

Data-Driven RANS Modelling of Junction Flows

An Evaluation of SpaRTA on the Wing-Body
Geometry

Matthijs E. van Ede

Data-Driven RANS Modelling of Junction Flows

An Evaluation of SpaRTA on the Wing-Body
Geometry

by

Matthijs E. van Ede

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday August 28, 2023 at 10:00 AM.

Student number: 4662768
Project duration: September, 2022 – August, 2023
Thesis committee: Dr. R. P. Dwight, TU Delft, supervisor
Prof.dr.-ing. G. Eitelberg, TU Delft
Dr. D. Modesti, TU Delft

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

Preface

Dear reader,

In front of you lies the thesis that completes my academic journey at the Delft University of Technology as an aerospace student. Since, as with most things in life, I did not go into this adventure as a sole individual, I would like to express my gratitude to a few people.

First of all Dr. Richard Dwight, my supervisor for this project. I would like to thank you for the support you gave me when conducting this research through the numerous meetings and discussions. Above that, I would like to thank you for treating me as an equal with respect to the other (PhD) students and allowing me to actively take part in your research group consisting of Tyler, Renzhi, Kherlen and Kaj whom I would like to thank as well. The equal treatment gave me a level of confidence that allowed me to critically look at my work and never to be afraid to ask stupid questions. Additionally, taking an active part in a research group made me enjoy my work even more due to the collaborative search for a common goal

Secondly, I would like to thank my friends that I have made before and during my studies. This really allowed me to enjoy life outside of my academic work. Therefore, I would specifically like to thank Reint, Igor, Kim, Rudi, Rutger and Rinke for the fun we had, and hopefully will have. Next to that, I would like to thank a lot of people who were a member of DSEA during my study period. Having an association where I could relax, have a laugh and a beer, really enhanced my study experience. I would also like to thank Kaj and Luuk for the fun drinks and conversations we had during our MSc projects in the basement.

Finally, I am also expressing my great gratitude to my parents and sisters (Meeke and Jitske) for their continuous support that I already experienced way before I started my journey at the TU Delft.

*Matthijs E. van Ede
Delft, August 2023*

Abstract

Classical RANS (Reynolds-Averaged Navier-Stokes) turbulence models have limited accuracy in the prediction of the flow over the wing-body geometry. Therefore, this work focuses on improving the prediction accuracy of the classical $k-\omega$ SST turbulence model for the junction flow by means of the data-driven method SpaRTA. In the SpaRTA methodology higher-fidelity data, in this work LES (Large Eddy Simulation) data, is leveraged to find correction models that enhance the baseline turbulence model. These correction models aim at correcting the Reynolds stress anisotropy and the turbulent kinetic energy (k) equation. The correction models are obtained by sparsely regressing flow features to the correction fields that are found by the k -corrective frozen approach.

The results showed that the correction fields determined by the frozen method are very effective. Direct propagation of these fields resulted in a solution very similar to the LES data. In contrast to the model search for the k correction, finding a model for the anisotropy correction by means of sparse regression was difficult. The latter is reflected in the CFD (Computation Fluid Dynamics) propagation of the found models on the same flow case as the training of the models. This model propagation showed an improvement in the placement of the horseshoe vortex, however, the vortex topology in the corner region was unsatisfactory. Which is linked to the poor model fit of the anisotropy correction field.

Besides the search for an improved turbulence model, this work also encapsulates an evaluation of the drag reduction by the anti-fairing geometry for the wing-body junction flow. This evaluation was achieved by comparing available wall-resolved LES data sets. The evaluation revealed that the anti-fairing reduces the drag force by a propulsive pressure mechanism over the bottom wall for a specific geometry over which the drag is computed. This geometry included only 5% of the complete wing span, for larger wing spans the drag reduction was not evident.

Contents

Preface	iii
Abstract	v
Nomenclature	ix
1 Introduction	1
2 Literature Study	3
2.1 Wing-Body Flow Physics	3
2.1.1 Horseshoe Vortex	3
2.1.2 Corner Separation	5
2.1.3 Undesirable Effects	7
2.1.4 Wing-Body Fairings	7
2.1.5 Anti-Fairing	8
2.2 Numerical Simulation	9
2.2.1 RANS Simulations	9
2.2.2 RANS Uncertainty Sources	10
2.2.3 LES Simulations	11
2.3 Data-Driven Turbulence Modelling	11
2.3.1 General Effective-Viscosity Hypothesis	12
2.3.2 Explicit Algebraic Models	13
2.3.3 Tensor Basis Neural Network	15
2.3.4 Gaussian Process Regression	15
2.3.5 Flow Classifier	16
2.4 Literature Study Conclusion	16
2.4.1 Research Gap	16
2.4.2 Research Questions	17
3 Data-Driven Modelling Methodology	19
3.1 Reynolds-Averaged Navier-Stokes	19
3.1.1 Governing Equations	19
3.1.2 Computational Setup	21
3.1.3 Fluid Solver	25
3.2 SpaRTA Framework for the Junction Flow	26
3.2.1 k-Corrective-Frozen	28
3.2.2 Propagation	29
3.2.3 Sparse Regression	30
3.2.4 Feature Library and Selection	33
3.2.5 Library Generation	36
4 Results and Discussion	39
4.1 Baseline RANS Results and LES Data	39
4.1.1 Mean Flow Fields	39
4.1.2 Laminar Separation Bubble	45

4.2	k-Corrective Frozen Results	47
4.2.1	Turbulent Kinetic Energy Correction	47
4.2.2	Anisotropy Correction	50
4.3	Correction Field Propagation	54
4.4	Model Training Results	59
4.4.1	Turbulent Kinetic Energy Residual	59
4.4.2	Turbulence Anisotropy Correction	68
4.5	Model Propagation	79
4.6	Discussion on SpaRTA Modifications	89
4.6.1	Altering the Transport Equation	89
4.6.2	Production Limiter	90
4.6.3	Initial Conditions	91
4.6.4	Propagation for Regression	92
4.6.5	Spatial Blending Functions	92
5	LES Drag Comparison Anti-Fairing	95
5.1	Computational Setup	95
5.2	Drag Results	97
5.3	Concluding Remarks	99
6	Conclusion	101
6.1	Recommendations	103
	References	105
A	SpaRTA Nomenclature	109
B	Mesh Study	111
C	Simulation Residual Plots	115
D	Code Implementations	119

Nomenclature

Abbreviation	Definition
BF	Bluntness Factor
BL	Boundary Layer
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
EASM	Explicit Algebraic Stress Model
EARSM	Explicit Algebraic Reynolds Stress Model
GP	Gaussian Processes
Hifi	High fidelity
HPC	High-Performance Cluster
HSV	Horseshoe Vortex
LES	Large Eddy Simulation
LEVMM	Linear Eddy Viscosity Model
LSB	Laminar Separation Bubble
MDF	Momentum Deficit Factor
MGEP	Multidimensional Gene Expression Programming
MI	Mutual Information
MSE	Mean Squared Error
ML	Machine Learning
NN	Neural Network
NS	Navier-Stokes
QCR	Quadratic Constitutive Relation
RANS	Reynolds-Averaged Navier-Stokes
Re	Reynolds Number
SpaRTA	Sparse Regression of Turbulent Stress Anisotropy
TBNN	Tensor Basis Neural Network

Symbol	Definition	Unit
a_{ij}^x	Anisotropic Reynolds stress correction term	$[m^2/s^2]$
b_{ij}^Δ	Non-dimensional anisotropic Reynolds stress correction term	[-]
c	Chord length	$[m]$
c_{nl1}	QCR modelling constant	[-]
C_d	Drag coefficient	[-]
C_f	Skin friction coefficient	[-]
C_p	Pressure coefficient	[-]
F_d	Drag force	$[(kg \cdot m)/s^2]$

f_i	Body force vector	$[(kg \cdot m)/s^2]$
$G^{(\lambda)}$	Production term of the base tensors of Pope	$[m^2/s^3]$
I, II, III, \dots	Tensor invariants	[-]
k	Turbulent kinetic energy	$[m^2/s^2]$
\mathcal{M}	RANS closure model	[-]
\mathcal{N}	Navier-Stokes equations	[-]
u, v, w	Velocity in x -, y - and z -direction	$[m/s]$
u', v', w'	Reynolds velocity fluctuation in x -, y - and z -direction	$[m/s]$
u^+	Non-dimensional velocity	[-]
u_τ	Friction velocity	$[m/s]$
U, V, W	Reynolds averaged velocity in x -, y - and z -direction	$[m/s]$
P	Reynolds averaged pressure	$[m^2/s^2]$
P_k	Turbulent production term	$[m^2/s^3]$
P_k^Δ	Turbulent production term correction	$[m^2/s^3]$
p_i	Primary feature	$[m^2/s^3]$, [-]
p'	Reynolds pressure fluctuation	$[m^2/s^2]$
$p(x)$	Marginal probability density function	[-]
q_i	Regression scalar feature	[-]
R	Turbulent kinetic energy correction term	$[m^2/s^3]$
R_0	Leading-edge radius	$[m]$
s_i	Secondary feature	[-]
S_{ij}	Mean strain-rate tensor	[-]
S_T	Distance between leading edge and X_T	$[m]$
T	Maximum wing thickness	$[m]$
$T_{ij}^{(\lambda)}$	Pope tensor basis	[-]
X_T	Chordwise position of maximum wing thickness	$[m]$
y^+	Non-dimensional wall distance	[-]
<hr/>		
α	Angle of attack	[-]
δ_{ij}	Kronecker delta	[-]
δ	Boundary layer thickness	$[m]$
ϵ	Dissipation rate of k	$[m^2/s^3]$
θ	Boundary layer momentum thickness	$[m]$
κ	von Kármán constant	[-]
ν	Kinematic viscosity (laminar viscosity)	$[m^2/s]$
ν_t	Turbulent kinematic viscosity (eddy viscosity)	$[m^2/s]$
ξ, η, ζ	Flow vorticity in x -, y - and z -direction	$[1/s]$
ρ	Air density	$[kg/m^3]$
τ	Reynolds stress tensor	$[m^2/s^2]$
$\bar{\tau}$	τ determined with the Boussinesq closure	$[m^2/s^2]$
τ_w	Wall shear stress	$[kg/(m \cdot s^2)]$
ω	Specific rate of dissipation	$[1/s]$
Ω_{ij}	Mean rotation-rate tensor	[-]
Ω_x, Ω_z	Vorticity in x - and z -direction	$[1/s]$
<hr/>		
Subscripts:		
k	Tensor invariants based on the gradient of k	
s	Non-dimensionalised using the mean flow time-scale	

1

Introduction

The drag force is defined as "the component of the aerodynamic force parallel to the relative wind" Anderson [1]. In many modern-day aerospace vehicles, such as cars and civil aircraft, it is desirable to minimise this drag force in normal operation. Since minimising the drag force will reduce the energy required to propel the vehicle, a favourable effect in the current fight against climate change. Often, modern aerospace vehicles encounter a type of flow somewhere on the external surface called the junction flow. As described by Simpson [2], in this type of flow a boundary layer, developed for example over a flat surface, encounters an object protruding the surface. A common example of a junction flow is the flow around the wing-fuselage connection on aircraft. The interaction of the boundary layer with the flow over the protruding object in this junction region causes the boundary layer to separate and form a system of unsteady vortices. This flow behaviour generates an extra drag component, this is often referred to as the interference drag. This extra drag component results in an increased drag force encountered by the complete aerodynamic vehicle. Thus reducing the interference drag will increase the energy efficiency of the vehicle.

As shown by Devenport et al. [3] a current solution, also seen on many present-day aircraft, is to design a fairing around the junction region to allow for a more smooth interaction of the two flows. Devenport et al. [3] managed to remove the separation of the boundary layer upstream of the wing by the use of a leading-edge fillet in an experimental setup. The design of this junction region using computational methods is, however, difficult due to the complex vortex behaviour in the region connecting the two geometries. The main reason for this is that the exploration of the design space using computational fluid dynamics (CFD) is currently computationally too costly. These high costs are generated since only Large Eddy Simulations (LES) or higher fidelity CFD codes are able to generate accurate solutions on the junction flow due to its highly unsteady and anisotropic nature, Ryu et al. [4].

There is, however, a possible solution to reduce these costs. Which is to increase the prediction accuracy of lower fidelity CFD solvers, such as the industry-popular Reynolds-averaged Navier-Stokes (RANS) methodology, by means of data-driven methods. As shown by Zhang et al. [5] aerodynamic shape optimisation using CFD for flow problems where RANS is lacking can become feasible by combining high-fidelity solutions with a lower-fidelity RANS solver. In Zhang et al. [5] it is shown that the design space of a shape optimisation problem for the periodic hill geometry can be explored significantly faster by using a RANS solver for which the accuracy of the turbulence model has been improved by means of a machine learning approach. This is because the modelling of turbulence is one of the key error sources for predicting flow separation in the RANS methodology (Duraismy et al. [6]). Therefore the objective of this MSc project is:

To improve a RANS turbulence model by means of a machine learning algorithm such that the flow field generated by the RANS equations for the wing-body junction flow problem is close to that of high-fidelity solvers.

If this objective is completed it will allow for aerodynamic shape optimisation of the wing-body junction flow using a low-cost RANS CFD solver. Hence the shape can be optimised such that the interference drag is minimal.

This thesis is structured as follows. First in Chapter 2 an elaborate literature study is presented on the various literature regarding wing-body junction flows and data-driven turbulence modelling. This chapter also describes the important flow phenomena involved in the junction flow. Secondly, in Chapter 3, the numerical methodology behind this work is described. This includes the RANS simulation technique as well as the machine-learning methodology. In Chapter 4, the results of this methodology will be displayed and discussed. This chapter is followed by Chapter 5 in which the drag reduction system of the anti-fairing geometry, which is covered in Chapter 2, is discussed by comparing two high-fidelity simulations. Finally, in Chapter 6, a conclusion on the work is given as well as a set of recommendations for future work.

2

Literature Study

This chapter covers an important portion of the available literature regarding the wing-body junction flow as well as data-driven turbulence modelling techniques. First in Section 2.1 the physical flow characteristics of the wing-body junction flow are discussed. Secondly, Section 2.2 investigates available numerical simulations of the wing-body geometry. Consecutively, Section 2.3 covers literature on data-driven turbulence modelling techniques. And finally, in Section 2.4, the research gap is identified and the research question is formulated.

2.1. Wing-Body Flow Physics

As described in the introduction, Chapter 1, the junction flow encapsulates a large range of geometries. Think of wall-mounted cubes, wall-mounted cylinders or any other combination which involves a relatively flat surface and a bluff body. Therefore, to gain a better understanding of the external junction flow many authors in the research field have simplified the study to a wing mounted perpendicular to a flat plate (Gand et al. [7]). This junction flow is often referred to as the wing-body junction flow. The flow physics involved in the wing-body junction flow is described by two correlated phenomena. That is the horseshoe vortex (HSV) system and the corner separation. Literature regarding these phenomena is studied in this section to build a foundation for the work described in this thesis, note that mostly incompressible flows with turbulent boundary layers are investigated.

