensors listed in (2.17) are also repeated for convenience

$$\begin{aligned}
\mathbf{T}^{(1)} &= \mathbf{S} & \mathbf{T}^{(6)} &= \boldsymbol{\Omega}^2 \mathbf{S} + \mathbf{S} \boldsymbol{\Omega}^2 - \frac{2}{3} \{\mathbf{S} \boldsymbol{\Omega}^2\} \mathbf{I} \\
\mathbf{T}^{(2)} &= \mathbf{S} \boldsymbol{\Omega} - \boldsymbol{\Omega} \mathbf{S} & \mathbf{T}^{(7)} &= \boldsymbol{\Omega} \mathbf{S} \boldsymbol{\Omega}^2 - \boldsymbol{\Omega}^2 \mathbf{S} \boldsymbol{\Omega} \\
\mathbf{T}^{(3)} &= \mathbf{S}^2 - \frac{1}{3} \{\mathbf{S}^2\} \mathbf{I} & \mathbf{T}^{(8)} &= \mathbf{S} \boldsymbol{\Omega} \mathbf{S}^2 - \mathbf{S}^2 \boldsymbol{\Omega} \mathbf{S} \\
\mathbf{T}^{(4)} &= \boldsymbol{\Omega}^2 - \frac{1}{3} \{\boldsymbol{\Omega}^2\} \mathbf{I} & \mathbf{T}^{(9)} &= \boldsymbol{\Omega}^2 \mathbf{S}^2 + \mathbf{S}^2 \boldsymbol{\Omega}^2 - \frac{2}{3} \{\boldsymbol{\Omega}^2 \mathbf{S}^2\} \mathbf{I} \\
\mathbf{T}^{(5)} &= \boldsymbol{\Omega} \mathbf{S}^2 - \mathbf{S}^2 \boldsymbol{\Omega} & \mathbf{T}^{(10)} &= \boldsymbol{\Omega} \mathbf{S}^2 \boldsymbol{\Omega}^2 - \boldsymbol{\Omega}^2 \mathbf{S}^2 \boldsymbol{\Omega}
\end{aligned} \tag{5.6}$$

5.2.4. Additional Invariants

In the original non-linear eddy viscosity model [59], it is assumed that turbulence is only affected by the strain-rate \mathbf{S} and rotation-rate tensor $\boldsymbol{\Omega}$ and that turbulence production is balanced by dissipation [94]. However, pressure gradients also influence turbulence because a favorable pressure gradient decreases the turbulent kinetic energy and an adverse gradient increases it. On top of that, strong non-equilibrium effects can exist, especially in turbine wake applications where the production of turbulence is not in equilibrium with the dissipation rate. To take these invalid assumptions into account, the pressure gradient ∇p and turbulence kinetic energy gradient ∇k are included in the form of invariants.

The gradients ∇p and ∇k are both Galilean invariant, but not rotational invariant. This can be achieved by constructing invariants based on ∇p and ∇k , which satisfy the rotational invariant condition [94]. To use the scalar gradients, they are normalized and transformed to anti-symmetric tensors

$$\mathbf{A}_k = -\mathbf{I} \times \left(\frac{\sqrt{k}}{\varepsilon} \nabla k \right) \quad \mathbf{A}_p = -\mathbf{I} \times \left(\frac{1}{\rho |D\mathbf{U}/Dt|} \nabla p \right) \tag{5.7}$$

and subsequently used to construct a set of invariants as was done by Wu et al. [94] to ensure the rotational invariance requirement. The invariants are depicted in Table 5.2. There is a finite number of invariants as the listed invariants can represent any polynomial invariant. Such a finite number of invariants is referred to as a minimal integrity basis. The invariants based on ∇p and ∇k are in this work referred to as *additional invariants* to separate them from the invariants in Section 5.2.2.

Table 5.2: Invariant bases, number of symmetric and antisymmetric tensors for each invariant are indicated by n_s and n_A , respectively. The invariant bases are the trace of the tensors listed. The asterisk on an invariant bases indicates that also the cyclic permutation of the antisymmetric tensors are included.

(n_s, n_A)	Feature index	Invariant bases
(0, 1)	1-2	$\mathbf{A}_p^2, \mathbf{A}_k^2$
(1, 1)	3-9	$\boldsymbol{\Omega}^2 \mathbf{S} \boldsymbol{\Omega} \mathbf{S}^2$ $\mathbf{A}_p^2 \mathbf{S}, \mathbf{A}_p^2 \mathbf{S}^2, \mathbf{A}_p^2 \mathbf{S} \mathbf{A}_p \mathbf{S}^2$ $\mathbf{A}_k^2 \mathbf{S}, \mathbf{A}_k^2 \mathbf{S}^2, \mathbf{A}_k^2 \mathbf{S} \mathbf{A}_k \mathbf{S}^2$
(0, 2)	10-12	$\boldsymbol{\Omega} \mathbf{A}_p, \mathbf{A}_p \mathbf{A}_k, \boldsymbol{\Omega} \mathbf{A}_k$
(1, 2)	13-36	$\boldsymbol{\Omega} \mathbf{A}_p \mathbf{S}, \boldsymbol{\Omega} \mathbf{A}_p \mathbf{S}^2, \boldsymbol{\Omega}^2 \mathbf{A}_p \mathbf{S}^*, \boldsymbol{\Omega}^2 \mathbf{A}_p \mathbf{S}^{2*}, \boldsymbol{\Omega}^2 \mathbf{S} \mathbf{A}_p \mathbf{S}^{2*}$ $\boldsymbol{\Omega} \mathbf{A}_k \mathbf{S}, \boldsymbol{\Omega} \mathbf{A}_k \mathbf{S}^2, \boldsymbol{\Omega}^2 \mathbf{A}_k \mathbf{S}^*, \boldsymbol{\Omega}^2 \mathbf{A}_k \mathbf{S}^{2*}, \boldsymbol{\Omega}^2 \mathbf{S} \mathbf{A}_k \mathbf{S}^{2*}$ $\mathbf{A}_p \mathbf{A}_k \mathbf{S}, \mathbf{A}_p \mathbf{A}_k \mathbf{S}^2, \mathbf{A}_p^2 \mathbf{A}_k \mathbf{S}^*, \mathbf{A}_p^2 \mathbf{A}_k \mathbf{S}^{2*}$
(0, 3)	37	$\boldsymbol{\Omega} \mathbf{A}_p \mathbf{A}_k$
(1, 3)	38-42	$\boldsymbol{\Omega} \mathbf{A}_p \mathbf{A}_k \mathbf{S}, \boldsymbol{\Omega} \mathbf{A}_k \mathbf{A}_p \mathbf{S}, \boldsymbol{\Omega} \mathbf{A}_p \mathbf{A}_k \mathbf{S}^2, \boldsymbol{\Omega} \mathbf{A}_k \mathbf{A}_p \mathbf{S}^2, \boldsymbol{\Omega} \mathbf{A}_p \mathbf{S} \mathbf{A}_k \mathbf{S}^2$

5.2.5. Physical Flow Features

Besides the listed basis tensors and invariants, features can be included that are more physically interpretable, listed in Table 5.3. All of the mentioned features comprise physical intuition to relate physical trends to the

correcting terms. All of the listed features, excluding q_6 and q_7 , are taken from Ling and Templeton [48] and Wang et al. [90]. The shear parameter q_7 is used by van der Laan et al. [83] to limit the eddy viscosity within the wind turbine wake. The force coefficient q_6 is included by the author himself to take into account the forcing applied by the turbines.

Table 5.3: Physical flow features. For each feature q_i the physical description is denoted including the raw feature with its normalization. The features that are not Galilean invariant are marked with †. Reynolds stress $u'_i u'_j$ is based only on Boussinesq eddy viscosity assumption.

Feature	Description	Raw feature	Normalization factor
q_1	Wall-distance Reynolds number	$\frac{\sqrt{k}d}{50\nu}$	-
q_2	Turbulence intensity 1	k	$\nu \ \frac{\partial U_i}{\partial x_j}\ $
q_3^\dagger	Turbulence intensity 2	k	$\frac{1}{2} U_i U_i$
q_4	Ratio of excess rotation rate to strain rate (Q criterion)	$\frac{1}{2} (\ \boldsymbol{\Omega}\ ^2 - \ \mathbf{S}\ ^2)$	$\ \mathbf{S}\ ^2$
q_5^\dagger	Pressure gradient along streamline	$U_k \frac{\partial P}{\partial x_k}$	$\sqrt{\frac{\partial P}{\partial x_j} \frac{\partial P}{\partial x_j} U_i U_i}$
q_6^\dagger	Force coefficient	$ F $	$0.001 (\frac{1}{2} / R U_i ^2)$
q_7	Shear parameter	$\ \frac{\partial U_i}{\partial x_j}\ $	$\frac{\varepsilon}{k}$
q_8	Viscosity ratio	ν_t	100ν
q_9^\dagger	Nonorthogonality between velocity and its gradient	$ U_i U_j \frac{\partial U_i}{\partial x_j} $	$\sqrt{U_l U_l U_i \frac{\partial U_i}{\partial x_j} U_k \frac{\partial U_k}{\partial x_j}}$
q_{10}	Ratio of total to normal Reynolds stresses	$\ \overline{u'_i u'_j}\ $	k
q_{11}^\dagger	Ratio of convection to production of TKE	$U_i \frac{dk}{dx_i}$	$ \overline{u'_j u'_k} S_{jk} $

6

Feature Selection

The performance of different features will be analyzed based on the data from case A using mutual information. A broad range of features is considered which were listed in Section 5.2. First the invariants and basis tensors in (2.19) and (2.17) are investigated, followed by the additional invariants based on $\nabla \mathbf{p}$ and $\nabla \mathbf{k}$. Finally, the dependency of the correction terms to the physical flow features is discussed. The goal is to get a better understanding of the relations between the correction terms (R and $\Pi_{\mathbf{b}^\Delta}$) and the features. This ensures that only the most relevant features are used in the remaining parts of this work, increasing the performance of the constructed data-driven models.

6.1. Invariants

The mutual information estimates between the invariants and the correction terms are listed in Figures 6.1 and 6.2.

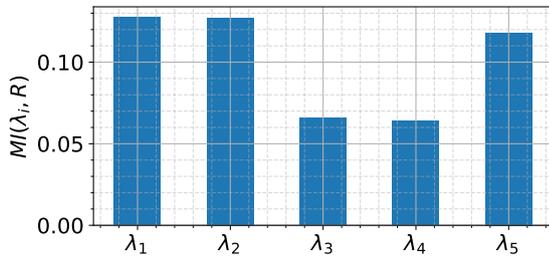


Figure 6.1: Mutual information between invariants and R .

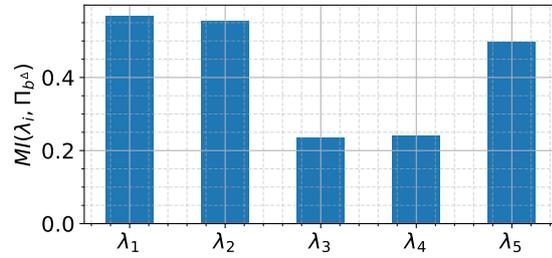


Figure 6.2: Mutual information between invariants and $\Pi_{\mathbf{b}^\Delta}$.

Clearly, three invariants λ_1 , λ_2 and λ_5 have the largest dependency with both correction terms while invariants λ_3 and λ_4 show a smaller dependency. This implies that from the set of five invariants, mainly λ_1 , λ_2 and λ_5 show to be important to R and \mathbf{b}^Δ .

While the first, second and fifth invariant show all a large dependency with the correction terms, it is not clear yet if there is any redundancy among the invariants, indicating that the invariants have a high dependency between themselves. In Figure 6.3, the mutual information between individual invariants is depicted as a heatmap, relations with large mutual dependency are indicated in red and low in blue. It is clearly shown that invariants λ_1 , λ_2 and λ_5 all have a high dependency between themselves, indicating that using all three results in redundancy. The invariants λ_3 and λ_4 , also have a high mutual dependency.

In Section 4.2.4, it was stated that when $\frac{\partial U}{\partial z}$ and $\frac{\partial U}{\partial y}$ are the most dominating gradients, the mathematical forms of the invariants can be significantly simplified. By assuming that $\frac{\partial U}{\partial z}$ and $\frac{\partial U}{\partial y}$ are the only non-zero velocity gradients, the mean rate of strain and rotation tensors reduce to their simplified forms

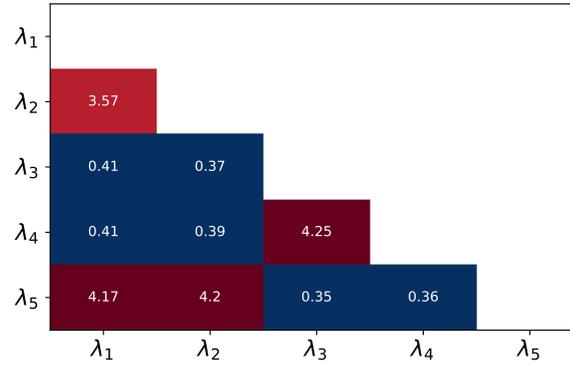


Figure 6.3: Mutual information between individual invariants

$$\mathbf{S} = \frac{1}{2} \frac{k}{\varepsilon} \begin{bmatrix} 0 & \frac{\partial U}{\partial y} & \frac{\partial U}{\partial z} \\ \frac{\partial U}{\partial y} & 0 & 0 \\ \frac{\partial U}{\partial z} & 0 & 0 \end{bmatrix} \quad \mathbf{\Omega} = \frac{1}{2} \frac{k}{\varepsilon} \begin{bmatrix} 0 & \frac{\partial U}{\partial y} & \frac{\partial U}{\partial z} \\ -\frac{\partial U}{\partial y} & 0 & 0 \\ -\frac{\partial U}{\partial z} & 0 & 0 \end{bmatrix},$$

respectively. By using the reduced form of these tensors, the expression of the first invariant λ_1 is simplified to

$$\{\mathbf{S}^2\} = \frac{k}{\varepsilon} \left[\left(\frac{\partial U}{\partial z} \right)^2 + \left(\frac{\partial U}{\partial y} \right)^2 \right]. \quad (6.1)$$

Clearly, λ_1 is a positive scalar quantity and its magnitude grows for increasing gradients. By applying the same procedure to the other invariants, the following relations between the invariants are obtained

$$\lambda_2 = -\lambda_1 \quad \lambda_3 = \lambda_4 = 0 \quad \lambda_5 = -\frac{1}{2} \lambda_1^2. \quad (6.2)$$

The second invariant λ_2 is linearly related to the first invariant while the fifth invariant λ_5 has a quadratic relation to λ_1 . The third and fourth invariant are both equal to zero under the velocity gradient assumption. This conclusion is consistent with the dependency analysis using mutual information.

To investigate if the above-described relations correspond to the data, a linear regression line is fit to $(\lambda_2, -\lambda_1)$ for the data partitioned in wake and free-stream region. The linear fit and the data are depicted in Figure 6.4 with the wake left and free-stream on the right. Although visually the data seems to not fit the linear line very well, the majority of the samples are densely located close to the linear fit resulting in a coefficient of determination R^2 that is close to unity. For the samples in the wake, the linear regression fits the data slightly less than for the free-stream samples.

The coefficient R^2 quantifies how strong the linear relationship is between two variables and equals 1.0 for a perfect fit and shrinks for increasing residuals between the data and the linear line. The coefficient R^2 is computed by

$$R^2 = 1 - \frac{\sum_{i=1}^N (x(i) - f(i))^2}{\sum_{i=1}^N (x(i) - \bar{x})^2} \quad (6.3)$$

where $x(i)$ are the individual data points, \bar{x} the data mean and $f(i)$ the regression value. Because R^2 is not equal to unity, there are a number of samples that differ from the trend and so where the velocity-gradient assumption does not hold, and it is therefore important to identify these regions of deviation.

From the velocity gradient assumption $\lambda_2 = -\lambda_1$. To detect the deviation from this assumption, a new feature is introduced as $\lambda_{1+2} = (|\lambda_1 + \lambda_2|)$ which should equal zero when the assumption is valid and deviates where it fails. A Contour showing λ_{1+2} for a vertical slice at turbine location is depicted in Figure 6.5. There are particular regions where λ_{1+2} is non-zero: in the center of the near wake and near the rotor. It is also noteworthy that λ_{1+2} is close to zero in wake regions not near the rotor axis and further downstream ($(x - x_{T1})/D > 7$), explaining why R^2 in the wake (Figure 6.4) is still reasonable large.

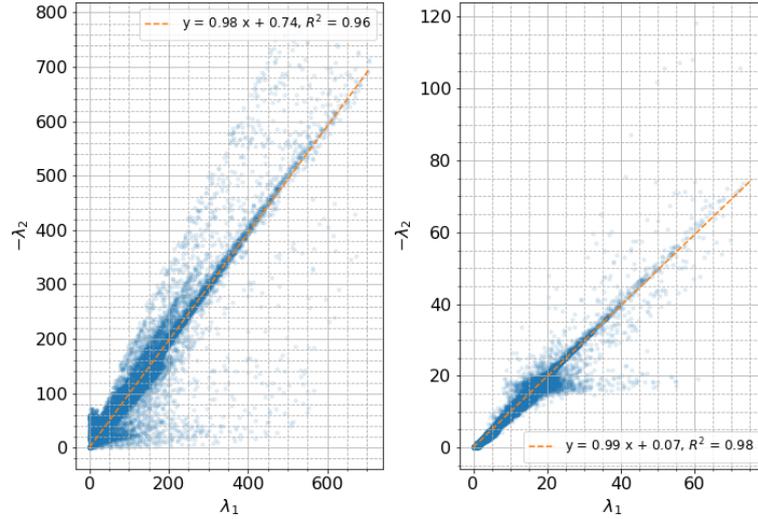


Figure 6.4: Linear fit between invariant $(\lambda_1, -\lambda_2)$ for data partitioned in wake and free-stream, left: wake, right: free-stream. Samples are indicated in blue.

The center of the wake, which is aligned with the rotor axis, experiences a small shear $\frac{\partial U}{\partial z}$ because the gradient gradually switches from negative to positive. On top of that, the center of the wake encounters a strong stream-wise gradient $\frac{\partial U}{\partial x}$. Near the turbines, the flow is influenced by the forcing of the rotor which significantly affects the velocity gradients.

The velocity-gradient assumption suggests that the set of five invariants can be reduced to a set of only one, either λ_1 , λ_2 or λ_5 . In Figure 6.7, the mutual information between the five invariants in the wake region is listed. From this can be concluded that, on average, the dependencies between the individual invariants remain strong and that the velocity-gradient assumption holds for the most part of the wake. However, the assumption fails in the specific above-described regions of the flow, and therefore the full effects of the five invariants cannot be accounted for by only one single invariant in these regions. This is confirmed by the heatmap in Figure 6.6 where the mutual information between the five invariants conditioned on $\lambda_{1+2} > 75$ is listed. Clearly, the strong relations as were observed in Figure 6.3 are not valid for this condition. Only the dependency between λ_2 and λ_5 remains reasonably strong.

Consequently, selecting more than one invariant can lead to capturing more flow characteristics in the domain near the turbines, often the first two invariants are chosen [69, 92]. Selecting the λ_1 and λ_2 is reasonable as λ_2 remains to have significant dependency with λ_5 .

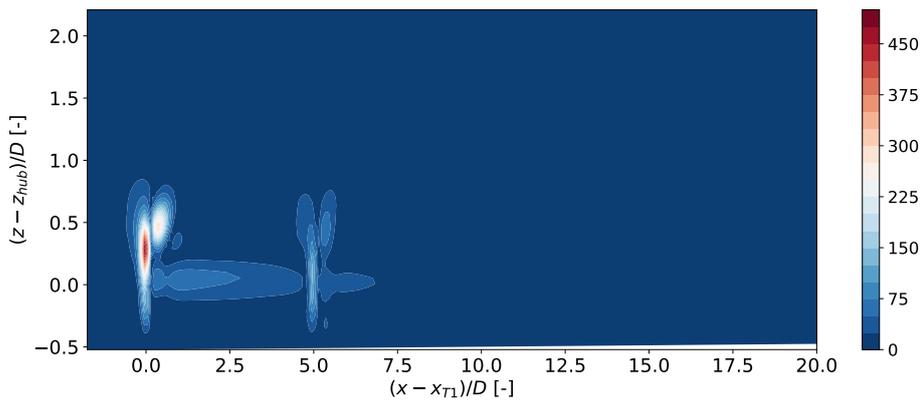


Figure 6.5: Contours of λ_{1+2} for vertical slice at turbine location.

Besides investigating the mutual dependency between invariants with respect to different regions in the flow, it is also interesting to define the dependency of invariants to the correction terms for different flow regions. In Figures 6.8 and 6.9, the mutual information is denoted between the five invariants with both

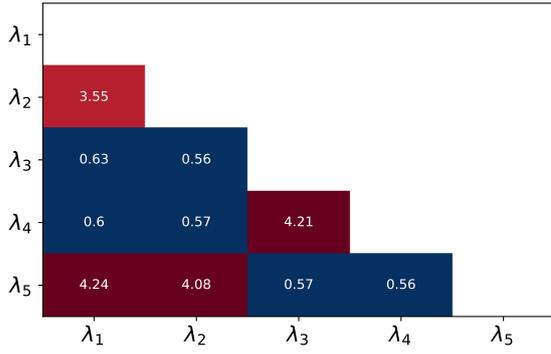


Figure 6.6: Mutual information between individual invariants in wake region.

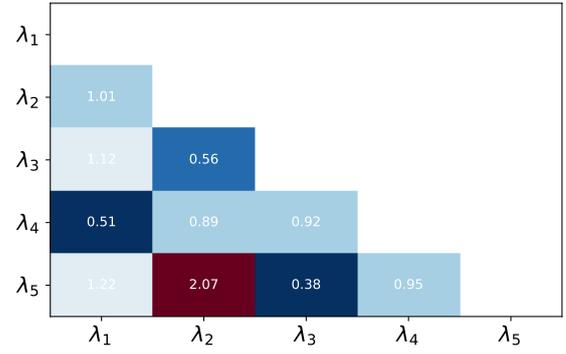


Figure 6.7: Mutual information between individual invariants conditioned on $\lambda_{1+2} > 75$.

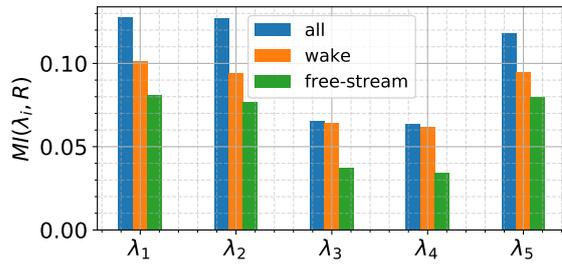


Figure 6.8: Mutual information between invariants and R .

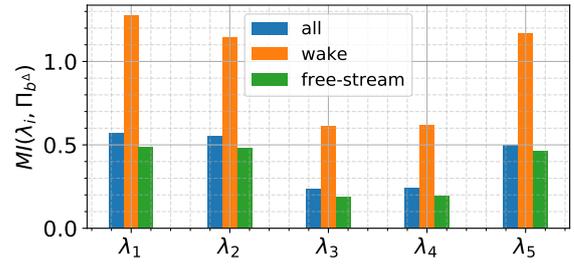


Figure 6.9: Mutual information between invariants and $\Pi_{b^{\Delta}}$.

correction terms for the wake and free-stream region. For the correction term R , the dependency with the invariants is slightly larger inside the wake than in the free-stream. It is noteworthy that the dependencies for both regions are smaller than for the whole domain. This is possible because the probability density functions in (4.4) are significantly distant for different flow regions. In Appendix B.1 a theoretical example is provided proving it is mathematically possible that two subsets have a smaller mutual information than the entire set.

For $\Pi_{b^{\Delta}}$, the dependencies inside the wake are significantly larger than outside the wake. For both corrections, λ_1 , λ_2 and λ_5 remain the most important invariants.

The relation between λ_1 and the correction terms is visualized in Figure 6.10. The relation of λ_1 with R shows to be very noisy when compared to the relation with $\Pi_{b^{\Delta}}$, which explains the difference in magnitude between $MI(\lambda_i, R)$ and $MI(\lambda_i, \Pi_{b^{\Delta}})$.

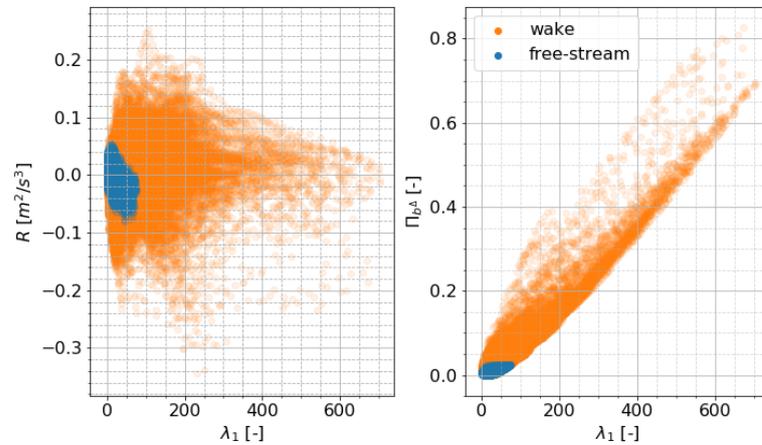


Figure 6.10: Correction terms R and $\Pi_{b^{\Delta}}$ as function of invariant λ_1 . Data is partitioned in wake and free-stream region.

6.2. Basis tensors

Applying the same procedure to the basis tensors as was done to the invariants will lead to a reduction in number of basis tensors by only selecting the ones which are most relevant to the correction terms. To compare tensors to other tensors or to scalars, the second invariant can be compared to give an indication of the tensor dependency. However, it is found that the second invariants of the basis tensors show very high mutual dependency, meaning it is inadequate to use these as a representation of the basis tensors because all basis tensors will have similar dependency with R and $\Pi_{\mathbf{b}^\Delta}$. For example, the second invariants of the first four basis tensors are related through

$$-4\Pi_{\mathbf{T}^{(1)}}^2 = \Pi_{\mathbf{T}^{(2)}} = \frac{1}{12}\Pi_{\mathbf{T}^{(3)}} = \frac{1}{12}\Pi_{\mathbf{T}^{(4)}} \quad (6.4)$$

based on the velocity gradient assumption.

Instead of comparing the second invariant of the basis tensors, individual tensor components can be compared. In Figure 6.11, the mutual information between individual tensor components of the basis tensors and \mathbf{b}^Δ are compared for the wake region in a heat map, bright colors correspond to strong relations. Both the first and sixth basis tensor $\mathbf{T}^{(1)}$, $\mathbf{T}^{(6)}$ have a very large mutual information estimate with two of the diagonal components (1,2) and (1,3). It is observed that basis tensors $\mathbf{T}^{(2)}$, $\mathbf{T}^{(7)}$ and $\mathbf{T}^{(8)}$ have some dependency with the components (1,1) and (2,3) and (3,3) while tensors $\mathbf{T}^{(3)}$, $\mathbf{T}^{(4)}$ and $\mathbf{T}^{(9)}$ only with (1,1) and (2,3). None of the basis tensor shows to have reasonable dependency with the (2,2) component.

Because many of the tensors show similar performance towards \mathbf{b}^Δ , a broad range of tensors are chosen for further analysis in Chapter 9. In Chapter 9, the effects of the chosen basis tensors on the model performance will be investigated. Basis tensors $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(6)}$ are chosen for their very strong relation with the off-diagonal components, $\mathbf{T}^{(2)}$ and $\mathbf{T}^{(8)}$ because of their relation to (1,1), (2,3) and (3,3). Finally, $\mathbf{T}^{(3)}$ and $\mathbf{T}^{(4)}$ are picked because of their dependency with (1,1) and (2,3) and because they are frequently used in literature.

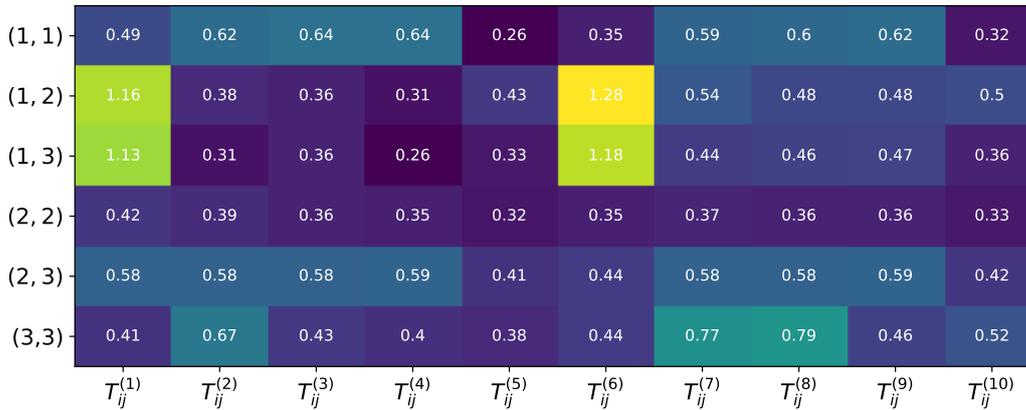


Figure 6.11: Mutual information heat map for basis tensor components with \mathbf{b}^Δ components.

The comparisons of the basis tensors components to \mathbf{b}^Δ for all samples and the free-stream region are listed in Appendix B. The dependency inside the wake is significantly larger than in the free-stream.

To compute the dependency between the basis tensors and R , the correction term can be compared to the double inner product between the velocity gradient tensor and the basis tensors. This is convenient as this product is a scalar and because R is partly modeled as a turbulent production term in which the same product is computed. Because this double inner product equals zero for basis tensors $\mathbf{T}^{(2)}$, $\mathbf{T}^{(5)}$ and $\mathbf{T}^{(7)}$, they are discarded from the analysis.

The dependencies are listed in Figure 6.12. Tensors $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(6)}$ have the largest dependency with R , while $\mathbf{T}^{(3)}$, $\mathbf{T}^{(4)}$ and $\mathbf{T}^{(9)}$ score slightly better than $\mathbf{T}^{(8)}$ and $\mathbf{T}^{(10)}$. Also the dependencies conditioned on wake or free-stream samples are listed. In both domains the $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(6)}$ remain the most dominant basis tensors.

The mutual dependency between the basis tensors for the wake region is computed and depicted as a heat map in Figure 6.13. There is clearly dependency between several basis tensors, especially between $\mathbf{T}^{(3)}$ and $\mathbf{T}^{(4)}$. The dependency between $\mathbf{T}^{(3)}$ and $\mathbf{T}^{(4)}$ is strong because under the velocity-gradient assumption

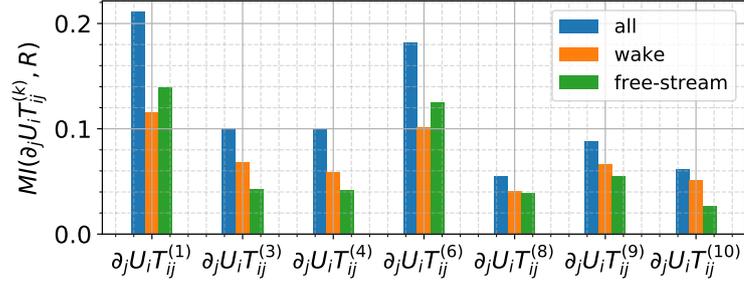


Figure 6.12: Mutual information between R and double inner dot product basis tensor and velocity gradients for different regions of the domain.

$$\mathbf{T}^{(3)} = -\mathbf{T}^{(4)} = \frac{1}{6} \frac{k}{\varepsilon} \begin{bmatrix} \frac{\partial U}{\partial y} \frac{\partial U}{\partial y} + \frac{\partial U}{\partial z} \frac{\partial U}{\partial z} & 0 & 0 \\ 0 & \frac{\partial U}{\partial y} \frac{\partial U}{\partial y} - 2 \frac{\partial U}{\partial z} \frac{\partial U}{\partial z} & 3 \frac{\partial U}{\partial y} \frac{\partial U}{\partial y} \\ 0 & 3 \frac{\partial U}{\partial y} \frac{\partial U}{\partial z} & -2 \frac{\partial U}{\partial y} \frac{\partial U}{\partial y} + \frac{\partial U}{\partial z} \frac{\partial U}{\partial z} \end{bmatrix}.$$

This explains the large value for the mutual information, but is only valid where the velocity-gradient assumption holds. Also noteworthy is that there is dependency between $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(6)}$, which is expected as both tensors contributed mainly towards the same components of \mathbf{b}^Δ . Also $\mathbf{T}^{(9)}$ is related to $\mathbf{T}^{(3)}$ and $\mathbf{T}^{(4)}$, explaining there similar contribution to R .