2.1.1. Horseshoe Vortex

The HSV is present due to the adverse pressure gradient imposed by the wing on the incoming flow over the flat plate. The adverse pressure gradient causes the incoming flow to separate from the surface and roll up into multiple horseshoe vortices (Simpson [2]). The recognisable horseshoe shape is formed as the vortices are stretched and convected around the wing with the flow. Since the boundary layer detaches from the flat plate a separation point upstream of the wing on the plate can be identified. Together with the HSV this separation point is convected around the wing forming the separation line, see Figure 2.1a.

In the time average of a turbulent flow on the wing-body geometry, the flow topology is characterised as displayed in Figure 2.1b [8]. This topology consists of a main HSV, a counter-rotating corner vortex (w.r.t. to the main HSV) and so-called stress induced vortices (Gand et al. [8]). However, the vortex topology is dependent on the geometry of the surfaces as well as the Reynolds number of the incoming flow, as shown by Simpson [2]. Therefore, a general mean topology for the junction flow is difficult to define. Nevertheless, it is shown that for many flows the time-mean topology often consists of one main HSV, hereafter referred to as the HSV,

together with multiple smaller co- and counter-rotating vortices, such as the corner vortex and stress induced vortex.

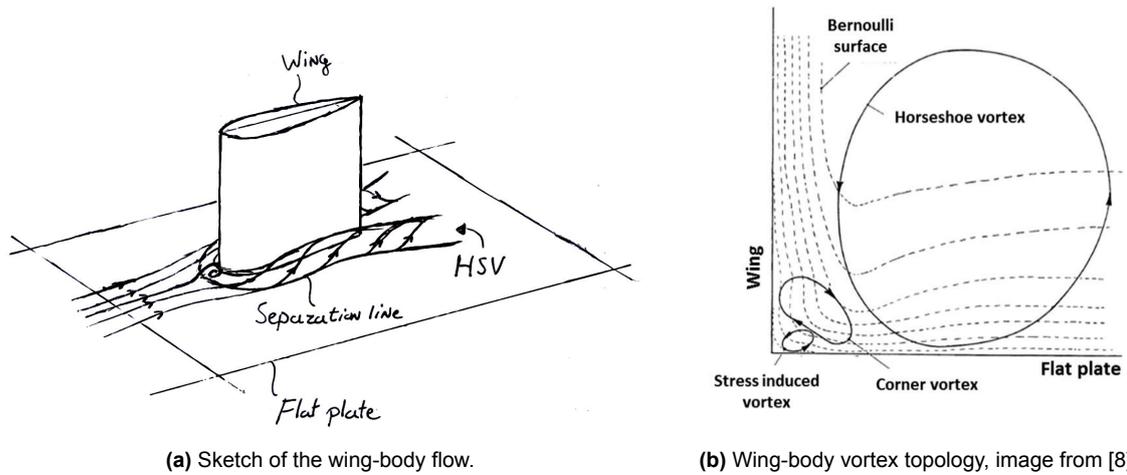


Figure 2.1: Schematics of the wing-body junction flow.

The system of vortices present on the wing-body geometry is unsteady in both strength and shape. Only for very low Reynolds numbers, that is $Re_\theta \leq 100$ (θ equals the boundary layer momentum thickness), the system is steady (Gand et al. [9]). Therefore, for most engineering examples, in which the flow is turbulent, the HSV and the coexisting secondary vortices are unsteady. As described by Simpson and Gand et al. [2, 9, 10], for turbulent flows the HSV translates forward and backwards upstream of the wing between two quasi-steady locations. These locations are defined as the backflow mode (upstream of the corner) and zeroflow mode (close to the corner), see Figure 2.2 [10]. The aforementioned names originate from the probability density functions of the streamwise velocity (u) measured at a fixed point upstream of the wing beneath the time mean location of the HSV in the symmetry plane. These functions show a peak at a negative value of u and a peak close to zero, indicating the two preferred locations of the HSV. This bimodal behaviour is aperiodic for turbulent flows. However, as reported by Gand et al. [9], who performed a literature survey on the periodicity, the bimodal behaviour is found to be periodic for laminar to transitional boundary layers for which $Re_\theta \leq 1500$.

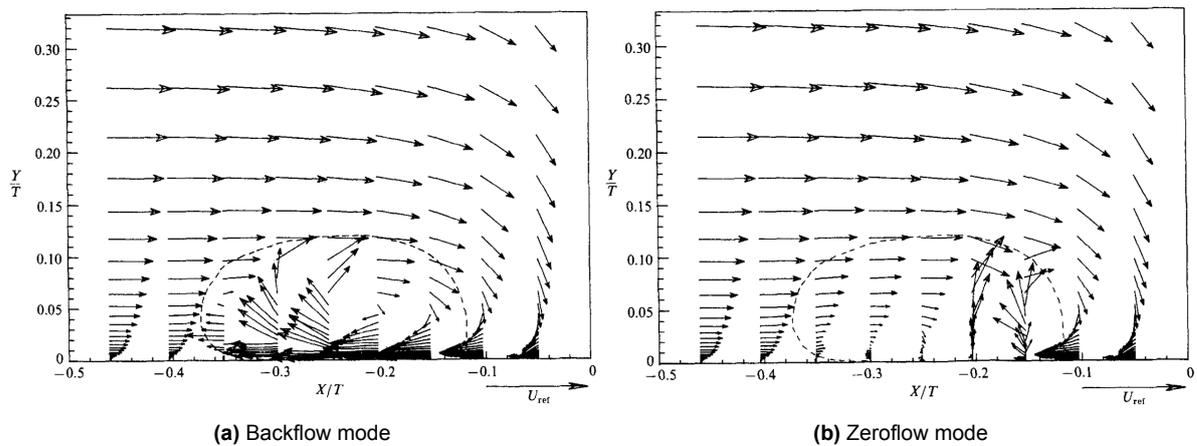


Figure 2.2: Velocity contours of the modes in the symmetry plane upstream of the wing, images from Devenport and Simpson [10]. The dashed line indicates the bimodal region.

In Elahi et al. [11] a study on the effect of the Reynolds number on the HSV is given. In the experiment, particle image velocimetry (PIV) measurements of the Rood wing [12] (the same geometry as in Deveport and Simpson [10]) at incompressible flow conditions were taken for wing-thickness based Reynolds numbers ranging from 6920 to 75000. This corresponded to a Re_θ range from 550 to 5740 using fully turbulent boundary layers. From the study, Elahi et al. concluded that the time-mean vortex core location is not strongly dependent on the Reynolds number. It was, however, found that at higher Reynolds numbers the motion of the HSV system upstream of the leading edge was more chaotic compared to lower Reynolds numbers. This indicates that the stability of the core decreases with increasing Reynolds number. Next to that, with increasing Reynolds number it was found that the shape of the turbulent kinetic energy in the symmetry plane at the time-mean HSV core location deforms from an elliptic shape to a 'C'-shape. Resulting in more kinetic energy close to the wall. This shape deformation is a result of the increased magnitude of the velocity fluctuations in the freestream direction (u') beneath the time-mean position of the HSV.

As previously mentioned, the geometry of the protruding object, for example the wing, influences the characteristics of the HSV. As described by Simpson [2], the strength of the HSV system increases with the bluntness of the leading edge of the object. To quantify this bluntness many authors use the bluntness factor (BF) introduced by Fleming [13]:

$$BF = \frac{1}{2} \frac{R_0}{X_T} \left[\frac{T}{S_T} + \frac{S_T}{X_T} \right] \quad (2.1)$$

in which R_0 equals the leading-edge radius, X_T the chordwise position of maximum thickness T and S_T the distance between the leading edge and X_T over the surface of the wing. A higher BF indicates a blunter leading edge and hence results in a stronger HSV.

Besides the bluntness factor, the characteristics of the incoming boundary layer dictate the shape of the vortex system as well. Therefore authors describe the incoming flow using the momentum deficit factor (MDF_x). Which is defined as the product of the Reynolds number based on the wing thickness (Re_T) and the Reynolds number based on the boundary layer momentum thickness (Re_{θ_x}) Gand et al. [7]:

$$MDF_x = Re_T \cdot Re_{\theta_x} \quad (2.2)$$

Fleming et al. [14] described that a larger MDF will result in stronger near-wall vortices and smaller flow distortions. And thus, in contrast, a smaller MDF indicates larger flow distortions.

2.1.2. Corner Separation

The second physical phenomenon of interest on the wing-body flow is the corner separation. The corner separation is a region of separated flow at the trailing edge of the wing where it makes contact with the flat plate. This separation is influenced by the interaction of the two boundary layers and the momentum transportation by the HSV. A visualisation of the corner separation is given in Figure 2.3 [15].



Figure 2.3: Corner separation on the trailing edge of a DLR-F6 wing connected to a fuselage at angle of attack of 5.0 degrees, image from Kegrise et al. [15].

As described in Gand et al. [7], the HSV is competing with the corner separation. Meaning that a strong HSV co-exists with a smaller corner separation, while a weak HSV results in a larger corner separation. This is because a stronger HSV transports high-momentum air into the corner region which suppresses separation. Thus the hypothesis was established by Gand et al. that the combination of MDF and BF control whether a corner separation is present or not. Since the MDF and BF indicate the strength of the HSV. This hypothesis was, however, invalidated in Gand et al. [7]. In their experiment, a configuration of BF and MDF was chosen such that a corner separation would be present according to previous experiments. Despite the weaker HSV, the corner separation was not observed in the experiment, indicating that other flow parameters are of importance as well.

More recently the idea is established that for turbulent flows the corner separation is also dependent on the gradients of the Reynolds stresses in the corner region. As described in Gand et al. [8] the corner separation depends on the shape and strength of the stress induced vortices (see Figure 2.1b). This is because these vortices influence the boundary layers in the corner region by transporting momentum. The idea is based on a theoretical foundation published by Perkins [16] in which the vorticity equation in free-stream direction is analysed in flows with anisotropic turbulence. This vorticity equation is derived from the steady incompressible RANS momentum equation and formulated as:

$$\begin{aligned}
 U \frac{\partial \xi}{\partial x} + V \frac{\partial \xi}{\partial y} + W \frac{\partial \xi}{\partial z} = \nu \nabla^2 \xi + \underbrace{\xi \frac{\partial U}{\partial x} + \eta \frac{\partial U}{\partial y} + \zeta \frac{\partial U}{\partial z}}_{P1} \\
 + \underbrace{\frac{\partial}{\partial x} \left(\frac{\partial \overline{u'v'}}{\partial z} - \frac{\partial \overline{u'w'}}{\partial y} \right)}_{P2} + \underbrace{\frac{\partial^2}{\partial y \partial z} (\overline{v'v'} - \overline{w'w'})}_{P3} + \underbrace{\left(\frac{\partial^2}{\partial z^2} - \frac{\partial^2}{\partial y^2} \right) \overline{v'w'}}_{P4}
 \end{aligned} \quad (2.3)$$

in which the vorticity vector is defined as $[\xi, \eta, \zeta]^T$, the Reynolds average velocity vector as $[U, V, W]^T$ and the Reynolds fluctuating velocity vector as $[u', v', w']^T$. In Gand et al. [8], it is described that the stress induced vortices are dependent on the turbulent terms (P2, P3 and P4). And since the stress induced vortices influence the boundary layers in the corner, the stress terms influence the corner separation. This idea is echoed by Rumsey et al. [17], in which numerical simulations are performed on the NASA Juncture Flow (JF) experiment [18]. The latter experiment is aimed at improving drag prediction on aircraft, hence the importance of correctly predicting corner separation.

It should be noted, however, that the stress induced vortex (or vortices) is not an isolated phenomenon. Meaning that the shape and size of the co-existing larger vortices, such as the

corner vortex and HSV, influence the characteristics of the stress induced vortex as well (and vice versa). Next to that, the HSV is also responsible for transporting high momentum flow into the corner region and therefore influences the corner separation. It is thus difficult to isolate the phenomena that directly dictate the size of the corner separation since it appears to be a combination of multiple flow and geometry parameters.

2.1.3. Undesirable Effects

The presence of a junction flow can lead to undesirable effects. As previously mentioned, the connection of two bodies causes an extra drag component to arise, the interference drag. The latter is due to the presence of the HSV system and the corner separation. As described by Simpson [2] the rotational sense of the HSV causes high-momentum fluid to mix into the boundary layers, which increases characteristics such as drag and heat transfer. The latter could be an undesirable effect depending on the application of the geometry. Besides, the HSV system also impacts the structural integrity of for example bridges as the HSV scours away soil and rocks surrounding the bridge pier.

Devenport et al. [19] mention how the bimodal behaviour of the HSV can lead to unwanted noise generation on submarines. Next to that, the HSV legs can persist far downstream. This has the consequence that objects in the wake of a junction flow are affected. It is thus important to be able to accurately predict junction flows such that these undesirable effects can be mitigated or controlled.

2.1.4. Wing-Body Fairings

A passive strategy to mitigate the undesirable effects described above in subsection 2.1.3 is by altering the geometry in the corner region. In Devenport et al. [19] a wing-body junction flow experiment is carried out to investigate the effects of a constant radius fillet wrapping around the entire wing, see Figure 2.4a [19]. The flow conditions and baseline geometry are similar to Simpson and Devenport [10]. In the latter study, the flow was characterised by a Re_θ of 6700, a free-stream velocity U_{ref} of 27.0 [m/s] and a fully turbulent boundary layer which was tripped on the plate upstream of the wing and on the wing itself close to the leading edge. The conclusion from the experiment was that a constant radius fillet does neither eliminate the HSV nor the unsteady behaviour. On the contrary, it amplified those effects, presumably because it increases the bluntness of the leading edge. From velocity fluctuation measurements in the wake of the wing, Devenport showed how the fillet magnifies the size and strength of the HSV legs. Besides, using oil flow visualisations, it is shown that the fillet does not eliminate the separation from the flat plate, only an upstream displacement of the HSV was noted.

In a consecutive paper, Devenport et al. [3] investigated a leading-edge fillet, see Figure 2.4b [3], in similar flow conditions as the constant radius fillet. From the results, it was deduced that the leading edge fillet at zero angle of attack reduces the magnitude and extent of the adverse pressure gradient imposed by the wing on the wall. As a consequence, the HSV and its coexisting bimodal behaviour are eliminated. At non-zero angles of attack, a vortex leg has been measured on the pressure side of the wing, which is caused by a stronger adverse pressure gradient located on the fillet at non-zero angles of attack. It should be noted, however, that the effects are still less severe compared to the baseline configuration (i.e. no fillet). It was thus concluded by Devenport et al. [3] that a leading edge fillet modifies the flow over the wing-body junction flow in a desirable way. That is, desirable with respect to the undesirable effects described in subsection 2.1.3.

Globally seen, the undesirable effects caused by the connection of a wing to a flat plate can be passively mitigated by altering the geometry of the leading edge. Although one has to be careful with the design as it can also work counterproductive, see the constant radius fillet.

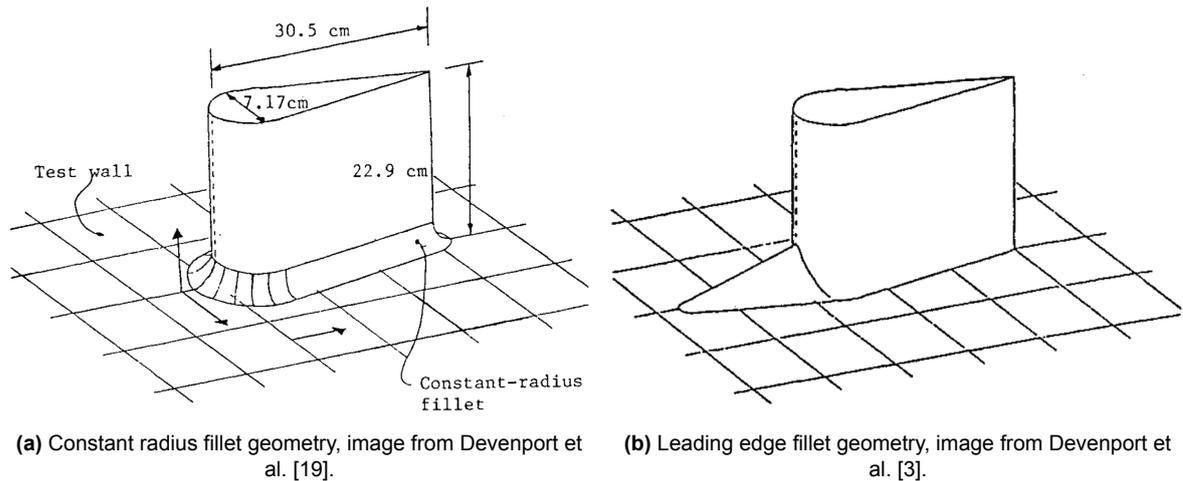


Figure 2.4: Wing-body geometry changes to passively influence the junction flow.

2.1.5. Anti-Fairing

Recently Belligoli et al. [20] found a new drag-reducing geometry for the wing-body junction flow. This new geometry is characterised by two indents in the flat plate, hence it is referred to as the anti-fairing, see Figure 2.5 [20]. The geometry was found by performing aerodynamic shape optimisation in a RANS solver (SU2¹). Belligoli et al. used the SU2 solver to solve the adjoint equations for the design parameters. With the solution to these adjoint equations, i.e. the gradients, the next design iteration could be determined. The full shape optimisation process is described in Belligoli et al. [20]. It should be noted that the original idea for the anti-fairing geometry originates from an aerodynamic shape optimisation by Brezillon and Dwigth [21] of a half-body aircraft (DLR-F6²).

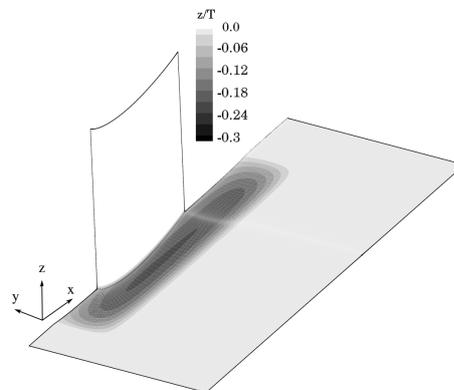


Figure 2.5: Anti-fairing geometry, only one half of the geometry is shown due to the symmetry in the y -axis, image from Belligoli et al. [20].

Multiple numerical and experimental studies have been performed for the anti-fairing which are summarised in Belligoli et al. [22]. Even though most studies in this overview cannot be directly compared to one another due to the mismatch in wing geometry and flow conditions, it is evident that the anti-fairing has a drag-reducing mechanism. Note that all studies in Belligoli et al. [22] were performed in the incompressible flow regime using fully turbulent boundary layers. The drag-reducing mechanism is described by a primary effect, namely a 'thrusting' force

¹<https://su2code.github.io/>, Last accessed: 22-05-2023

²<https://aiaa-dpw.larc.nasa.gov/Workshop2/DLR-F6-geom.html>, Last accessed: 03-04-2023

generated by the pressure distribution over the anti-fairing, as well as a secondary viscous effect that seems to reduce the viscous stress over the flat plate. The primary effect is caused by the incomplete pressure recovery over the indent of the anti-fairing resulting in a net force in the opposite direction of the drag force. The secondary effect, according to Belligoli et al., is caused by the adverse pressure gradient in the upstream convex curve. This adverse gradient reduces the velocity gradient at the wall and hence the wall shear stress, which reduces the total drag. Besides, it was also found that the anti-fairing reduces the turbulent kinetic energy of the flow in the first half of the anti-fairing geometry.

Due to the uncertainties in the studies described in Belligoli et al. [22] complete evidence of the drag reduction was still missing. Hence Belligoli et al. recommended performing wall-resolved LES simulations on a baseline geometry and an anti-fairing geometry, freezing all flow and geometry parameters except for the flat plate geometry. These simulations have been performed by Alberts [23] (baseline) and Vigner [24] (anti-fairing), unfortunately however, drag comparisons between the two simulations have not been published yet. This will therefore be addressed in Chapter 5 in this thesis.

2.2. Numerical Simulation

In this section, various numerical studies regarding the wing-body junction flow are discussed. Both Reynolds-Average Navier Stokes (RANS) and Large Eddy Simulations (LES) will be covered as well as uncertainty sources in the RANS modelling approach.

2.2.1. RANS Simulations

Numerous literature report the poor performance of the RANS simulation technique for the wing-body junction flow (Ryu et al. [4], Gand et al. [7], Rumsey et al. [17], Alberts [23]). This is mainly due to the highly unsteady nature of the problem as well as the anisotropic behaviour of the turbulence near the corner region.

A study on the accuracy of different RANS turbulence models for the wing-body geometry (using the Rood wing [12]) is given in Apsley and Leschziner [25]. In this study, linear and non-linear eddy viscosity models were tested as well as second-moment closure models. The wing geometry and boundary layer thickness (at the location of measurement) for this study were identical to Simpson and Devenport [10], the Reynolds number based on wing thickness was equal to $Re_T = 1.15 \cdot 10^5$. From the extensive research, to which six partners contributed, it was concluded that the second-moment closure models have a better accuracy compared to the (non-) linear eddy viscosity models. Although, the $k-g$ model (Kalitzin et al. [26]) and $k-\omega$ *SST* model (Menter [27]) came close to the results of the second-moment closure models for the mean flow quantities, but failed to correctly predict the anisotropy of the turbulence. It should be noted, however, that even with the better accuracy of the second-moment closure models, the results had noticeable discrepancies in both the mean-field and turbulence quantities with respect to the experimental reference data. Besides, using second-moment closure models (or Reynolds stress models) will increase the computational cost and decrease the convergence rate compared to linear eddy viscosity models (Hickel [28]).

Regarding eddy viscosity models, in Gand et al. [8] an investigation was conducted on the predicting performance of the quadratic constitutive relation (QCR) (Spalart [29]) for the corner separation of the wing-body junction flow. This QCR modification modifies the Boussinesq closure by adding non-linear terms. The QCR modification is formulated as (Gand et al. [8]):

$$\tau_{ij} = \bar{\tau}_{ij} - c_{nl1} (O_{ik}\bar{\tau}_{jk} + O_{jk}\bar{\tau}_{ik}) \quad \text{with} \quad O_{ik} = \frac{\partial_k U_i - \partial_i U_k}{\sqrt{\partial_n U_m \partial_n U_m}} \quad (2.4)$$

In which τ_{ij} equals the Reynolds stress tensor, $\bar{\tau}_{ij}$ the Reynolds stress tensor calculated with the Boussinesq closure, U_i the mean velocity and c_{nl1} a modelling constant. The aim of the

QCR addition is to better predict the components of the Reynolds stress tensor. This would result in better predicted stress induced vortices, see Equation 2.3. Which in turn will result in a more accurate corner separation prediction, see subsection 2.1.2.

For the RANS simulation in Gand et al. [8] the Spalart Allmaras (SA) model, see Spalart and Allmaras [30], was used with and without QCR modification. The flow was characterised by a Reynolds number based on the chord length of $Re_c = 3 \cdot 10^5$ and a boundary layer thickness of $\delta_0/T = 0.42$ measured 0.3 chord lengths upstream of the wing. The wing geometry was based on a NACA 0015 airfoil. From the mean-field results, it was concluded that the QCR closure brings significant improvements compared to the SA model without QCR. The SA with QCR model showed only minor discrepancies with respect to the experimental reference data on the aft part of the wing for the mean-field quantities. From the Reynolds stress analysis, it was deduced that, in particular, the QCR closure was able to more correctly predict the $\overline{v'w'}$ and $\overline{v'v'} - \overline{w'w'}$ terms in the corner region, resulting in a more accurate prediction of P3 and P4 in Equation 2.3. This correctly influences the boundary layer in the corner region resulting in a realistic corner separation (compared to the standard SA prediction). It was therefore concluded that the QCR modification to Boussinesq's hypothesis improves the RANS prediction for wing-body junction flows.

More recently a RANS simulation on a junction flow with an updated version of the SA model with QCR modification is given by Rumsey et al. [17]. The updated version of QCR, QCR2020 instead of QCR2000 used by Gand et al., applies the non-linear correction to the Boussinesq hypothesis using a wall distance function (high values close to the wall) instead of a constant (c_{nl1}). Another discrepancy with Gand et al. is that Rumsey et al. use a half-aircraft geometry instead of a wing mounted to a flat plate. Similar to Gand et al., Rumsey et al. concluded that the QCR modification improved the predicted corner separation with respect to the experimental data, also describing that the spread between the normal stresses (for example $\overline{v'v'} - \overline{w'w'}$) have moved into the right direction. Even though, the individual normal stresses still showed discrepancies compared to the experiment.

These two individual simulations show that using the QCR modification on the Boussinesq closure will improve the RANS prediction of the corner separation. Although it must be noted that comparisons with experimental data close to the leading edge of the wing were not given. And hence no conclusion can be drawn on the prediction performance of the origin and location of the HSV which is typically mispredicted by RANS.

2.2.2. RANS Uncertainty Sources

As described above it is difficult to get accurate flow solutions using the RANS simulation technique on the wing-body junction flow. This is due to systematic errors in the formulation of the RANS equations. In Duraisamy et al. [6] an overview of the RANS simplification levels is given. The first error source comes from level 1 (L1) and is due to the fact that the Reynolds average of the Navier Stokes equations (\mathcal{N}) is not equal to the Reynolds averaged flow variables substituted into the Navier Stokes:

$$\langle \mathcal{N}(\cdot) \rangle \neq \mathcal{N}(\langle \cdot \rangle) \quad (2.5)$$

It is thus required to define closure models $\mathcal{M}(\cdot)$:

$$\langle \mathcal{N}(\cdot) \rangle = \mathcal{N}(\langle \cdot \rangle) + \mathcal{M}(\cdot) \quad (2.6)$$

The development of these closure models, such as the Boussinesq hypothesis, defines the simplification level 2 (L2). The closure model $\mathcal{M}(\cdot)$ consists of (differential) equations for the independent variables w . Simplification level 3 (L3) is defined by establishing these equations $\mathcal{P}(w)$, an example of $\mathcal{P}(w)$ are the k and ω equations by Wilcox [31]. Finally, level 4 (L4) is

defined by the calibration of the model coefficients c used in the model equations $\mathcal{P}(w)$. The ensemble of the four levels defines the error source made in the RANS modelling technique in CFD. The decomposition of the error source into these four levels allow authors to indicate which specific error source or uncertainty they are reducing.

2.2.3. LES Simulations

A more accurate technique for simulating the wing body junction flow is by using Large Eddy Simulations. In contrast to RANS, in LES a portion of the turbulent scales, that is the most energetic scales, are resolved rather than modelled (Hickel [32]). This has the benefit that the anisotropy of the turbulence is much better predicted compared to RANS turbulence models. Which improves the accuracy of the flow prediction on the wing-body junction flow. The downside of LES, as mentioned in the introduction (Chapter 1), is the significantly higher computational cost compared to RANS. Hence the LES method is less popular in the industry where often multiple simulations are required for a single project.

In Ryu et al. [4], as well as the similar simulation by Alberts [23], a wall-resolved LES simulation of the wing-body junction flow was carried out. For the setup of the simulation, the geometry and flow characteristics were matched to the experiment by Devenport and Simpson [10]. In general, the mean flow structures by the LES were in accordance with the experimental result, indicating that LES simulations are superior to RANS simulations for the wing-body junction flow in terms of accuracy. This conclusion was also drawn in Gand et al. [7], in which a wing-body LES was performed on a wing with a different geometry compared to Ryu et al. [4]. Next to that, by computing the statistics in the corner region upstream of the wing, Alberts [23] was able to replicate the bimodal behaviour of the HSV as reported by Devenport and Simpson [10]. This is another indication that wall-resolved LES is able to reproduce experimental results. However, it is noticed by Ryu et al. that there still exist small discrepancies in the Reynolds stresses in the corner region. According to Ryu et al. this is likely due to the inaccuracy of the subgrid-scale model used, the Vreman model [33].

Besides the comparison study, an investigation by Ryu et al. [4] was performed in the leading edge corner region of the wing-body geometry, a region which is complicated to measure in an experimental setup. In this region, the authors showed that the anisotropy of the Reynolds normal stresses showed a resemblance with the secondary vortex structures in this region. Indicating again that this anisotropy, not well captured by RANS, is important for the formation of these smaller secondary vortex structures close to the walls. The latter observation is in coherence with the paper published by Gand et al. [8] see subsection 2.1.2.

An alternative to wall-resolved LES is wall-modelled LES, in which the flow in close distance to a wall is modelled by a wall model instead of resolved. This allows for a coarser mesh and hence a cheaper simulation. In Belligoli et al. [22] a wall-modelled LES is described for the anti-fairing geometry (see subsection 2.1.5) using a $y^+ = 50$ mesh. From the results it was concluded that the wall-modelled LES showed good agreement in the boundary-layer development, wall pressure contours and vorticity fields in comparison with experimental data. Although it was noticed that the turbulent kinetic energy had half the magnitude of the experimental data. This is due to the insufficient mesh resolution according to Belligoli et al. [22]. Hence for accurate turbulence fields, it is desirable to wall resolve the LES simulation.

2.3. Data-Driven Turbulence Modelling

As mentioned in subsection 2.2.2 in RANS a model for the Reynolds stress tensor is required to close the Reynolds-averaged Navier Stokes. These models are often referred to as turbulence models. Classical models might use the Boussinesq hypothesis, in which the Reynolds stresses are modelled based on the local velocity gradients and a turbulent viscosity. The

latter is often determined by solving extra transport equations, e.g. the k - ω model by Wilcox [31]. Although these classical models are effective, they can introduce significant errors into the flow field, such as for the junction flow as described in subsection 2.2.1. Hence, in the modern era, more focus is pointed towards data-driven techniques to improve these classical turbulence models. In these techniques, high-fidelity turbulence data is used to improve the turbulence models. Due to the high variety of data-driven techniques, the emphasis in this section is on supervised machine learning methods that are already applied to turbulence modelling problems to narrow down the research area.