Because a lot of the basis tensors show to have similar performance, it is difficult to rule out most of them. Therefore the basis tensors $\mathbf{T}^{(1)}$, $\mathbf{T}^{(3)}$, $\mathbf{T}^{(4)}$, $\mathbf{T}^{(6)}$ and $\mathbf{T}^{(9)}$ are selected for further analysis in Chapter 8.

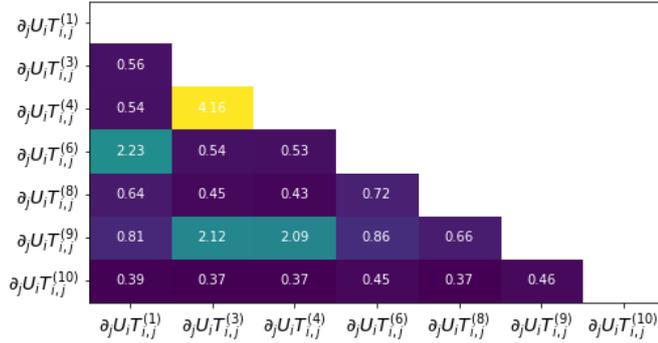


Figure 6.13: Heat map of mutual dependency between double inner product between basis tensors and velocity gradient for wake samples. Measured by mutual information.

6.3. Additional invariants

The additional invariants were listed in Section 5.2.4. The mutual information between the three best scoring additional invariants and R are listed in Figure 6.14 and Figure 6.15 for the entire domain and wake region, respectively. Contrary to the invariants and basis tensors discussed earlier, the dependency of the additional invariants depends significantly on the selected region as different additional invariants are designated with the highest dependency for the two considered regions. Although the dependencies are relatively small, it is observed that the invariants based on $\nabla \mathbf{k}$ are considered to have a higher relation to the correction factor than the invariants based on $\nabla \mathbf{p}$.

The mutual information between the three best scoring invariants and $\Pi_{\mathbf{b}^\Delta}$ for the entire domain and wake region are depicted in Figure 6.16 and Figure 6.17, respectively. Again, the dependencies are relatively small when compared to the invariants in Section 6.1. For this correction term, the designated additional invariants are also different for the wake region and the entire domain.

The mutual dependency between the best scoring invariants is depicted in Figure 6.18 for the wake region. Although the mathematical form of the invariants is very similar, they do not necessarily have a large

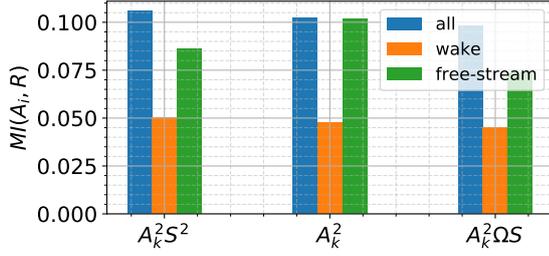


Figure 6.14: Mutual information between invariants and R for three best scoring invariants when considering all samples.

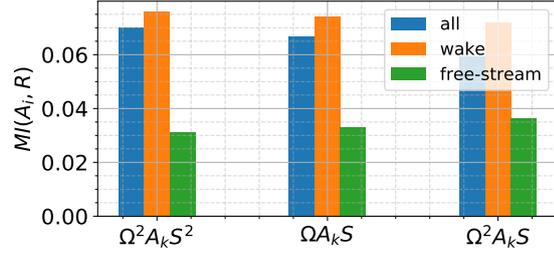


Figure 6.15: Mutual information between invariants and R for three best scoring invariants when considering wake samples.

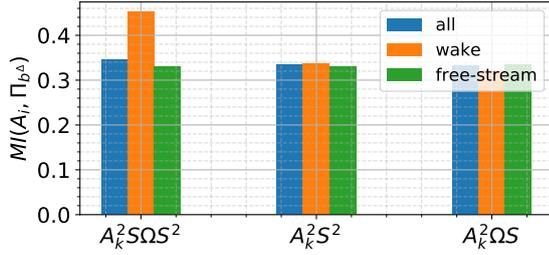


Figure 6.16: Mutual information between invariants and Π_{b^Δ} for three best scoring invariants when considering all samples.

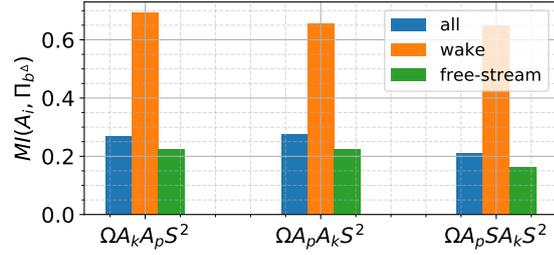


Figure 6.17: Mutual information between invariants and Π_{b^Δ} for three best scoring invariants when considering wake samples.

dependency. The additional invariants chosen for further analysis are $\Omega^2 A_k S^2$ and $\Omega A_k S$ for R and $\Omega A_k A_p S^2$ and $\Omega_p A_k S^2$ for b^Δ .

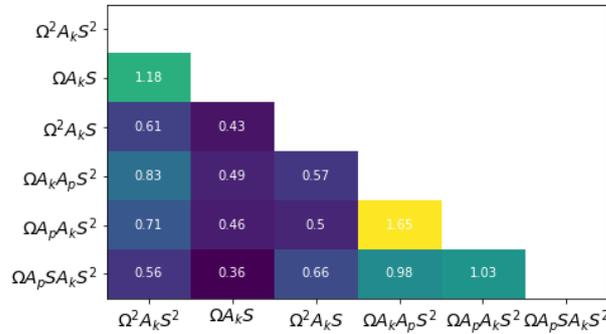


Figure 6.18: Mutual information between individual invariants for wake samples. Invariants are the traces of the listed expressions.

6.4. Physical flow features

The physical flow features analysed were listed and described in Table 5.3. The estimated mutual information between the physical flow features and the correction term R for samples from the entire domain, wake and free-stream are listed in Figure 6.19. For the correction R , q_8 (viscosity ratio), q_1 (Wall-based Reynolds number), q_{10} (ratio of total to normal Reynolds stresses) and q_7 (shear parameter) are found to have the largest dependency in the wake region. Clearly, the mutual information estimates (and therefore the feature importance) is dependent on the region selected. While for the entire domain, q_3 (turbulence intensity 2) was found to have the highest mutual information, its value for the wake region is significantly smaller.

In feature q_{10} , the Reynolds stress is defined only by the Boussinesq hypothesis (deviatoric part is proportional to strain-rate tensor and eddy viscosity), meaning q_{10} can be reformulated as

$$q_{10} = \frac{\overline{\|u'_i u'_j\|}}{k} = \frac{\| -2\frac{\nu_t \epsilon}{k} S_{ij} + \frac{2}{3} k \delta_{ij} \|}{k} = \frac{\| -2k C_\mu S_{ij} + \frac{2}{3} k \delta_{ij} \|}{k} = \| -2C_\mu S_{ij} + \frac{2}{3} \delta_{ij} \| \approx -2C_\mu \|S\| \propto \|S\|, \quad (6.5)$$

showing its form is very similar to the first invariant λ_1 , but instead the l_2 -norm of the strain-rate tensor is taken while λ_1 is computed by the trace of the strain-rate tensor squared $\{S^2\}$. Only the Reynolds stress anisotropy part related to the eddy viscosity is taken for q_{10} because including \mathbf{b}^Δ could result in misleading results when compared to $\Pi_{\mathbf{b}^\Delta}$, as the correction term is then part of the feature. The mutual information between (λ_1, q_{10}) equals 9.96, meaning q_{10} is redundant when λ_1 is already included.

Therefore, the set of best scoring features for further analysis is reduced to q_1, q_7 and q_8 .

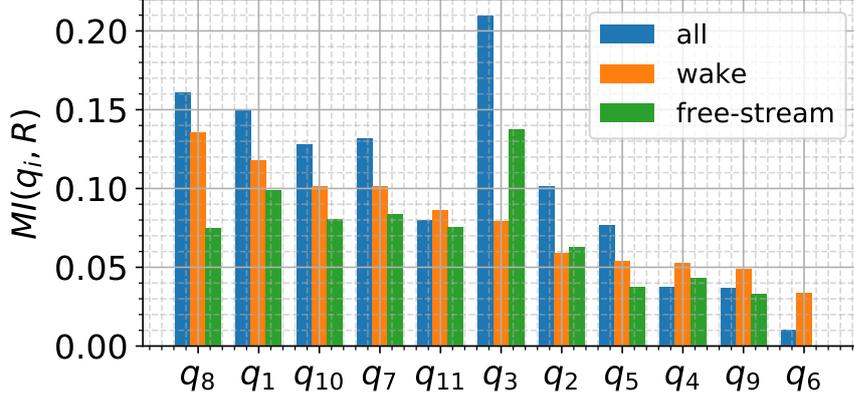


Figure 6.19: Mutual information between physical flow features and R for all samples, wake samples and free-stream samples. Features are ordered by mutual information in wake region.

The same analysis is applied to $\Pi_{\mathbf{b}^\Delta}$, in Figure 6.20 barplots of the mutual information between the physical flow features and the second invariant are depicted for different flow regions. Again, the dependency depends significantly on the select region, while q_3 shows to have the largest dependency for the entire domain, its value for the wake region is much smaller. It is found that q_2 (turbulence intensity 1), q_7 (shear parameter) and q_{10} (ratio of total to normal Reynolds stresses) have the largest dependency with the correction term in the wake region. Because the form of q_{10} is very similar to that of λ_1 , it is discarded.

For $\Pi_{\mathbf{b}^\Delta}$, q_2 and q_7 are the features which will be used for further analysis Chapter 9.

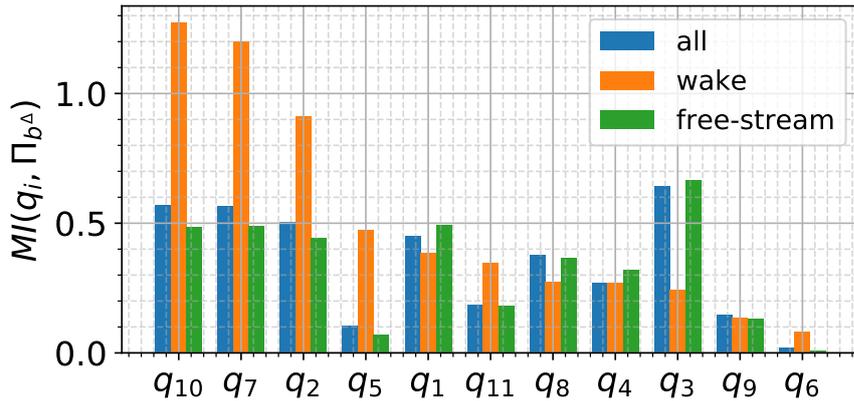


Figure 6.20: Mutual information between physical flow features and $\Pi_{\mathbf{b}^\Delta}$ for all samples, wake samples and free-stream samples. Features are ordered by mutual information in wake region.

The mutual dependencies between the selected physical flow features in the wake region are listed in Figure 6.21. The dependencies are relatively small except for (q_1, q_8) as both the magnitude of the wall-based Reynolds number and viscosity ratio depends significantly on k . Similarly, there is some dependency between (q_2, q_7) because both depend on the k and the velocity gradient tensor.

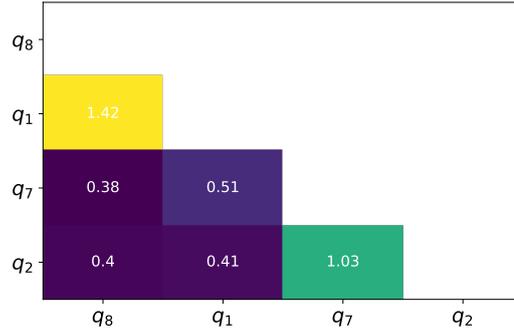


Figure 6.21: Mutual information between selected physical flow features for wake region.

6.5. Domain specific feature importance

Mutual information determines the general dependency between variables. The analysis has already been narrowed to wake samples to determine which features are most influential to the correction terms in this region. Still, if there are strong trends between the features and the correction terms in a small portion of the wake region, while in the remainder of the wake the relation is mainly noise, the mutual information will be small. This is because the probability distributions are based on all wake samples, meaning the strong trends in a small portion of the wake samples will only be moderately represented in the probability distributions. Two examples are given where this can lead to complications.

For the force coefficient q_6 , it is found that the mutual information can significantly increase when only a portion of the samples is considered. For example, q_6 is zero for most of the samples because the turbine forcing only affects samples close to the turbine. However, there is a clear trend between (q_6, R) when only considering the samples that have $q_6 > 0$, as can be observed in Figure 6.22.

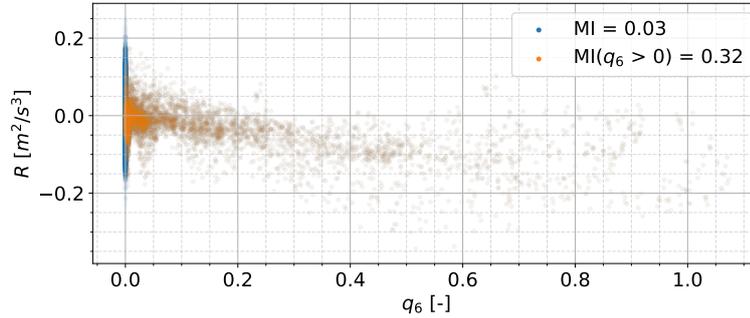


Figure 6.22: Correction R as function of force coefficient q_6 . Blue indicates wake samples where orange is wake samples including specific condition in legend.

Secondly, the estimated mutual information for the additional invariants depends significantly on the selected region. In Section 6.3, it was already found that different additional invariants were described as *most important* for the entire domain and the wake region. Similarly, when considering only samples in the upper part of the near-wake ($(x - x_{T2})/D < 5.5$ and $(z - z_{hub})/D > 0.1$, referred to as near-wake samples), \mathbf{A}_k^2 is found to be the most important additional invariant. The reason its mutual information estimate is very region specific is because in the far-wake R contains a lot of noise. This noise affects the joint probability distribution $p(x, y)$ significantly and thus also the mutual information estimate. In Figure 6.23, a trend is visible for the near-wake data. However, this trend is vanished when all wake samples are considered. This effect is quantified by the mutual information for the two regions, where the mutual information for the samples in the near-wake is much larger than for all wake samples. The far-wake noise present in R did not have significant effect on the other considered features in this research.

Both q_6 and \mathbf{A}_k^2 are found to improve the predictions of R significantly (as will be discussed in Chapter 8), however their importance can only be quantified with mutual information by considering small regions of the flow. This is because for the whole wake domain, on average, their contribution is small. Although selecting q_6 and \mathbf{A}_k^2 is rather ad-hoc, it is important to also show the shortcomings of a methodology so that these can

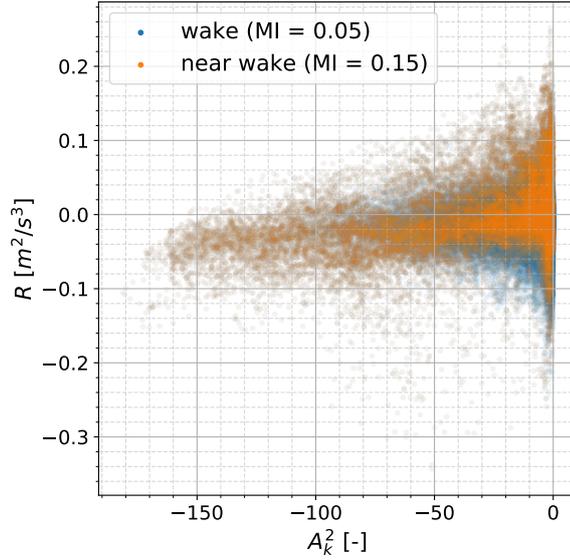


Figure 6.23: Correction term R as function of A_k^2 for two flow regions (wake and near-wake).

be taken into account in future research. Based on the observations, it is chosen to define both q_6 and A_k^2 as important features for the correction term R . Using a similar analysis q_6 is also found important to Π_{b^Δ} .

6.6. Conclusion

In this chapter, the importance of different features with respect to the correction terms R and Π_{b^Δ} was investigated using mutual information.

In research [49, 67, 91], often the invariants and basis tensors discussed in Section 6.1 and Section 6.2 are used as features. This is theoretically the most sound as these only depend on the local rate of strain and rotation $f(\mathbf{S}, \boldsymbol{\Omega})$ and are an inherent part of the effective-viscosity formulation proposed by Pope [58]. However, (as analysed for this specific turbine case) it seems that this gives only limited possibilities because the sets can be reduced to only a smaller set that is actually relevant. By loosening the assumption that the Reynolds stresses only depend on the local rate of strain and rotation, additional invariants and physical flow features can be included. It is found that especially certain physical flow features have a large dependence with the correction terms, implying that including these will potentially lead to improved model predictions.

Mutual information gives a clear quantitative picture of the dependencies between features and the correction terms. Similarly, mutual information is used to quantify the relation between different features to detect redundancies. Given the mutual information results, it remains however to the user to make a choice about which features to use for the construction of the correction models. This is not always easy, for example, there are a number of basis tensors with equal dependency towards the correction terms, making it difficult to rule out most of them. Also for the invariants, the mutual information between the invariants is significant, indicating redundancy. However the level of redundancy is very region specific, making it difficult to determine whether it is necessary to include more than one invariant.

Similarly, the mutual information estimate is significantly dependent on the selected region. As the physics inside the wind turbine wake are very different compared to those in the free-stream, the relations between the features and the correction terms are also affected by the chosen region. Because the wake region samples will be used to train the algebraic models later in this work, the derived dependencies inside the turbine wake are leading when defining the most important features. However, this can still lead to missing features that are important to more specific flow regions, such as the force coefficient q_6 .

Concluded from the analysis in this chapter, the features that will be used for further analysis are those listed in Table 6.1. These will be used in Chapter 8 to improve the predictions of R and in Chapter 9 the predictions of b^Δ . The total feature set has been reduced from 68 to 14 and 13 for R and b^Δ , respectively.

Table 6.1: Features used for further investigation with respect to the correction terms R and \mathbf{b}^Δ .

n	Feature type	Features R	Features \mathbf{b}^Δ
7	Basis tensors	$\mathbf{T}^{(1)}, \mathbf{T}^{(3)}, \mathbf{T}^{(4)}, \mathbf{T}^{(6)}, \mathbf{T}^{(9)}$	$\mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}, \mathbf{T}^{(4)}, \mathbf{T}^{(6)}, \mathbf{T}^{(8)}$
2	Invariants	λ_1, λ_2	λ_1, λ_2
5	Additional invariants	$\boldsymbol{\Omega}^2 \mathbf{A}_k \mathbf{S}^2, \boldsymbol{\Omega} \mathbf{A}_k \mathbf{S}, \mathbf{A}_k^2$	$\boldsymbol{\Omega} \mathbf{A}_k \mathbf{A}_p \mathbf{S}^2, \boldsymbol{\Omega} \mathbf{A}_p \mathbf{A}_k \mathbf{S}^2$
5	Physical features	q_1, q_6, q_7, q_8	q_2, q_6, q_7

7

Wake Classification

In this chapter, a classification model is constructed that distinguishes the turbine wake from the remaining part of the domain. This wake classification model can be used in combination with an algebraic model for R and b^A that is specifically trained on the wake samples to enhance the predictions.

Correction terms trained specifically on the data inside the turbine wake can reach higher accuracy in this region because they do not have to cope with the trends and noise in the correction terms in the free-stream flow. This can lead to models that are simpler but can reach the same or even higher accuracy as more complex models that are trained on the entire data set. Correction terms trained specifically on the data inside the turbine wake are computationally less demanding, as the data set is significantly reduced. Therefore, the list of candidate functions, used to train the model, can be extended. This improves the flexibility of the model to fit the data.

Because the correction models are solely trained upon the samples inside the turbine wake, it is not guaranteed that it fits the free-stream data because this region is not used for training. Therefore, the classification model is introduced that turns *on* the correction model in the turbine wake and *off* in the free-stream. Because the correction models are turned off outside the turbine wake, the correction models are not constrained to fit the data outside the wake, meaning it can *focus* entirely on the data inside the turbine wake, resulting in a better fit than when trained on the entire sample set. In this research, it is assumed that the baseline $k-\varepsilon$ turbulence model is able to simulate the free-stream flow accurately, meaning it is acceptable that the correction terms are turned off in this region. This assumption is valid because a free-stream correction is applied to the baseline $k-\varepsilon$ turbulence model such that the velocity and turbulent kinetic profiles match those from LES.

For the classifier, the wake has as class label "1" or "True", while data in the free-stream is labeled with "0" or "False". The wake classification model must meet a number of requirements:

1. classification model must be able to accurately predict the turbine wake region
2. classification model must predict a smooth transition at wake boundaries to avoid irregularities
3. classification model must preferably be a simple/sparse algebraic expression

The second requirement implies that the confidence of the classification model at the wake boundaries should gradually decrease from close to 1.0 to close to 0.0 to avoid irregularities and non-smooth behavior in the predictions of the correction terms R and b^A . The third requirement can be split up in two parts, sparsity ensures the model is less prone to overfitting and that the numerical stiffness is reduced, which promotes convergence when implemented into a CFD solver. The model should preferably be algebraic, resulting in easy implementation in a solver and fast execution while it also offers better interpretability. By utilizing the sparse logistic regression algorithm, an algebraic expression is guaranteed.

First the features that show a high mutual dependency with the wake label are discussed in Section 7.1. Second, the results are discussed in Sections 7.2 to 7.4.

7.1. Wake Dependency

The features that are best suited to distinguish whether samples are located in the turbine wake or not, can be determined by computing the mutual information between individual features and the class labels [85]. The class label is 1 for samples in the wake and 0 for samples outside the wake, then the expected value of the conditional differential entropy is

$$E[h(X|Y)] = p(y=1) \cdot h(X|y=1) + p(y=0) \cdot h(X|y=0) \quad (7.1)$$

such that the mutual information is

$$MI(X; Y) = h(X) - h(X|Y). \quad (7.2)$$

The five physical flow features having the largest dependency with the class label are listed in Figure 7.1. The mutual information estimations indicate that the features q_3 and q_8 , the turbulence intensity 2 and viscosity ratio respectively, are most important, which is reasonable as their magnitude grows significantly in the majority of the turbine wake.

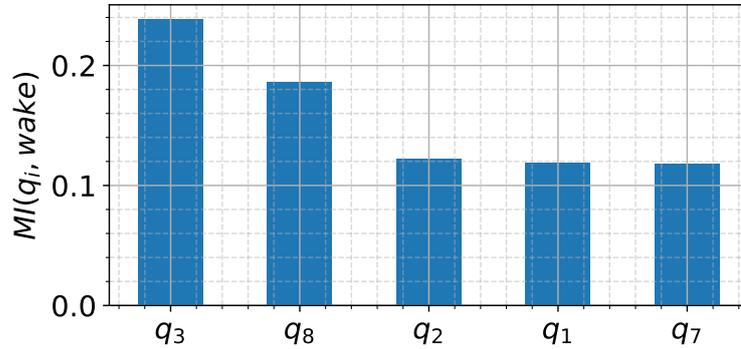


Figure 7.1: Mutual information between physical flow features and wake classes.

It is noted that the above-listed features mainly correlate with wake samples downstream of the turbines and are not able to identify all wake points near the turbines itself. Therefore, two more features are introduced that are active near the turbines, the force coefficient q_6 and non-orthogonality between velocity and its gradient q_9 . It is also noted that, from the features listed in Figure 7.1, q_3 , q_7 and q_8 are most effective, leading to the final feature set $[q_3, q_6, q_7, q_8, q_9]$.

Sparse logistic regression is chosen as machine learning algorithm as the classification model should be simple and algebraic. The advantage over ordinary logistic regression is that it can incorporate pre-defined non-linear feature combinations, while ordinary regressions can only cope with linear relations. The number of pre-defined non-linear combinations is although very limited, meaning sparse symbolic regression is not able to capture as many detailed flow details as more complex machine learning classification models such as neural networks and random forests.

The sparse logistic regression algorithm constructs a library of candidate functions, which are non-linear combinations of the pre-defined set of features. The feature set \mathcal{B} is

$$\mathcal{B} = [q_3, q_6, q_7, q_8, q_9] \quad (7.3)$$

and the list of non-linear candidate functions is constructed by using the exponent 1.0, meaning the features are multiplied by each other up to a maximum degree of 3. This results in the vector \mathbf{x}

$$\mathbf{x} = [q_3, q_3 q_7, q_3 q_7 q_8, \dots, q_7 q_9^2]^T. \quad (7.4)$$

7.2. Model Discovery

In the model discovery phase, the library of candidate functions \mathbf{x} in (7.4) is standardized and subjected to the regression in (4.37) to discover the most important model coefficients. For the regularization parameters,

a broad set of values are used to cover different regularization types. For the mixing parameter ρ and the inverse regularization strength C , the following values are used

$$\rho = [0.01, 0.1, 0.2, 0.5, 0.7, 0.9, 0.95, 0.99, 1.0]^T \quad (7.5)$$

$$C = [10^{-7}, \dots, 10^{-2}]^T, \quad (7.6)$$

respectively to set up a grid of regularization parameters (C, ρ) . For C , 50 entries are used uniformly spaced on log-scale between the limits depicted in (7.6), meaning in total 450 models are discovered. Because different combinations of the regularization parameters (C, ρ) can result in identical non-zero model coefficients, only the unique sets of non-zero model coefficients are forwarded to the model inference phase. The models are trained on the data from case B and C, such that it can later be used to test on case A.

In Figure 7.2, the percentage is depicted how often each candidate function is selected in the model discovery phase. The features q_3 and q_8 are present in most of the functions that are selected most frequently, which is consistent with the conclusion in Figure 7.1 that these two features are most important.

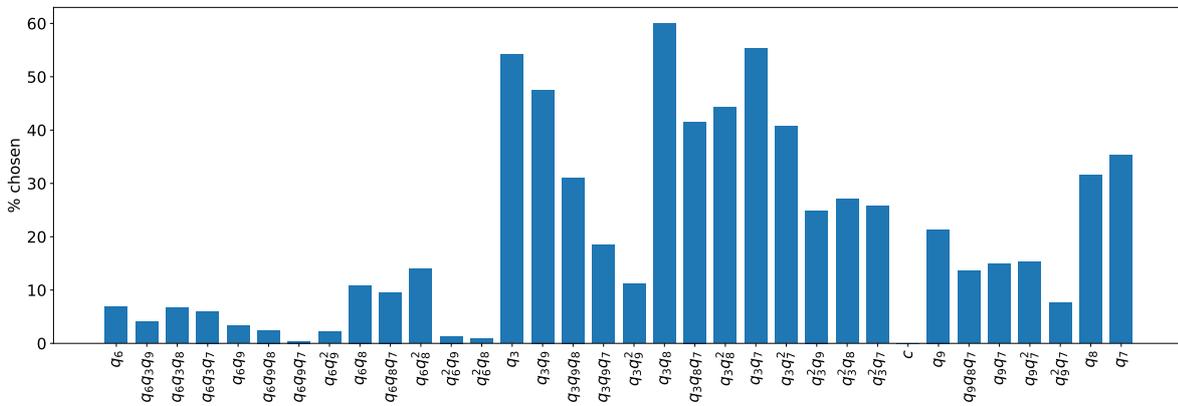


Figure 7.2: Occurrence of candidate functions in model discovery phase expressed in percentages.

7.3. Model Inference using cross-validation

The unique sets of non-zero model coefficients obtained in the model discovery phase are undergoing an additional regression using unstandardized functions, such that models with correct units are obtained. In the model discovery phase, the model performance on the training data was irrelevant, as the intention was to discover model coefficients for different regularization strengths.

In the model inference phase, the ultimate goal is to select regression models that have high predictive capabilities. Therefore in the model inference phase, cross-validation is applied in which models are trained and tested on separate data to assess the predictive performance. In this section, the models are trained on case B and C and tested on case A.

The functions corresponding to the discovered non-zero coefficients in Section 7.2 are subjected to the regression in (4.38) with l_2 regularization through parameter C . For the regularization parameter, the following values are used

$$C = [10^0, 10^1, 10^2]^T. \quad (7.7)$$

The performances of the obtained models are depicted in Figure 7.3 using the metrics discussed in Section 4.4.1. The log loss, accuracy and recall are denoted as a function of complexity. Only the models with complexity $n_C < 15$ are depicted. On average, the model performance increases for increasing complexity which is mainly visible in the log loss, although this is not always valid as more complex models are generally more prone to overfitting. Although the performance increases for more complex models, it is only modest, meaning simple models do have almost the same predictive performance as more complex models. This can be because only five features are considered, meaning adding more non-linear combinations of the same features does not result in significant improvements. On top of that, complex models are more prone

to overfitting, meaning increasing model complexity does not guarantee increased performance on unseen test data.

Overall the performance of all models are good, as the accuracy and recall are both high. The final chosen model is depicted by the green star, it is chosen because it has low complexity (only four non-zero terms) and has good predictive performance as it has a high accuracy and recall and low log loss. The accuracy is 99%, meaning 99 out of 100 samples are predicted correctly. The recall is 94%, meaning 94 out of 100 wake samples are predicted correctly.

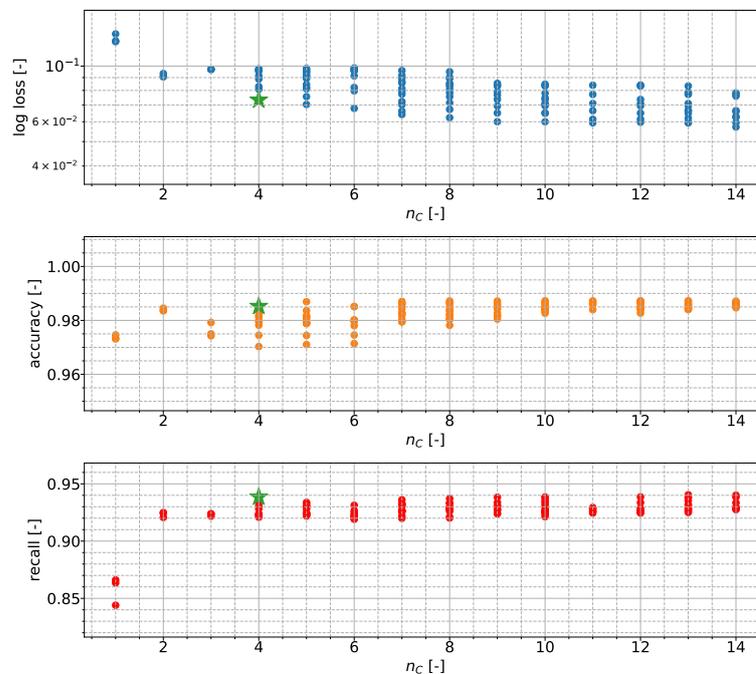


Figure 7.3: Log loss (top), accuracy (middle) and recall (bottom) as function of complexity n_c for all models analysed in the inference phase. Final chosen model high-lighted with green star.

The model is formulated as

$$\mathcal{M} = \sigma \left[\mathbf{w}_u^T \mathbf{x}_u + 1.02 \right] \quad (7.8)$$

$$\mathbf{x}_u = \left[q_3 q_9, q_3 q_8, q_3 q_7^2, q_8 \right] \quad (7.9)$$

$$\mathbf{w}_u = \left[1471, 248.8, 4.77, -6.51 \right]. \quad (7.10)$$

The same cross-validation procedure has been applied to case B and C as test sets (described in Appendix C), the results are denoted in Table 7.1.

Table 7.1: Performance classifier when tested on specific cases

	case A	case B	case C
Log loss	0.07	0.12	0.09
Accuracy	0.99	0.96	0.97
Recall	0.94	0.80	0.89
Complexity	4	6	4

7.4. Performance visualization

The performance of the model can be visualized using contour plots of slices of the domain. When tested on case A, the vertical slice in the $x-z$ plane at turbine location is depicted in Figure 7.4 showing the true wake label and wake classifier confidence predictions. Overall the classifier gives good results, outside the wake the prediction is close to zero while inside the wake close to one. There are a few regions where the classifier fails, in the near-wake it fails in the lower part of the wake and at a small region in the wake center. Besides that, the confidence is under-predicted in a small region upstream of the top part of the rotor and over-predicted upstream of the lower part of the rotor.