2.3.1. General Effective-Viscosity Hypothesis

Before going over the various machine learning methods applied in the field of RANS modelling it is important to note that many of them rely on the general effective-viscosity hypothesis established by Pope [34]. In this hypothesis, it is stated that under certain conditions the Reynolds stresses can be determined by a ten-term tensor polynomial based on the turbulent kinetic energy (k), the turbulent energy dissipation rate (ϵ) and the gradients of the mean velocity ($\partial_i U_j$). Expressed as an equation this tensor polynomial yields:

$$\overline{u'_i u'_j} = \frac{2}{3} k \delta_{ij} + k \sum_{\lambda=1}^{10} G^{(\lambda)} T_{ij}^{(\lambda)} \quad (2.7)$$

in which k equals the turbulent kinetic energy, δ_{ij} equals the Kronecker delta, G a scalar function based on the tensor invariants and $T_{ij}^{(\lambda)}$ a set of tensors based on the mean strain-rate and rotation-rate tensors:

$$\mathbf{S} = S_{ij}^* = \frac{1}{2} \tau \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad \text{and} \quad \mathbf{\Omega} = \Omega_{ij}^* = \frac{1}{2} \tau \left(\frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i} \right) \quad (2.8)$$

The tensors $T_{ij}^{(\lambda)}$ and invariants ($I \dots V$) are defined as follows (Pope [34]):

$$\begin{aligned} T_{ij}^1 &= \mathbf{T}^1 = \mathbf{S} & \mathbf{T}^2 &= \mathbf{S}\mathbf{\Omega} - \mathbf{\Omega}\mathbf{S} \\ \mathbf{T}^3 &= \mathbf{S}^2 - \frac{1}{3} \mathbf{I} \{ \mathbf{S}^2 \} & \mathbf{T}^4 &= \mathbf{\Omega}^2 - \frac{1}{3} \mathbf{I} \{ \mathbf{\Omega}^2 \} \\ \mathbf{T}^5 &= \mathbf{\Omega}\mathbf{S}^2 - \mathbf{S}^2\mathbf{\Omega} & \mathbf{T}^6 &= \mathbf{\Omega}^2\mathbf{S} + \mathbf{S}\mathbf{\Omega}^2 - \frac{2}{3} \mathbf{I} \{ \mathbf{S}\mathbf{\Omega}^2 \} \\ \mathbf{T}^7 &= \mathbf{\Omega}\mathbf{S}\mathbf{\Omega}^2 - \mathbf{\Omega}^2\mathbf{S}\mathbf{\Omega} & \mathbf{T}^8 &= \mathbf{S}\mathbf{\Omega}\mathbf{S}^2 - \mathbf{S}^2\mathbf{\Omega}\mathbf{S} \\ \mathbf{T}^9 &= \mathbf{\Omega}^2\mathbf{S}^2 + \mathbf{S}^2\mathbf{\Omega}^2 - \frac{2}{3} \mathbf{I} \{ \mathbf{S}^2\mathbf{\Omega}^2 \} & \mathbf{T}^{10} &= \mathbf{\Omega}\mathbf{S}^2\mathbf{\Omega}^2 - \mathbf{\Omega}^2\mathbf{S}^2\mathbf{\Omega} \end{aligned} \quad (2.9)$$

$$I = \{ \mathbf{S}^2 \} \quad II = \{ \mathbf{\Omega}^2 \} \quad III = \{ \mathbf{S}^3 \} \quad IV = \{ \mathbf{\Omega}^2\mathbf{S} \} \quad V = \{ \mathbf{\Omega}^2\mathbf{S}^2 \} \quad (2.10)$$

Note that \mathbf{S}^2 describes a tensor product, i.e. $S_{ik}^* S_{kj}^*$, $\{ \cdot \}$ represents the trace of a tensor and τ the timescale used for non-dimensionalisation. Note that by using \mathbf{T}^1 only, the linear eddy-viscosity model is found, frequently referred to as the Boussinesq hypothesis. As described by Pope [34], for the general effective-viscosity hypothesis to be correct it is important that the rates of strain are nearly homogeneous in the flowfield. Only then the Reynolds-stresses can be determined locally, i.e. by k , ϵ and $\partial_i U_j$, as no turbulence is transported through the flowfield. Therefore, corrections might be required for turbulence models that are based on this hypothesis when applied to flows with significant velocity gradients, such as flows over wings, cylinders etc.

2.3.2. Explicit Algebraic Models

A popular turbulence model for data-driven techniques is an explicit algebraic Reynolds stress model, often referred to as EARSM or EASM. As the name suggests in these types of models the Reynolds stress is determined via an explicit algebraic equation. This often comes down to using the general effective hypothesis (see subsection 2.3.1), hence the goal of turbulence modellers is to find the scalar functions G in Equation 2.7.

The origin of EARSMs was established by authors such as Gatski and Speziale [35] and Wallin and Johansson [36]. In both [35] and [36] EARSMs are found by analytical methods for both 2-Dimensional (2D) and 3D flows. For the derivation of these models, the general form of the algebraic (Reynolds) stress model is used as a starting point. This model is derived from the Reynolds stress transport equation for which it is assumed that the turbulent flow is in equilibrium and homogeneous. It should be noted that this is quite a significant assumption, however, similar to the underlying assumption of the general effective viscosity hypothesis. In this algebraic stress model, a tensorially linear model is substituted for the pressure-strain term. The selection of such a linear pressure-strain model is up to the turbulence modeller and one of the differences between the EARSM derived by Gatski and Speziale [35] and the EARSM by Wallin and Johansson [36]. Consecutively, Pope's tensor basis (Equation 2.7) is substituted into the stress model which can be rewritten as a linear system. This system is then solved for the scalar functions G which in combination with the tensor basis results in an EARSM. Note that the full derivation is given in Gatski and Speziale [35].

Although both 2D and 3D flow EARSMs are derived by Gatski and Speziale [35], only results of the 2D version are shown. This is, according to Gatski and Speziale, due to the (numerical) regularisation required for the found EARSM when applied to non-equilibrium turbulent flows. As these flows provoke the underlying assumptions and hence cause numerical instabilities. According to Gatski and Speziale finding a regularised version of the 3D EARSM is complicated and hence only a demonstration of the 2D version is given. In this successful demonstration, the 2D EARSM was applied to a homogeneous shear flow in a rotating frame and showed a good resemblance with the reference LES data.

In Wallin and Johansson [36] an EARSM is presented that gives the exact solution to the algebraic stress model in 2D and a good, and numerically stable, solution in 3D. Wallin and Johansson showed a successful demonstration of the EARSM for an axially rotating pipe flow. A 3D type of flow for which the prediction accuracy of two-equation linear eddy viscosity models is low. Besides [36] also reports improved prediction of the EARSM in adverse pressure gradient boundary layers.

Multidimensional Gene Expression Programming

A data-driven method of establishing an EARSM is for example the Multidimensional Gene Expression Programming (MGEP) algorithm proposed by Weatheritt and Sandberg [37]. This is a symbolic regression algorithm based on a survival of the fittest methodology. In MGEP an explicit expression for a correction term in the anisotropic part of the Boussinesq hypothesis is aimed to be found (Weatheritt and Sandberg [37]):

$$\tau_{ij} = \bar{\tau}_{ij} + a_{ij}^x \quad (2.11)$$

In this equation $\bar{\tau}_{ij}$ equals the Boussinesq Reynolds stress and a_{ij}^x the anisotropic correction term. The explicit expression for this correction term is found by creating a random set of pre-selected candidate functions which are tested on their fitness via a fitness function. The fittest candidates then randomly reproduce themselves with each other to generate a new generation. This new generation then undergoes the same reproduction process, which continues until convergence is achieved. By incorporating plasmids into the standard GEP algorithm

Weatheritt and Sandberg [37] were able to regress tensors with the GEP algorithm, hence Multidimensional GEP. As candidate functions for the correction model Weatheritt and Sandberg use a combination of the tensors and tensor invariants defined by Pope [34] (see subsection 2.3.1). And hence, in essence, an EARSM is aimed to be found. However, instead of using an analytical approach a data-driven approach is used. The advantage of using MGEP over for example black box machine learning methods (e.g. Neural Networks) is that it generates an explicit function. It, therefore, has the possibility to contribute to the understanding of turbulence modelling since it can be understood by the creator.

In Weatheritt and Sandberg [37] it is shown that using the MGEP algorithm the *a posteriori* results of a periodic hill flow can be significantly improved compared to baseline RANS. Note that the algorithm was trained using high-fidelity data on a different geometry namely the backwards facing step.

In a follow-up publication, Weatheritt and Sandberg [38], an extension is made to the above described evolutionary algorithm. Instead of training the algorithm with RANS results as inputs and high fidelity (hifi) results as the target, the authors opt for the frozen approach. In this approach the transport equation for the specific dissipation rate ω (from the k - ω SST RANS closure by Menter [27]) is solved passively by freezing the velocity field U , turbulent kinetic energy field k and Reynold stresses, these variables are provided by the hifi simulation. The resulting U , k (from the hifi results) and ω (from the frozen approach) are then used as inputs for the MGEP algorithm. The authors have chosen this approach since the physical meaning of ω , that is the turbulent time scale, of an eddy-resolving simulation is not representative of the scale of ω in a RANS simulation. The frozen approach, therefore, acts as a transformation of the hifi simulation to a RANS simulation with equivalent flow fields for U , k and τ_{ij} . The extension of the original MGEP method allowed the authors to successfully improve more complex flows such as a diffuser.

Sparse Regression of Turbulent Stress Anisotropy

A similar data-driven method to MGEP is the Sparse Regression of Turbulent Stress Anisotropy (SpaRTA) method by Schmelzer et al. [39]. This method tries to find explicit tensor polynomials that aim to improve the closure of RANS based on hifi data. In other words, the method tries to find an EARSM based on hifi turbulence data. In the baseline version of SpaRTA, explicit models are regressed that predict correction fields for the anisotropic part of the Reynolds stress tensor (see Equation 2.11) and for the turbulent kinetic energy. These correction fields are found by using the k -corrective-frozen-RANS approach, which is similar to the frozen approach described above by Weatheritt and Sandberg [38]. The only difference is that besides solving the ω equation, the residual with respect to the hifi data of the k transport equation is calculated as well. As the k residual represents the correction field for k . Finally, a library of candidate functions, based on the set of tensors defined by Pope [34], is sparsely regressed to these two correction fields using the elastic net regression algorithm. These two models correct the anisotropic part of the Reynolds stress and the value of k in the standard RANS equations using the k - ω SST model. By comparing this method to the MGEP method many similarities can be identified. The two main differences are that SpaRTA also corrects the k -equation instead of only the Reynolds stress tensor and that MGEP uses an evolutionary algorithm for the regression instead of the elastic net algorithm.

The advantage of the SpaRTA method, similar to MGEP, is that by finding sparse explicit correction functions a better physical understanding of the shortcomings of a standard closure model, in this case, the k - ω SST, can be created.

In Schmelzer et al. [39] a successful demonstration of SpaRTA on the Periodic Hill, Curved Backward Facing Step and Converging Diverging channel flow are shown by implementing the found correction models into a solver. A true (unseen) prediction on a significantly increased

Reynolds number is also utilised to assess performance.

An important base for the research in this work is the thesis by Vigner [24]. Vigner showed a successful propagation of the correction fields found by the frozen approach for the wing-body junction flow. This resulted in mean fields that look very similar to the original LES. Note that the geometry and flow characteristics are matched to Devenport and Simpson [10] since the wall-resolved LES training data was provided by Alberts [23]. Thus, the first steps of the SpaRTA method were shown to be effective for this type of three-dimensional flow and hence provide a promising basis for this work. This is because Vigner did not publish a complete set of explicit models for the two SpaRTA correction fields nor a propagation of such models.

In Huijing [40] a successful operation of SpaRTA on three three-dimensional flows is shown. The three geometries include a wall-mounted cube, an infinite cylinder and an idealised rotating wheel. All three geometries were simulated in the incompressible flow regime using Reynolds numbers varying from 10^4 to 10^6 . Huijing showed that the regressed models improved especially the separated areas over the geometries significantly. Next to that, by applying the models trained on one geometry to another it was found that the wall-mounted cube model improved the flow with respect to the baseline over all three geometries tested.

2.3.3. Tensor Basis Neural Network

The Tensor Basis Neural Network (TBNN) described in Ling et al. [41] is a neural network algorithm that combines the learning capability of a neural network with the tensor decomposition of the anisotropic part of the Reynolds stress described in Pope [34], see Equation 2.7. The multi-level neural network (NN) proposed by Ling et al. [41] takes the flow invariants as the input layer and multiplies the output of the neural network ($G^{(\lambda)}$) by the tensor set ($T_{ij}^{(\lambda)}$) to generate the anisotropic stress tensor. The used NN had eight hidden layers each with thirty nodes. It is important to note that during the training phase RANS results were used as inputs and DNS/LES results as truth labels. This is because the NN should be able to predict $\overline{u'_i u'_j}$ in the absence of DNS data. The advantage of using the TBNN structure is the fact that the algorithm remains Galilean invariant. This is desirable since the Navier-Stokes equations are Galilean invariant. The downside of using a NN is that it remains a black box. This means that although the NN might improve CFD results, its workings can not be understood by its creator. Therefore the NN can not directly improve the understanding of turbulence modelling.

Note that in contradiction to SpaRTA, Ling et al. are not adapting an existing turbulence model to improve results. But rather modelling the Reynolds stresses directly using the general-effective viscosity hypothesis.

The goal of the paper was to generate a network that would be generally applicable to multiple flows to predict b_{ij} . And therefore disregard the use of a linear eddy viscosity model (LEVM). Ling et al. [41] concluded from the *a posteriori* results that the network was not able to reproduce the direct numerical simulation (DNS) data with high fidelity. However, the network did generate correct flow structures that were not predicted by the LEVMs.

2.3.4. Gaussian Process Regression

In Zhang and Duraisamy [42] Gaussian processes (GP) and NN's have been used to improve turbulence modelling for a 1D channel flow case and a 2D flat plate flow. In the channel flow the k transport equation in the k - ω model (Wilcox [31]) was modified by multiplying the production term P_k with an adjustment term α . For the flat plate flow, the focus was put on the transition of the flow. Hence, the transport equation for the intermittency factor γ was modified by replacing the sum of the production and destruction terms with α . The adjustment factor α for both flows was found by solving the inverse problem with Bayesian inversion for which DNS or experimental data has been used. It was concluded by Zhang and Duraisamy [42]

that both GP and NN were able to correctly predict α on test cases for both flows. These were training data cases on which the model has not been trained. However, using training data as test cases implied that a CFD prediction utilising the GP and NN models was missing.

2.3.5. Flow Classifier

A different approach to using ML in RANS modelling is by means of flow classifiers. In Ling and Kurzawski [43] a flow classifier is presented that can identify regions within a flow for which the turbulence is significantly anisotropic. This information can be used to identify regions in which the classical linear eddy viscosity model (the Boussinesq hypothesis) breaks down. In Ling and Kurzawski [43] a non-linear eddy viscosity model is applied in these regions to improve the flow solution. As described by the authors, applying the non-linear model in the entire flow field is unfeasible due to stability issues. Therefore, in the regions not tagged by the classifier, the linear eddy viscosity model is applied. By using this approach Ling and Kurzawski are reducing the level 2 uncertainty of RANS modelling (see subsection 2.2.2).

The development of the flow classifier is described in Ling and Templeton [44] in which different machine learning methods are used together with DNS and LES data to generate the classifier. Ling and Kurzawski [43] tested the method on the fully developed turbulent duct flow and on the wavy wall with scalar injection flow. From the results, it was concluded that, for both flow fields produced with the classifier model, the results have improved accuracy compared to the standard linear eddy viscosity model. However, the results did not perfectly match the DNS data. Ling and Kurzawski propose this is due to the simplicity of the non-linear model used for the tested flows. And describe that this paper is rather a demonstration of the usefulness of using flow classifiers, that is improved accuracy while keeping good stability and convergence, than a proposal for a more accurate RANS model.

Taking this line of thought a step further is done by Steiner et al. [45]. In Steiner et al. [45] the SpaRTA methodology is combined with a flow classifier for a wind turbine application. To recall, the SpaRTA method aims at finding two correction models for a baseline two-equation RANS turbulence model by leveraging high-fidelity data. Steiner et al. [45] noticed that the baseline model was accurate enough in certain regions and therefore there was no need to apply any correction model, similar to Ling and Kurzawski [43]. However, in the wake of the turbines corrections were required. Hence Steiner et al. [45] trained a classifier model using a sparse regression method that identified regions in which the SpaRTA corrections should be activated. From the results, it was concluded that this method was as effective as some significantly more complex models. However, the combined classifier-correction models did decrease the stability of the simulations. The latter was solved by applying numerical limiters.

2.4. Literature Study Conclusion

The aim of this chapter was to get an overview of the literature, and thus the research, available in the field of wing-body junction flows and turbulence modelling. In this section, the research gap concerning this work will be extracted from the literature study. And consecutively the research question will be formulated.

2.4.1. Research Gap

An important task when conducting scientific research is to identify the research gap. Which in this case comes from multiple angles. First of all, it is recognised that an accurate prediction of the mean junction flow using a steady RANS solver is missing. From several experimental and numerical studies, it is shown that especially the unsteady behaviour of the junction flow as well as the dependence on the anisotropy of the turbulence makes it difficult to get an accurate prediction with the current RANS turbulence models. This leaves an opportunity for finding an

improved turbulence model via a data-driven approach.

Regarding these data-driven methods, it has been shown that these can be successful for 2D and simple 3D cases. However, the amount of successful and/or unsuccessful demonstrations on more complex 3D cases, such as the junction flow, is somewhat limited. This leaves another opportunity for research, namely to test if it is possible to apply a machine learning technique to the junction flow. The latter also partly rests on the NASA Symposium on Turbulence Modelling 2022 (Rumsey and Coleman [46]), in which it is concluded that machine learning in the field of turbulence modelling still has a long way to go. And, therefore, both successful and especially unsuccessful attempts in this field should be published in order to increase common knowledge. Since frankly the improvements in turbulence modelling have stalled from the late 1990s onwards (see Figure 2 in Xiao and Cinnella [47]).

Finally, if an enhanced turbulence model is found for the junction flow it will create an opening for more accurate design optimisations of the junction geometry using RANS solvers. However, this is rather a futuristic goal than an actual gap for this work.

These three opportunities combined generate a rather large research space, which is too big to explore within one Master's thesis. Hence the objective stated in Chapter 1 is reformulated into a more precise form:

To improve the $k-\omega$ SST turbulence model by means of the SpaRTA machine learning approach such that the flow field generated by the RANS equations for the wing-body junction flow is close to that of a wall-resolved LES simulation.

The choice for the SpaRTA method was made based on the fact that in-house expertise is available on this method, as well as the partly-successful demonstration by Vigner [24] and the more complete demonstration by Huijing [40] for 3D geometries.

2.4.2. Research Questions

On the basis of the research gap rests the main research question, which for this work is defined as follows:

Can the discrepancy in the flow field of a standard RANS simulation with respect to a large eddy simulation for the wing-body junction flow be reduced using the SpaRTA machine learning algorithm?

This main question is accompanied by the following sub-questions:

- *Which regions in the RANS solution require significant corrections?*
- *What is the error between the regressed model and the training data?*
- *What are the discrepancies in the flow fields between the propagated correction fields and the high-fidelity simulation?*
- *Do the regressed models run stable in a RANS simulation?*
- *What are the discrepancies in the flow fields between RANS with correction models and the high-fidelity simulation?*
- *Is the vortex topology different between the LES and the baseline RANS?*
- *Is the vortex topology present in the LES solution predicted by the custom turbulence model?*
- *Is the new turbulence model generally applicable? Will the new model also improve wing-body junction flows with different types of fairings and wing geometries?*

Next to these sub-questions is a question not directly related to the main research question:

- *What is the difference in drag force between the LES on the baseline wing-body geometry by Alberts [23] and the LES on the anti-fairing geometry by Vigner [24]?*

3

Data-Driven Modelling Methodology

In this chapter, the numerical methodology behind the results displayed in this work is discussed. First in Section 3.1 the equations behind the baseline Reynolds-Averaged Navier-Stokes (RANS) are described. Next to that, this section also describes the computational setup to solve the RANS equations. Secondly, in Section 3.2, the SpARTA approach is explained, which includes the k -corrective frozen approach, the propagation and model regression.

3.1. Reynolds-Averaged Navier-Stokes

The simulations described in this work all rely on the fundamentals of the Reynolds-Averaged Navier-Stokes (RANS) equations. Which is a set of equations that describe the mean flow. The derivation of this set of equations, as well as the computational setup to solve them will be discussed in the following subsections.

3.1.1. Governing Equations

The derivation of the RANS equations starts with the definition of the Navier-Stokes (NS) equations. The corresponding NS equations for a flow problem in the incompressible flow regime with constant density and constant viscosity are given by (Pope [48]):

$$\begin{aligned} \frac{\partial u_i}{\partial x_i} &= 0 \\ \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} &= f_i - \frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \end{aligned} \quad (3.1)$$

In which the top equation represents the conservation of mass and the bottom equation the conservation of momentum. Note that u_i , x_i and f_i represent the velocity, position and body force vector respectively in index notation. Next to that, t , p and ν equal the time, air pressure and kinematic viscosity respectively.

As the naming suggests to obtain the RANS equations the variables should be Reynolds-averaged. To achieve this the two flow variables are decomposed using Reynolds decomposition into a mean component (U_i and P) and a fluctuating component (u'_i and p'):

$$u_i = U_i + u'_i \quad \text{and} \quad p = P + p' \quad (3.2)$$

The RANS equations are subsequently achieved by substituting Equation 3.2 into the NS equations and ensemble averaging the resulting system. The derivation for each term in Equa-

tion 3.1 is given below, note that $\langle \cdot \rangle$ operator indicates taking the ensemble average.

$$\langle u_i \rangle = \frac{1}{N} \sum_{\mu=1}^N u_i|_{\mu} \quad (3.3)$$

$$\begin{aligned} \left\langle \frac{\partial(U_i + u'_i)}{\partial x_i} \right\rangle &= \frac{\partial(\langle U_i \rangle + \langle u'_i \rangle)}{\partial x_i} = \frac{\partial U_i}{\partial x_i} \quad \text{since} \quad \langle U_i \rangle = U_i \quad \text{and} \quad \langle u'_i \rangle = 0 \\ \left\langle \frac{\partial(U_i + u'_i)}{\partial t} \right\rangle &= \frac{\partial(\langle U_i \rangle + \langle u'_i \rangle)}{\partial t} = \frac{\partial U_i}{\partial t} \\ \left\langle \frac{1}{\rho} \frac{\partial(P + p')}{\partial x_i} \right\rangle &= \frac{1}{\rho} \frac{\partial(\langle P \rangle + \langle p' \rangle)}{\partial x_i} = \frac{1}{\rho} \frac{\partial P}{\partial x_i} \\ \left\langle \nu \frac{\partial^2(U_i + u'_i)}{\partial x_j \partial x_j} \right\rangle &= \nu \frac{\partial^2(\langle U_i \rangle + \langle u'_i \rangle)}{\partial x_j \partial x_j} = \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} \end{aligned} \quad (3.4)$$

The non-linear term in the momentum equation is rewritten in order to perform the averaging:

$$\begin{aligned} \left\langle (U_j + u'_j) \frac{\partial(U_i + u'_i)}{\partial x_j} \right\rangle &= \left\langle \frac{\partial(U_i U_j + U_i u'_j + u'_i U_j + u'_i u'_j)}{\partial x_j} \right\rangle = U_j \frac{\partial U_i}{\partial x_j} + \frac{\partial \langle u'_i u'_j \rangle}{\partial x_j} \\ &\quad \text{since} \quad \langle U_i u'_j \rangle = \langle u'_i U_j \rangle = 0 \quad \text{and} \quad \langle U_i U_j \rangle = U_i U_j \end{aligned} \quad (3.5)$$

Combining Equation 3.4 and Equation 3.5, assuming zero body forces ($f_i = 0$) and assuming that the solution is steady state ($\partial U_i / \partial t = 0$) results in the steady state incompressible RANS equations:

$$\begin{aligned} \frac{\partial U_i}{\partial x_i} &= 0 \\ U_j \frac{\partial U_i}{\partial x_j} &= -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} - \frac{\partial \overline{u'_i u'_j}}{\partial x_j} \end{aligned} \quad (3.6)$$

Note that the average operator has been replaced, i.e. $\langle u'_i u'_j \rangle = \overline{u'_i u'_j}$. Comparing Equation 3.6 with Equation 3.1 shows that the system of equations is very similar. Yet an extra term is present in the RANS formulation, that is the divergence of the Reynolds stress tensor $\overline{u'_i u'_j}$. The RANS system of equations is usually solved for the mean flow quantities, U_i and P , since this reduces the computational cost significantly compared to eddy-resolving techniques such as LES or DNS. This does, however, mean that the Reynolds stress tensor has to be modelled using a turbulence model as there is no information available on the fluctuating velocity components. As described by Duraisamy et al. [6] (subsection 2.2.2) this modelling step is one of the significant error sources within a RANS simulation.

In this work the k - ω SST model by Menter [27] has been used as the baseline turbulence model to predict the Reynolds stress tensor. The k - ω SST model is a Linear Eddy Viscosity Model (LEVM) and relies on the Boussinesq closure to model the Reynolds stress tensor. This closure model is given by (Pope [48]):

$$\overline{u'_i u'_j} = -2\nu_t \left(S_{ij} - \frac{1}{3} \frac{\partial U_k}{\partial x_k} \delta_{ij} \right) + \frac{2}{3} k \delta_{ij} \quad \text{with} \quad S_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad (3.7)$$

In which ν_t equals the kinematic eddy viscosity, k the turbulent kinetic energy and δ_{ij} the Kronecker delta. Note that since the flow is incompressible $\partial U_k / \partial x_k = 0$. To obtain ν_t and k the transport equations for k and ω (specific dissipation rate) as defined by Menter [27] are numerically solved:

$$\frac{\partial k}{\partial t} + \frac{\partial U_j k}{\partial x_j} = P_k - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right] \quad (3.8)$$

$$\frac{\partial \omega}{\partial t} + \frac{\partial U_j \omega}{\partial x_j} = \frac{\gamma}{\nu_t} P_k - \beta \omega^2 + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_\omega \nu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \frac{\sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \quad (3.9)$$

In which the production term (P_k) and eddy viscosity are determined via:

$$P_k = -\overline{u'_i u'_j} \frac{\partial U_i}{\partial x_j} \quad \text{and} \quad \nu_t = \frac{a_1 k}{\max(a_1 \omega, \sqrt{2\Omega_{ij}\Omega_{ij} F_2})} \quad \text{with} \quad \Omega_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i} \right) \quad (3.10)$$

For numerical stability a limiter is used for the production term:

$$P_k = \min(P_k, 10\beta^* \omega k) \quad (3.11)$$

The additional functions in the transport equations are given by (d equals the wall distance):

$$\begin{aligned} CD_{k\omega} &= \max \left(2\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-20} \right) \\ F_1 &= \tanh \left(\min \left[\max \left(\frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\nu}{d^2 \omega} \right), \frac{4\sigma_{\omega 2} k}{CD_{k\omega} d^2} \right]^4 \right) \\ F_2 &= \tanh \left(\max \left[2 \frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\nu}{d^2 \omega} \right]^2 \right) \end{aligned} \quad (3.12)$$

The modelling constants, referred to as ϕ , are determined via a blending function between ϕ_1 and ϕ_2 :

$$\phi = F_1 \phi_1 + (1 - F_1) \phi_2 \quad (3.13)$$

The modelling constants are:

$$\begin{aligned} \sigma_{k1} &= 0.85 & \sigma_{\omega 1} &= 0.500 & \beta_1 &= 0.0750 \\ \sigma_{k2} &= 1.00 & \sigma_{\omega 2} &= 0.856 & \beta_2 &= 0.0828 \\ \beta^* &= 0.09 & \kappa &= 0.41 & a_1 &= 0.31 \end{aligned}$$

And:

$$\gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1} \kappa^2}{\sqrt{\beta^*}} \quad \text{and} \quad \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2} \kappa^2}{\sqrt{\beta^*}}$$

3.1.2. Computational Setup

In order to solve the differential equations described above, that is the conservation of mass, conservation of momentum, the k -equation and ω -equation, the computational domain should be defined. This includes defining the boundaries, the flow conditions and the discretization of the flow domain. The setup described in this subsection applies to all RANS simulations performed in this work, that is the baseline RANS, the k -corrective-frozen approach and the (model) propagation.

As described in Chapter 1 the flow problem of interest is the wing-body junction flow. This flow is characterised by a wing connected perpendicular to a flat plate. below in Figure 3.1 a sketch of the flow domain is given, the flat plate is visualised by the grey area at $y = 0$. Note that in Figure 3.1 the coordinates are non-dimensionalised by the maximum wing thickness.

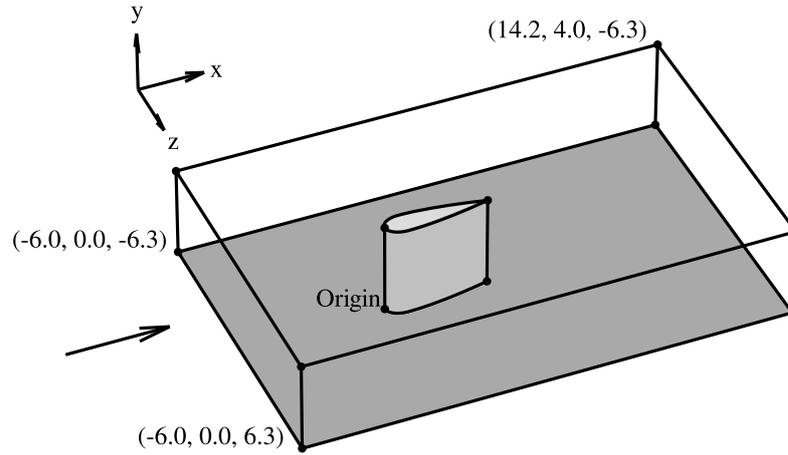


Figure 3.1: Sketch of the computational domain, coordinates ' (x, y, z) ' are given in maximum wing thicknesses.