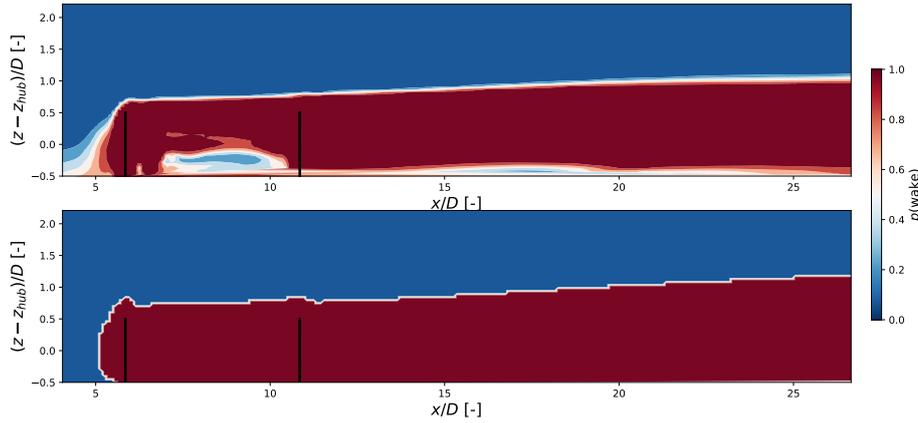


Figure 7.4: Wake classifier predictions compared to true label for case A on $x-z$ plane at turbine location $(y - y_{\text{hub}})/D = 0$. Upper figure: classifier predictions, lower figure: true label. Turbines indicated in black.

The performance of the classifier can also be expressed by its discrete predictions (when rounded to either 0 or 1) and labeling samples to either true positive (TP), true negative (TN), false positive (FP) or false negative (FN). The labels are depicted in Figure 7.5, showing indeed that the errors occur along the edges of the wake, in front of the first turbine and in the lower part of the near-wake.

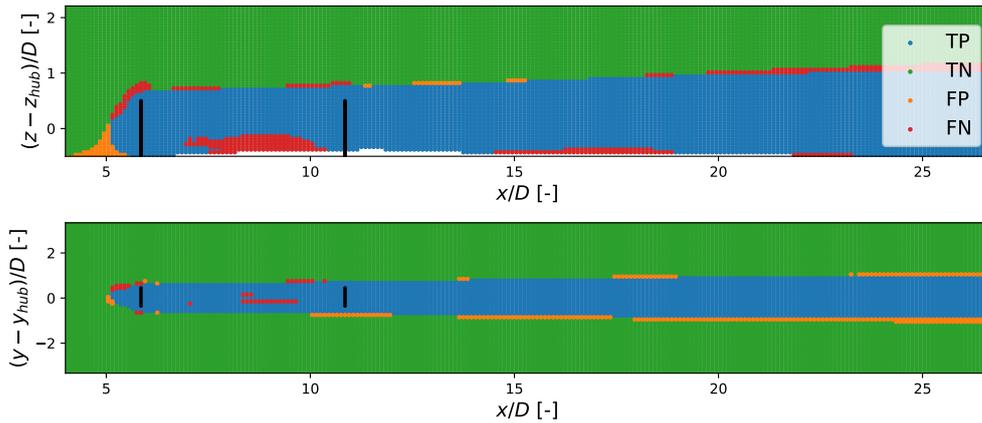


Figure 7.5: Wake classifier discrete predictions compared to true discrete labels for case A. Upper figure: $x-z$ plane at $(y - y_{\text{hub}})/D = 0$, lower figure: $x-y$ plane at $(z - z_{\text{hub}})/D = 0$. Turbines indicated in black.

The classifier predictions are compared to the contours of R and $\Pi_{b\Delta}$ in Figure 7.6 for different slices in the $y-z$ plane. It can be clearly seen that close to the first turbine $((x - x_{T1})/D = 3)$, the classifier fails to identify the lower part of the wake. However, when comparing the classifier predictions to the contours of R and $\Pi_{b\Delta}$, it is seen that both correction terms are close to zero in the lower part of the near-wake, meaning the

shortcoming of the classifier in not able to identify this region does not result in the classifier being turned *off* where the correction terms are large. One final observation is that in the free-stream, the prediction grows when approaching the wall. This is due to the turbulence intensity q_3 feature, which magnitude grows near the wall.

The contourplots for case B and C are denoted and described in Appendix C.

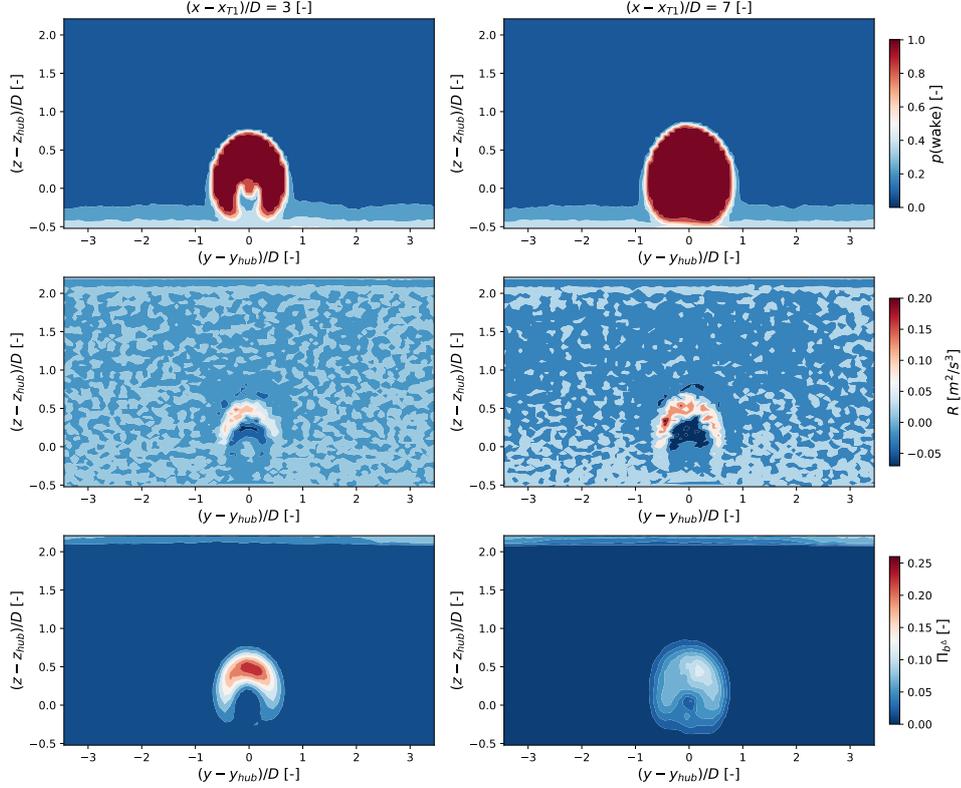


Figure 7.6: Comparison of classifier predictions to contours of R and $\Pi_{\mathbf{b}^\Delta}$ for case A at different slices in the $y-z$ plane, x -location depicted in title upper row. Upper row: classifier predictions, middle row: R and lower row: $\Pi_{\mathbf{b}^\Delta}$.

7.5. Conclusion

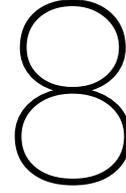
In this chapter, a wake classification model is constructed using sparse logistic regression which can act as an activation of the correction terms R and \mathbf{b}^Δ .

First, the most important features were selected and used to construct a library of candidate functions. Subsequently in the model discovery phase, the model coefficients were selected for different regularization parameters. The final unstandardized models were constructed in the model inference phase and the predictive performance was tested using cross-validation.

It was said that the classification model should meet three requirements: accurately predict the turbine wake, smooth transition at wake boundaries and low complexity. The complexity is definitely low, as the model for case A only consists of four terms, making it easy to interpret and applicable for implementation in a CFD solver.

The predictions at the wake boundaries does not always have the desired smoothness. For example, when observing the prediction boundaries (where the predictions go from 1 to 0) in Figure 7.6, the classification predictions quickly goes from 1 to 0. However, the magnitude of the correction terms R and $\Pi_{\mathbf{b}^\Delta}$ at these boundaries are already small, so no significant irregularities are expected. Because the predictions mainly depend on the velocity (gradients), turbulent kinetic energy and the dissipation rate, whose magnitudes increase abruptly at the wake edges, it is difficult to obtain a classification model that has a reasonable amount of smoothness in this region.

The classification model can accurately predict most regions of the wake, although it fails in particular regions such as the lower part of the near-wake. It is noted that both the magnitude of both R and \mathbf{b}^Δ are not significant in these regions.



Transport equation deficit R

The investigated features in Chapter 6 and the wake classification model predictions in Chapter 7 are used to enhance the regression models for R . In addition, the modeling-form of R is extended using a source/sink term proportional to the local dissipation rate to improve the predictions of the correction to the transport equations.

First in Section 8.1 the features, derived from the mutual information analysis are denoted that will be used in this chapter to improve the predictions of R . Secondly in Section 8.2, the coupling between the wake classification results and the regression model is explained such that the regression results are mainly influenced by the estimated wake samples. Subsequently, the specifications of the machine learning algorithm are discussed in Section 8.3 and the regression results for different feature sets in Section 8.4. In Section 8.5, a comparison is made of training the regression models either on the predicted wake samples or all samples. Also the effects of applying the wake classification model is described.

The best-performing feature set is subsequently used for cross-validation in Section 8.6 to assess the model's predictive performance when trained and tested on different turbine cases. Finally, the chapter is ended with a conclusion in Section 8.7.

8.1. Features

From the analysis in Chapter 6, the total list of available features was reduced to a smaller set using the mutual information estimates. The remaining features are summarized in Table 8.1.

The entire feature set listed in Table 8.1 cannot be used all at once due to the computational limitations of the machine learning algorithm. The algorithm constructs a library of non-linear candidate functions based on the pre-defined features, this means that introducing new features results in an exponential growth of the library of functions. Besides that, adding more features does not necessarily result in a reduction of the error and can even cause an increase in error, this is further elaborated in Section 8.4.3. It is also interesting to observe whether the features appointed by mutual information actually result in a prediction improvement or not. In this way it can be validated if mutual information is a reliable tool for feature selection.

Therefore, the analysis will start with a feature set of moderate size (based on the minimal integrity basis used by [68]) which will be iteratively extended and shrunk based on the performance of the specific considered set. In this way, it can be observed what the effects of different features are on the model performance.

Table 8.1: Features used for modeling correction term R .

n	Feature type	Features
5	Basis tensors	$\mathbf{T}^{(1)}, \mathbf{T}^{(3)}, \mathbf{T}^{(4)}, \mathbf{T}^{(6)}, \mathbf{T}^{(9)}$
2	Invariants	λ_1, λ_2
3	Additional invariants	$\Omega^2 \mathbf{A}_k \mathbf{S}^2, \Omega \mathbf{A}_k \mathbf{S}, \mathbf{A}_k^2$
4	Physical features	q_1, q_6, q_7, q_8

8.2. Classification and regression coupling

The results of the classification model are incorporated into the regression algorithm such that the regression model is mainly influenced by the samples predicted to be located in the turbine wake. The wake prediction results \mathbf{p} are coupled to their corresponding samples \mathbf{x} . This coupling is incorporated by the following, considering the residual of an ordinary least square problem as

$$r = \sum_{i=1}^n (y_i - \mathbf{w}\mathbf{x}_i)^2, \quad (8.1)$$

in which the sample wake prediction p_i can be introduced as a weight to the importance of the sample residual as

$$\sum_{i=1}^n p_i (y_i - \mathbf{w}^T \mathbf{x}_i)^2. \quad (8.2)$$

This can be rewritten to

$$\sum_{i=1}^n (\sqrt{p_i} y_i - \sqrt{p_i} \mathbf{w}^T \mathbf{x}_i)^2 \quad (8.3)$$

which implies the wake predictions can be incorporated in the regression algorithm by multiplying in advance the individual samples \mathbf{x}_i and correction term y_i with the corresponding sample's wake prediction $\sqrt{p_i}$. Preliminary, all samples with p_i smaller than 0.30 are removed from the data set as it is expected that their effect is only marginal on the regression outcome. In addition, the predictions of those samples are significantly damped by the classification model. The samples for which $p_i > 0.30$ holds are referred to as the predicted wake samples. The wake predictions \mathbf{p} are also included in the computation of the mean-squared error, which measures the model performance, as

$$MSE = \frac{1}{N} \sum_{i=1}^N p_i (y_i - \mathbf{w}^T \mathbf{x}_i)^2. \quad (8.4)$$

8.3. Sparse regression

The SpaRTA algorithm is applied as discussed in Section 4.3. The pre-defined features are combined with exponent 1.0 such that feature combinations up to cubic terms are constructed. In the model discovery phase, the regularization mixing parameter ρ is assigned to the following values

$$\rho = [0.01, 0.1, 0.2, 0.5, 0.7, 0.9, 0.95, 0.99, 1.0]^T \quad (8.5)$$

such that a broad range of different regularization combinations is applied. For the regularization strength α_i , 20 entries are used uniformly spaced on a log-scale between α_0 and α_{\max}

$$\boldsymbol{\alpha} = [\alpha_0, \dots, \alpha_{\max}]^T, \quad (8.6)$$

where $\alpha_0 = 10^{-3}$, α_{\max} and $\alpha_{\max} = \max(\mathbf{w}^T \mathbf{x}) / (K\rho)$ where K is total number of samples. The regularization strength α_{\max} is defined such that any $\alpha > \alpha_{\max}$ results in all coefficients \mathbf{w} to be set to zero.

In the model discovery and inference phases in Section 8.4, the predicted wake samples of the three turbine cases are combined to one data set and randomly divided into a training and testing set, containing 75% and 25% of the total data set, respectively. It is chosen to combine the three cases to one set to diminish the effects of training and test set composition on the feature set performance. The model inference phase is performed using ordinary least squares regression because the only goal is to compare the performance of different feature sets.

For the model discovery and inference phases during cross-validation in Section 8.6, the models are trained on the predicted wake samples of case B and C, and tested on those of case A. In the model inference phase, the models are subjected to an additional l_2 regularisation using the regularization parameter λ_r . When applying regularization to unstandardized data, candidate functions with small magnitudes (coupled to a large coefficient) will be more affected by the regularization than candidate functions with large

magnitude (coupled to a small coefficient), which affects the model performance. Generally, models obtained through large values for λ_r have a larger likelihood to result in converged solutions when incorporated into a CFD solver. For λ_r , the following values are used

$$\boldsymbol{\lambda}_r = [0, 10^{-1}, 1]^T. \quad (8.7)$$

8.4. Results

The features listed in Table 8.1 are used to learn sparse regression models for R . The procedure is started with the reduced integrity basis consisting of the first four basis tensors and two invariants, as used by Schmelzer et al. [68] and step-by-step expanded. The correction term is expanded with the source/sink term R_c in Section 8.4.4. Only models with complexity $n_C \leq 20$ are used for the analysis to enhance the model performance visualization and because overly complex models impede convergence and increase the numerical stiffness in CFD [69].

8.4.1. Basis tensors

This section investigates the observations from Section 6.2 in which it was concluded that the basis tensors with highest dependency to R are $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(6)}$. Besides that, also $\mathbf{T}^{(3)}$, $\mathbf{T}^{(4)}$ and $\mathbf{T}^{(9)}$ had an equal estimated mutual information which was significantly smaller.

These observations are now tested by constructing sparse regression models using different feature sets. They are then either validated or disproved by observing the mean-squared error as function of model complexity. The considered feature sets are listed in Table 8.2.

Table 8.2: Features used for modeling correction term $R_{\mathcal{P}}$.

Feature set	Features	# functions
$\mathcal{B}_{R_{\mathcal{P}}}^{(1)}$	$\lambda_1, \lambda_2, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}, \mathbf{T}^{(4)}$	18
$\mathcal{B}_{R_{\mathcal{P}}}^{(2)}$	$\lambda_1, \lambda_2, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}, \mathbf{T}^{(6)}$	18
$\mathcal{B}_{R_{\mathcal{P}}}^{(3)}$	$\lambda_1, \lambda_2, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}$	12
$\mathcal{B}_{R_{\mathcal{P}}}^{(4)}$	$\lambda_1, \lambda_2, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}, \mathbf{T}^{(9)}$	18

The mean-squared error as function of model complexity is depicted in Figure 8.1. Clearly, $\mathcal{B}_{R_{\mathcal{P}}}^{(1)}$, $\mathcal{B}_{R_{\mathcal{P}}}^{(3)}$ and $\mathcal{B}_{R_{\mathcal{P}}}^{(4)}$ all have similar performance, meaning adding $\mathbf{T}^{(4)}$ and $\mathbf{T}^{(9)}$ does not yield any improvements. This is consistent with the mutual information estimates, as there was a significant dependency measured between $\mathbf{T}^{(3)}$ and $\mathbf{T}^{(4)}$ and to a lesser amount with $\mathbf{T}^{(9)}$, the mutual information of these basis tensors with the correction term R was also small.

Adding basis tensor $\mathbf{T}^{(6)}$ results in improved model predictions for more complex models, which is also consistent with the mutual information results as it was the second ranked tensor after $\mathbf{T}^{(1)}$.

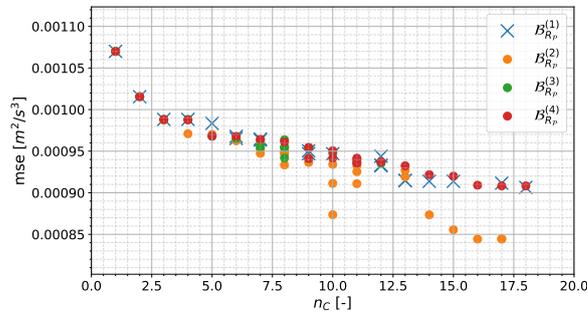


Figure 8.1: Mean-squared error in R as function of model complexity n_C for three different feature sets described in Table 8.2.

8.4.2. Additional invariants

The best scoring feature set in Section 8.4.1, $\mathcal{B}_{R\mathcal{P}}^{(2)}$, is expanded with the additional invariants. The considered combinations are listed in Table 8.3. The mean-squared error as function of complexity for the different feature combinations is depicted in Figure 8.2. The feature combinations are compared to the best performing combination from Section 8.4.1. Clearly the introduction of the additional invariants gives a slight improvement in the predictions and it is seen that the effect of invariant \mathbf{A}_k^2 is most significant as $\mathcal{B}_{R\mathcal{P}}^{(6)}$ and $\mathcal{B}_{R\mathcal{P}}^{(7)}$ perform better than $\mathcal{B}_{R\mathcal{P}}^{(5)}$. It should be noted that \mathbf{A}_k^2 was however not identified directly as an important feature by mutual information.

Table 8.3: Features used for modeling correction term $R_{\mathcal{P}}$.

Feature set	Features	# functions
$\mathcal{B}_{R\mathcal{P}}^{(5)}$	$\lambda_1, \lambda_2, \mathbf{\Omega}^2 \mathbf{A}_k \mathbf{S}^2, \mathbf{\Omega} \mathbf{A}_k \mathbf{S}, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}, \mathbf{T}^{(6)}$	54
$\mathcal{B}_{R\mathcal{P}}^{(6)}$	$\lambda_1, \lambda_2, \mathbf{\Omega}^2 \mathbf{A}_k \mathbf{S}^2, \mathbf{A}_k^2, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}, \mathbf{T}^{(6)}$	54
$\mathcal{B}_{R\mathcal{P}}^{(7)}$	$\lambda_1, \lambda_2, \mathbf{A}_k^2, \mathbf{\Omega} \mathbf{A}_k \mathbf{S}, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}, \mathbf{T}^{(6)}$	54

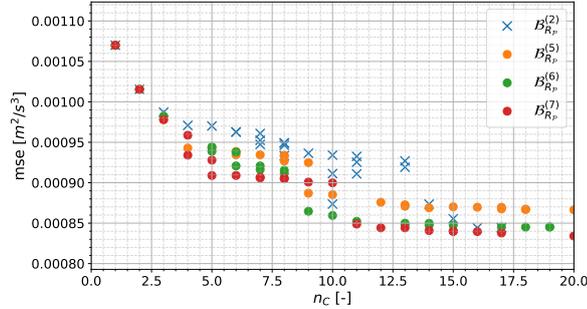


Figure 8.2: Mean-squared error R as function of model complexity n_C for feature sets described in Table 8.3.

8.4.3. Physical flow features

The feature sets in Section 8.4.2 are expanded using the physical flow features listed in Table 8.1. The examined features are q_1 (Reynolds number), q_6 (force coefficient), q_7 (shear parameter) and q_8 (viscosity ratio) and the considered feature sets are listed in Table 8.4. It must be noted that invariants ($\lambda_2, \mathbf{\Omega} \mathbf{A}_k \mathbf{S}$) and basis tensor $\mathbf{T}^{(6)}$ are removed from the feature set to reduce the computational costs of the algorithm and because they have become redundant with the addition of the physical flow features. The invariant λ_2 is removed because it has a significant dependency with λ_1 , only \mathbf{A}_k^2 is used as additional invariant because it was found that this invariant is responsible for the largest prediction improvements in Figure 8.2. Basis tensor $\mathbf{T}^{(6)}$ is removed because most of the selected terms in Figure 8.2 contain either $\mathbf{T}^{(1)}$ or $\mathbf{T}^{(3)}$.

The mean-squared error as function of complexity for the different feature sets is depicted in Figure 8.3 and it is obviously visible that the addition of physical flow features results in a significant error reduction. This is also consistent with the mutual information estimates as these had the largest measured dependency with R .

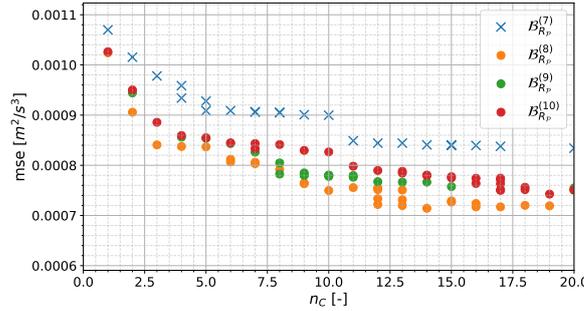
Clearly, the addition of the force coefficient q_6 results in an error reduction as both $\mathcal{B}_{R\mathcal{P}}^{(8)}$ and $\mathcal{B}_{R\mathcal{P}}^{(9)}$ perform better than $\mathcal{B}_{R\mathcal{P}}^{(10)}$. The models constructed using the feature set $\mathcal{B}_{R\mathcal{P}}^{(8)}$ performs slightly better than $\mathcal{B}_{R\mathcal{P}}^{(9)}$, meaning that with the physical flow features q_1, q_6 and q_7 the best performing $R_{\mathcal{P}}$ prediction model can be constructed.

Removing features

One could argue that removing features from the feature set results in decreased model performance. Instead, one can use less data samples for model training to compensate the computational costs.

Table 8.4: Features used for modeling correction term $R_{\mathcal{P}}$.

Feature set	Features	# functions
$\mathcal{B}_{R_{\mathcal{P}}}^{(8)}$	$\lambda_1, \mathbf{A}_k^2, q_1, q_6, q_7, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}$	68
$\mathcal{B}_{R_{\mathcal{P}}}^{(9)}$	$\lambda_1, \mathbf{A}_k^2, q_6, q_7, q_8, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}$	68
$\mathcal{B}_{R_{\mathcal{P}}}^{(10)}$	$\lambda_1, \mathbf{A}_k^2, q_1, q_7, q_8, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}$	68

Figure 8.3: Mean-squared error R as function of model complexity n_C for feature sets described in Table 8.4.

The performance of $\mathcal{B}_{R_{\mathcal{P}}}^{(8)}$ is compared to a feature set that includes, on top of the features used by $\mathcal{B}_{R_{\mathcal{P}}}^{(8)}$, λ_2 , $\mathbf{\Omega A}_k \mathbf{S}$ and q_8 and uses a third of the data for training compared to $\mathcal{B}_{R_{\mathcal{P}}}^{(8)}$ in the model discovery phase. Also, the effects of including $\mathbf{T}^{(6)}$ and excluding $\mathbf{T}^{(3)}$ are investigated. The feature sets are listed in Table 8.5 and the mean-squared error as function of model complexity is depicted in Figure 8.4.

First, it is found that the addition of features and reduction of data samples rather results in a decreased model performance. Second, it is also observed that removing $\mathbf{T}^{(3)}$ from the feature set does not result in an increase in error, meaning that when a sufficient number of physical flow features and additional invariants is used, the use of more basis tensors than $\mathbf{T}^{(1)}$ becomes redundant. By using $\mathbf{T}^{(6)}$ instead of $\mathbf{T}^{(3)}$, the error is slightly increased.

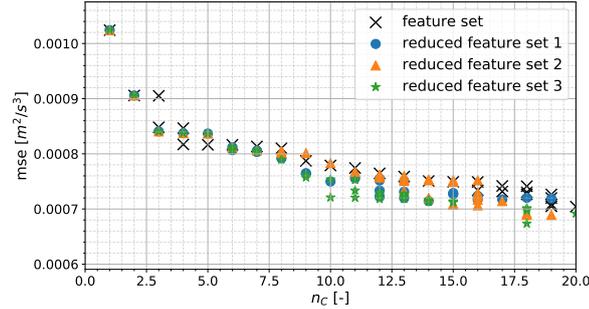
There are a couple of reasons why the prediction error can increase for a larger feature set, such as

- **Reduced training set.** The feature set with 256 functions in Table 8.5 uses a third of the training data compared to the other feature sets in the model discovery phase. However, it is not expected that this results in a significant difference.
- **Training and test data.** The training and test data are not identical, this means that specific chosen functions can result in overfitting. However, this is not expected as the training and test data are very similar given they both originate from the data set in which the three turbine cases are combined.
- **Coordinate descent.** The optimisation algorithm used is coordinate descent that finds the minimum of a function by successively minimizing along the coordinate directions (candidate functions). The coordinate descent algorithm is guaranteed to find the global optimum only when the function space is strictly convex and differentiable [52]. This means that adding more features can result in the coordinate descent algorithm converging to worse local optima.
- **Standardized/unstandardized.** The SpaRTA algorithm consists of two regression steps, model discovery using standardized functions and model inference using unstandardized functions. It is possible that certain candidate functions perform well when standardized while having worse performance unstandardized.

It is not investigated what the exact reason is for the increase in prediction error. However, it is expected that it is a combination of the above-described reasons.

Table 8.5: Features used for modeling correction term $R_{\mathcal{P}}$.

Feature set	Features	# functions
feature set	$\lambda_1, \lambda_2, \mathbf{A}_k^2, \mathbf{\Omega A}_k \mathbf{S}, q_1, q_6, q_7, q_8, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}$	256
reduced feature set 1 ($\mathcal{B}_{R_{\mathcal{P}}}^{(8)}$)	$\lambda_1, \mathbf{A}_k^2, q_1, q_6, q_7, \mathbf{T}^{(1)}, \mathbf{T}^{(3)}$	68
reduced feature set 2	$\lambda_1, \mathbf{A}_k^2, q_1, q_6, q_7, \mathbf{T}^{(1)}, \mathbf{T}^{(6)}$	68
reduced feature set 3	$\lambda_1, \mathbf{A}_k^2, q_1, q_6, q_7, \mathbf{T}^{(1)}$	34

Figure 8.4: Mean-squared error in R as function of model complexity for four different feature sets listed in Table 8.5.

8.4.4. Source/sink term

The correction model is expanded by the source/sink term R_c . The proposed feature set is listed in Table 8.6. Two separate libraries of candidate functions are constructed, the first in the form of turbulent production ($R_{\mathcal{P}}$) and the second as the source/sink term (R_c). The listed feature set in Table 8.6 is found to result in the lowest mean-squared error compared to other tested feature combinations.

The mean-squared error of $\mathcal{B}_{R_{\mathcal{P}}+R_c}^{(11)}$ as function of model complexity is shown in Figure 8.5. Clearly the addition of the source/sink term R_c results in a further reduction in error compared to the feature set only based on $R_{\mathcal{P}}$. Interestingly to indicate is that a model constructed using $R_{\mathcal{P}} + R_c$ of complexity $n_c = 8$ has better performance as much more complex models constructed using only $R_{\mathcal{P}}$.

It must be noted that the physical flow features used for the $R_{\mathcal{P}}$ part are slightly different than the ones used in the best performing feature set in Table 8.4. When solely $R_{\mathcal{P}}$ is used for modeling R , the physical flow features q_1, q_6 and q_7 result in the lowest error. However when R is modeled by both $R_{\mathcal{P}}$ and R_c , using q_6 and q_8 for $R_{\mathcal{P}}$ result in the lowest error for R . This means that introducing the R_c part affects the best set of features for $R_{\mathcal{P}}$.

Secondly, for R_c the function $q_8^{1.0}$ is added manually to the discovered models. It was observed that more complex models had a significant smaller mean-squared error than simple models. This was because complex models had the function $q_8^{1.0}$ included while simple models did not. Subsequently, it was found that including $q_8^{1.0}$ manually also for simple models resulted in a reduction in mean-squared error for those simple models. The reason the algorithm did not pick this function itself for simple models could be because the solver did not reach the optimal solution. It could also be that this function has a worse performance when standardized, causing it to not be selected in the model discovery phase.

Table 8.6: Features used for modeling correction term R .

Feature set	Features ($R_{\mathcal{P}}$)	Features (R_c)	# functions
$\mathcal{B}_{R_{\mathcal{P}}+R_c}^{(11)}$	$\lambda_1, q_6, q_8, \mathbf{T}^{(1)}$	$\mathbf{A}_k^2, \{\mathbf{\Omega A}_k \mathbf{S}\}, q_6, q_7, q_8$	42

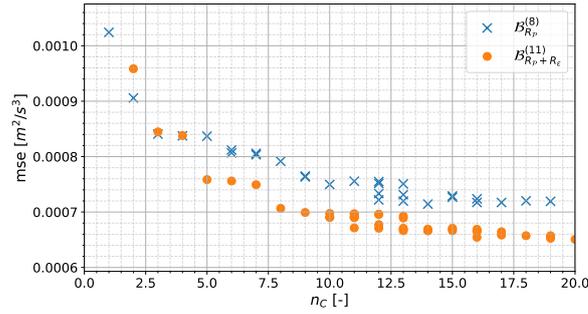


Figure 8.5: Mean-squared error R as function of model complexity n_C for feature sets described in Table 8.6.

8.5. Comparison training on predicted wake/all samples

By training the regression models for R only on the predicted wake samples, potential improvements in R inside the wake can be made because the regression model does not have to cope with the free-stream data. When there are trends in R in the free-stream, the regression model will try to fit both the trends inside and outside the wake, resulting in diminished predictions inside the wake.

Two frameworks are considered, framework 1 trains the regression models for R on all samples of case B and C and tests on the predicted wake samples of case A. Framework 2 trains on the predicted wake samples of case B and C and tests on the predicted wake samples of case A.

The mean-squared and maximum-squared errors for both frameworks are depicted in Figures 8.6 and 8.7. It can be seen that for simple models ($n_C < 13$), framework 2 results in only very marginal improvements compared to framework 1. For more complex models ($n_C > 12$), the improvements are only in maximum-squared error.

The reason that the prediction improvements are marginal is because R in the free-stream is mainly noise without any significant trends. This means that the regression models, when R is trained on all samples, will automatically also focus on the trends inside the wake region.

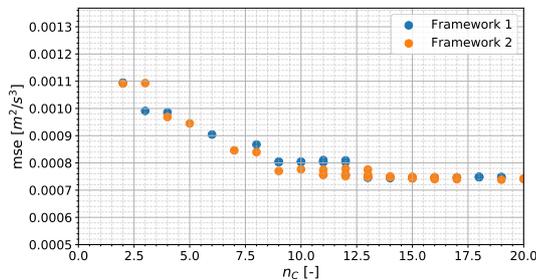


Figure 8.6: Mean squared error as function of model complexity for models trained on predicted wake samples and models trained on all samples.

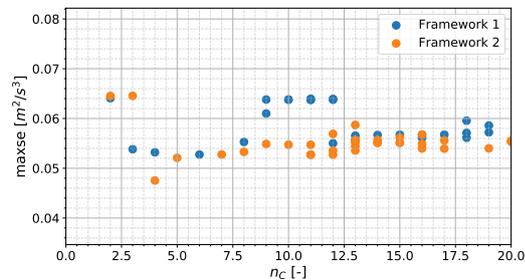


Figure 8.7: Maximum squared error as function of model complexity for models trained on predicted wake samples and models trained on all samples.

By using the regression models obtained through framework 2, it is not guaranteed that the prediction models will accurately fit the data in the free-stream (which fluctuates around zero) because only wake samples are used for training. To visualize this consequence, the predicted R by a model with $n_C = 9$ trained on the predicted wake samples only, is depicted in Figure 8.8 when multiplied with the classification model (' C ') and when not multiplied with the classification model ('no C ').