The wing used for the numerical setup is the Rood wing, as used by Alberts and Vigner [23, 24]. Note that the wing is at zero angle of attack without any twist or taper. The flow and geometry characteristics as well as the airfoil profile are displayed in Table 3.1 and Figure 3.2 respectively.

Table 3.1: Flow and geometry characteristics.

Characteristic	Symbol	Value
Chord length	c	0.3050 [m]
Maximum wing thickness	T	0.0717 [m]
Bluntness factor	BF	0.32 [-]
Free-stream velocity	U_∞	27.0 [m/s]
Angle of attack	α	0.0 [deg]
Inlet boundary layer thickness	δ_{inlet}	0.0316 [m]
Kinematic viscosity	ν	$1.41 \cdot 10^{-5}$ [m ² /s]
Chord length Reynolds number	Re_c	$5.84 \cdot 10^5$ [-]
Momentum deficit factor	MDF	$8.1 \cdot 10^8$ [-]

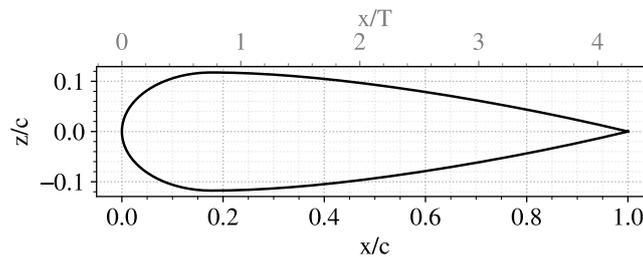


Figure 3.2: Rood wing airfoil profile.

The type of boundary conditions used in this work to solve the RANS equations numerically are displayed in Table 3.2. The inlet patch is located at the most upstream boundary in the yz -plane (at $x = -6.0$ in Figure 3.1) and is split into two patches parallel to the plate, *inlet top* and *inlet bottom*. The generation of the boundary layer (BL) profile specified at *inlet bottom* will be discussed below. The *outlet* patch is the most downstream boundary in the yz -plane.

The *bottom* patch represents the flat plate surface, and the *wing* patch the surface of the wing. Finally, the *top* patch represents the boundary opposite of the flat plate and the *sides* the side boundaries parallel to the flow.

Table 3.2: Boundary conditions of the computational domain.

Patch	Velocity boundary condition	Pressure boundary condition
Inlet bottom	Fixed value ($U = \text{BL-profile}$)	Zero-gradient
Inlet top	Fixed value ($U = 27.0$)	Zero-gradient
Outlet	Zero-gradient	Fixed value ($p = 0$)
Wing	No-slip wall	Zero-gradient
Bottom	No-slip wall	Zero-gradient
Top	Symmetry	Symmetry
Sides	Symmetry	Symmetry

Next to defining the boundary conditions, is to discretise the flow domain into finite volumes, i.e. mesh generation. The mesh used for all RANS simulations in this work is displayed in Figure 3.3. The mesh is an adaptation of the C-grid hexahedron RANS mesh used by Alberts [23]. The first modification is a refinement towards the inlet, see bottom right in Figure 3.3. This was done to increase the numerical stability of ω . The instability was caused since the inlet boundary had a relatively low value for ω prescribed, while the bottom (wall) boundary had a high value. The refined mesh at the inlet presumably allows for a smoother interaction between the two boundaries which removed the instability. Secondly, the cells close to the walls (bottom and wing) have been refined to achieve $y^+ \approx 1$. From testing, it was found that a $y^+ = 1$ mesh was required to resolve the smaller vortex structures in the corner region (for more details please refer to Appendix B). Finally, the mesh has been coarsened towards the top and outlet to reduce the computational cost. The total number of cells is equal to 16.4m which is similar to the wing-body RANS simulation by Gand et al. [7] (12m cells).

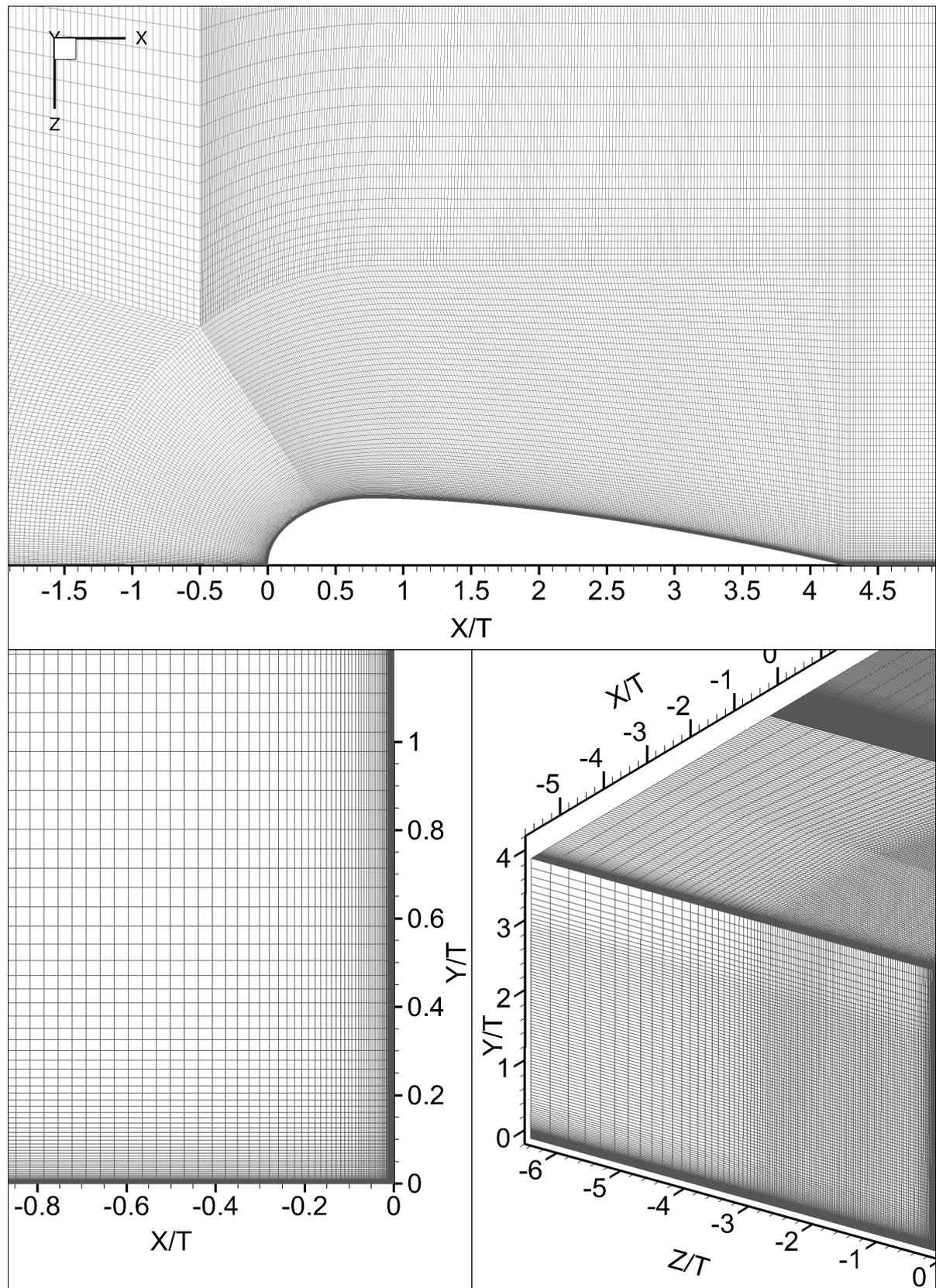


Figure 3.3: Illustrations of the computational mesh, (top) the mesh seen from above, (bottom left) mesh in the symmetry plane ($Z/T = 0$) at the LE of the wing, (bottom right) inlet section of the domain.

To generate the steady inlet boundary layer profile, which is applied as a fixed boundary condition on the *inlet bottom* patch (Table 3.2), the approach used by Alberts [23] is followed. In this approach the BL-profile is approximated using the Reichardt profile (Reichardt [49]), which is given by:

$$u^+ = \frac{1}{\kappa} \ln(1 + \kappa y^+) + 7.8 \left[1 - \exp\left(\frac{-y^+}{11}\right) - \frac{y^+}{11} \exp\left(\frac{-y^+}{3}\right) \right] \quad (3.14)$$

For which the non-dimensional velocity u^+ and non-dimensional wall distance y^+ are given by:

$$u^+ = \frac{u}{u_\tau} \quad \text{and} \quad y^+ = \frac{y u_\tau}{\nu} \quad (3.15)$$

Note that κ equals the von Kármán constant, u_τ the friction velocity and ν the kinematic viscosity. Hence, in order to generate a correct turbulent boundary layer the value of u_τ has to be determined. This is done using an optimisation procedure. First, it should be noted that u_τ is a direct function of the friction coefficient (C_f) since:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad \text{with} \quad \tau_w = C_f \frac{1}{2} \rho U_\infty^2 \quad \text{and} \quad \rho = \text{const.} \quad (3.16)$$

Thus a value of C_f shall be found such that the fixed inlet boundary layer thickness (δ_{inlet}) of 0.0316 [m] determined by Alberts [23] (according to the measurement by Devenport and Simpson [10]) is achieved. Using the Python³ optimisation library from SciPy [50], a function that determines the difference between $0.99U_\infty$ and u according to the Reichardt profile at the mesh cell closest to $y = \delta_{inlet}$ was minimised by varying C_f . This means that a value of C_f is found such that:

$$\text{abs}(0.99U_\infty - u_{Reichardt}(C_f, y = \delta_{inlet})) \approx 0 \quad (3.17)$$

Then accordingly, the found value of C_f is used to generate the entire profile for $y \leq \delta_{inlet}$ using Equation 3.14, for $y > \delta_{inlet}$ the velocity is set to the free stream velocity ($u = U_\infty$). The resulting velocity profile is displayed in Figure 3.4. It is important to match the δ_{inlet} to the LES by Alberts since this LES is used as training data in the SpaRTA approach and it has been shown in Chapter 2 that the BL-profile influences the behaviour of the junction flow significantly.

3.1.3. Fluid Solver

To solve the RANS equations on the discretized mesh the fluid solver OpenFOAM 7 [51] has been utilised. For all simulations reported in this work the finite volume solver simpleFoam has been used to find the flow solution. This solver is based on the SIMPLE algorithm [52] in which the solution of the momentum equation is iteratively corrected using the solution of the pressure correction equation. The latter correction equation enforces the conservation of mass on the momentum equation.

In order to enhance stability and therefore convergence, under-relaxation factors were used to control the magnitude at which the variables update from one iteration to the other. For U_i this factor was set to 0.7, P used a factor of 0.5 and the turbulence quantities ω and k used a value of 0.6.

Next to that, the P correction equation was solved using the OpenFOAM GAMG solver. The equations for U_i , k and ω were solved using the OpenFOAM smoothSolver using the symmetric Gauss Seidel smoother. Note that in this work no sensitivity study is performed for the different available linear system solvers in OpenFOAM since it is outside the scope of this

¹<https://www.python.org/>, Last accessed: 31-05-2023

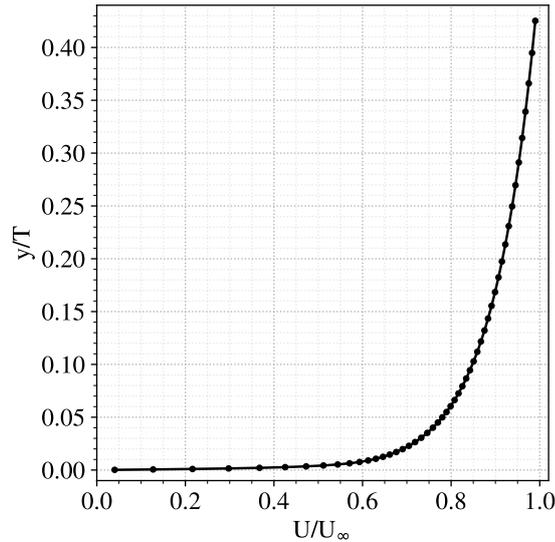


Figure 3.4: The fixed velocity profile used as boundary condition on the patch *Inlet bottom*, note $\delta_{inlet}/T \approx 0.44$.

research. The solvers were set identically to the junction-flow RANS simulation performed by Alberts [23].

To interpolate the flow values from the cell centres to the cell faces, as required for the finite volume method using Gaussian integration, second-order linear schemes were used. To be specific: the gradient terms used the Gauss linear scheme, the divergence terms for k and ω the bounded Gauss limitedLinear scheme for U the bounded Gauss linearUpwind scheme and for the Reynolds stress the Gauss linear corrected scheme was applied.

Since the computational domain is too big to run on a conventional personal computer the simulations were performed on a high-performance cluster (HPC). For this work, the simulations were run on the HPC HPC12 provided by the TU Delft. Within this cluster, two types of processors were utilised: the Intel(R) Xeon(R) Gold 6130 ² and the Intel(R) Xeon(R) E5-2640v4 ³. Although dependent on the availability, approximately 200 cores were used for each simulation. As a rough indication, since the time depended on the type of simulation and the type of CPU, the simulation duration was 3 to 6 hours which comes down to 600 to 1200 core-hours.

3.2. SpaRTA Framework for the Junction Flow

As briefly described in the literature study, Chapter 2, the goal of the SpaRTA method (Schmelzer et al. [39]) is to improve the RANS approach by utilising high-fidelity data, in this work LES data (by Alberts [23]). This is done using the steps described in the flowchart displayed below in Figure 3.5. Note that this flow chart is aimed towards the junction flow in this work, but can be generalised for any type of flow.

In detail, the SpaRTA method can be described by the following steps. First, a baseline RANS simulation is performed and the LES data is interpolated onto the RANS mesh. With these two starting points the k -corrective frozen approach can be executed. Note that the P

²<https://www.intel.com/content/www/us/en/products/sku/120492/intel-xeon-gold-6130-processor-22m-cache-2-10-ghz/specifications.html>, Last accessed: 27-06-2023

³<https://www.intel.com/content/www/us/en/products/sku/92984/intel-xeon-processor-e52640-v4-25m-cache-2-40-ghz/specifications.html>, Last accessed: 27-06-2023

and ω fields from the baseline simulation are used for the initial condition of the frozen simulation to improve the convergence of the solution. The frozen method generates a correction field for the anisotropy of the Reynolds stress tensor called b_{ij}^{Δ} and a correction field for the turbulent kinetic energy equation called R . These two correction fields describe the discrepancy in the turbulence prediction between the baseline RANS and the LES. The equations used to find these correction fields are described in subsection 3.2.1.

The outputted correction fields are then used as the machine learning target for the two to-be-regressed models ($M_{b_{ij}^{\Delta}}$, M_R). The (flow) features that are used for the sparse regression are generated using the propagated frozen fields. The latter denotes that the exact correction fields found by the frozen method are propagated in the RANS solver to obtain the theoretically highest achievable result with respect to flow prediction. This is because the to-be-regressed models are presumably not able to perfectly match the targets. The propagation process is described in subsection 3.2.2 and the regression as well as the generation of the regression features are described in subsection 3.2.3 and subsection 3.2.4 respectively. Finally, the regressed models are propagated in the RANS solver as described in subsection 3.2.2 and evaluated against the LES and or (frozen field) propagation results.

As can be observed in the flow chart for both the propagation of the frozen fields and the model propagation the k and ω fields from the frozen approach are used for the initial condition of the simulations. This was done in order to achieve satisfactory convergence of the solution. The impact of this decision will be discussed later in Chapter 4.

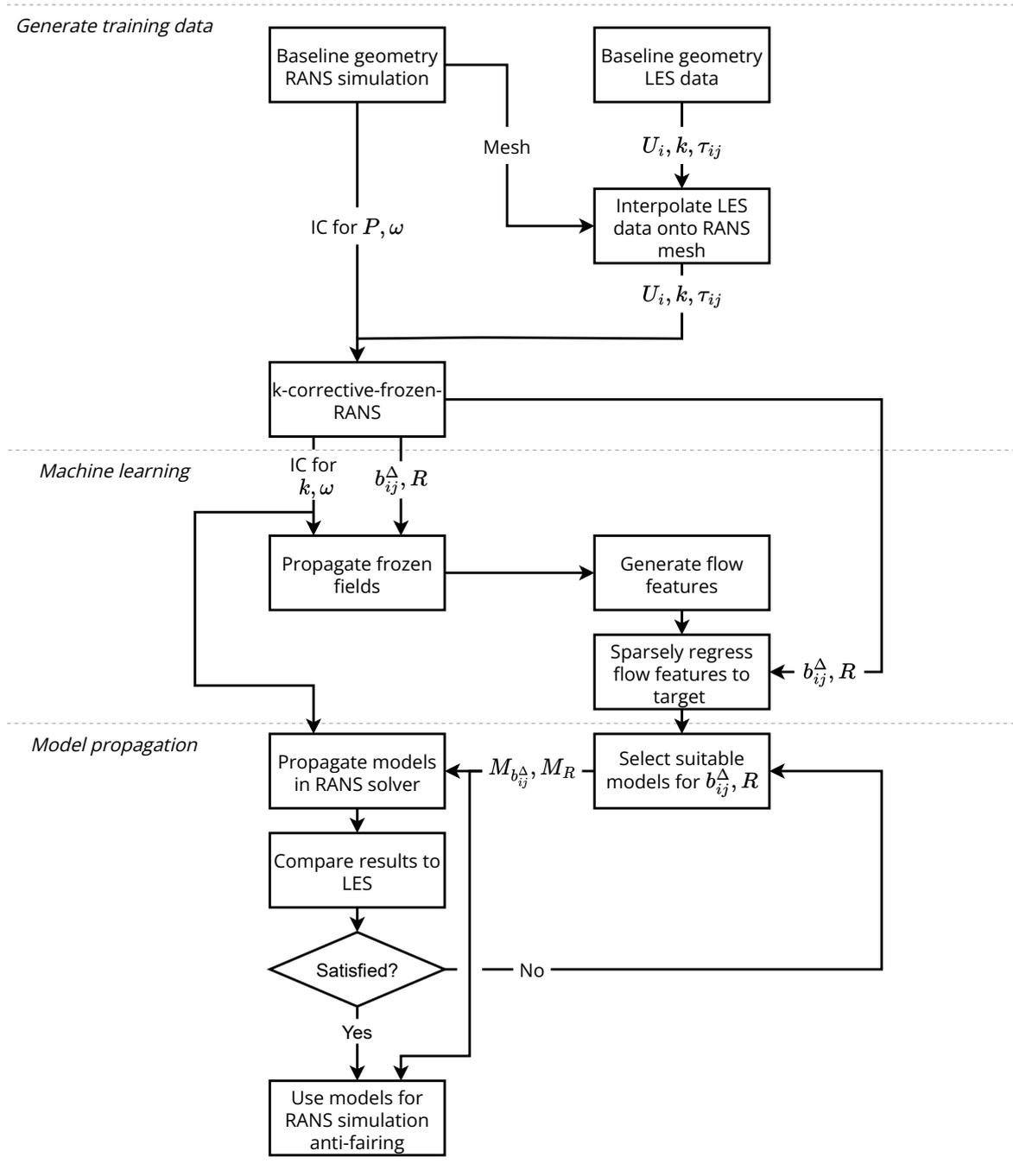


Figure 3.5: Flowchart of SpARTA for the junction flow.

3.2.1. k-Corrective-Frozen

In the k -corrective-frozen approach the correction fields b_{ij}^Δ and R are determined. These correction fields describe the discrepancies of the RANS turbulence quantities with respect to the LES values. The first step in this approach is to 'freeze' the following three variables which are supplied by the LES solution:

$$U_i = u_{i_{LES}} \quad \text{and} \quad k = k_{LES} \quad \text{and} \quad b_{ij} = \frac{1}{2k} \tau_{ij_{LES}} - \frac{1}{3} \delta_{ij} \quad (3.18)$$

Note that the above notation is only used in this subsection. With the frozen k and b_{ij} the value of b_{ij}^Δ can be determined using Equation 3.19. In this equation $\overline{b_{ij}}$ equals the anisotropic part of the Reynolds stress calculated using the Boussinesq closure.

$$b_{ij}^\Delta = b_{ij} - \overline{b_{ij}} \quad \text{with} \quad \overline{b_{ij}} = -\frac{\nu_t}{k} S_{ij} \quad (3.19)$$

However, to determine $\overline{b_{ij}}$ the value of ν_t should be evaluated and thus the ω equation has to be solved. But before doing so it is important to note that the b_{ij}^Δ field does not only correct the Reynolds stress tensor but also the turbulence production term in the transport equations for k and ω . The new production term is defined as:

$$\tilde{P}_k = \max \left[0, \min \left(-2k (\overline{b_{ij}} + b_{ij}^\Delta) \frac{\partial U_i}{\partial x_j}, 10\beta^* \omega k \right) \right] \quad (3.20)$$

In comparison to the standard SpaRTA method, an extra limit has been applied, that is $\tilde{P}_k \geq 0$. This limit has been used to achieve numerical stability of the solver. With the new production term defined, the transport equation for ω is solved to determine the value of ν_t .

$$\frac{\partial \omega}{\partial t} + \frac{\partial U_j \omega}{\partial x_j} = \frac{\gamma}{\nu_t} \tilde{P}_k - \beta \omega^2 + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_\omega \nu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \frac{\sigma_\omega}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \quad (3.21)$$

Finally, since k is frozen and ω and thus ν_t is known, the correction field for k -equation, that is R , can be determined via the explicit equation:

$$R = \frac{\partial k}{\partial t} + \frac{\partial U_j k}{\partial x_j} - \tilde{P}_k + \beta^* k \omega - \frac{\partial}{\partial x_j} \left[(\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right] \quad (3.22)$$

In short, R describes the residual of the k -equation when substituting the LES data. The original k - ω SST definitions for ν_t , the model constants and blending functions are used in the transport equations displayed above. Next to that, solving the ω equation is an iterative process since ν_t is dependent on ω and ω on ν_t . Therefore, in each iteration, the value of b_{ij}^Δ and R is updated until convergence is achieved.

3.2.2. Propagation

In the (model) propagation the correction fields, or models that determine the correction fields, are propagated in the RANS CFD solver. Hereafter, the term propagation will refer to the propagation of the exact correction fields (as calculated by the frozen solver) and the term model propagation will refer to the propagation of the found models by the regression algorithm. In the (model) propagation the equations for the Reynolds stress, ω and k are altered to incorporate the correction terms.

First of all, the Reynolds stress tensor is modified such that the anisotropy correction b_{ij}^Δ is included, that is (note $\partial_k U_k = 0$):

$$\overline{u'_i u'_j} = 2k \left(-\frac{\nu_t}{k} S_{ij} + b_{ij}^\Delta + \frac{1}{3} \delta_{ij} \right) \quad (3.23)$$

Secondly, the ω equation is modified to the version displayed by Equation 3.21, including the newly defined production term \tilde{P}_k . Finally, the k -equation is modified to include the correction field R and the new \tilde{P}_k :

$$\frac{\partial k}{\partial t} + \frac{\partial U_j k}{\partial x_j} = \tilde{P}_k + R - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right] \quad (3.24)$$

All other equations and model constants of the baseline RANS with $k-\omega$ SST model are unchanged. Note that compared to the original SpaRTA equations published by Schmelzer et al. [39] the ω equation is slightly altered. Namely, the correction field R is not included in the ω equation to enhance numerical stability. A discussion regarding this change will be given in Chapter 4.

Besides the updated equations to accommodate the correction fields, an extra numerical trick in the (model) propagation was used to increase stability. Namely, spatial blending functions for the corrections fields were used. These functions return a factor between $[0, 1]$ depending on the spatial location in the domain. This factor is multiplied with b_{ij}^Δ and R in each iteration before being substituted into the equations. Two spatial blending functions were used, one that wraps around the area close to the wing and one for the far-field.

The blending function that wraps the wing is displayed in Figure 3.6. As can be seen in Figure 3.6a, the area where the corner vortex and (smaller) secondary vortices exist is not blended since the blending starts at $y/T \approx 0.06$ (as will be shown in Chapter 4). Above this y level a blending factor linearly dependent on wall normal distance is applied (see Figure 3.6b). Note that the blending function extends all the way up to the top of the computational domain. The spatial blending around the wing is used to filter out a laminar separation bubble, this is discussed in detail in Chapter 4.

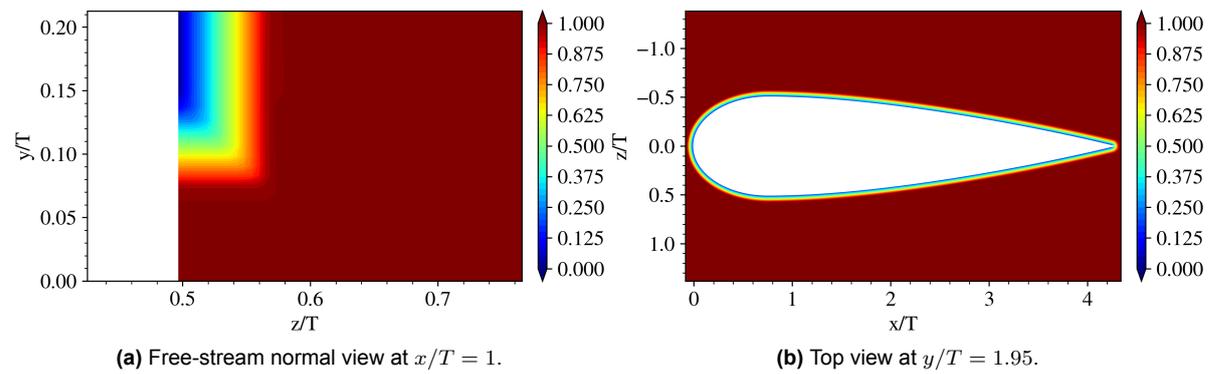


Figure 3.6: Spatial blending factor applied close to the airfoil.

Next to the blending applied close the wing, is the spatial blending applied in the far-field. This blending is displayed in Figure 3.7. As can be seen correction fields at the top and outlet of the computational domain are linearly blended to zero. This is again performed to enhance the numerical stability of the solver.

3.2.3. Sparse Regression

In order to generate the models that determine the turbulence corrections, sparse regression is utilized. The latter means that a set of (flow) features are sparsely regressed to the correction fields. Resulting in two separate explicit functions for the tensor field b_{ij}^Δ and the scalar field R . The features that are used for this regression are described in subsection 3.2.4, for the description in this subsection it is assumed that the features exist and are discretized over the RANS mesh resulting in a feature matrix C_Δ of shape $(n_{cells}, n_{basisfunctions})$. Note that $\Delta = b_{ij}^\Delta$ or R .

As described by Schmelzer et al. [39], the regression occurs in two steps. First elastic net regression is applied to the linear problem:

$$\Delta = C_\Delta \Theta \quad (3.25)$$

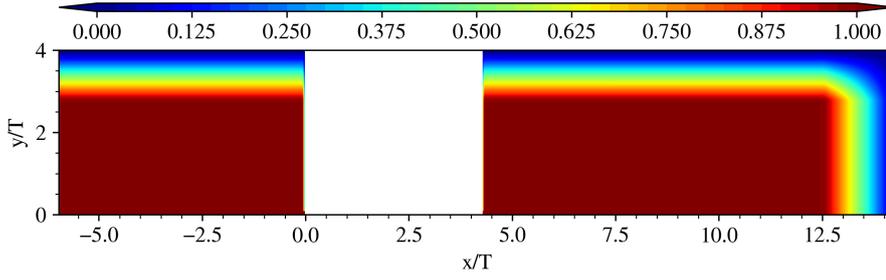


Figure 3.7: Spatial blending factor applied in the far-field of the domain, $z/T = 0$.

In which Δ equals the target (b_{ij}^{Δ} or R), C_{Δ} the feature matrix and Θ the to-be-found set of coefficients. In elastic net regression, the linear system is solved according to the following optimisation formulation ([39]):

$$\Theta = \arg \min_{\hat{\Theta}} \left(\left\| C_{\Delta} \hat{\Theta} - \Delta \right\|_2^2 + \lambda \rho \left\| \hat{\Theta} \right\|_1 + 0.5 \lambda (1 - \rho) \left\| \hat{\Theta} \right\|_2^2 \right) \quad (3.26)$$

In which λ equals the regularisation weight and ρ the mixing parameter valued between 0 and 1. This mixing parameter controls the blend between LASSO regression and ridge regression. Using only the l_1 norm, defined by the subscript 1 in Equation 3.26, will result in LASSO regression which promotes sparsity of the solution. Yet using only the l_2 norm will result in ridge regression which promotes small coefficients (Schmelzer et al. [39]). And thus using the combination, by controlling the value of λ and ρ , will allow for a sparse solution with relatively small coefficients. The goal of the elastic net regression is to find the sparse set of basis functions in C_{Δ} that describe the target to a good extent, not necessarily to find the correct coefficients. Hence in the original paper, it is referred to as model selection. Besides, since the optimal set of λ and ρ is unknown a brute force technique is used in this work that tests all combinations:

$$\begin{aligned} \lambda &= [0.01, 0.1, 1.0, 10.0, 100.0] \\ \rho &= [0.01, 0.1, 0.2, 0.5, 0.7, 0.9, 0.95, 0.99, 0.999, 1.0] \end{aligned}$$

By doing this a set of multiple candidate models consisting of only a few basis functions is found, each for a different combination of λ and ρ .

The second step in the regression is the model inference, in which the optimal set of coefficients is found for the set of sparse candidate models. For this step ridge regression is used since it promotes small coefficients. The optimisation problem for each individual model candidate is formulated as follows ([39]):

$$\Theta^i = \arg \min_{\hat{\Theta}^i} \left(\left\| C_{\Delta}^i \hat{\Theta}^i - \Delta \right\|_2^2 + \lambda_r \left\| \hat{\Theta}^i \right\|_2^2 \right) \quad (3.27)$$

In which superscript i denotes the candidate model, and thus the submatrix and subvector of C_{Δ} and Θ that are non-zero according to the model selection step. Besides, λ_r equals the regularisation parameter. Similar to the elastic net regression several values of λ_r were tried, namely:

$$\lambda_r = [0.001, 0.01, 0.1, 1.0, 10.0]$$

Then finally a set of correction models for b_{ij}^Δ ($M_{b_{ij}^\Delta}$) and R (M_R) is formed by multiplying the non-zero coefficients in Θ^i with the corresponding basis functions.