As can be seen, both predictions are relatively small compared to the true values. This is because the candidate functions are multiplied either with $2k\partial_j U_i$ or ε (depending on the library), which magnitude shrinks significantly in the free-stream flow. By using the classification model, the magnitude of the correction term is damped, reducing the effects on the free-stream flow. However, because the magnitude is also small when the classification model is not used, it is not expected that it is necessary to damp the R predictions in the free-stream.

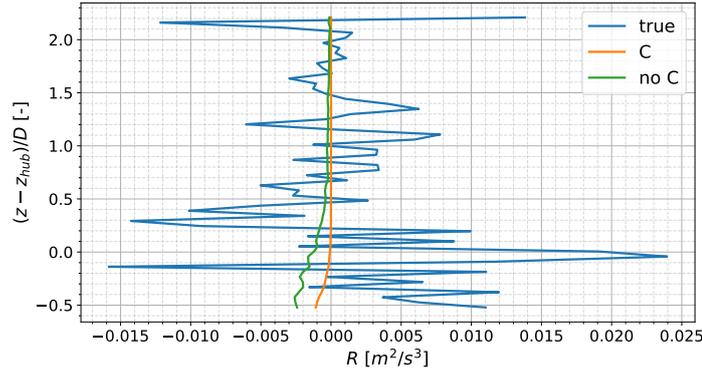


Figure 8.8: Predictions in R (no C) and predictions in R multiplied with classification model (C) compared to true values as function of domain height in the free-stream. Location in y : $(y - y_{\text{hub}})/D = -3$, location in x : $(x - x_{T2})/D = 10$.

8.6. Cross-validation

The established feature set in Table 8.6 is trained on the wake predictions of case B and C and tested on the wake predictions of case A to quantify the predictive capabilities on unseen data. Different sparse models of varying complexity are obtained by applying the model discovery and model inference steps, as described in Section 4.3. The regularization parameters in (8.7) are used in the inference phase.

The mean-squared error as function of model complexity is depicted in Figure 8.9, in which it becomes clear that the model's performance increases rapidly for low complexity but stagnates after $n_C > 9$. The model chosen for further inspection is depicted with an orange star, which is a good trade-off between complexity and accuracy. In Figure 8.10 the maximum-squared error is shown, it is observed that, contrary to the mean-squared error, it increases with model complexity. Although the increase is only moderate, it should be taken into account that large maximum-squared errors could indicate that specific regions in the domain are not well predicted.

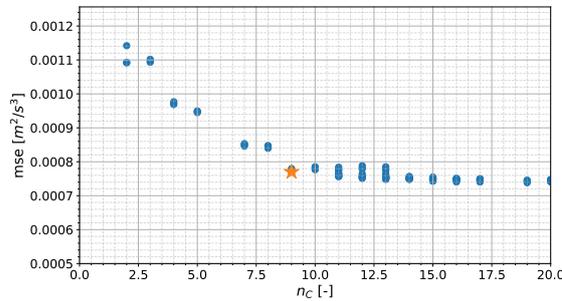


Figure 8.9: Mean squared error R as function of model complexity. Model for further inspection indicated by orange star.

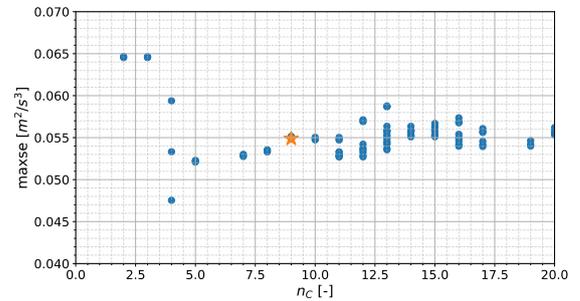


Figure 8.10: Maximum squared error R as function of model complexity. Model for further inspection indicated by orange star.

The model depicted with the orange star has the form

$$\begin{aligned} \mathcal{M}(R_{\mathcal{P}} + R_{\mathcal{E}}) = & 2k\partial_j U_i (\mathbf{T}^{(1)} \cdot [0.017q_6 + 0.003q_8]) + \\ & \epsilon (-0.06q_6q_7q_8 - 7.19q_6q_8 + 0.05q_7 - 2.0 \cdot 10^{-5}q_7A_1^2 - 0.36q_8 + 0.00011q_8A_1A_2 + 0.041A_2) \end{aligned} \quad (8.8)$$

where $A_1 = \mathbf{A}_k^2$ and $A_2 = \mathbf{\Omega A}_k \mathbf{S}$.

The model predictions on case A are compared to the true values in Figure 8.11 for different one-dimensional lines in the domain. It can be seen that the model predictions are able to capture most of the trends in the correction term, although the trends in vertical direction are captured better than in transverse direction. In Figure 8.12, contour-plots in the $y - z$ plane of the true and predicted values of R are depicted for different x -locations. Again, it is observed that most of the trends in the true values are also captured by the predictions.

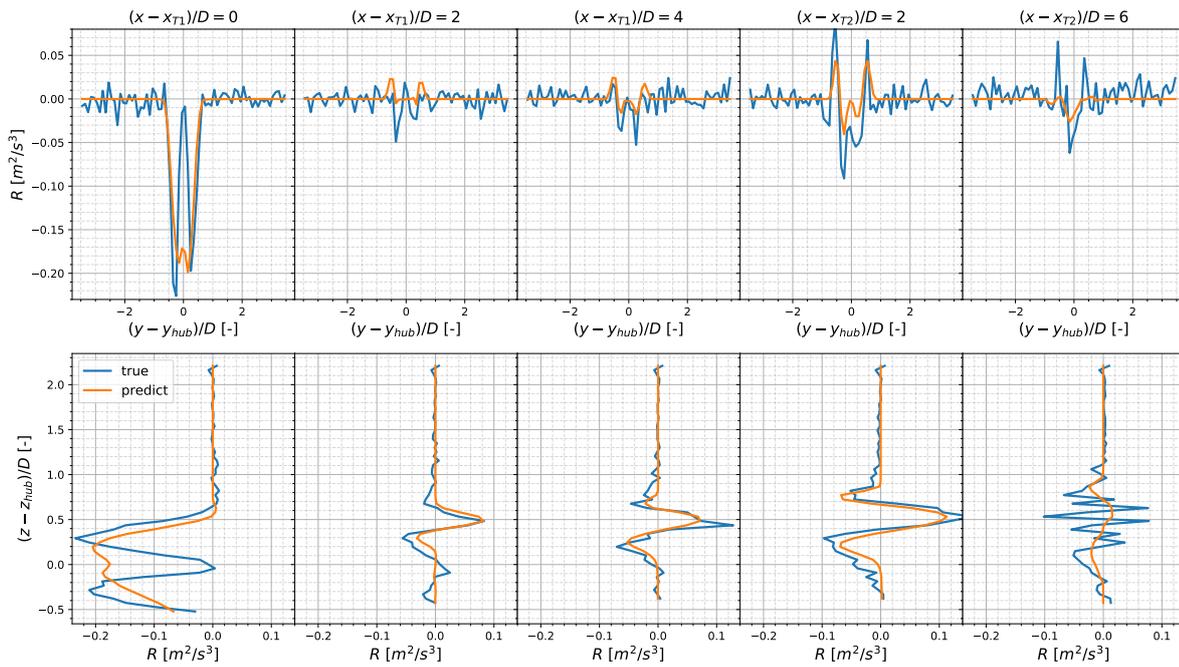


Figure 8.11: Case A. True (blue) and prediction (orange) of R as function of domain width (top row for $(z - z_{hub})/D = 0$) and height (bottom row for $(y - y_{hub})/D = 0$). Relative locations in x with respect to turbines are denoted in title.

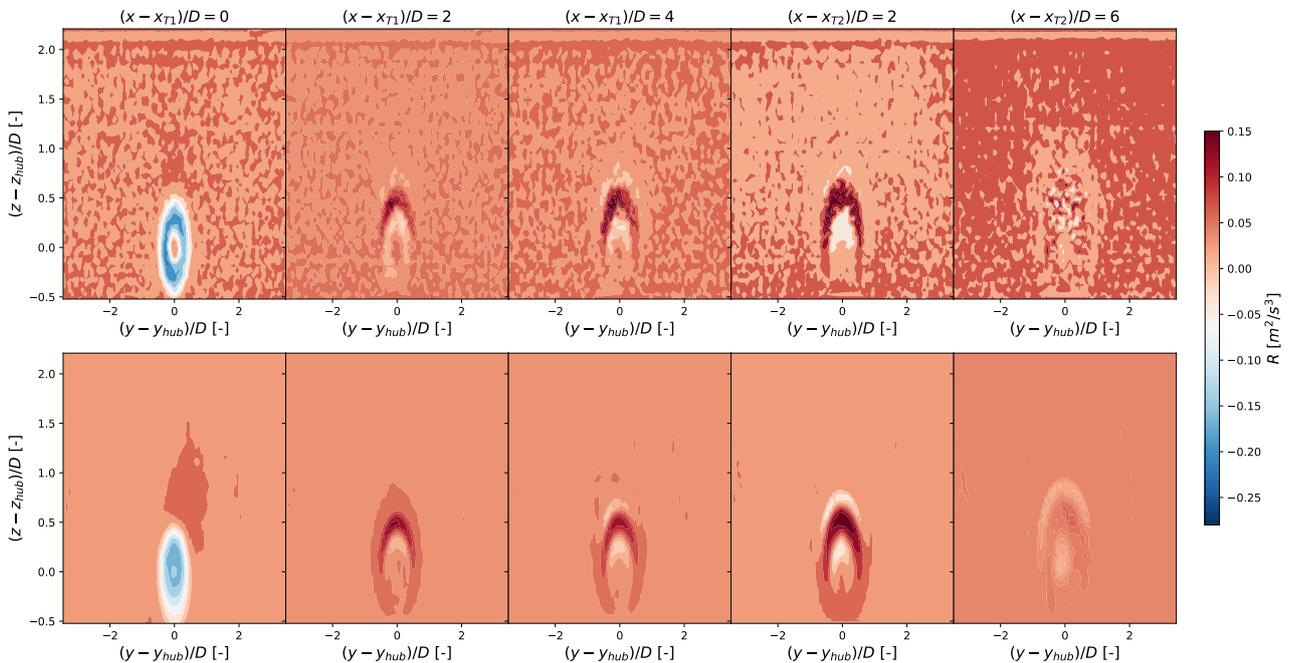


Figure 8.12: True (top row) and prediction (bottom row) of R for $y - z$ planes. Relative locations in x with respect to turbines are denoted in title.

8.7. Conclusion

In this chapter, the aim was to improve the predictions of R by introducing new features, training solely on the predicted wake samples and expanding the modeling-form with R_c . It was concluded that training models on the predicted wake samples does only yield a very slight improvement in the predictions inside the wake when compared to models trained on all domain samples. This is because R does not contain any significant trends in the free-stream as it is mainly noise. It is also found that R remains small in the free-stream even when the classification model is not used. This indicates that, with respect to R , the use of the classification model is not per se necessary.

The introduction of new features, that were identified by mutual information as having a high dependency with the correction term, significantly affect the model performance. Especially the addition of physical flow features results in large improvements. It was chosen to iteratively introduce new features to be able to compare the mutual information results to the corresponding model performance. It was found that when a significant number of scalar features (additional invariants, physical flow features) is used, all basis tensors except $\mathbf{T}^{(1)}$ become redundant. On top of that, using more features does not necessarily result in better predictions.

The addition of the R_c term resulted in further improvements, showing that expanding the modeling-form gives possibilities for better predictions.

During cross-validation, the regression model was trained and tested on separate cases to investigate the predictive performance. It was observed that most of the dominant trends in R were captured by the regression model using only a moderately complex model. It is therefore concluded that by introducing features with high dependency with the correction term and by expanding the modeling-form, good results can be accomplished. It must be noted that this is not yet a guarantee that these models will lead to good predictions when implemented in CFD.

9

Anisotropy correction \mathbf{b}^Δ

Similar to the analysis in Chapter 8, the predictions of \mathbf{b}^Δ will be enhanced using the features discussed in Chapter 6 and improved by using only the predicted wake samples for training. The same algorithm specifications will be used as in Chapter 8 meaning the regularization parameters and training and testing sets are identical. The threshold of \mathbf{p} is set to 0.5 (instead of 0.3 for R) because it was found that this significantly improved the predictions inside the wake.

In Section 9.1, the features are denoted that will be used in the subsequent analysis. Thereafter in Section 9.2, the predictions for different feature sets will be analysed. The effect of using only the predicted wake samples for training is discussed in Section 9.3. The best-performing features are used for cross-validation in Section 9.4, to test the model performance when trained and tested on different cases. The chapter is finished with a conclusion in Section 9.5.

9.1. Features

The analysed features are listed in Table 9.1. The physical flow features are q_2 (turbulence intensity 1), q_6 (force coefficient) and q_7 (shear parameter). Both the turbulence intensity 1 and shear parameter depend on the norm of the velocity gradient tensor (mathematical form listed in Table 5.3), which explains their relevance to \mathbf{b}^Δ as Pope [58] hypothesized that the Reynolds stresses are related to the rates of strain. The force coefficient q_6 is included because it is expected that the magnitude of \mathbf{b}^Δ is dependent on the applied forcing of the turbine.

Table 9.1: Features used for modeling correction term \mathbf{b}^Δ .

n	Feature type	Features
6	Basis tensors	$\mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}, \mathbf{T}^{(4)}, \mathbf{T}^{(6)}, \mathbf{T}^{(8)}$
2	Invariants	λ_1, λ_2
2	Additional invariants	$\mathbf{\Omega A}_k \mathbf{A}_p \mathbf{S}^2, \mathbf{\Omega A}_p \mathbf{S A}_k \mathbf{S}^2$
3	Physical features	q_2, q_6, q_7

9.2. Results

The features denoted in Table 9.1 will be used to learn sparse regression models for \mathbf{b}^Δ . The procedure is initiated with the reduced integrity basis consisting of the first four basis tensors and two invariants, as used by Schmelzer et al. [68] and step-by-step expanded. For the construction of the candidate functions, only the exponent 1.0 is used. Similar as in Section 8.4, the three turbine cases are combined of which 75% is used for training and 25% for testing. The same regularization parameters are used.

9.2.1. Basis tensors

The effects of using different basis tensors are analysed according to the feature sets in Table 9.2. The resulting mean-squared error as function of complexity is depicted in Figure 9.1. Clearly, all feature sets have similar performance, indicating that $\mathbf{T}^{(6)}$ and $\mathbf{T}^{(8)}$ are redundant. Similarly, discarding $\mathbf{T}^{(4)}$ does not result in any diminished predictions, meaning $\mathbf{T}^{(4)}$ can be removed from the feature set safely. This is consistent with the conclusion that $\mathbf{T}^{(3)}$ and $\mathbf{T}^{(4)}$ have a large mutual dependency, from the mutual information estimates.

Table 9.2: Features used for modeling correction term \mathbf{b}^Δ .

Feature set	Features	# functions
$\mathcal{B}_{\mathbf{b}^\Delta}^{(1)}$	$\lambda_1, \lambda_2, \mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}, \mathbf{T}^{(4)}$	24
$\mathcal{B}_{\mathbf{b}^\Delta}^{(2)}$	$\lambda_1, \lambda_2, \mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}, \mathbf{T}^{(6)}$	24
$\mathcal{B}_{\mathbf{b}^\Delta}^{(3)}$	$\lambda_1, \lambda_2, \mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}$	18
$\mathcal{B}_{\mathbf{b}^\Delta}^{(4)}$	$\lambda_1, \lambda_2, \mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}, \mathbf{T}^{(8)}$	24

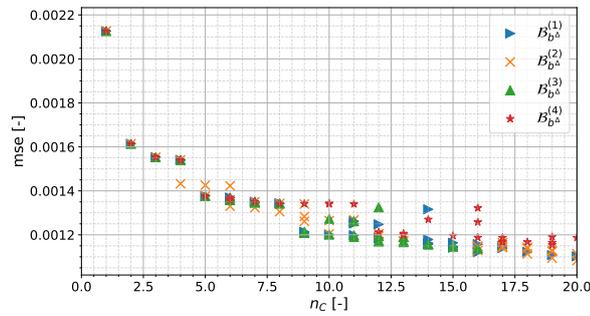


Figure 9.1: Mean-squared error of components \mathbf{b}^Δ as function of model complexity n_C for feature sets described in Table 9.2.

It is noteworthy that $\mathbf{T}^{(6)}$ does not result in any error reduction while the estimated mutual information with the off-diagonal components of \mathbf{b}^Δ was even slightly larger than for $\mathbf{T}^{(1)}$. Scatter plots of the off-diagonal component (1,2) between $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(6)}$ are depicted in Figures 9.2 and 9.3, respectively. Clearly, the relation for $\mathbf{T}^{(1)}$ is roughly linear while for $\mathbf{T}^{(6)}$ it is more S-shaped. Because \mathbf{b}^Δ is built from a linear combination of the basis tensors, it can be more difficult to incorporate the non-linear trends. Another reason can be that $\mathbf{T}^{(6)}$ results in a good fit with certain components of \mathbf{b}^Δ , but in a weak fit to the other components. Meaning that the average performance remains low.

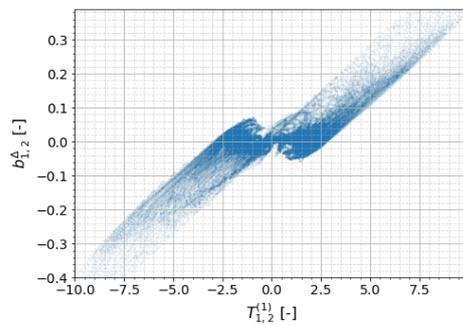


Figure 9.2: Component $b_{1,2}^\Delta$ as function of $T_{1,2}^{(1)}$.

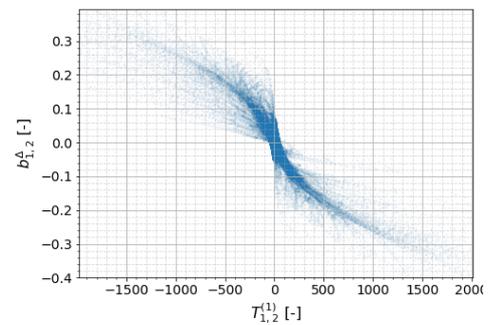


Figure 9.3: Component $b_{1,2}^\Delta$ as function of $T_{1,2}^{(6)}$.

9.2.2. Additional invariants

The two additional invariants in Table 9.1 are introduced and the subsequent feature sets are listed in Table 9.3. The mean-squared error of the \mathbf{b}^Δ components as function of complexity is depicted in Figure 9.4. The addition of the invariants does not result in a reduction in error, it is even observed that it can result in an increase in error (due to the reasons listed in Section 8.4.3). It is not a surprise that the additional invariants do not contribute to the predictions of \mathbf{b}^Δ , as the estimated mutual information was low.

Table 9.3: Features used for modeling correction term \mathbf{b}^Δ .

Feature set	Features	# functions
$\mathcal{B}_{\mathbf{b}^\Delta}^{(5)}$	$\lambda_1, \lambda_2, \{\mathbf{\Omega A}_k \mathbf{A}_p \mathbf{S}^2\}, \mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}$	42
$\mathcal{B}_{\mathbf{b}^\Delta}^{(6)}$	$\lambda_1, \lambda_2, \{\mathbf{\Omega A}_p \mathbf{S A}_k \mathbf{S}^2\}, \mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}$	42

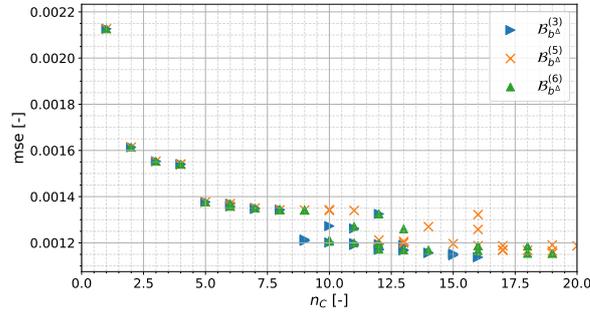


Figure 9.4: Mean-squared error of \mathbf{b}^Δ components as function of model complexity for feature sets described in Table 9.3.

9.2.3. Physical flow features

The physical flow features used are q_2 (turbulence intensity 1), q_6 (force coefficient) and q_7 (shear parameter). The considered feature sets are listed in Table 9.4 and the mean-squared error of the \mathbf{b}^Δ components as function of model complexity is depicted in Figure 9.5. It can be clearly seen that the introduction of q_2 results in significant reduction of mean-squared error as both $\mathcal{B}_{\mathbf{b}^\Delta}^{(7)}$ and $\mathcal{B}_{\mathbf{b}^\Delta}^{(9)}$ have a smaller error compared to the other feature sets. The shear parameter q_7 does not result in similar improvements as when q_2 is used. This is because the mathematical definition of q_7 is very similar to that of λ_1 and λ_2 . While q_7 is constructed from the norm of the velocity gradient tensor, λ_1 and λ_2 are formed by the trace of the strain-rate tensor squared and rotation-rate tensor squared, respectively. Both tensors added together equals the velocity gradient tensor. Because λ_1 and λ_2 are already incorporated in the feature set, it is no surprise that the addition of q_2 results in more reduction of mean-squared error than q_7 .

Although feature sets $\mathcal{B}_{\mathbf{b}^\Delta}^{(7)}$ and $\mathcal{B}_{\mathbf{b}^\Delta}^{(9)}$ have similar values regarding the mean-squared error, $\mathcal{B}_{\mathbf{b}^\Delta}^{(7)}$ has significantly smaller maximum-squared error, as depicted in Figure 9.6. Feature set $\mathcal{B}_{\mathbf{b}^\Delta}^{(7)}$ includes the force coefficient q_6 , meaning it has improved predictions in the region near the turbine. This explains the reduction in maximum-squared error, as the magnitude of \mathbf{b}^Δ near the turbine is significant. It is therefore decided that the final feature set used is $\mathcal{B}_{\mathbf{b}^\Delta}^{(7)}$.

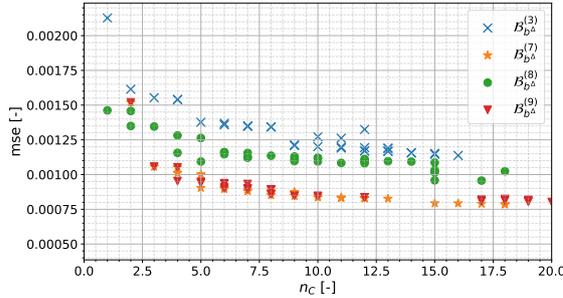
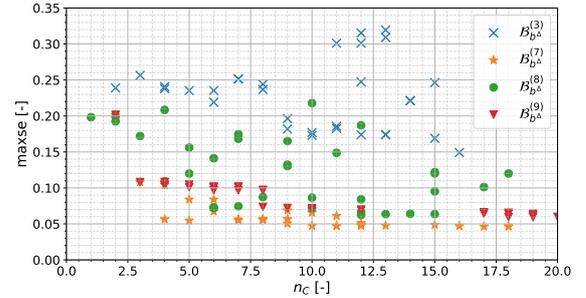
It must be noted that the function $q_2^{1.0} \cdot \mathbf{T}^{(1)}$ is added manually to the discovered models. This is done for the same reason as why $q_8^{1.0}$ was added manually to R_ϵ .

9.3. Comparison training on predicted wake/all samples

Similarly as in Section 8.5, two frameworks are considered. Framework 1 trains the regression models for \mathbf{b}^Δ on all samples of case B and C and tests on the predicted wake samples of case A. Framework 2 trains on the predicted wake samples of case B and C and tests on the predicted wake samples of case A. The feature set used is $\mathcal{B}_{\mathbf{b}^\Delta}^{(7)}$ and the regularization parameters are described in Section 8.3. The mean-squared and

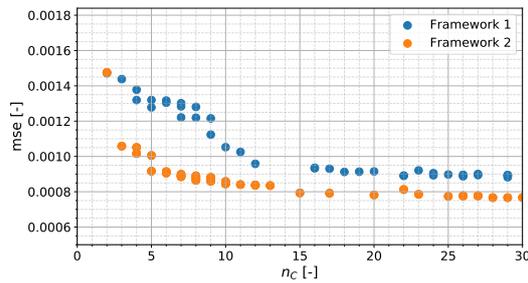
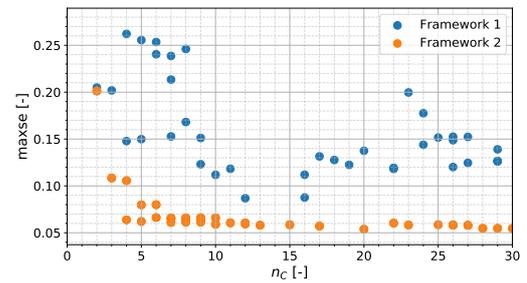
Table 9.4: Features used for modeling correction term \mathbf{b}^Δ .

Feature set	Features	
$\mathcal{B}_{\mathbf{b}^\Delta}^{(7)}$	$\lambda_1, \lambda_2, q_2, q_6, \mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}$	56
$\mathcal{B}_{\mathbf{b}^\Delta}^{(8)}$	$\lambda_1, \lambda_2, q_6, q_7, \mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}$	56
$\mathcal{B}_{\mathbf{b}^\Delta}^{(9)}$	$\lambda_1, \lambda_2, q_2, \mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \mathbf{T}^{(3)}$	38

Figure 9.5: Mean squared error in \mathbf{b}^Δ as function of model complexity.Figure 9.6: Maximum squared error in \mathbf{b}^Δ as function of model complexity.

maximum-squared errors of \mathbf{b}^Δ as function of model complexity are depicted in Figures 9.7 and 9.8, respectively.

It is observed that there is a significant difference in prediction error. While the mean-squared error of framework 1 is high for simple models ($n_C < 10$), framework 2 achieves already reasonable accuracy's already for simple models. It is interesting to note that a model from framework 2 with complexity $n_C = 7$ has a lower mean-squared error than very complex models trained of framework 1. In Figure 9.8, the maximum-squared error in \mathbf{b}^Δ is depicted as function of model complexity. The maximum-squared error is significantly larger for framework 1 compared to framework 2. This entails that, when the regression model is trained on all samples, there remain regions in the turbine wake where the predictions deviate significantly from the true value. This could not only lead to inaccurate predictions, but will also enhance model instability when implemented into a CFD solver. It must be noted that, due to computational limitations, framework 1 uses a larger subsample for training compared to framework 2. However it is not expected that this affects the results significantly.

Figure 9.7: Mean-squared error \mathbf{b}^Δ as function of complexity (number of non-zero components) for models trained on all samples and models trained on wake samples.Figure 9.8: Maximum squared error in \mathbf{b}^Δ as function of complexity (number of non-zero components) for models trained on all samples and models trained on wake samples.

The main effects of \mathbf{b}^Δ on the mean flow fields such as velocity U_i and turbulent kinetic energy k is through the turbulent production term. The turbulent production is affected by \mathbf{b}^Δ through

$$\mathcal{P}_k^\Delta = -2kb_{ij}^\Delta \partial_j U_i. \quad (9.1)$$

The mean-squared error and maximum-squared error in \mathcal{P}_k^Δ are depicted in Figures 9.9 and 9.10, respectively. Again, it is observed that the regression models from framework 2 have a significantly smaller error compared to the models of framework 1. The reductions in mean-squared error can build up to roughly 60% while the reduction of maximum-squared error can go up to more than 90%. Given that the maximum-squared error in \mathcal{P}_k^Δ is very low indicates that the models from framework 2 are able to fit all predicted wake samples to a reasonable level.

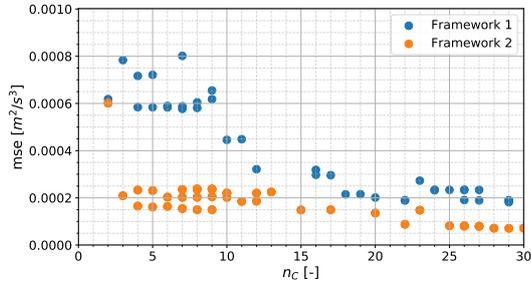


Figure 9.9: Mean-squared error \mathcal{P}_k^Δ for models trained on all samples and models trained on predicted wake samples.

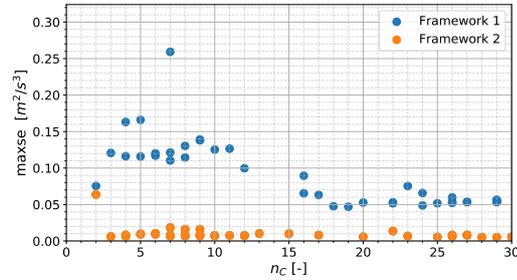


Figure 9.10: Maximum-squared error \mathcal{P}_k^Δ for models trained on all samples and models trained on predicted wake samples.

9.4. Cross validation

The regression models are trained and tested on separate cases to quantify the predictive capabilities on unseen data. The feature set utilized is $\mathcal{B}_{b^\Delta}^{(7)}$ and used to build a library of candidate functions. This library is built using both exponents 0.5 and 1.0. The exponent 0.5 is included because it is found that adding functions containing square-roots lead to better predictions of b^Δ . Because only predicted wake samples are used, the computational costs of the training phase are reduced, meaning there is this possibility to extend the library.

Different sparse models are obtained by applying the model discovery and model inference steps, as described in Section 4.3. The training data consists of the predicted wake samples from case B and C, while the predicted wake samples from case A are used for testing. The regularization parameters are identical to the ones listed in Section 8.3. The mean-squared error in b^Δ as function of model complexity is depicted in Figure 9.11. For simple models, the mean-squared error drops significantly for increasing complexity while it stagnates after roughly $n_C = 7$. The maximum-squared error is shown in Figure 9.12 and behaves similar as the mean-squared error and stagnates after roughly $n_C = 10$. The model chosen for further analysis is depicted by the orange star.

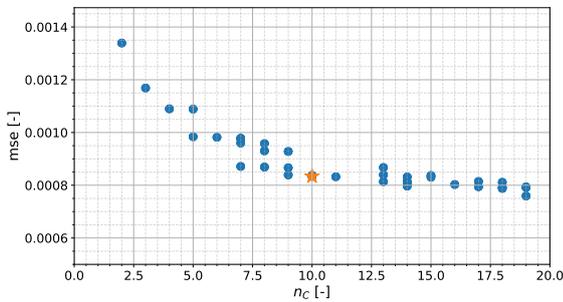


Figure 9.11: Mean squared error b^Δ as function of complexity.

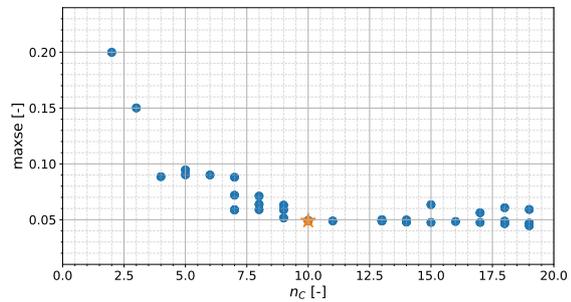


Figure 9.12: Max squared error b^Δ as function of complexity.

The model-form is

$$\begin{aligned}
\mathcal{M}(\mathbf{b}^\Delta) = & \mathbf{T}^{(1)} \cdot [0.059 - 6.93 \cdot 10^{-5} \lambda_1^{0.5} - 1.70 \cdot 10^{-5} \lambda_1^{0.5} q_2^{0.5} + 2.12 \cdot 10^{-8} \lambda_1^{0.5} q_2^2 - \\
& 1.55 \cdot 10^{-4} q_2 + 6.76 \cdot 10^{-6} q_2^{1.5} q_6^{0.5}] + \\
& \mathbf{T}^{(2)} \cdot [2.34 \cdot 10^{-10} \lambda_1^{0.5} q_2^2 + 2.31 \cdot 10^{-11} \lambda_1 q_2^2 - 7.63 \cdot 10^{-8} q_3^2 q_6] + \\
& \mathbf{T}^{(3)} \cdot [-5.03 \cdot 10^{-5} q_2^{1.5} q_6^{0.5}]
\end{aligned} \tag{9.2}$$

Because it is cumbersome to visually compare all \mathbf{b}^Δ tensor components individually, the turbulent production scalar \mathcal{P}_k^Δ is depicted in Figure 9.13 as function of height and width. Clearly, the predictions fit the true values accurately, although downstream of the second turbine, the turbulent production is slightly underestimated.

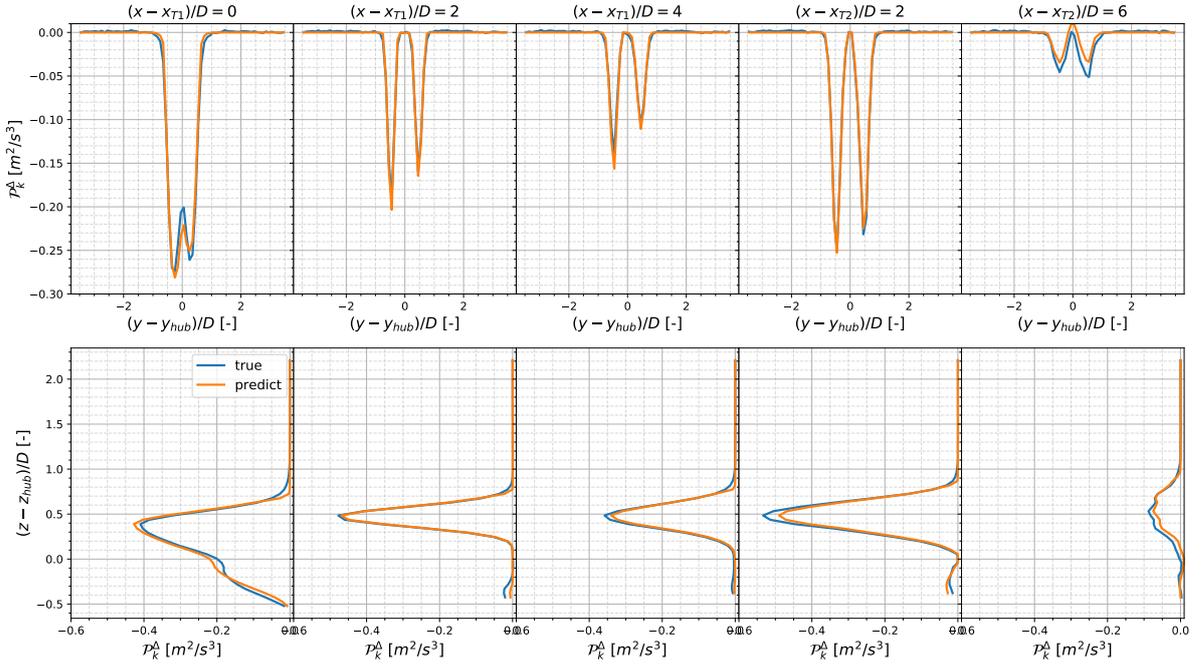


Figure 9.13: Case A. True (blue) and prediction (orange) of \mathcal{P}_k^Δ as function of height z (top row for $(y - y_{hub})/D = 0$) and y (bottom row for $(z - z_{hub})/D = 0$). Relative locations in x with respect to most upstream turbine are denoted in title.

The turbulent production \mathcal{P}_k^Δ on multiple $x - y$ planes is depicted in Figure 9.14. Again, it can be seen that there are no significant discrepancies between the estimated and true values on the depicted planes.

The effect of multiplying the \mathcal{P}_k^Δ results with the classification model is shown in Figure 9.15 for a one-dimensional line in the free-stream flow. Clearly, the true values are slightly positive and grow in magnitude near the wall. The predictions behave similar, however the values are negative. It can be seen that the coupling with the classification model damps this magnitude, although not fully near the wall. Similar to R , the magnitude of \mathcal{P}_k^Δ remains small in the free-stream as it is multiplied with $2k\partial_j U_i$.

9.5. Conclusion

The aim of this chapter was to obtain accurate predictions of \mathbf{b}^Δ using the features discussed in Chapter 6 and training it only on the predicted wake samples. The introduction of physical flow features caused a significant improvement in the predictions, while the effect of the additional invariants was negligible. Mutual information identified in Chapter 6 that a significant number of basis tensors have a dependency with the components of \mathbf{b}^Δ . However, it was found that many of these basis tensors do not contribute to the predictions of \mathbf{b}^Δ , which is probably because the basis tensors are incorporated linearly.

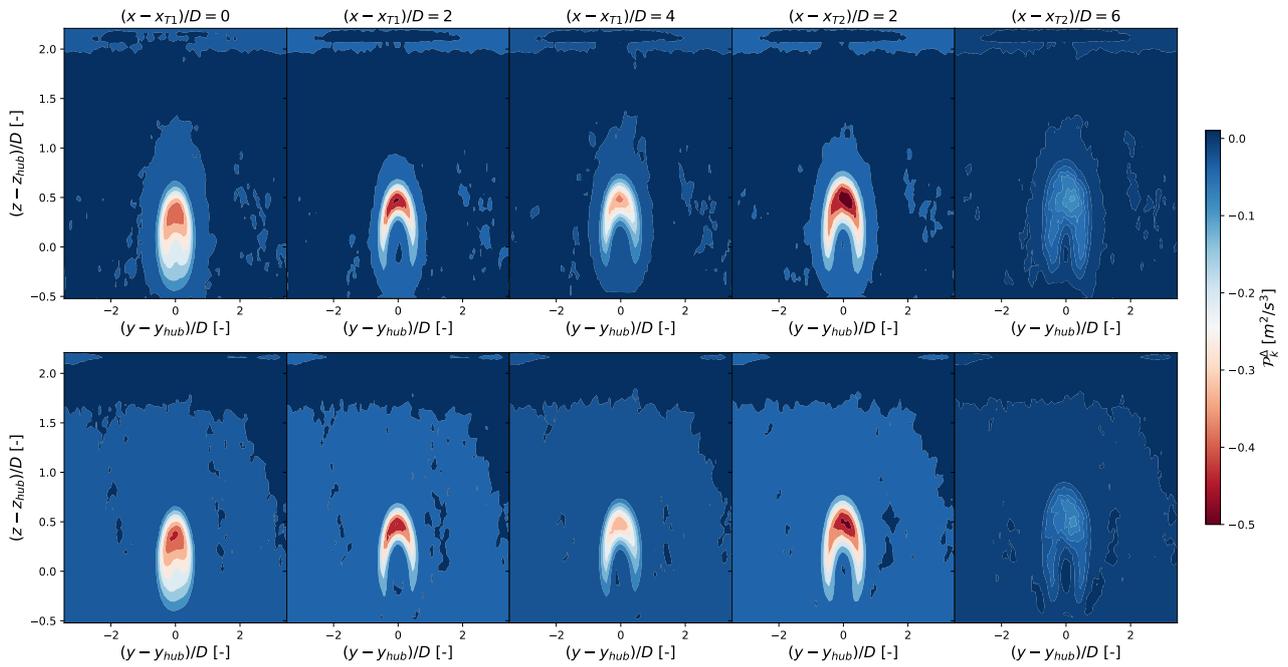


Figure 9.14: True (top row) and prediction (bottom row) of \mathcal{P}_k^Δ for $y-z$ plane. Relative locations in x with respect to turbines are denoted in title.

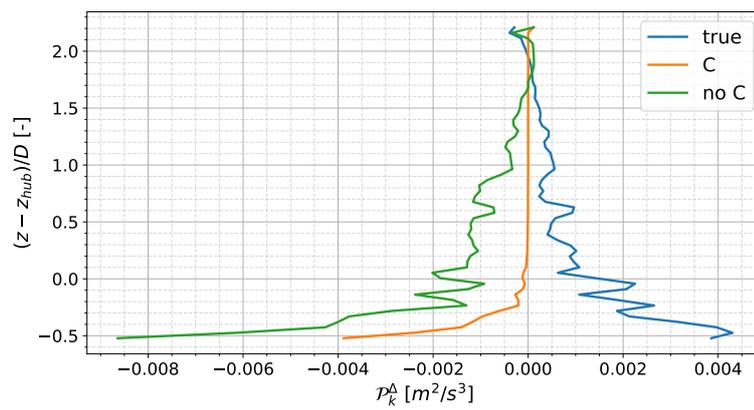


Figure 9.15: Predictions in \mathcal{P}_k^Δ (no C) and predictions in \mathcal{P}_k^Δ multiplied with classification model (C) compared to true values as function of domain width. Location in y : $(y - y_{hub})/D = -3$, location in x : $(x - x_{T2})/D = 10$.

By training the regression model solely on the predicted wake samples, compelling improvements were made compared to when trained on all samples. Not only the mean-squared error was reduced, but also the maximum-squared error was significantly reduced. This also enhances stability when incorporated into a CFD solver.

When applying cross-validation by training and testing the regression models on different turbine cases, already accurate models were obtained for low model complexity. Although the results look promising, this is not yet a guarantee that the models will lead to good predictions when implemented in CFD.

By applying the wake classification model, the predictions in the free-stream are damped. Although the predictions remain relatively small when the classification model is not used, it could cumulatively have an effect on the mean-flow predictions.

10

CFD Implementation

The constructed algebraic models for the wake classification, R and \mathbf{b}^A are implemented into the OpenFOAM RANS CFD solver for wind turbine case A. First, a general description of the RANS setup is provided in Section 10.1. Second, the algebraic models used are denoted in Section 10.2. The predictions are then first injected as a static correction field to validate that the correction models can lead to improved mean-flow predictions, described in Section 10.3. The final phase is to integrate and couple the algebraic models to the turbulence model equations, described in Section 10.4.

10.1. CFD setup

The RANS simulations are initially setup by Steiner et al. [74] while the author of this thesis has implemented the (in this work) derived data-driven algebraic models.

The physical lay-out of case A is depicted in Figure 10.1. The flow enters the domain at the western boundary and leaves at the eastern. The case parameters are identical to the LES parameters listed in Table 5.1. This also means the same grid for RANS is used as for LES, a uniform spaced grid with resolution of $360 \times 120 \times 64$. Normalized by the rotor diameter, the uniform grid spacing is $0.1D \times 0.1D \times 0.05D$. Although RANS simulations could be performed on coarser grids, it is chosen to use the same grid as in LES for simplicity and to avoid interpolation errors. Due to time constraints, no grid refinement study is performed. However, a grid refinement study is important to verify a RANS setup, as many NLEVMs are prone to instabilities for finer grids. For example, van der Laan [79] tested numerous NLEVMs and found that all of them behaved unstable for small grid spacing. The instability was dependent on the number of higher order terms in the stress-strain relation. For future work it is therefore recommended to perform an extensive grid refinement study.

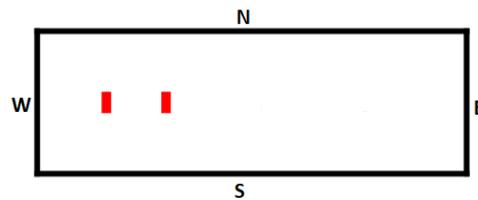


Figure 10.1: Lay out case A. Turbine's are depicted in red. The boundaries are denoted by their cardinal direction. Inflow is from west.

The inflow and turbine characteristics correspond to the properties of the wind tunnel experiment from Chamorro and Porté-Agel [12]. The inflow properties are extracted from the LES precursor. The turbine forces are represented by an actuator disk model, which calculates the lift and drag forces for each cell to determine the body force vector. The rotational speed of the turbines is remained constant.

The boundary conditions are listed in Table 10.1. For all RANS variables, the northern and southern boundaries are coupled through the `Cyclic` boundary condition. The variables U_i , k and ϵ have a `FixedValue` inflow at the western boundary, extracted from the LES precursor. For the outlet at the eastern boundary, either a zero gradient condition is applied or the values are fixed to zero. For U_i , the `inletOutlet` boundary condition is used, which is similar to the zero gradient condition but it additionally sets reversed flows to

zero. For k , the `kqRWallFunction` boundary condition is implemented that sets the gradient to zero at the wall while for the dissipation rate ε the `epsilonWallFunction` condition at the wall is used to compute ε either from viscous or inertial sublayer assumptions. The turbulent viscosity ν_t at the wall is defined by the log-law where the local friction velocity is related to the turbulent kinetic energy at the first cell height. This is done through the `nutkAtmRoughWallFunction` boundary condition [56].

Table 10.1: Boundary conditions RANS solver.

Boundary	Lower	Upper	West	South	East	North
U_i	0	Slip	FixedValue	Cyclic	inletOutlet	Cyclic
p_{rgh}	fixedFluxPressure	fixedFluxPressure	fixedFluxPressure	cyclic	$\frac{\partial p}{\partial x} = 0$	Cyclic
k	kqRWallFunction	Slip	FixedValue	Cyclic	$\frac{\partial k}{\partial x} = 0$	Cyclic
ε	epsilonWallFunction	Slip	FixedValue	Cyclic	$\frac{\partial \varepsilon}{\partial x} = 0$	Cyclic
ν_t	nutkAtmRoughWallFunction	Slip	$\frac{\partial \nu_t}{\partial x} = 0$	Cyclic	$\frac{\partial \nu_t}{\partial x} = 0$	Cyclic
τ_{ij}	0	0	0	Cyclic	0	Cyclic
b_{ij}^Δ	0	0	0	Cyclic	0	Cyclic
R	0	0	0	Cyclic	0	Cyclic

The numerical schemes used for the different mathematical operators are listed in Table 10.2. An elaborate explanation of the schemes is described by OpenCFD [56].

Table 10.2: Numerical schemes RANS solver.

Operator	Scheme	Order
$\frac{\partial}{\partial t}$ (time)	steadyState	
∇ (gradient)	cellLimited Gauss linear 1.0	second
$\nabla \cdot$ (divergence)	U_i : bounded Gauss linearUpwind grad(U) $k, \varepsilon, \tau_{ij}$: bounded Gauss upwind remaining: Gauss linear	second first second
∇^2 (Laplacian)	Gauss linear corrected	second
Surface-normal gradient	corrected	second

The model coefficients of the transport equations in Section 2.2.3 are

$$C_\mu = 0.055, \quad C_{\varepsilon 1} = 1.42, \quad C_{\varepsilon 2} = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3, \quad \kappa = 0.4, \quad (10.1)$$

where C_μ is defined by the wall friction and turbulent kinetic energy at the wall.

The correction terms b^Δ and R are directly implemented in the transport equations. Because the algebraic corrections will have an exponentially growing magnitude when approaching the wall, its values near the wall are damped using a blending term F . The values of the correction terms will grow exponentially because they dependent highly on $\partial_j U_i$, k and ε , which magnitudes grow significantly near the wall. The blending term is added for practical purposes to improve numerical stability. The blending term F is defined as

$$F = \begin{cases} \tanh \left[\left(\frac{z}{z_{\text{lower}}} \right)^\alpha \right] & \text{for } z \leq z_{\text{mid}} \\ \tanh \left[\left(\frac{z_{\text{max}} - z}{z_{\text{max}} - z_{\text{upper}}} \right)^\alpha \right] & \text{for } z > z_{\text{mid}} \end{cases} \quad (10.2)$$

where $\alpha = 4$ specifies how quick the term grows from 0 to 1, $z_{\text{mid}} = 0.23$ and $z_{\text{max}} = 0.46$ are obtained from the domain dimensions while $z_{\text{lower}} = 0.05$ and $z_{\text{upper}} = 0.44$ are the threshold parameters. The correction terms are then blended through

$$\tilde{R} = F \cdot R \quad (10.3)$$

$$\tilde{\mathbf{b}}^\Delta = F \cdot \mathbf{b}^\Delta. \quad (10.4)$$

Correction R is only blended near the lower wall (so only the upper criteria in (10.2) is applied) while \mathbf{b}^Δ is blended both near the wall and the top of the domain.

As was mentioned in Section 4.1, free-stream corrections are added to all RANS simulations such that the free-stream profiles of velocity and turbulent kinetic energy match those from LES. These corrections are incorporated into the transport equations as R_{FS} and \mathbf{b}_{FS}^Δ and obtained from a LES precursor.

10.2. Algebraic models

The algebraic expression for the wake classification model is

$$\mathcal{M}_{\text{wake}} = \sigma \left[1471 q_2 q_9 + 248.8 q_2 q_8 + 4.77 q_2 q_7^2 - 6.51 q_8 + 1.02 \right] \quad (10.5)$$

where $\sigma(x) = 1/(1 + \exp(-x))$ and is identical to the model discussed in Section 7.3.

Three combinations of models for R and \mathbf{b}^Δ are used, hand-picked based on performance and complexity. The models are referred to as simple, medium and complex, based on their number of non-zero components.

The simple models for \mathbf{b}^Δ and R are

$$\begin{aligned} \mathcal{M}_{\mathbf{b}^\Delta}^{(1)} = & \mathbf{T}^{(1)} \cdot [4.08 \cdot 10^{-4} \lambda_1^{0.5} + 5.21 \cdot 10^{-4} \lambda_1^{0.5} q_2^{0.5} - 1.52 \cdot 10^{-6} \lambda_1 q_2 - \\ & 7.08 \cdot 10^{-5} q_2 - 1.70 \cdot 10^{-5} q_2 q_6^{0.5} + 1.11 \cdot 10^{-5} q_2^{1.5} q_6^{0.5}] + \\ & \mathbf{T}^{(2)} \cdot [6.87 \cdot 10^{-9} \lambda_1^{0.5} q_2^{1.5}] \end{aligned} \quad (10.6)$$

and

$$\begin{aligned} \mathcal{M}_R^{(1)} = & 2k\partial_j U_i (\mathbf{T}^{(1)} \cdot [-0.018 q_6 + 0.0031 q_8]) + \\ & \varepsilon (-0.055 q_6 q_7 q_8 - 6.87 q_6 q_8 + 0.0055 q_7 - 0.25 q_8 - 3.17 \cdot 10^{-4} q_8 A_1 A_2), \end{aligned} \quad (10.7)$$

both have 7 non-zero components.

The medium models are

$$\begin{aligned} \mathcal{M}_{\mathbf{b}^\Delta}^{(2)} = & \mathbf{T}^{(1)} \cdot [0.059 - 6.93 \cdot 10^{-5} \lambda_1^{0.5} - 1.70 \cdot 10^{-5} \lambda_1^{0.5} q_2^{0.5} + 2.12 \cdot 10^{-8} \lambda_1^{0.5} q_2^2 - \\ & 1.55 \cdot 10^{-4} q_2 + 6.76 \cdot 10^{-6} q_2^{1.5} q_6^{0.5}] + \\ & \mathbf{T}^{(2)} \cdot [2.34 \cdot 10^{-10} \lambda_1^{0.5} q_2^2 + 2.31 \cdot 10^{-11} \lambda_1 q_2^2 - 7.63 \cdot 10^{-8} q_3^2 q_6] + \\ & \mathbf{T}^{(3)} \cdot [-5.03 \cdot 10^{-5} q_2^{1.5} q_6^{0.5}] \end{aligned} \quad (10.8)$$

and

$$\begin{aligned} \mathcal{M}_R^{(2)} = & 2k\partial_j U_i (\mathbf{T}^{(1)} \cdot [-0.017 q_6 + 0.003 q_8]) + \\ & \varepsilon (-0.06 q_6 q_7 q_8 - 7.19 q_6 q_8 + 0.049 q_7 - 1.5 \cdot 10^{-5} q_7 A_1^2 - 0.36 q_8 + 1.05 \cdot 10^{-4} q_8 A_1 A_2 + 0.04 A_2) \end{aligned} \quad (10.9)$$

with complexity 10 and 9, respectively.

Finally, the complex models are

$$\begin{aligned} \mathcal{M}_{\mathbf{b}^\Delta}^{(3)} = & \mathbf{T}^{(1)} \cdot [5.82 \cdot 10^{-2} - 1.38 \cdot 10^{-4} \lambda_1^{0.5} - 7.80 \cdot 10^{-6} \lambda_1^{0.5} q_2^{0.5} + 2.06 \cdot 10^{-8} \lambda_1^{0.5} q_2^2 + \\ & 5.28 \cdot 10^{-12} \lambda_1^{1.5} \lambda_2 - 1.53 \cdot 10^{-4} q_2 + 7.0 \cdot 10^{-6} q_2^{1.5} q_6^{0.5}] + \\ & \mathbf{T}^{(2)} \cdot [1.14 \cdot 10^{-7} \lambda_1^{0.5} \lambda_2 - 2.71 \cdot 10^{-10} \lambda_1^{0.5} q_2^2 - 3.74 \cdot 10^{-9} \lambda_1 \lambda_2 + 1.01 \lambda_1 q_2^2 - 8.10 \cdot 10^{-8} q_2^2 q_6] + \\ & \mathbf{T}^{(3)} \cdot [-1.25 \cdot 10^{-4} - 3.81 \cdot 10^{-7} q_2^{1.5} q_6^{0.5}] \end{aligned} \quad (10.10)$$

and

$$\begin{aligned} \mathcal{M}_R^{(3)} = & 2k\partial_j U_i (\mathbf{T}^{(1)} \cdot [-1.47 \cdot 10^{-8} \lambda_1^2 q_8 - 0.007 q_6 + 0.0056 q_8]) + \\ & \varepsilon(0.49 - 6.86 q_6 - 0.37 q_6 q_7 q_8 + 0.0013 q_6 A_1 A_2 - 6.23 \cdot 10^{-6} q_7 A_1^2 - \\ & 2.33 \cdot 10^{-6} q_7 A_2^2 + 7.40 \cdot 10^{-5} q_7^2 A_1 - 0.58 q_8 - 4.33 \cdot 10^{-5} q_8 A_2^2 + 0.022 A_2) \end{aligned} \quad (10.11)$$

with complexity 14 and 13, respectively. The algebraic wake classification model and the algebraic models for the correction terms are combined through

$$\mathcal{M}_{R(w)/\mathbf{b}^\Delta(w)}^{(i)} = \mathcal{M}_{\text{wake}} \cdot \mathcal{M}_{R/\mathbf{b}^\Delta}^{(i)} \quad (10.12)$$

such that the correction terms are turned on only inside the turbine's wake.

10.3. Static correction

The static correction fields are obtained by inserting the time-averaged LES variables into the algebraic equations. This means that the corrections R and \mathbf{b}^Δ are fixed. The fields are then added to the turbulence model equations without any coupling to the RANS variables. By using a relaxation factor, the correction fields are incrementally introduced to the model equations to promote convergence. The fields are added to the equations after 100 iterations. The relaxation factor then starts from zero and increases with 0.01 with each iteration, meaning the static correction field is fully implemented after 200 iterations. For the wake classification model, (10.5) is used while for R and \mathbf{b}^Δ the medium algebraic models in (10.9) and (10.8) are used, respectively.

The results for the mean stream-wise velocity component U are depicted in Figure 10.2 as function of domain width and height. It can be seen that the static correction gives a significant improvement over the baseline k - ε turbulence model and matches the LES profiles closely. Only far downstream ($(x - x_{T2})/D = 6$) it is seen that the correction starts to deviate from LES, showing that the predictions in this region are not accurate enough.

The turbulent kinetic energy as function of domain width and height is shown in Figure 10.3. Again significant improvements are made compared to the baseline turbulence model, however the predictions also have a discrepancy with the LES profiles which grows in magnitude when moving downstream. Because k is over-predicted with respect to LES, mixing is enhanced in the turbine wake. This can cause the small discrepancy in the velocity profiles, observed in Figure 10.2.

10.4. Coupled correction

The obtained algebraic models are coupled to the k - ε turbulence model equations in OpenFOAM. For both R and \mathbf{b}^Δ , the three model types are implemented and the effects on mean-flow variables are assessed.

10.4.1. Performance

The performances of the three model sets are listed in Table 10.3 as normalized error of velocity and turbulent kinetic energy. The normalized error is the mean-squared error of the coupled correction relative to the mean-squared error of the baseline k - ε turbulence model. The mean-squared error is the mean of the difference squared to the LES variables. For all three model sets the error is reduced, especially for the medium and complex sets the error is significantly smaller than the baseline turbulence model. This means the coupled corrections are able to partly reproduce the model-form error of the k - ε turbulence model, although there remains a portion of the error which also the corrections are not able to account for. While the simple models lead to the smallest error reduction, the medium model performs slightly better than the complex model, indicating that the complex model is numerically more stiff.

Although the error is significantly reduced, the wake deficit (especially more downstream) is still under-predicted compared to LES, as can be seen in Figure 10.4 for the model with medium complexity. This is partly because k remains over-predicted, as is shown in Figure 10.5, meaning the coupled correction models do not drain enough turbulent kinetic energy from the flow.

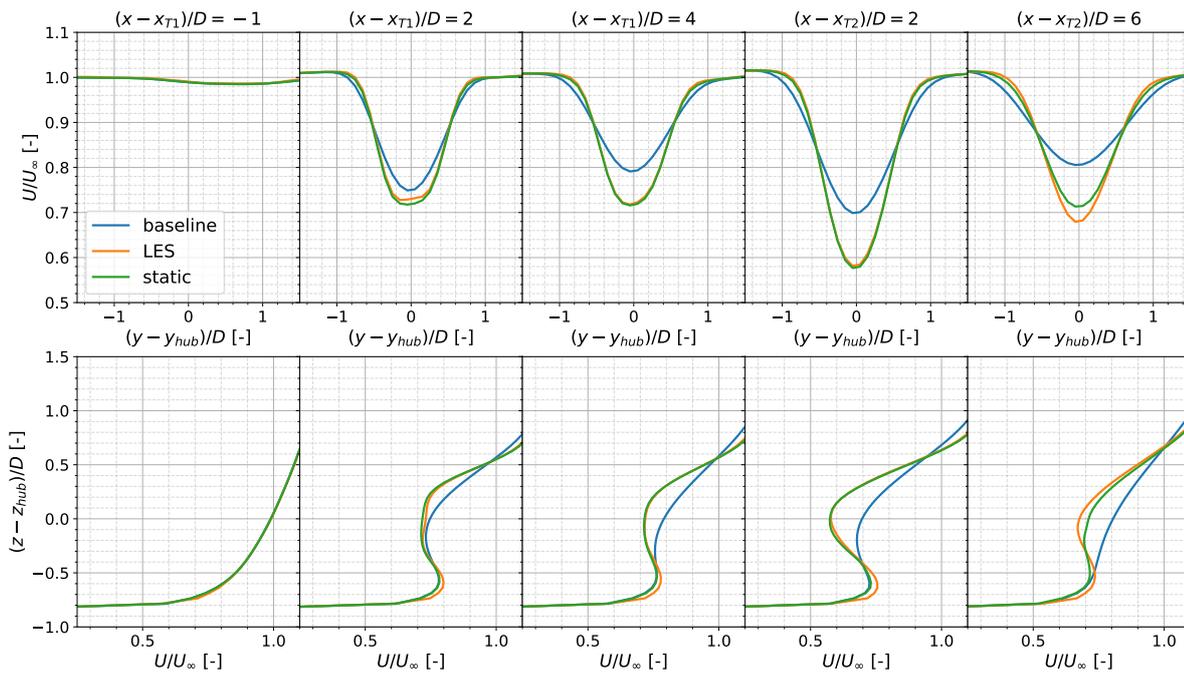


Figure 10.2: Comparison of baseline $k-\epsilon$ RANS, LES and static correction RANS via normalized streamwise velocity as function of domain width and height for different downstream locations with respect to turbines.

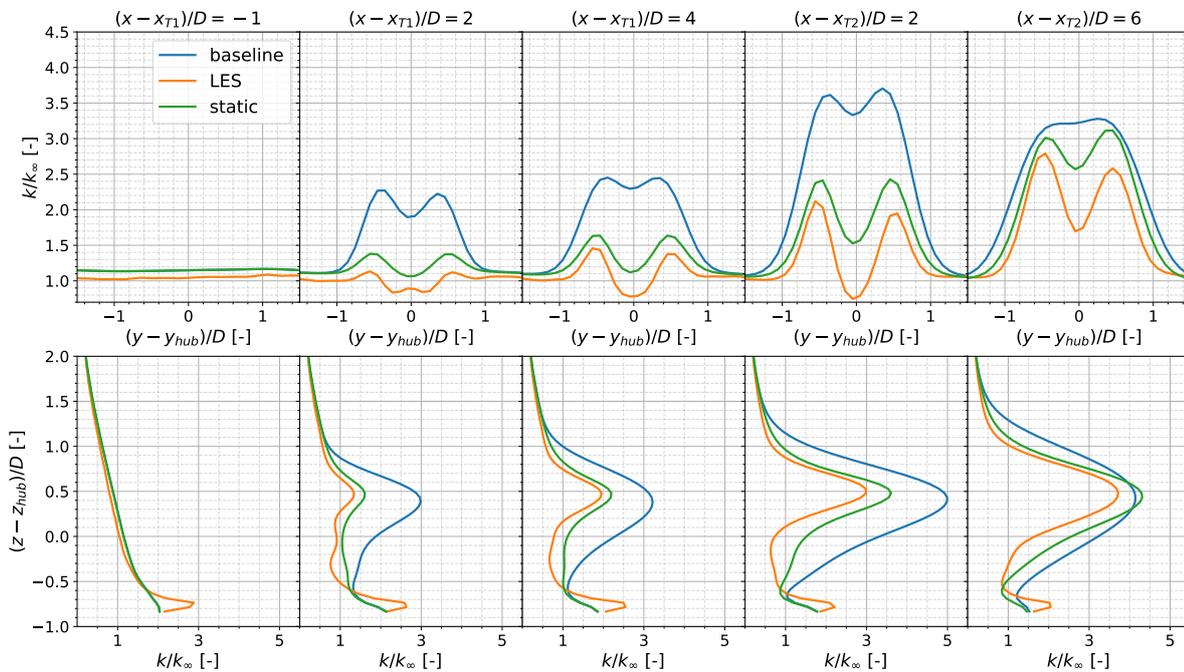
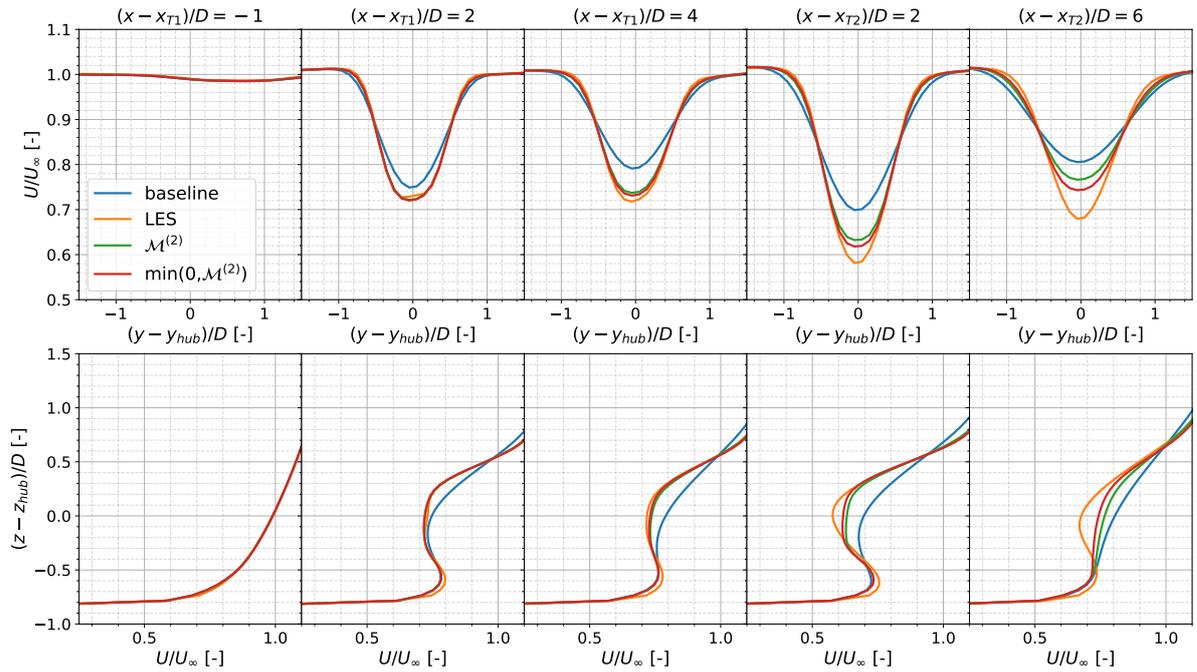


Figure 10.3: Comparison of baseline $k-\epsilon$ RANS, LES and static correction RANS via normalized turbulent kinetic energy as function of domain width and height for different downstream locations with respect to turbines.

Table 10.3: Performance coupled algebraic models on case A using normalized error for velocity and turbulent kinetic energy.

Model	$\epsilon(U)/\epsilon(U^0)$	$\epsilon(k)/\epsilon(k^0)$
Simple $(\mathcal{M}_{b^A(w)}^{(1)}$ and $\mathcal{M}_{R(w)}^{(1)})$	0.66	0.52
Medium $(\mathcal{M}_{b^A(w)}^{(2)}$ and $\mathcal{M}_{R(w)}^{(2)})$	0.38	0.32
Complex $(\mathcal{M}_{b^A(w)}^{(3)}$ and $\mathcal{M}_{R(w)}^{(3)})$	0.39	0.34

Figure 10.4: Comparison of baseline $k-\epsilon$ RANS, LES and two coupled RANS corrections via normalized streamwise velocity as function of domain width and height for different downstream locations with respect to turbines.

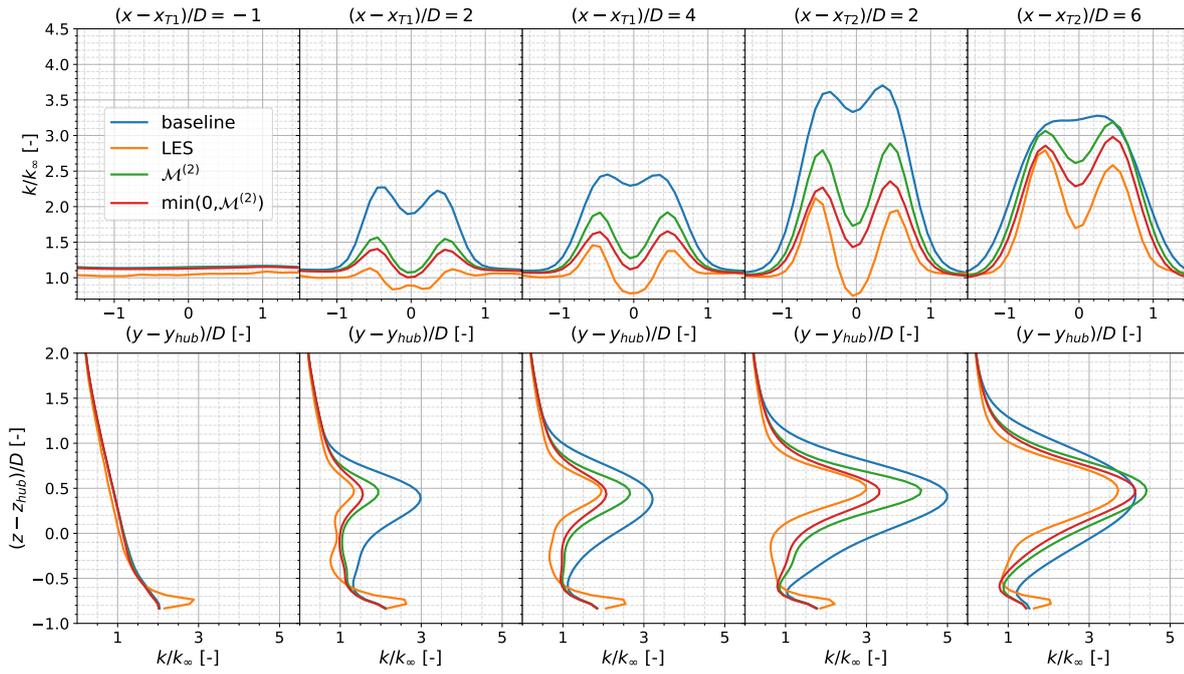


Figure 10.5: Comparison of baseline k - ϵ RANS, LES and two coupled RANS corrections via normalized turbulent kinetic energy as function of domain width and height for different downstream locations with respect to turbines.

Fixed and coupled correction comparison

The training process of the correction terms is performed using the LES variables. This means, when the correction terms are coupled to the RANS solver, the model outcome can be different because the RANS variables are likely to be different than the LES variables. In the ideal case, the outcome of the coupled corrections will converge to the outcome when subjected to the LES variables (fixed), however this cannot be guaranteed.

The robustness (how well the coupled corrections converge to the fixed corrections) of R when coupled to the transport equations is assessed in Figure 10.6. The coupled predictions (predict RANS) are compared to the fixed predictions (when subjected to LES variables) and the true values. Ideally the fixed and RANS predictions would overlap. It is observed that in most regions both predictions are relatively close, although far downstream ($(x - x_{T2})/D = 6$) R is slightly underpredicted.

The robustness of \mathcal{P}_k^Δ is assessed in Figure 10.7. It is clear that the RANS predictions underestimate the magnitude, especially in the horizontal profiles. On top of that, \mathcal{P}_k^Δ in RANS becomes positive far downstream ($(x - x_{T2})/D = 6$) while it actually should be negative.

Although the corrections seem to be reasonable robust as the RANS predictions do not deviate significantly from the fixed predictions, it is observed that the magnitude of \mathcal{P}_k^Δ is systematically underpredicted. This can be one of the reasons why the velocity deficit remains to be underpredicted in the far wake.

Model constraint

As is observed in Figure 10.6, the R correction is actually positive in certain regions of the domain and therefore adds energy to the flow, counteracting the effects of \mathcal{P}_k^Δ . The turbulent production \mathcal{P}_k^Δ was found to be almost entirely negative. However, further downstream it actually becomes positive, as can be seen in Figure 10.7, while it should be negative.

Based on these observations, a fourth model set is constructed, $\mathcal{M}^{(2)*}$, in which all positive model outcomes for R and \mathcal{P}_k^Δ are constraint to zero, implying the correction models can only drain turbulent kinetic energy from the flow and not add energy to it

$$\mathcal{M}^{(2)*} = \min(0, \mathcal{M}^{(2)}). \quad (10.13)$$

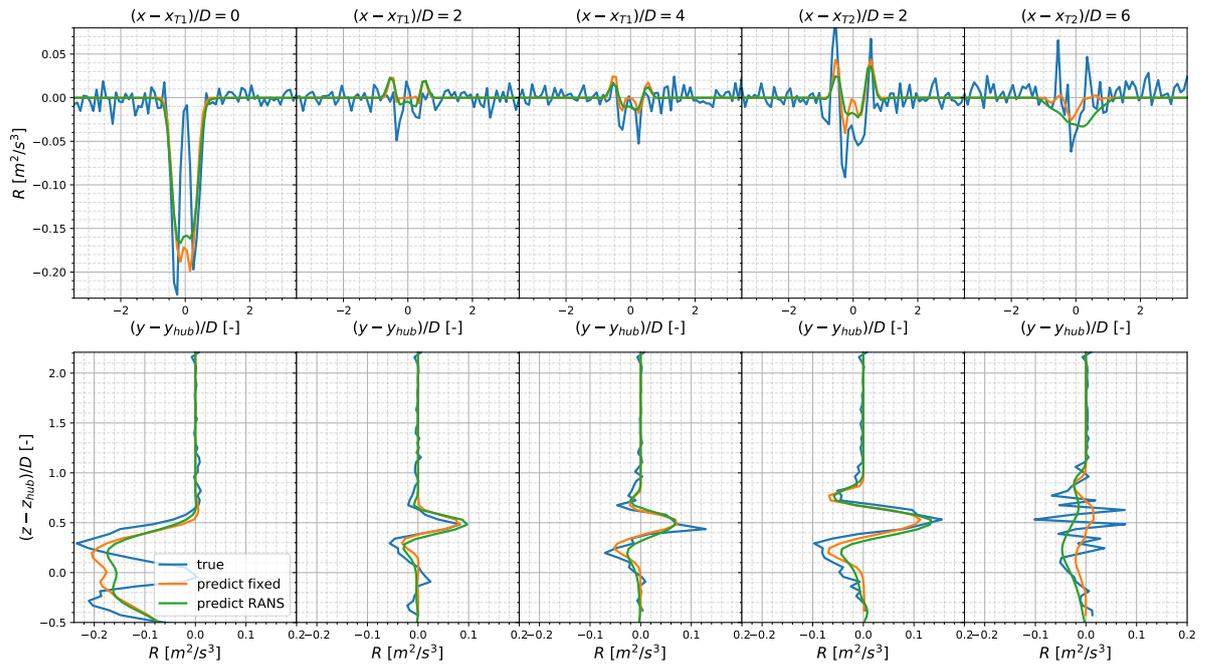


Figure 10.6: Comparison of true, fixed predictions (subjected to LES variables) and RANS predictions of R as function of domain width and height for different downstream locations with respect to turbines.

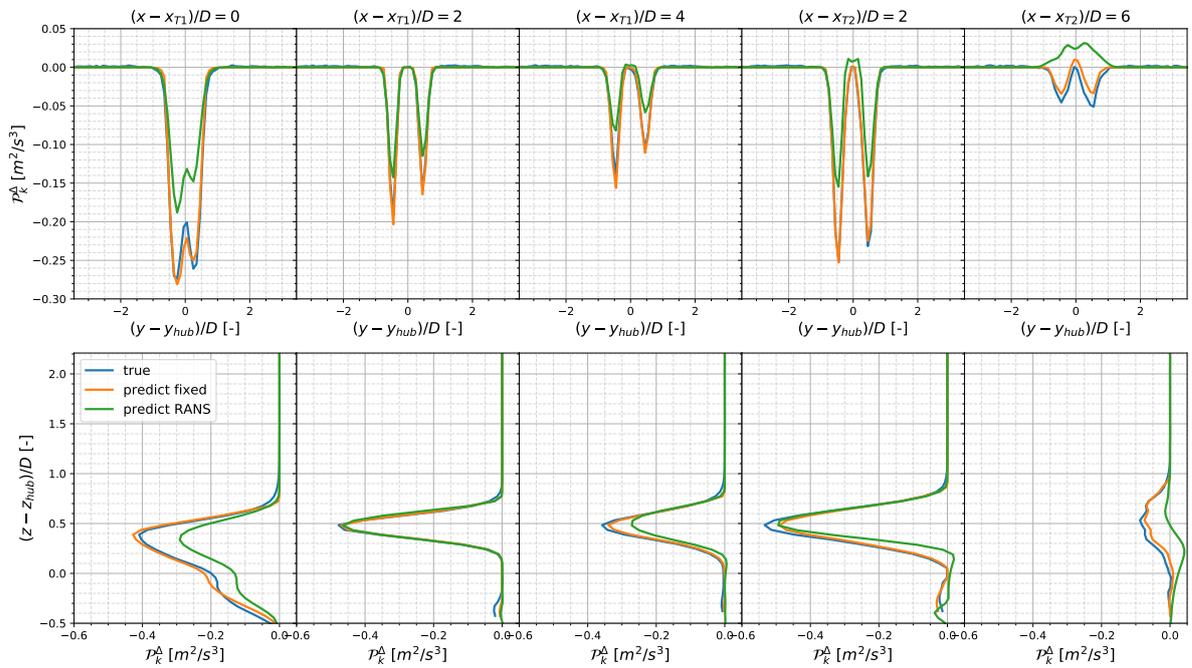


Figure 10.7: Comparison of true, fixed predictions (subjected to LES variables) and RANS predictions of P_k^Δ as function of domain width and height for different downstream locations with respect to turbines.

The normalized mean-squared error for $\mathcal{M}^{(2)*}$ is included in Table 10.4. By limiting the correction models to providing only negative values for R and \mathcal{P}_k^Δ results in a further reduction of the normalized error. Although the results are improved, this approach is slightly ad-hoc, because in other turbine cases it could be that the wake deficit is over-predicted, meaning it is desirable that the correction terms actually add energy to the flow.

Table 10.4: Performance coupled algebraic models on case A using normalized error for velocity and turbulent kinetic energy.

Model	$\epsilon(U)/\epsilon(U^0)$	$\epsilon(k)/\epsilon(k^0)$
Medium $(\mathcal{M}_{\mathbf{b}^\Delta(w)}^{(2)}$ and $\mathcal{M}_{R(w)}^{(2)})$	0.38	0.32
Medium* $(\mathcal{M}_{\mathbf{b}^\Delta(w)}^{(2)*}$ and $\mathcal{M}_{R(w)}^{(2)*})$	0.23	0.15
Medium [†] $(\mathcal{M}_{\mathbf{b}^\Delta(w)}^{(2)}$ and $\mathcal{M}_{R(w)}^{(4)})$	0.45	0.50
Medium ^{††} $(\mathcal{M}_{\mathbf{b}^\Delta(w)}^{(2)}$ and $\mathcal{M}_{R(w)}^{(5)})$	0.34	0.25
Medium ^{†††} $(\mathcal{M}_{\mathbf{b}^\Delta(w)}^{(4)}$ and $\mathcal{M}_{R(w)}^{(6)})$	0.28	0.26

Effect classifier

The algebraic classification model is coupled to the correction models to damp the predictions of R and \mathbf{b}^Δ in the free-stream flow. This is potentially necessary because the correction models are not trained on the free-stream data, and therefore its behaviour in this region is unknown.

To investigate the additional value of the classification model, the medium models in (10.8) and (10.9) are coupled to the transport equations without intervene from the classification model. This simulation did however not sufficiently converge. Because the correction models are not damped outside the wake, they affect the mean flow fields in the free-stream. This causes instabilities with the result that the simulation does not properly converge. The velocity residuals of the simulations with and without classification model are depicted in Figure 10.8 (sufficient convergence is obtained when residuals drop below 10^{-5}). Clearly, the simulation with classifier converges quickly while for the simulation without the classifier, the residuals remain constant.

The effect of the classification model to the correction terms can be analysed by looking at the final time-step of both simulations. In Figure 10.9, \mathcal{P}_k^Δ is depicted as function of domain height in the free-stream flow ($(x - x_{T2})/D = 10$ and $(y - y_{hub})/D = -3$) for both simulations and compared to the values obtained from LES (optimal). The optimal values are slightly positive while for both RANS simulations \mathcal{P}_k^Δ is negative. Using the classification model, the values of \mathcal{P}_k^Δ are significantly damped, bringing them much closer to the optimal values.

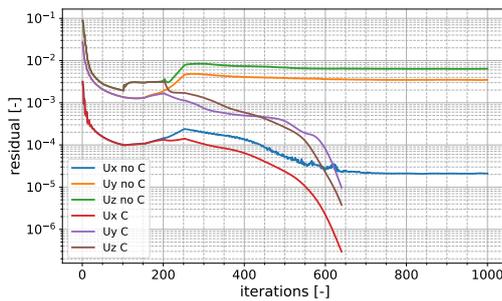


Figure 10.8: Residuals as function of iteration for simulation with classification model (C) and simulation without classification model (no C).

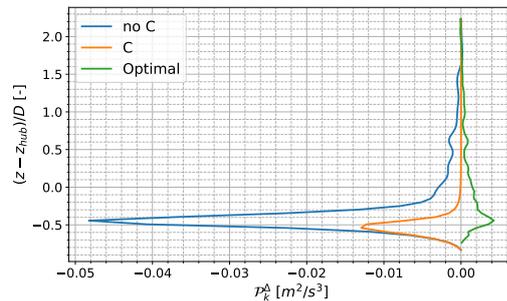


Figure 10.9: Correction \mathcal{P}_k^Δ in free-stream ($(x - x_{T2})/D = 10$ and $(y - y_{hub})/D = -3$) with classification (C) and without classification (no C) compared to true values.

The effect of using the classification model on the turbulent kinetic energy in the free-stream is visualized in Figure 10.10. When no classification model is used, \mathcal{P}_k^Δ has a negative magnitude in the free-stream, which affects the turbulent kinetic energy profile. The same observation is made for a horizontal profile in Figure 10.11. While k is not affected by the wake classification model inside the wake, it can be seen that k is

much smaller outside the wake when no classification model is used. The effects on the velocity profiles was negligible.

An additional note must be made about convergence. It is well known about non-linear eddy viscosity models that they are prone to instabilities and can easily diverge. However, all algebraic models used in this work that are coupled together with the wake classification model converged, implying that damping the correction models in the free-stream flow enhances the stability of the simulations.

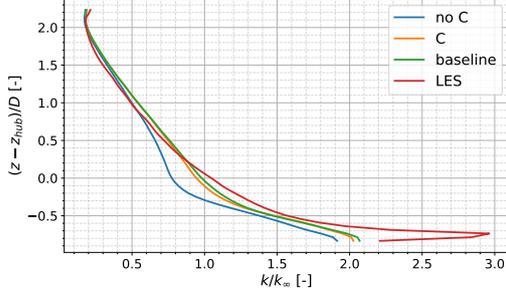


Figure 10.10: Turbulent kinetic energy in free-stream $((x - x_{T2})/D = 10$ and $(y - y_{hub})/D = -3$) with classification (C) and without classification (no C) compared to baseline RANS and LES.

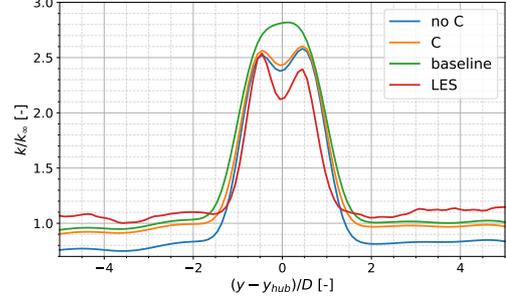


Figure 10.11: Turbulent kinetic energy at horizontal slice of the domain $((x - x_{T2})/D = 10$ and $(z - z_{hub})/D = 0$) with classification (C) and without classification (no C) compared to baseline RANS and LES.

Effect physical form R

In this research, it has been proposed to extend the physical form of R with an additional source/sink term proportional to the local dissipation rate ε . To determine the additional value of this term, an algebraic model $\mathcal{M}_R^{(4)}$ for R is constructed that only consists of the production part using the feature set $\mathcal{B}_{R_P}^{(8)}$ in Table 8.4. The number of non-zero components equals 9, similar as the medium model for R in (10.9)

$$\begin{aligned} \mathcal{M}_R^{(4)} = & 2k\partial_j U_i(\mathbf{T}^{(1)}) \cdot [-0.011 - 1.65 \cdot 10^{-7} \lambda_1 q_7^2 + 7.10 \cdot 10^{-9} \lambda_1^2 q_7 + 4.80 \cdot 10^{-5} q_1 q_7 - \\ & 3.72 \cdot 10^{-7} q_1^2 q_7 - 0.047 q_6 + 1.16 \cdot 10^{-4} q_7 A_1 + 1.44 \cdot 10^{-4} A_1] + \\ & \mathbf{T}^{(3)} \cdot [-5.20 \cdot 10^{-5} \lambda_1 q_6^2]. \end{aligned} \quad (10.14)$$

The performance is listed in Table 10.4 (Medium[†]). Clearly, this model results in much less error reduction than the medium model with the extended physical-form for R . This implies that extending the physical-form of R with the source/sink term proportional to ε does improve the prediction results.

Similarly, an algebraic model for R is constructed, consisting only of the ε term and of equal complexity

$$\begin{aligned} \mathcal{M}_R^{(5)} = & \varepsilon [-12.15 q_6 - 0.603 q_6 q_7 q_8 + 5.84 q_6 q_8 + 0.180 q_7 - 6.39 \cdot 10^{-6} q_7 A_1^2 - \\ & 0.601 q_8 - 9.85 \cdot 10^{-5} q_8 A_2^2 + 0.012 A_1 + 0.018 A_2]. \end{aligned} \quad (10.15)$$

The model performance is also listed in Table 10.4 (Medium^{††}). By only using the source/sink term to model R , the relative error in both U and k is further reduced.

Effect training data

The previous mentioned models were all obtained through cross-validation e.g. trained and tested on different data sets. When training and testing the models on the same data, models are obtained that have a better fit with the test data. When comparing the performance of such models to models obtained through cross-validation, it can be quantified how well the cross-validated models can actually generalize to other cases. For both correction terms, models with medium complexity are obtained that were trained on the predicted wake samples from case A. The model-forms are

$$\begin{aligned}
\mathcal{M}_{b^\Delta}^{(4)} = & \mathbf{T}^{(1)} \cdot [1.37 \cdot 10^{-6} \lambda_1 q_2 - 2.50 \cdot 10^{-9} \lambda_1^2 q_2 + 2.81 \cdot 10^{-10} \lambda_1 q_6 - \\
& 7.26 \cdot 10^{-5} \lambda_2 - 2.67 \cdot 10^{-5} q_2 + 4.28 \cdot 10^{-2} q_6] + \\
& \mathbf{T}^{(2)} \cdot [3.29 \cdot 10^{-11} \lambda_1 q_2^2 + 2.60 \cdot 10^{-7} q_2 - 1.08 \cdot 10^{-3} q_6] + \\
& \mathbf{T}^{(3)} \cdot [-1.33 \cdot 10^{-6} q_2]
\end{aligned} \tag{10.16}$$

and

$$\begin{aligned}
\mathcal{M}_R^{(6)} = & 2k\partial_j U_i (\mathbf{T}^{(1)} \cdot [8.96 \cdot 10^{-3} - 1.45 \cdot 10^{-6} \lambda_1 q_8^2 - 3.74 \cdot 10^{-3} q_6]) + \\
& \varepsilon (-22.72 q_6 + 0.92 q_6 q_7 q_8 + 19.44 q_6 q_8 - 1.35 \cdot 10^{-5} q_7 A_1^2 - 0.31 q_8 + 0.045 A_2).
\end{aligned} \tag{10.17}$$

The performance of the models are listed in Table 10.4 (Medium^{†††}). Clearly and as expected, it performs better than models obtained through cross-validation (Medium) as both the relative error in U and k have decreased. This means there is clearly a loss in predictions due to training and testing on separate data, however the loss is not significant. This either means the models generalize well or that the training and test data are relatively similar.

The predictions of the best performing models (Medium*) for U are depicted as contour plot in Figure 10.12. Clearly, the predictions in U are improved compared to the baseline k - ε turbulence model, although in the far-wake the improvements are much less significant. Similarly, the contours of k are depicted in Figure 10.13 showing that the use of the algebraic data-driven models results in better resemblance with the high-fidelity LES data.

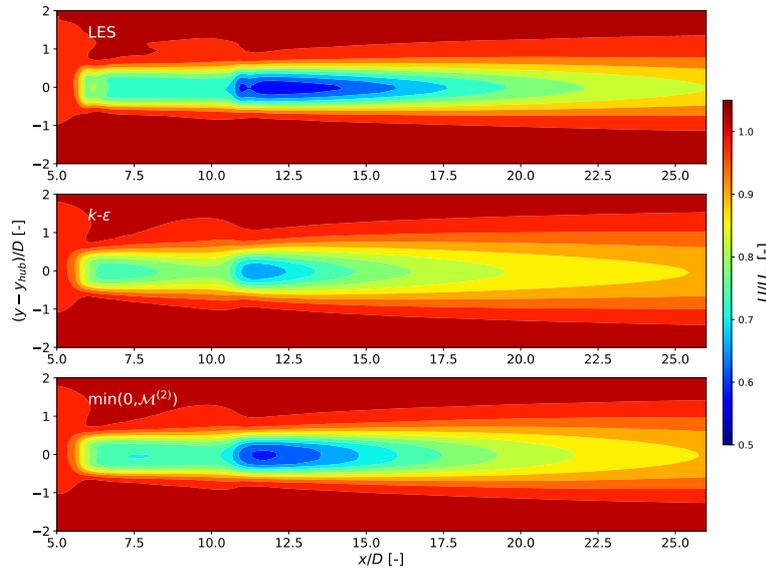


Figure 10.12: Comparison of normalized streamwise velocity of LES, baseline k - ε model and coupled correction at hub-height.

10.4.2. Turbulent stress anisotropy

The turbulence anisotropy componentality is depicted in Figure 10.14 for a horizontal slice at hub-height. In LES, the turbulence in the center of the wake is close to isotropic (blue, as in Figure 2.8) while at the wake edges it is more a mixture of the different turbulence components.

When comparing the baseline k - ε and $\mathcal{M}^{(2)}$ componentality to LES, it is observed that they both do not agree very well. Although $\mathcal{M}^{(2)}$ seems to resemble LES slightly better than the k - ε turbulence model, the

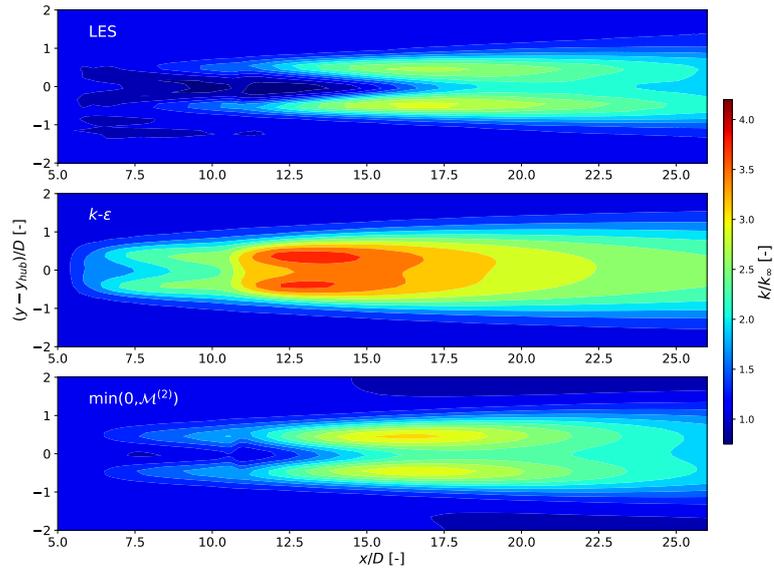


Figure 10.13: Comparison of normalized turbulent kinetic energy of LES, baseline $k\text{-}\varepsilon$ model and coupled correction at hub-height.

main trends are still not captured. This is a disadvantage of using SpARTA as machine learning algorithm as it only identifies the most important trends in the \mathbf{b}^Δ components. This means the more complex flow details are not captured well, which are necessary to correctly predict the turbulence anisotropy.

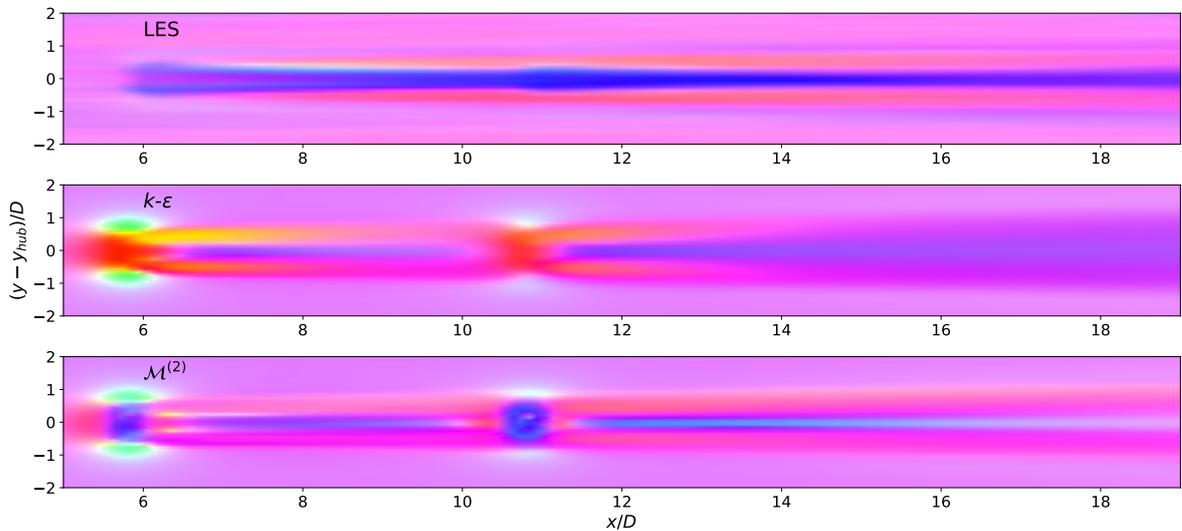


Figure 10.14: Turbulence state depicted by RGB color system at hub height. Top: LES, middle: baseline $k\text{-}\varepsilon$ turbulence model, bottom: $\mathcal{M}^{(2)}$.

10.5. Conclusion

The obtained algebraic models for the wake classification, R and \mathbf{b}^Δ are coupled to the transport equations in OpenFOAM. First a static field has been injected showing that the constructed correction models perform well when LES variables are implemented. Secondly, the algebraic models are coupled to the transport equations. Three model combinations for R and \mathbf{b}^Δ are chosen, based on model complexity. It is found that more

complex models do not necessarily lead to better predictions in mean-flow quantities. This is because increasing complexity can result in more numerically stiff models (less robust) that are less able to cope with the changing input variables in a RANS simulation.

Although already significant improvements were obtained by the selected models, the turbulent kinetic energy is still over-predicted and the velocity deficit remains under-predicted in the far-wake. By setting the positive predictions of R and \mathcal{P}_k^Δ to zero, the predictions are further improved. Besides that, it has been shown that the constructed correction models only lead to the required convergence when the algebraic wake classification model is used. Also, the extension of the physical form of R with the source/sink term results in improved predictions when compared to when only the turbulent production term is used.

It must be noted that the conclusions drawn in this chapter are preliminary, given only a limited number of models have been tested. To get a better understanding of, for example, the effects of model complexity and the physical modeling form on the model performance, a broader range of different models must be tested. Similarly, other features can be considered to observe if they lead to more robust predictions of the correction terms. However due to time constraints, this is out of the scope of the current research.

Conclusion and Recommendations

This work aimed to contribute to the enhancement of data-driven turbulence models for wind turbine wake applications. The principle contribution lies in two aspects: (1) the application of mutual information to measure feature importance and (2) the introduction of the wake classification model framework to specifically construct correction models for the turbine's wake region.

Mutual information is used to measure the general dependency between various features and the correction terms R and \mathbf{b}^Δ . By constructing a sparse logistic regression framework, which is an extension of the SpaRTA algorithm by Schmelzer et al. [69], algebraic classification models are constructed that predicts whether or not a sample location is inside the turbine's wake. The selected features are used to discover sparse algebraic models for both R and \mathbf{b}^Δ which are specifically trained on the predicted wake samples to improve the predictions in this specific region. By coupling the algebraic wake classifier to the models for R and \mathbf{b}^Δ , it is enforced that the corrections are set to close to zero in the freestream flow, as the models are not trained for this region.

All algebraic models are obtained through the cross-validation strategy, by training and testing on different turbine cases. Thereby, the predictive performance of the models is assessed. Finally, the models are implemented into the OpenFOAM CFD solver to assess the improvements in mean-flow data. Based on this work's results, conclusions are drawn in Section 11.1 and recommendations for future research are provided in Section 11.2.

11.1. Conclusions

The conclusions follow the research questions described in Chapter 1. Therefore, the conclusions are drawn in answering the main research questions and are discussed underneath.

RQ1: "In the context of data-driven turbulence modeling, what are the advantages and disadvantages of using mutual information for feature selection?"

Mutual information is chosen as the measure for feature importance as it quantifies the general dependency between two variables. There are multiple methods to rank features, but they all have their limitations (linearity and monotonicity for the correlation coefficient) or are specific to a certain machine learning algorithm (feature ranking in random forests). Given that mutual information depends on the probability distributions of variables, quantifying it exact for continuous variables is not possible. A k NN method is chosen as an estimator based on accuracy and efficiency and used to estimate the dependencies between the features and the correction terms R and \mathbf{b}^Δ . The k NN method relies on two assumptions, that the entropy is estimated by N realizations and that within distance $\varepsilon/2$ around the considered sample, the probability density remains constant.

A broad range of features is obtained from literature. Also, a feature is proposed by the author himself, the force coefficient. Mutual information is used to measure the dependency between the features and the correction terms, but also to measure the dependency between individual features to detect redundancies. It was found that mutual information gives a clear quantitative picture of the importance of the features, but depends largely on the selected region.

From the effective viscosity hypothesis proposed by Pope [59], invariants λ_1 , λ_2 and λ_5 have the largest dependence with both the correction terms. By quantifying the dependencies between the invariants themselves, it was found that the invariants are strongly related to each other in the free-stream flow and also in a large portion of the turbine's wake. The dependencies diminish in regions where $\partial U/\partial z$ and $\partial U/\partial y$ are not the only dominating velocity gradients. This occurs in particular in the regions near the turbines. Similar analysis was performed to the basis tensors, concluding that a significant number of basis tensors show dependency with the components of \mathbf{b}^Δ . With respect to the correction term R , basis tensors $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(6)}$ were found to be most important. The importance of the additional invariants based on ∇k and ∇p depends significantly on the selected region and is affected by the far-wake noise present in R . The physical flow features shear parameter, turbulence intensity, viscosity ratio and wall-based Reynolds number were identified by mutual information as important features towards the correction terms. The measured dependency was significant, implying that introducing them to the feature set leads to improved model predictions.

Although mutual information is a fast way to identify feature importance, it is based on the contribution of all considered samples. It is therefore plausible that features are important in a small portion of the domain while irrelevant in the remaining, resulting in a small mutual information estimate. The choice was therefore made to focus on the wake region, thus waiving the effects of the samples in the free-stream region. Still, the importance of certain features is missed such as of the force coefficient, which remains zero in almost the entire domain except near the turbine, where the dependency is very significant.

Given that mutual information measures the general dependency, which can be highly non-linear, it is not guaranteed that the performance of a machine learning model is improved when a feature is introduced that is highly non-linear. Especially in SpaRTA, only a very limited amount of non-linearity can be included.

Although, there are some limitations to the mutual information methodology, it performs well in general and can be used efficiently to define feature importance. Given the mutual information results, it remains however to the user to make a choice about which features to use for further processing.

RQ2: "Can an algebraic wake classification model be constructed such that the algebraic models for \mathbf{b}^Δ and R are accurately turned off in regions where they were not trained on?"

The predictions of \mathbf{b}^Δ inside the turbine wake are significantly improved when discarding the free-stream samples in the training phase. Thereby it cannot be guaranteed that the predictions of R and \mathbf{b}^Δ are accurate in the free-stream and to prevent large discrepancies in this region, the prediction models can be turned off using a wake classification model. The choice was made to develop a sparse logistic regression framework because it can incorporate a limited amount of non-linearity and produces a sparse algebraic classification model that is easy interpretable and can be straight-forwardly implemented into a CFD solver.

From a limited feature set, non-linear functions are constructed, which are used to learn the algebraic classification models. It was found that good performance, measured by the log loss, accuracy and recall, can already be achieved for simple models. There are certain regions of the wake where the predictions are *false negative*, meaning the classification model labels the samples to the free-stream flow while they actually belong to the wake domain. However, it was observed that the magnitude of the correction terms R and \mathbf{b}^Δ in these regions is small, meaning the misclassification has no severe effects. Therefore, it is concluded that, although based on the limited data sets used in this work, sparse logistic regression as a framework is suitable for classifying the turbine wake.

RQ3: "By learning algebraic models for the correction terms \mathbf{b}^Δ and R specifically for the wind turbine wake region using the most relevant features and combining it with the algebraic wake classification model, what improvements in mean-flow conditions can be achieved when compared to the standard $k-\varepsilon$ turbulence model?"

The feature set derived using the mutual information measure is used to build regression models for both R and \mathbf{b}^Δ using SpaRTA. This is done iteratively using different feature combinations because the sparse regression algorithm can only cope with a limited set of features. Also, this was done to observe the importance of specific features and verify the mutual information results.

It is found that for R the additional invariants result in prediction improvements, but the most significant improvements are made by using the physical flow features. By extending the physical-form of R with a source/sink term proportional to ε , the predictions are further improved. For \mathbf{b}^Δ , the predictions are also improved using the physical flow features while it was found that the chosen additional invariants do not improve the predictions.

It has been shown that significant improvements in predictions inside the turbine wake can be accomplished when the data is only trained on the predicted wake samples. Especially for \mathcal{P}_k^Δ , the turbulent production due to \mathbf{b}^Δ , the mean-squared error and maximum-squared error can reduce up to 60% and 90%, respectively. For correction R , the improvements are much less prominent.

When the predictions for R and \mathbf{b}^Δ are injected into a RANS simulation as static fields, compelling improvements in mean-flow data are made, up to very close to the high-fidelity LES data. It is observed that the only region where the predictions show deviations from LES is the far-wake, indicating that the models lack some predictive performance in this region. When the algebraic models for the wake classifier, R and \mathbf{b}^Δ are coupled to the turbulence model equations, large improvements over the baseline k - ε turbulence model are made, both in mean stream-wise velocity and turbulent kinetic energy. Besides that, it has been shown that including the source/sink term in R results in better predictions compared to when R is modeled only as a turbulent production term. Without the inclusion of the wake classification model, the solver did not sufficiently converge. Because the correction terms are not damped outside the wake, too much turbulent kinetic energy is removed from the flow, which affects the turbulent kinetic energy profiles in the free-stream.

Although the discrepancy in mean velocity and turbulent kinetic energy with respect to the baseline k - ε turbulence model is reduced, there remains enough room for improvements especially in the far-wake predictions.

Finally, in wrapping up this thesis and returning to its main objective, it becomes clear that the k - ε turbulence model uncertainty in wind turbine wake applications can indeed be reduced in applying the two approaches analyzed. It thereby becomes clear that the model performance depends significantly on the features used, meaning it is of importance to apply a methodology that quantifies feature importance. Second, improvements are made in model performance when only trained on the predicted wake region samples. This shows the potential of coupling the correction models to a wake classification model to improve mean-flow predictions.

This research worked with time-averaged data from LES cases on wind-tunnel scale. The next step to discover the potential of the considered approaches, is to extent the work to full-scale cases which resemble more the physics of industrial operating turbines.

11.2. Recommendations

Based on the work performed in this thesis, recommendations can be made for future research.

Machine learning

In this thesis, mutual information is used to measure dependencies between variables. Although a thorough investigation has been performed to detect the importance of the different researched features, the features were then manually picked based on the estimated mutual information. A next step would be to develop an approach that, based on the estimated mutual information between features and correction term and between features themselves, automatically chooses the set of suitable features. This would be more systematic than the manual approach described in this work.

The developed sparse logistic regression framework showed to work well in predicting whether a sample location belongs to either the turbine's wake or the free-stream flow. It would be interesting to see if this framework is also able to detect locations of high turbulence model uncertainty in completely different flow cases. Data-driven turbulence modeling is applied in widespread applications, and this framework could also potentially lead to improved predictions in these. For example, given standard turbulence models are not able to capture well the separation zone of an aircraft wing in stall, it would be interesting to see whether sparse logistic regression can capture this region such that, similar as done in this work, a data-driven correction could be turned on to correct the baseline turbulence model.

There exist numerous machine learning algorithms for classification problems. Sparse logistic regression is chosen because an algebraic model expression is obtained that can easily be implemented into the numerical solver. However, there are also downsides of sparse logistic regression, as only a limited amount of non-linearity can be incorporated, which limits the model performance. To determine what improvements could be made by using more complex classification algorithms, a comparison between different algorithms, varying in complexity, can be made.

The current research focuses on turbines in neutral atmospheric conditions. Although in simulations, it is often assumed that the conditions are neutral, in reality there are always buoyancy effects, which affect the turbulence properties in the atmosphere and turbine wakes. Similarly as to the closure problem in the momentum equation, the temperature equation also requires a model for the heat flux. It would be relevant to investigate the performance of SpARTE to construct data-driven corrections for the heat flux term.

Computational Fluid Dynamics

Three different wind turbine cases have been used for training and testing of the data-driven corrections. The only difference between the cases is the location and orientation of the turbines, meaning testing the generalization of the constructed models is limited. Therefore, It would be of important value to include more diverse cases, such as with different Reynolds number and turbulence intensity. Similarly, the cases are on wind-tunnel scale. It would be interesting to see whether the same results could be obtained on cases where the turbines operate in planetary atmospheric conditions.

During the last decade, numerous (not necessarily data-driven) corrections to the $k-\varepsilon$ turbulence model have been proposed for wind turbine wake applications. It would be valuable to compare the performance of different proposals and make an extensive overview of the advantages and disadvantages of the investigated methods.

Finally, non-linear eddy viscosity models are prone to instabilities for finer grid sizes. To investigate whether the constructed algebraic correction model still provide converged solutions for finer grids, a grid refinement study must be performed. Due to limited amount of time, this has not been done in the current research. However to numerically verify (showing that the convergence is not dependent on the grid size) the algebraic models have to be tested for different (especially finer) grid sizes.

Bibliography

- [1] C. Alinot and C. Masson. " k - ϵ model for the atmospheric boundary layer under various thermal stratifications". *Journal of Solar Energy Engineering*, 127(4):438–443, 2005. URL <https://doi.org/10.1115/1.2035704>.
- [2] S.J. Andersen, J.N. Sørensen, and R. Mikkelsen. "Simulation of the inherent turbulence and wake interaction inside an infinitely long row of wind turbines". *Journal of Turbulence*, 14(4):1–24, 2013. URL <https://dx.doi.org/10.1080/14685248.2013.796085>.
- [3] D. D. Apsley and M. A. Leschziner. "A new low-reynolds-number nonlinear two-equation turbulence model for complex flows". *International Journal of Heat and Fluid Flow*, 19(3):209–222, 1998. URL [https://dx.doi.org/10.1016/S0142-727X\(97\)10007-8](https://dx.doi.org/10.1016/S0142-727X(97)10007-8).
- [4] M. Bastankhah and F. Porté-Agel. "A new analytical model for wind-turbine wakes". *Renewable Energy*, 70:116–123, 2014. URL <https://doi.org/10.1016/j.renene.2014.01.002>.
- [5] R. Battiti. "Using mutual information for selecting features in supervised neural net learning". *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994. URL <https://dx.doi.org/10.1109/72.298224>.
- [6] R.J. Beare, M.K. Macvean, A.A.M. Holtslag, J. Cuxart, I. Esau, J.C. Golaz, M.A. Jimenez, M. Khairoutdinov, B. Kosovic, D. Lewellen, T.S. Lund, J.K. Lundquist, A. McCabe, A.F. Moene, Y. Noh, S. Raasch, and P. Sullivan. "An intercomparison of large-eddy simulations of the stable boundary layer". *Boundary-Layer Meteorology*, 118(2):247–272, 2006. URL <https://dx.doi.org/10.1007/s10546-004-2820-6>.
- [7] A. Betz. "Das maximum der theoretisch möglichen ausnützung des windes durch windmotoren". *Zeitschrift für das gesamte Turbinenwesen*, 26:307–309, 1920.
- [8] J. Bleeg, D. Digraskar, J. Woodcock, and J.F. Corbett. "Modeling stable thermal stratification and its impact on wind flow over topography". *Wind Energy*, 18(2):369–383, 2015. URL <https://doi.org/10.1002/we.1692>.
- [9] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi. *Wind Energy Handbook*. Wiley, Baffins Lane, Chichester, England, 2001.
- [10] N. Carrara and J. Ernst. "On the estimation of mutual information". *Proceedings*, 33(1):31, 2020. URL <https://dx.doi.org/10.3390/proceedings2019033031>.
- [11] L.P. Chamorro and F. Porté-Agel. "A wind-tunnel investigation of wind-turbine wakes: Boundary-layer turbulence effects". *Boundary-Layer Meteorology*, 132(1):129–149, 2009. URL <https://dx.doi.org/10.1007/s10546-009-9380-8>.
- [12] L.P. Chamorro and F. Porté-Agel. "Effects of thermal stability and incoming boundary-layer flow characteristics on wind-turbine wakes: A wind-tunnel study". *Boundary-Layer Meteorology*, 136(3):515–533, 2010. doi: 10.1007/s10546-010-9512-1.
- [13] G. Chandrashekar and F. Sahin. "A survey on feature selection methods". *Computers Electrical Engineering*, 40(1):16–28, 2014. URL <https://dx.doi.org/10.1016/j.compeleceng.2013.11.024>.
- [14] A. Chaudhari, O. Agafonova, A. Hellsten, and J. Sorvari. "Numerical study of the impact of atmospheric stratification on a wind-turbine performance". *Journal of Physics: Conference Series*, 854(012007), 2017. URL <https://dx.doi.org/10.1088/1742-6596/854/1/012007>.
- [15] H.S. Chivae. "*Large Eddy Simulation of Turbulent Flows in Wind Energy*". DTU Wind Energy, 2014.

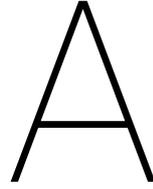
- [16] M. Churchfield. "NWTC information portal (SOWFA)". <https://nwtc.nrel.gov/SOWFA>. Accessed: 2019-11-24.
- [17] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley-Interscience, second edition, 2006.
- [18] K. Duraisamy, G. Iaccarino, and H. Xiao. "Turbulence modeling in the age of data". *Annual Review of Fluid Mechanics*, 51(1):357–377, 2019. URL <https://doi.org/10.1146/annurev-fluid-010518-040547>.
- [19] P.A. Durbin. "On the k - ϵ stagnation point anomaly". *International Journal of Heat and Fluid Flow*, 17, 1996.
- [20] A. El Kasmi and C. Masson. "An extended model for turbulent flow through horizontal-axis wind turbines". *Journal of Wind Engineering and Industrial Aerodynamics*, 96(1):103–122, 2008. URL <https://dx.doi.org/10.1016/j.jweia.2007.03.007>.
- [21] M. Emory and G. Iaccarino. "Visualizing turbulence anisotropy in the spatial domain with componentality contours". *Center for Turbulence Research Annual Research Briefs 2014*, pages 123–138, 2014.
- [22] T. M. K. G. Fernando, H. R. Maier, and G. C. Dandy. "Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach". *Journal of Hydrology*, 367(3-4):165–176, 2009. URL <https://dx.doi.org/10.1016/j.jhydro1.2008.10.019>.
- [23] C. Ferreira. "Gene expression programming: A new adaptive algorithm for solving problems". *Complex Systems*, 13(2):87–129, 2001.
- [24] S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann, S. Larsen, J. Højstrup, and M. Thøgersen. "Analytical modelling of wind speed deficit in large offshore wind farms". *Wind Energy*, 9(1-2):39–53, 2006. URL <https://dx.doi.org/10.1002/we.189>.
- [25] S. Gao, G. Ver Steeg, and A. Galstyan. "Efficient estimation of mutual information for strongly dependent variables". *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 38, 2015. URL <https://arxiv.org/abs/1411.2003>.
- [26] H. Glauert. "airplane propellers". *Aerodynamic Theory*, 4:169–360, 1935. URL http://dx.doi.org/10.1007/978-3-642-91487-4_3.
- [27] C. Gorié, J. Larsson, M. Emory, and G. Iaccarino. "The deviation from parallel shear flow as an indicator of linear eddy-viscosity model inaccuracy". *Physics of Fluids*, 26(5):051702, 2014. URL <https://dx.doi.org/10.1063/1.4876577>.
- [28] T. Göçmen, P.M. van der Laan, P.E. Réthoré, A.P. Diaz, G.C. Larsen, and S. Ott. "Wind turbine wake models developed at the technical university of denmark: A review". *Renewable and Sustainable Energy Reviews*, 60:752–769, 2016. URL <https://doi.org/10.1016/j.rser.2016.01.113>.
- [29] P. Hanchuan, L. Fuhui, and C. Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005. URL <https://dx.doi.org/10.1109/tpami.2005.159>.
- [30] P. E. Hancock and T. D. Farr. "Wind-tunnel simulations of wind-turbine arrays in neutral and non-neutral winds". *Journal of Physics: Conference Series*, 524, 2014. doi: 10.1088/1742-6596/524/1/012166.
- [31] P. E. Hancock and S. Zhang. "A wind-tunnel simulation of the wake of a large wind turbine in a weakly unstable boundary layer". *Boundary-Layer Meteorology*, 156(3):395–413, 2015. doi: 10.1007/s10546-015-0037-5.
- [32] Philip E. Hancock and Frauke Pascheke. "Wind-tunnel simulation of the wake of a large wind turbine in a stable boundary layer: Part 2, the wake flow". *Boundary-Layer Meteorology*, 151(1):23–37, 2013. doi: 10.1007/s10546-013-9887-x.
- [33] T. Herlau, M.N. Schmidt, and M. Mørup. "Introduction to machine learning and data mining". *DTU Lecture notes, Fall 2019, version 1.3*, 2019.

- [34] G.V. Iungo, F. Viola, U. Ciri, M. Rotea, and S. Leonardi. "Data-driven RANS for simulations of large wind farms". *Journal of Physics: Conference Series* 625, (012025), 2015. doi: doi:10.1088/1742-6596/625/1/012025.
- [35] N.O. Jensen. "A note on wind generator interaction". *Roskilde: Risø National Laboratory. Risø-M, No. 2411*, 1983.
- [36] N.E. Joukowsky. "Windmill of the nej type". *Transactions of the Central Institute for Aero-Hydrodynamics of Moscow*, 1920.
- [37] M.L.A. Kaandorp and R.P. Dwight. "Data-driven modelling of the reynolds stress tensor using random forests with invariance". *Computers Fluids*, 202:104497, 2020. URL <https://dx.doi.org/10.1016/j.compfluid.2020.104497>.
- [38] M. Kaandrop. "Machine learning for data-driven RANS turbulence modelling". Master Thesis Report, 2018.
- [39] R. N. King, C. Adcock, J. Annoni, and K. Dykes. "Data-driven machine learning for wind plant flow modeling". *Journal of Physics: Conference Series*, 1037:072004, 2018. URL <https://dx.doi.org/10.1088/1742-6596/1037/7/072004>.
- [40] T. Koblitz. "CFD modeling of non-neutral atmospheric boundary layer conditions". DTU Wind Energy, PhD, no. 0019(EN), 2013.
- [41] T. Koblitz, A. Bechmann, A. Sogachev, N. Sørensen, and P. E Réthoré. "Computational fluid dynamics model of stratified atmospheric boundary-layer flow". *Wind Energy*, (18):75–89, 2013. URL <https://dx.doi.org/10.1002/we.1684>.
- [42] A.N. Kolmogorov. "The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers". *Proceedings: Mathematical and Physical Sciences*, 434(1890):9–13, 1941.
- [43] L.F. Kozachenko and N.N. Leonenko. "Sample estimate of the entropy of a random vector". *Probl. Inf. Transm.*, 23:95–101, 1987.
- [44] A. Kraskov, H. Stögbauer, and P. Grassberger. "Estimating mutual information". *Physical Review E*, 69(6), 2004. URL <https://dx.doi.org/10.1103/PhysRevE.69.066138>.
- [45] N. Kwak and Choi Chong-Ho. "Input feature selection by mutual information based on parzen window". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1667–1671, 2002. URL <https://dx.doi.org/10.1109/TPAMI.2002.1114861>.
- [46] L. Ladha and T. Deepa. "Feature selection methods and algorithms". *International Journal on Computer Science and Engineering (IJCSE)*, 3(5):1887–1797, 2011. ISSN 0975-3397.
- [47] L. E. M. Lignarolo, D. Ragni, C. Krishnaswami, Q. Chen, C. J. Simão Ferreira, and G. J. W. van Bussel. "Experimental analysis of the wake of a horizontal-axis wind-turbine model". *Renewable Energy*, 70: 31–46, 2014. URL <https://doi.org/10.1016/j.renene.2014.01.020>.
- [48] J. Ling and J. Templeton. "Evaluation of machine learning algorithms for prediction of regions of high reynolds averaged navier stokes uncertainty". *Physics of Fluids*, 27(8):085103, 2015. URL <https://doi.org/10.1063/1.4927765>.
- [49] J. Ling, A. Kurzawski, and J. Templeton. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". *Journal of Fluid Mechanics*, 807:155–166, 2016. URL <https://doi.org/10.1017/jfm.2016.615>.
- [50] H. Lu and F. Porté-Agel. "Large-eddy simulation of a very large wind farm in a stable atmospheric boundary layer". *Physics of Fluids*, 23(6), 2011. URL <https://doi.org/10.1063/1.3589857>.
- [51] J.L. Lumley and G.R. Newman. "The return to isotropy of homogeneous turbulence". *Journal of Fluid Mechanics*, 82(01):161–178, 1977. URL <https://dx.doi.org/10.1017/S0022112077000585>.

- [52] Z. Q. Luo and P. Tseng. "On the convergence of the coordinate descent method for convex differentiable minimization". *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992. URL <https://dx.doi.org/10.1007/BF00939948>.
- [53] P.M. Milani, J. Ling, and J.K. Eaton. "Physical interpretation of machine learning models applied to film cooling flows". *Journal of Turbomachinery*, 141(1), 2019. URL <https://doi.org/10.1115/1.4041291>.
- [54] Y.I. Moon, B. Rajagopalan, and U. Lall. "Estimation of mutual information using kernel density estimators". *Physical Review E*, 52(3):2318–2321, 1995. URL <https://dx.doi.org/10.1103/physreve.52.2318>.
- [55] B. Mutlu Sumer. *Lecture notes on turbulence*. Technical University of Denmark, Section for Fluid Mechanics, Coastal Maritime Engineering, revised edition, 2013.
- [56] OpenCFD. *OpenFOAM Programmer's Guide*. Springer, 2004.
- [57] A. Papanas and D. Kugiumtzis. "Evaluation of mutual information estimators on nonlinear dynamic systems". *Nonlinear Phenomena in Complex Systems*, 11, 2008. URL <https://arxiv.org/abs/0809.2149v1>.
- [58] S. B. Pope. "A more general effective-viscosity hypothesis". *Journal of Fluid Mechanics*, 72(02):331, 1975. doi: 10.1017/s0022112075003382.
- [59] S.B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [60] F. Porté-Agel, M. Bastankhah, and S. Shamsoddin. "Wind-turbine and wind-farm flows: A review". *Boundary-Layer Meteorology*, 174:1–59, 2019. URL <https://dx.doi.org/10.1007/s10546-019-00473-0>.
- [61] J.M. Prospathopoulos, P.K. Chaviaropoulos, and Politis E.S. "CFD modeling issues of wind turbine wake under stable atmospheric conditions". *European Wind Energy Conference and Exhibition 2009, EWEC 2009*, 6, 2009.
- [62] Sanz R.J., M. Churchfield, and B. Kosovic. "A methodology for the design and testing of atmospheric boundary layer models for wind energy applications". *Wind Energy Science*, 2(1):35–54, 2017. URL <https://dx.doi.org/10.5194/wes-2-35-2017>.
- [63] B.C. Ross. "Mutual information between discrete and continuous data sets". *PLoS ONE*, 9(2):e87357, 2014. URL <https://dx.doi.org/10.1371/journal.pone.0087357>.
- [64] P.E. Réthoré. "Wind turbine wake in atmospheric turbulence". *Risø National Laboratory, phd thesis*, 2009.
- [65] B. Sanderse, S. P. Pijl, and B. Koren. "Review of computational fluid dynamics for wind turbine wake aerodynamics". *Wind Energy*, 14(7):799–819, 2011. URL <https://doi.org/10.1002/we.458>.
- [66] C. Sanz. "A note on $k-\epsilon$ modelling of vegetation canopy air-flows". *Boundary-Layer Meteorology* 2003, 108:191–197, 2002. doi: 10.1023/A:1023066012766.
- [67] M. Schmelzer, R. Dwight, and P. Cinnella. "Data-driven deterministic symbolic regression of nonlinear stress-strain relation for rans turbulence modelling". In *Fluid Dynamics Conference June*. URL <https://dx.doi.org/10.2514/6.2018-2900>.
- [68] M. Schmelzer, R. Dwight, and P. Cinnella. "Machine learning of algebraic stress models using deterministic symbolic regression". 2019. URL <https://arxiv.org/abs/1905.07510v1>.
- [69] M. Schmelzer, R.P. Dwight, and P. Cinnella. "Discovery of algebraic reynolds-stress models using sparse symbolic regression". *Flow, Turbulence and Combustion*, 104:579–603, 2020. URL <https://dx.doi.org/10.1007/s10494-019-00089-x>.
- [70] F.G. Schmitt. "About boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity". *Comptes Rendus Mécanique, Elsevier Masson*, 335(9-10):617–627, 2007. URL <https://doi.org/10.1016/j.crme.2007.08.004>.

- [71] D.W. Scott. *Multivariate Density Estimation. Theory, Practice and Visualization*. New York: Wiley, 1992.
- [72] W.Z. Shen, J.N. Sørensen, and J. Zhang. "Actuator surface model for wind turbine flow computations". *Proceedings of European Wind Energy Conference 2007*, 2, 2007.
- [73] T.-H. Shih, W.W. Liou, A. Shabbir, Z. Yang, and J. Zhu. "New k- ϵ eddy viscosity model for high reynolds number turbulent flows-model development and validation". *NASA Technical Memorandum ICOMP-94-21*, 1994.
- [74] J. Steiner, R.P. Dwight, and A. Viré. "Data-driven turbulence modeling for wind turbine wakes under neutral conditions". *PhD, Wind energy research group TU Delft*, 2019.
- [75] J.N. Sørensen and W.Z. Shen. "Numerical modeling of wind turbine wakes". *Journal of Fluids Engineering*, 124(2):393–399, 2002. URL <https://dx.doi.org/10.1115/1.1471361>.
- [76] Z. Ti, X.W. Deng, and H. Yang. "Wake modeling of wind turbines using machine learning". *Applied Energy*, 257, 2020. URL <https://doi.org/10.1016/j.apenergy.2019.114025>.
- [77] L. Tian, Y. Song, N. Zhao, T. Wang, and W. Zhong. "Numerical investigations of wind turbine wakes under neutral and convective atmospheric stability conditions". *Journal of Physics: Conference Series*, 1037:072044, 2018. URL <https://dx.doi.org/10.1088/1742-6596/1037/7/072044>.
- [78] A. M. Turing. "I.—computing machinery and intelligence". *Mind*, LIX(236):433–460, 1950. URL <https://dx.doi.org/10.1093/mind/LIX.236.433>.
- [79] M. P. van der Laan. "Efficient turbulence modeling for CFD wake simulations". *Risø National Laboratory, phd thesis*, 2014.
- [80] M. P. van der Laan and S. J. Andersen. "The turbulence scales of a wind turbine wake: A revisit of extended k-epsilon models". *Journal of Physics: Conference Series*, 1037:072001, 2018. URL <https://dx.doi.org/10.1088/1742-6596/1037/7/072001>.
- [81] M. P. Van der Laan, N.N. Sørensen, P. E Réthoré, J. Mann, M. Kelly, and J.G. Schepers. "Nonlinear eddy viscosity models applied to wind turbine wakes". *Proceedings of the 2013 International Conference on aerodynamics of Offshore Wind Energy Systems and wakes (ICOWES2013)*, pages 514–525, 2013.
- [82] M.P. van der Laan, N.N. Sørensen, P-E. Réthoré, J. Mann, M.C. Kelly, N. Troldborg, K.S. Hansen, and J.P. Murcia. "The k- ϵ - f_p model applied to wind farms". *Wind Energy*, 18(12):2065–2084, 2015. URL <https://doi.org/10.1002/we.1804>.
- [83] M.P. van der Laan, N.N. Sørensen, P-E. Réthoré, J. Mann, M.C. Kelly, N. Troldborg, J.G. Schepers, and E. Machefaux. "An improved k- ϵ model applied to a wind turbine wake in atmospheric turbulence". *Wind Energy*, 18(5):889–907, 2015. URL <https://doi.org/10.1002/we.1736>.
- [84] M. Vejmelka and M. Paluš. "Inferring the directionality of coupling with conditional mutual information". *Phys Rev E Stat Nonlin Soft Matter*, 77(2), 2008. URL <https://dx.doi.org/10.1103/physreve.77.026214>.
- [85] G.. Ver Steeg. "non-parametric entropy estimation toolbox (NPEET) 2014". <https://github.com/gregversteeg/NPEET>.
- [86] G. Ver Steeg and A. Galstyan. "Information-theoretic measures of influence based on content dynamics". *Proceedings of the 6th International Conference on Web Search and Data Mining*, pages 3–12, 2013. URL <https://doi.org/10.1145/2433396.2433400>.
- [87] L. J. Vermeer, J. N. Sørensen, and A. Crespo. "Wind turbine wake aerodynamics". *Progress in Aerospace Sciences*, 39(6-7):467–510, 2003. URL [https://doi.org/10.1016/S0376-0421\(03\)00078-2](https://doi.org/10.1016/S0376-0421(03)00078-2).
- [88] A. Viré and M. Zaaijer. Introduction to wind turbines: physics and technology TU Delft course notes. 2018.

- [89] J. Walters-Williams and L. Yan. "Estimation of mutual information: A survey". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5589:389–396, 2009. URL https://dx.doi.org/10.1007/978-3-642-02962-2_49.
- [90] J.X. Wang, J.L. Wu, and H. Xiao. "Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data". *Physical Review Fluids*, 2(3), 2017. URL <https://dx.doi.org/10.1103/PhysRevFluids.2.034603>.
- [91] J. Weatheritt and R. Sandberg. "A novel evolutionary algorithm applied to algebraic modifications of the rans stress-strain relationship". *Journal of Computational Physics*, 325:22–37, 2016. URL <https://doi.org/10.1016/j.jcp.2016.08.015>.
- [92] J. Weatheritt and R. D. Sandberg. "The development of algebraic stress models using a novel evolutionary algorithm". *International Journal of Heat and Fluid Flow*, 68:298–318, 2017. URL <https://doi.org/10.1016/j.ijheatfluidflow.2017.09.017>.
- [93] J. Weatheritt, Y. Zhao, R.D. Sandberg, S. Mizukami, and K. Tanimoto. "Data-driven scalar-flux model development with application to jet in cross flow". *International Journal of Heat and Mass Transfer*, 2019. URL <https://doi.org/10.1016/j.ijheatmasstransfer.2019.118931>.
- [94] J.L. Wu, H. Xiao, and E. Paterson. "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework". *Physical Review Fluids*, 3(7), 2018. URL <https://dx.doi.org/10.1103/physrevfluids.3.074602>.
- [95] J.C. Wyngaard. *Turbulence in the Atmosphere*. Cambridge University Press, first edition, 2010.



SpaRTA

In Algorithm 1, the procedure to construct the candidate features is described. Feature interaction is first introduced in \mathcal{B}_2 . Due to the fact that the range indicated in red only goes up to $k - 1$, not all possible feature interactions are constructed. It has been experimented to adjust this such that *all* feature interactions are constructed, but this did not lead to any noteworthy model prediction improvements but rather only to increased computational costs. It is therefore chosen to remain the algorithm as it is.

Algorithm 1 Algorithm for constructing library of candidate functions

Require: pre-defined features: f_1, \dots, f_n (e.g. λ_1, λ_2)

Require: chosen exponents: $\text{exp}_1, \dots, \text{exp}_m$ (e.g. 0.5, 1.0)

Ensure: Library of non-linear features of pre-defined features f

$\mathcal{B}_1 = []$

for $i = 1, \dots, n$ **do**

for $j = 1, \dots, m$ **do**

$\text{fexp} = \text{eval}(f[i]^{\text{exp}[j]})$

$\mathcal{B}_1.\text{append}(\text{fexp})$

end for

end for

$\mathcal{B}_2 = []$

for $k = 1, \dots, \text{len}(\mathcal{B}_1)$ **do**

$\text{bk} = \mathcal{B}_1[k]$

for l in **range**($k-1$) **do**

$\text{bl} = \mathcal{B}_1[l]$

$\text{fint} = \text{bk} \cdot \text{bl}$

$\mathcal{B}_2.\text{append}(\text{fint})$

end for

end for

$\mathcal{B}_3 = []$

for p in **range**($\text{len}(\mathcal{B}_2)$) **do**

$\text{bp} = \mathcal{B}_2[p]$

for q in **range**($\text{len}(\mathcal{B}_1)$) **do**

$\text{bq} = \mathcal{B}_1[q]$

$\text{bint} = \text{bp} \cdot \text{bq}$

$\mathcal{B}_3.\text{append}(\text{bint})$

end for

end for

$f = \mathcal{B}_1 + \mathcal{B}_2 + \mathcal{B}_3 + c$

B

Mutual Information

B.1. Theoretical Example

The mutual information definition for continuous variables is

$$MI(X; Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x, y) \log \frac{p_{(X,Y)}(x, y)}{p_X(x)p_Y(y)} dx dy, \quad (\text{B.1})$$

where $p_{(X,Y)}(x, y)$ is the joint probability density function and $p_X(x)$ and $p_Y(y)$ are marginal probability density functions of variables X and Y , respectively. In the case of discrete variables, the mutual information definition is calculated as a double sum

$$MI(X; Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \quad (\text{B.2})$$

Consider two data sets, each consisting of three points depicted in Figure B.1. The mutual information of the orange set equals

$$MI(X; Y) = \frac{1}{3} \log \left(\frac{1/3}{1/3 \cdot 1} \right) + \frac{1}{3} \log \left(\frac{1/3}{1/3 \cdot 1} \right) + \frac{1}{3} \log \left(\frac{1/3}{1/3 \cdot 1} \right) = 0, \quad (\text{B.3})$$

and for the blue set

$$MI(X; Y) = \frac{1}{3} \log \left(\frac{1/3}{1 \cdot 1/3} \right) + \frac{1}{3} \log \left(\frac{1/3}{1 \cdot 1/3} \right) + \frac{1}{3} \log \left(\frac{1/3}{1 \cdot 1/3} \right) = 0. \quad (\text{B.4})$$

When combining the blue and orange sets to one large set (consisting of six points), the mutual information is

$$MI(X; Y) = \frac{1}{6} \log \left(\frac{1/6}{1/6 \cdot 2/3} \right) + \frac{1}{6} \log \left(\frac{1/6}{1/6 \cdot 2/3} \right) + \frac{1}{3} \log \left(\frac{1/3}{2/3 \cdot 2/3} \right) + \frac{1}{6} \log \left(\frac{1/6}{2/3 \cdot 1/6} \right) + \frac{1}{6} \log \left(\frac{1/6}{2/3 \cdot 1/6} \right) = 0.17. \quad (\text{B.5})$$

Clearly the two sets separately lead to a mutual information of zero, indicating X and Y are independent. However there is dependency when both sets are combined, leading to a non-zero mutual information. This proves that it is possible that two subsets of a dataset can both have a lower mutual information than the entire dataset.

B.2. Tensor basis

The mutual information between individual tensor basis components and \mathbf{b}^A for the entire domain and free-stream region are listed in Figures B.2 and B.3. The mutual information estimates are significantly larger inside the turbine wakes than outside. Only for component (2,3) the mutual information outside the wake is larger than inside the wake.

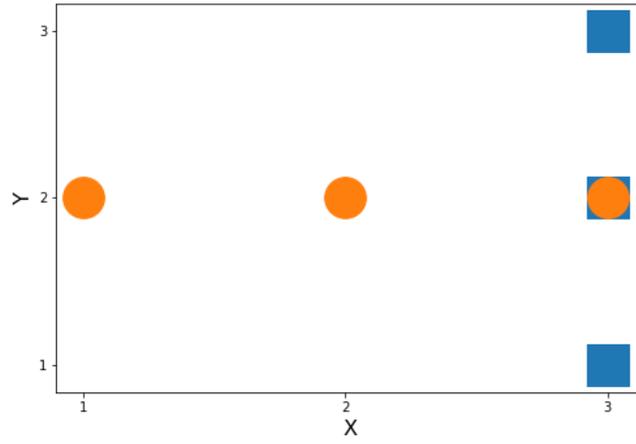
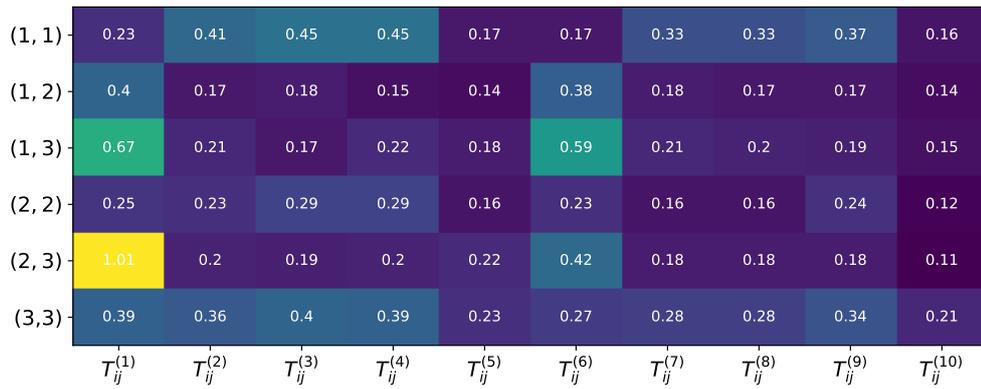
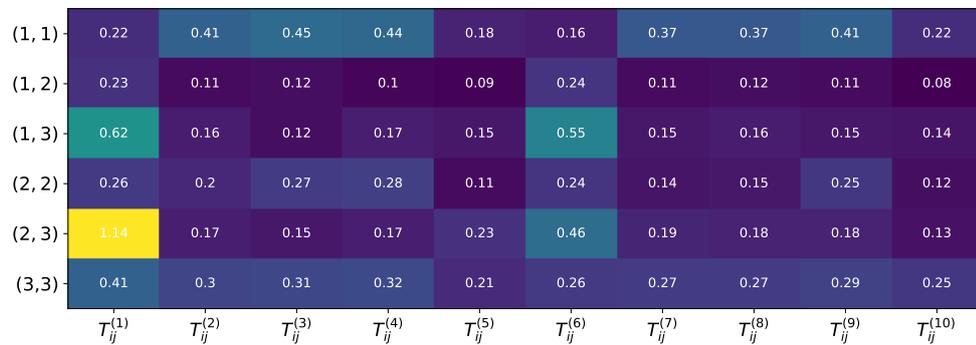
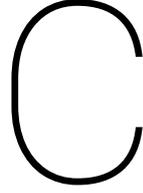


Figure B.1: Two discrete data sets depicted in orange and blue.

Figure B.2: Mutual information heat map for basis tensor components with b^Δ components for entire domain.Figure B.3: Mutual information heat map for basis tensor components with b^Δ components for free-stream region.



Wake Classification

Additional models for case A (obtained through cross-validation) are described in Appendix C.1. The model discovery and inference with cross-validation described in Section 7.3 is applied to case B and case C as test sets in Appendices C.2 and C.3, respectively.

C.1. Case A

A disadvantage of the wake classification model visualized in Section 7.4 is that the classifier predictions in the free-stream grow when approaching the wall. Although the predictions are closer to zero than to one, it would be more desirable when the predictions would remain closer to zero.

When selecting a different wake classification model, visualized in Figure C.1, the predictions remain closer to zero when approaching the wall when compared to the classification predictions in Figure 7.6. On the other hand, the recall will be smaller because more wake samples are identified incorrectly. This is therefore a trade-off that has to be made.

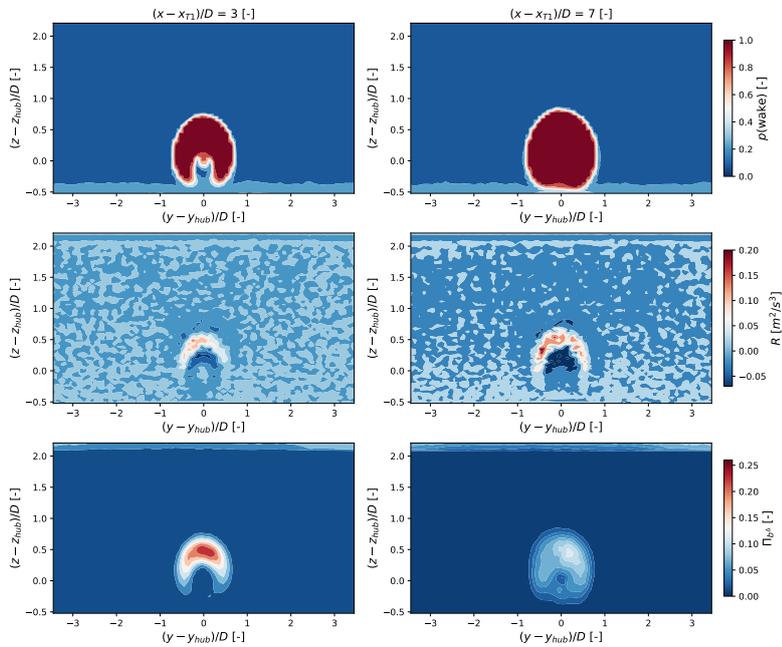


Figure C.1: Comparison of classifier predictions to contours of R and $\Pi_{b\Delta}$ for case A at different slices in the $y-z$ plane, x -location depicted in title upper row. Upper row: classifier predictions, middle row: R and lower row: $\Pi_{b\Delta}$.

C.2. Case B

The model performance when trained on case A and C and subsequently tested on case B is shown in Figure C.2. The model for further visualization is depicted by the green star, this model has the form

$$\mathcal{M} = \sigma \left[\mathbf{w}_u^T \mathbf{x}_u + 1.01 \right] \quad (\text{C.1})$$

$$\mathbf{x}_u = \left[q_2, q_2 q_9, q_2 q_8^2, q_2 q_7, q_8, q_7 \right] \quad (\text{C.2})$$

$$\mathbf{w}_u = \left[52.2, 1140, 495, -44.0, -13.5, 1.36 \right]. \quad (\text{C.3})$$

The classifier predictions are compared to the contours of R and $\Pi_{b\Delta}$ in Figure C.3. Clearly the regions where the correction terms are significant are captured by the classification model. The wake classification predictions are compared to the true wake values in Figure C.4. It can be seen that the model has difficulties identifying the lower part of the wake. The discrete outcomes are shown in Figure C.5.

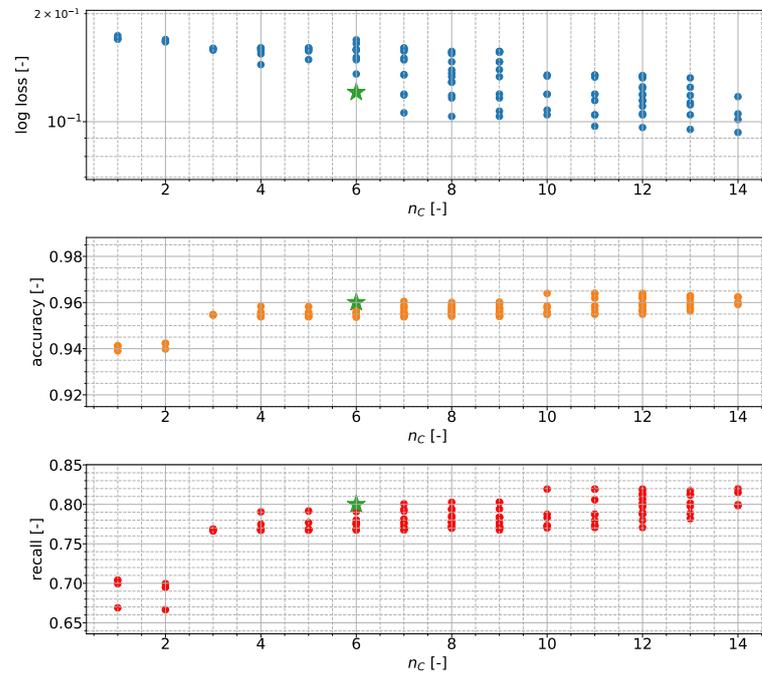


Figure C.2: Case B: log loss (top), accuracy (middle) and recall (bottom) as function of complexity n_c for all models analysed in the inference phase. Final chosen model high-lighted by green star.

C.3. Case C

The model performance when trained on cases A and B and subsequently tested on the unseen case C is depicted in Figure C.6.

The wake classification model tested on case C has the form:

$$\mathcal{M} = \sigma \left[\mathbf{w}_u^T \mathbf{x}_u + 0.20 \right] \quad (\text{C.4})$$

$$\mathbf{x}_u = \left[q_2 q_9, q_2 q_8^2, q_2 q_7^2, q_8 \right] \quad (\text{C.5})$$

$$\mathbf{w}_u = \left[1465, 274, 4.80, -6.29 \right]. \quad (\text{C.6})$$

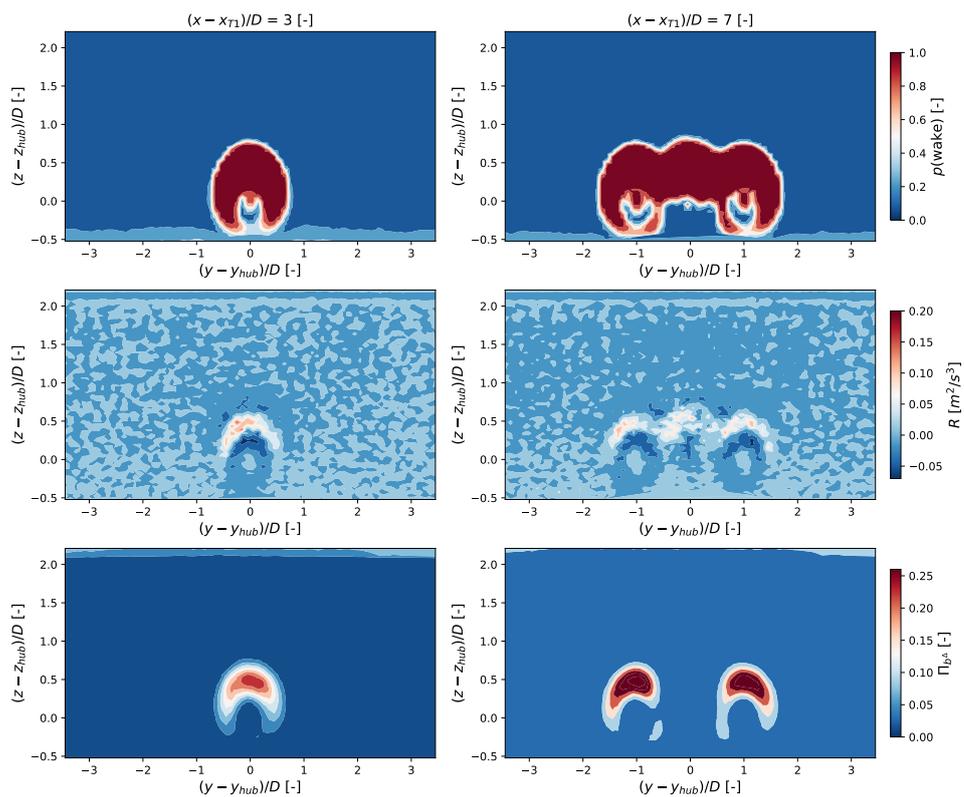


Figure C.3: Comparison of classifier predictions to contours of R and $\Pi_{b\Delta}$ for case B at different slices in the $y-z$ plane, x -location depicted in title upper row. Upper row: classifier predictions, middle row: R and lower row: $\Pi_{b\Delta}$.

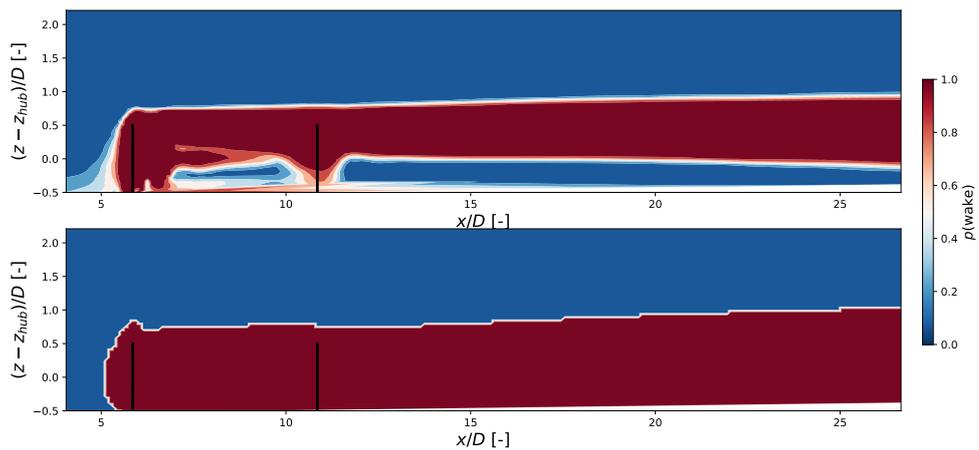


Figure C.4: Wake classifier predictions compared to true label for case B on $x-z$ plane at turbine location $(y - y_{hub})/D = 0$. Upper figure: classifier predictions, lower figure: true label.

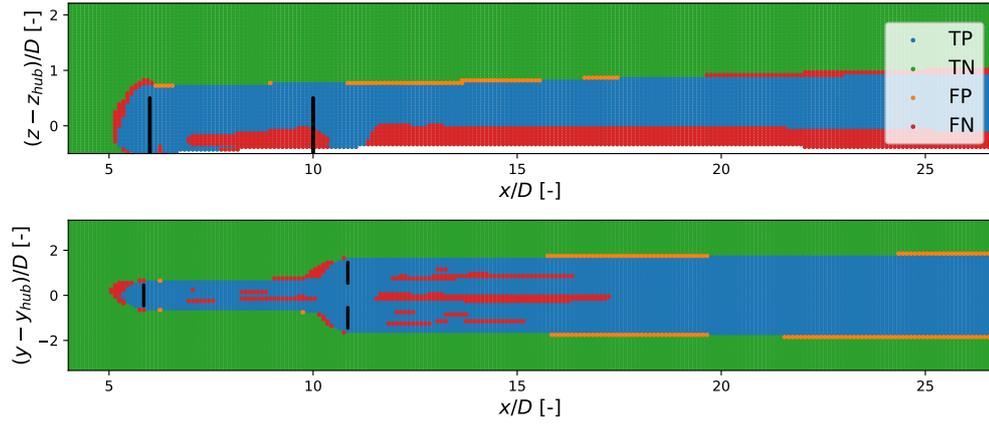


Figure C.5: Wake classifier discrete predictions compared to true discrete labels for case B. Upper figure: $x - z$ plane at $(y - y_{\text{hub}})/D = 0$, lower figure: $x - y$ plane at $(z - z_{\text{hub}})/D = 0$.

The classifier predictions are compared to the contours of R and $\Pi_{b\Delta}$ in Figure C.7. Clearly the regions where the correction terms are significant are captured by the classification model. The wake classification predictions are compared to the true wake values in Figure C.8. The discrete outcomes are shown in Figure C.9.

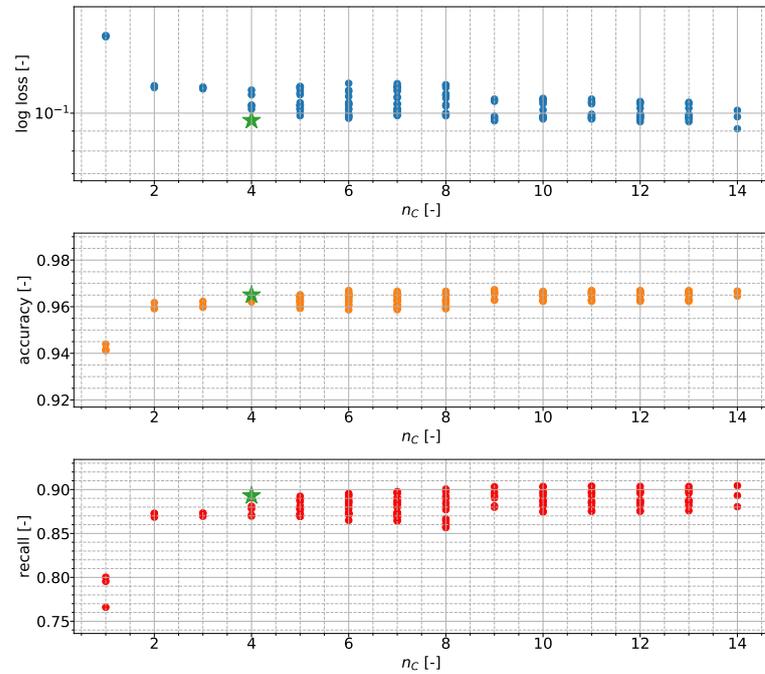


Figure C.6: Case C: log loss (top), accuracy (middle) and recall (bottom) as function of complexity n_c for all models analysed in the inference phase. Final chosen model high-lighted with green star.

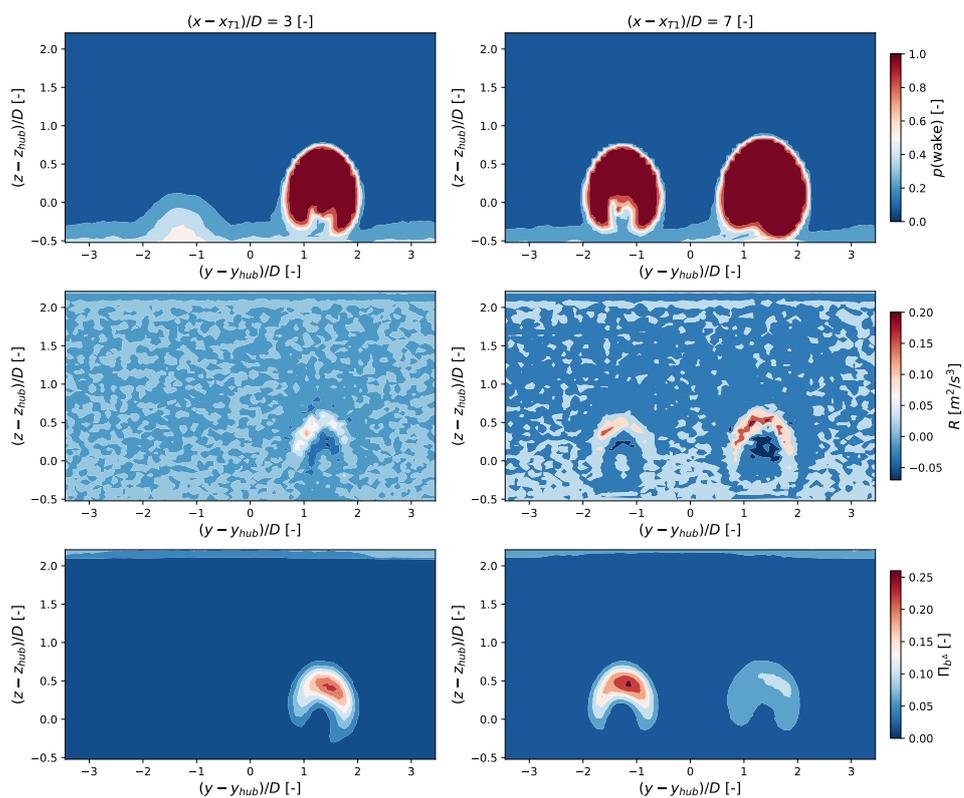


Figure C.7: Comparison of classifier predictions to contours of R and $\Pi_{b\Delta}$ for case C at different slices in the $y-z$ plane, x -location depicted in title upper row. Upper row: classifier predictions, middle row: R and lower row: $\Pi_{b\Delta}$.

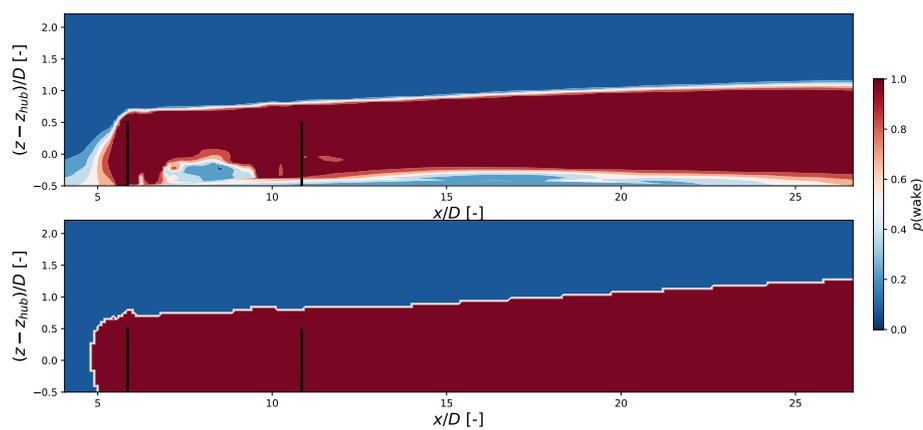


Figure C.8: Wake classifier predictions compared to true label for case C on $x-z$ plane at turbine location $(y - y_{hub})/D = 1.25$. Upper figure: classifier predictions, lower figure: true label.

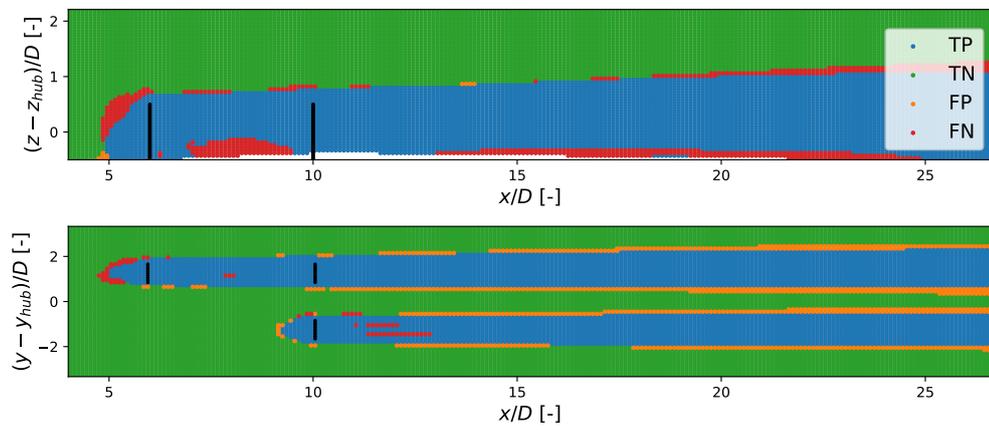


Figure C.9: Wake classifier discrete predictions compared to true discrete labels for case C. Upper figure: $x - z$ plane at $(y - y_{hub})/D = 1.25$, lower figure: $x - y$ plane at $(z - z_{hub})/D = 0$.

D

CFD implementations

D.1. Coupled corrections

The stream-wise velocity component as function of domain width and height is depicted in Figure D.1 for three models, varying in complexity. It is observed that the medium and complex model perform significantly better than the simple model, which was already verified by the normalized mean-squared error in Table 10.3. The same observation is made for the turbulent kinetic energy in Figure D.2.

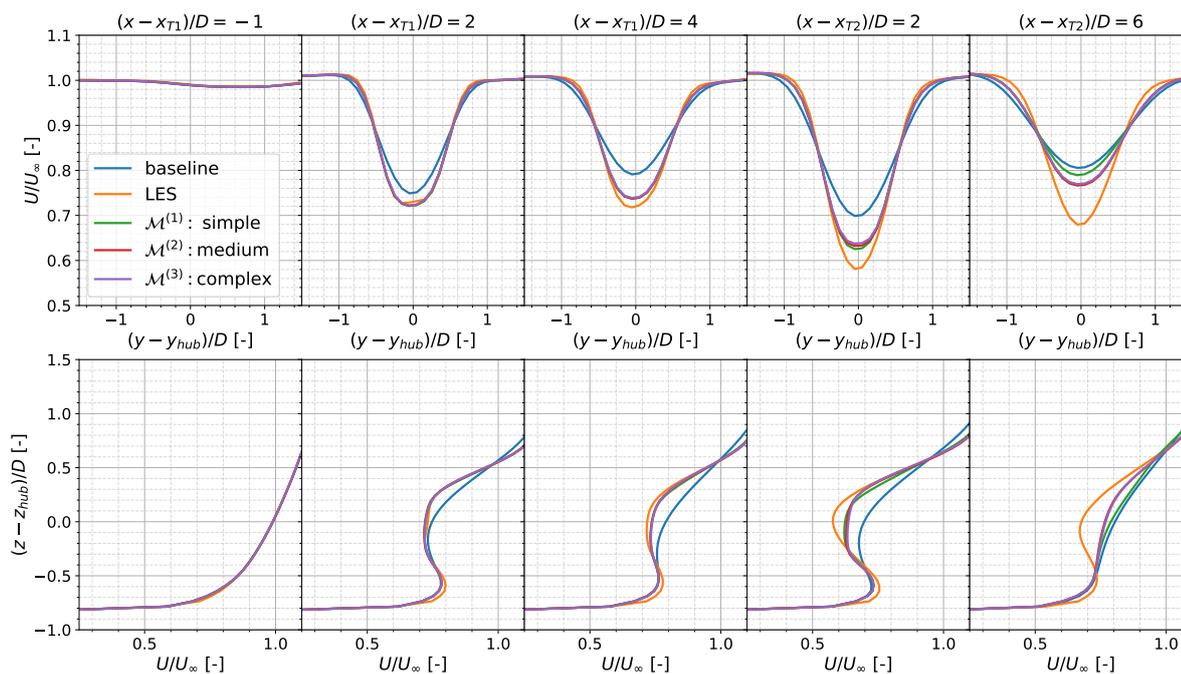


Figure D.1: Comparison of baseline k - ϵ RANS, LES and three coupled RANS corrections via normalized stream-wise velocity as function of domain width and height for different downstream locations with respect to turbines.

D.2. Physical-form R

The turbulent kinetic energy as function of domain width for three coupled corrections is depicted in Figure D.3. The three coupled corrections differ in what the physical-form of R is. Clearly when R is only mod-

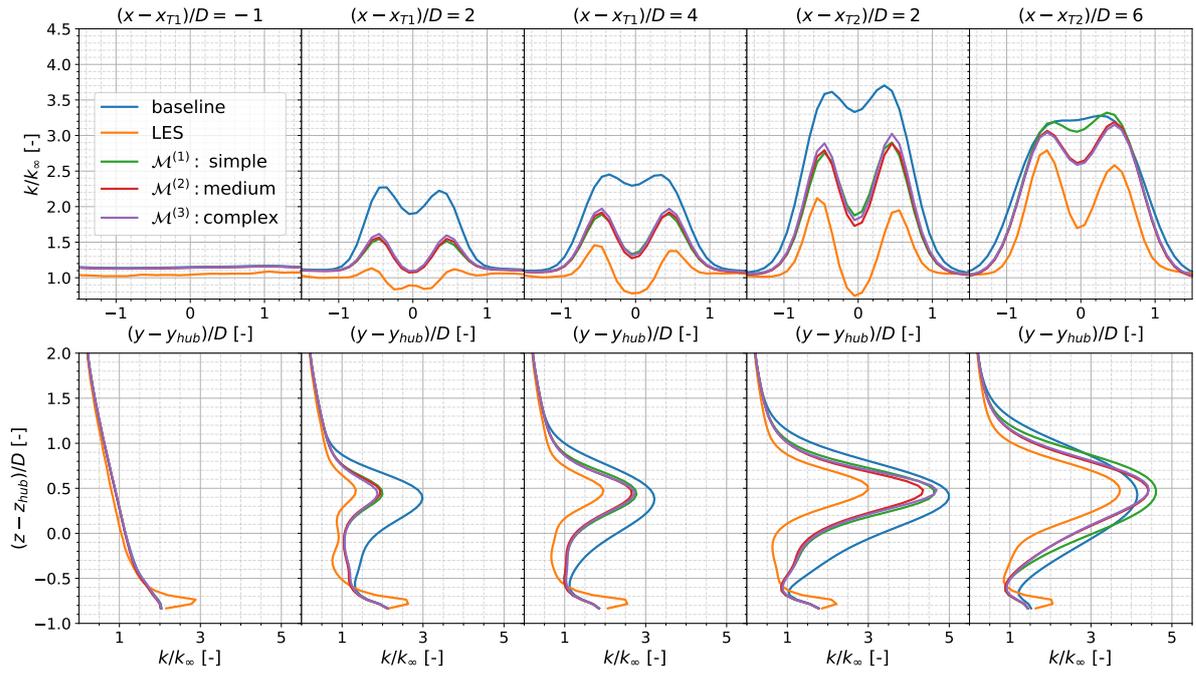


Figure D.2: Comparison of baseline k - ϵ RANS, LES and three coupled RANS corrections via normalized turbulent kinetic energy as function of domain width and height for different downstream locations with respect to turbines.

eled using the production part, k is more overestimated in the far-wake than when R is modeled only by the source/sink term.

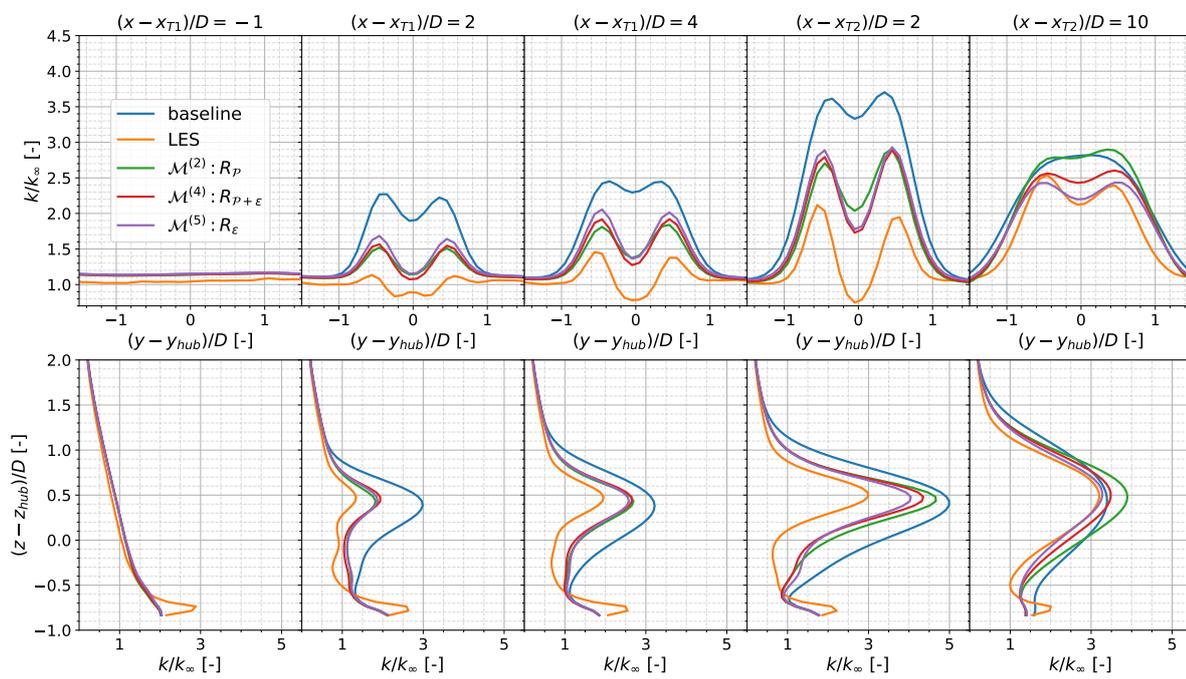


Figure D.3: Comparison of baseline k - ϵ RANS, LES and three coupled RANS corrections via normalized turbulent kinetic energy as function of domain width and height for different downstream locations with respect to turbines.