To perform the regression in this work the Python library Scikit-learn [53] is used together with a code under development by the research group of Dr. R.P. Dwight. The latter is mainly meant to automate the machine learning, generate the regression library and allow for smooth interaction with the OpenFOAM fluid solver. Due to the large computational domain, the TU Delft provided HPC 12 computing cluster was used to perform the machine learning, which enables the use of 128 gigabytes of memory for the learning process. Yet, especially for b_{ij}^Δ , reducing the cells used for the regression is required to meet the memory limit. To reduce the number of cells thinning and boxing were used.

Thinning and Boxing

As described above two methods were applied to reduce the number of cells used for the regression to reduce the memory requirement. This is because reducing the number of cells will reduce the number of rows of C_Δ and thus the memory required to store it.

The thinning process is rather simple. In this process, a random number of cells from the entire domain is picked. To control the number of random cells to take, a thinning factor between $[0.0, 1.0]$ is user defined which when multiplied by the total number of cells results in the number of cells to randomly choose. For both b_{ij}^Δ and R a thinning factor of 0.1 is used.

Besides thinning, boxing is applied as well. In the boxing process, only cells are used for the regression that are inside a user-defined rectangular box within the domain. This allows the regression to purely focus on a part of the flow. For the junction flow a large portion in the free-stream region is not of interest for the correction models. Hence defining the box to surround the interesting region allows to reduce the number of cells used for the regression. For both R and b_{ij}^Δ a box is used that extends in x/T from $[-1.4, 1.4]$ and in z/T from $[-2.1, 2.1]$. For R the box extends in y/T from $[0.0, 1.4]$, while for b_{ij}^Δ it extends from $[0.0, 0.7]$, see Figure 3.8. These boxes were determined using a trial and error approach, however, it was kept in mind to include a portion of free-stream area within the box to 'show' the regression areas with low to no corrections.

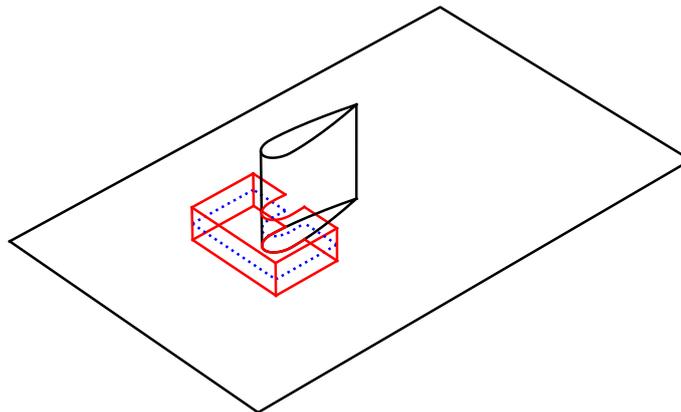


Figure 3.8: Drawing of the box used for reducing the data, figure is up to scale, (red) box used for R , (blue) box used for b_{ij}^Δ .

Culling

In order to decide which models go into the refit step, i.e. the ridge regression, a culling step is introduced which culls the models from the elastic net regression that are not of interest. For all for models that have non-zero coefficients the following two criteria must hold before

continuing into the refitting step:

- The R^2 -score must be above a certain user-defined value.
- The complexity of the model, that is the number of non-zeros in the Θ -vector, must be below a user-defined value.

The R^2 -score is computed as follows (Rawlings et al. [54]):

$$R^2\text{-score} = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (3.28)$$

in which y_i equals the target data, \bar{y} the mean of the target data and f_i the predicted value by the model. This score gives an indication of how well the model fits the target data, if the R^2 -score equals 1.0 then the model predicts the target perfectly.

Besides ranking models on the R^2 -score, the mean squared error (MSE) is also used as a performance indicator. The MSE of a model is computed as follows:

$$MSE = \frac{1}{n} \sum_i (y_i - f_i)^2 \quad (3.29)$$

in which n equals the length of the vector y_i , i.e. the number of data points.

3.2.4. Feature Library and Selection

In order to perform the regression described above in subsection 3.2.3 it is important to establish the feature matrix C_Δ . For both b_{ij}^Δ and R , Pope's tensor basis, as displayed in Section 2.3, will be used as a starting point. This means that, especially for b_{ij}^Δ since it impacts the Reynolds stress tensor directly, the underlying assumption of the near homogeneous rate of strain should be kept in mind.

Features for t.k.e. Residual

Since R is a scalar field, a set of scalar features is required. Hence as a primary set of features the tensors defined by Pope [34], see Equation 2.9, are multiplied by the velocity gradient tensor ∇U to generate the scalar features:

$$G^{(\lambda)} = 2kT_{ij}^{(\lambda)} \frac{\partial U_i}{\partial x_j} \quad (3.30)$$

This is similar to how the production term in the turbulent transport equations is determined, see Equation 3.20. Now to evaluate $T_{ij}^{(\lambda)}$ the strain rate and rotation rate tensors should be non-dimensionalised using a time-scale τ . In this work, two time scales are used. The first time scale is the conventional turbulent time scale $1/\omega$, also used by [38, 39]. And the second time scale is the mean flow time scale $1/\|\nabla U\|$ as for example used in the QCR model to non-dimensionalise O_{ik} (Equation 2.4). The subscript s is used to indicate that the mean flow time scale has been applied instead of the conventional turbulent time scale. Finally, the turbulence dissipation rate (ϵ) is added to the set of $G^{(\lambda)}$ functions to generate the list of primary features:

$$\text{Primary features} = \left[\epsilon, G^{(1)}, G_s^{(1)}, G^{(2)}, G_s^{(2)}, G^{(3)}, G_s^{(3)}, \dots, G^{(10)}, G_s^{(10)} \right] \quad (3.31)$$

To not solely rely on the primary features for the regression, the primary set is accompanied by a set of non-dimensional secondary features. This secondary set is based on several flow features and the tensor invariants corresponding to Pope's tensor basis, see Equation 2.10. These invariants were again computed with the two different time scales. The non-dimensional

flow features q_i are displayed below in Table 3.3. These flow features were previously defined in Steiner et al. [45] and Vigner [24].

Although the list displayed in Table 3.3 can be very useful to activate models in regions where the invariants defined by Pope do not activate one must be careful with the use of them. First of all, q_{TI} is not Galilean-invariant, which means it should preferably be avoided in a turbulence model. It is kept in the list since for design optimisation the Galilean-invariance argument is less of a problem as from iteration to iteration the reference frame will remain unchanged. However, it is tried to be avoided if possible. Secondly, as mentioned by Spalart [55], the use of the gradient of P should also be avoided in any turbulence model. Therefore, if possible q_{pS} and $q_{dp/ds}$ are omitted.

Table 3.3: List of non-dimensionalized q_i features.

Expression	Description
$q_{TI} = k / (\frac{1}{2}U_i^2)$	Turbulence intensity
$q_Q = \frac{1}{2} (\ \Omega_{ij}\ ^2 - \ S_{ij}\ ^2) / \ S_{ij}\ ^2$	Q-criterion
$q_T = \frac{1}{\omega} / \ S_{ij}\ $	Ratio of turbulent to mean time scale
$q_\nu = \nu_t / (100\nu)$	Ratio of turbulent to laminar viscosity
$q_{\tau kB} = \ \tau_{ij, Boussinesq}\ / k$	Ratio of total to normal Boussinesq Reynolds stresses
$q_\perp = \left U_i U_j \frac{\partial U_i}{\partial x_j} \right / \sqrt{U_n^2 U_i \frac{\partial U_i}{\partial x_j} U_m \frac{\partial U_m}{\partial x_j}}$	Deviation from parallel shear flow
$q_\gamma = \left\ \frac{\partial U_i}{\partial x_j} \right\ / \omega$	Shear parameter
$q_V = \sqrt{\omega_j \frac{\partial U_i}{\partial x_j} \omega_k \frac{\partial U_i}{\partial x_k}} / \ S_{ij}\ ^2$	Vortex stretching
$q_{Re} = \frac{\sqrt{k}d}{50\nu}$	Wall-distance based Reynolds number
$q_{pS} = \sqrt{\frac{\partial P}{\partial x_i} \frac{\partial P}{\partial x_i}} / \left U_i \frac{\partial U_i}{\partial x_i} \right $	Pressure to shear ratio
$q_{dp/ds} = U_k \frac{\partial P}{\partial x_k} / \sqrt{\frac{\partial P}{\partial x_j} \frac{\partial P}{\partial x_j} U_i U_i}$	Pressure gradient along a streamline

Finally, a set of tensor invariants based on the ∇k were added as defined by Wu et al. [56], the introduction of these invariants into SpaRTA is published by Steiner et al. [45]. The invariants are given by Equation 3.32 and Table 3.4, note that ϵ equals the dissipation rate and I the identity tensor. For Table 3.4, similar to Equation 2.10, $\{\cdot\}$ equals the trace and the bold symbols refer to conventional tensor multiplication. Note that \mathbf{S} and Ω are non-dimensionalised using two time scales, similar to the conventional tensor basis, subscript s refers to the mean flow time scale.

$$\mathbf{A}_k = -\frac{\sqrt{k}}{\epsilon} I \times \nabla k \quad (3.32)$$

Table 3.4: List of invariants based on \mathbf{A}_k , \mathbf{S} and Ω .

$I_k = \{\mathbf{A}_k^2\}$	$VIII_k = \{\Omega\Omega\mathbf{A}_k\mathbf{S}\}$
$II_k = \{\mathbf{A}_k\mathbf{A}_k\mathbf{S}\}$	$IX_k = \{\mathbf{A}_k\mathbf{A}_k\Omega\mathbf{S}\}$
$III_k = \{\mathbf{A}_k\mathbf{A}_k\mathbf{S}\mathbf{S}\}$	$X_k = \{\Omega\Omega\mathbf{A}_k\mathbf{S}\mathbf{S}\}$
$IV_k = \{\mathbf{A}_k\mathbf{A}_k\mathbf{S}\mathbf{A}_k\mathbf{S}\mathbf{S}\}$	$XI_k = \{\Omega\Omega\mathbf{S}\mathbf{A}_k\mathbf{S}\mathbf{S}\}$
$V_k = \{\Omega\mathbf{A}_k\}$	$XII_k = \{\mathbf{A}_k\mathbf{A}_k\Omega\mathbf{S}\mathbf{S}\}$
$VI_k = \{\Omega\mathbf{A}_k\mathbf{S}\}$	$XIII_k = \{\mathbf{A}_k\mathbf{A}_k\mathbf{S}\Omega\mathbf{S}\mathbf{S}\}$
$VII_k = \{\Omega\mathbf{A}_k\mathbf{S}\mathbf{S}\}$	

Note that Wu et al. [56] also specifies invariants based on ∇P yet these have not been included since Spalart ([55]) discourages the use of ∇P in turbulence models. This all combined establishes the set of secondary non-dimensional features:

$$\text{Secondary features} = [(q_{TI}, \dots, q_{dp/ds}), (I, I_s, \dots, V, V_s), (I_k, \dots, XIII_k, XIII_{k,s})] \quad (3.33)$$

Besides boxing the flow to reduce the memory requirement of the regression, it is also important to filter out the secondary features that hardly contain any information regarding R . This means that only a subset of the secondary features is actually used for the regression, which reduces the number of columns required for C_Δ and thus the required memory. For the regression of R the concept of mutual information is used.

By computing the mutual information of all secondary features with R a measure is computed that quantifies the amount of information a secondary feature gives about R . And thus the dependency of a secondary feature X with respect to R is discovered. If the feature X is independent of R then the mutual information value equals zero. The formal definition of the mutual information is given by Equation 3.34 (Kraskov et al. [57]).

$$MI(X, Y) = \int_y \int_x p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) dx dy \quad (3.34)$$

In Equation 3.34 $p(x)$ equals the marginal probability density function of random variable X and $p(x, y)$ the joint probability density function of random variables X and Y . However, since the probability density functions of the secondary features and R are unknown the Python library scikit-learn [53] is used which estimates the value of MI based on the method by Kraskov et al. [57].

In Kraskov et al. [57] a k-nearest neighbour method is utilised. First the value $\epsilon(i)$ is determined which is based on the distance of a point i to its kth nearest neighbour, let's say there are two random variables X and Y consisting of N points then the value of $\epsilon(i)$ yields:

$$\epsilon(i)/2 = \max(\|x_i - x'_i\|, \|y_i - y'_i\|) \quad (3.35)$$

in which (x_i, y_i) represent the point i , (x'_i, y'_i) the kth nearest neighbour of i and $\|\cdot\|$ the Euclidean distance. Sequentially the number of points (x_j) within X that satisfy the relation:

$$\|x_j - x_i\| < \epsilon(i)/2 \quad (3.36)$$

is defined as $n_x(i)$, similarly the value of $n_y(i)$ is determined. Finally, the value of $MI(X, Y)$ is determined as:

$$MI(X, Y) = \Psi(k) - \frac{1}{N} \sum_i^N [\Psi(n_x(i) + 1) + \Psi(n_y(i) + 1)] + \Psi(N) \quad (3.37)$$

in which Ψ equals the digamma function. As described by Goderie [58] the assumptions made to derive Equation 3.37 is that the entropy of variable X (and Y) can be determined with N realisations and that the marginal density function $p(x)$ is constant within a radius $\epsilon(i)/2$ from point i . Note that the value of MI can also be expressed using the entropy of X and Y defined by $H()$ (Kraskov et al. [57]):

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \quad (3.38)$$

With Equation 3.37, the value of MI for each secondary feature with R can be computed. Consequently, secondary features with a relatively low MI score can be removed from the set reducing the amount of memory required to store C_Δ . The only decision left is what value to use for k , as described by Kraskov et al. [57] lower values of k will reduce systematic errors in the estimation, while higher values of k reduce statistical errors. Kraskov et al. [57] advise to use a value between 2 and 4. In this work, it was found that for this particular problem the value of k (as will be shown in Chapter 4) was insensitive. Higher values would reduce the magnitude of MI compared to lower values, however, the relative difference between the MI scores of different secondary features would remain fairly constant.

Feature Variables for Anisotropy Correction

The regression features used for b_{ij}^Δ are very similar to the features used for R . For the primary features, instead of using the features $G^{(\lambda)}$, Pope's tensor basis can be used directly. Hence the primary set is defined as follows, again using the two time scales:

$$\text{Primary features} = \left[T_{ij}^{(1)}, T_{ij,s}^{(1)}, T_{ij}^{(2)}, T_{ij,s}^{(2)}, T_{ij}^{(3)}, T_{ij,s}^{(3)}, \dots, T_{ij}^{(10)}, T_{ij,s}^{(10)} \right] \quad (3.39)$$

The set of secondary scalar features is the same as for R , that is the features defined in Equation 3.33. Similar to R only a subset of the secondary features is used to limit the requirement on memory. However, using mutual information on the set of secondary features with respect to b_{ij}^Δ is difficult to define since b_{ij}^Δ is a tensor and the secondary features are scalars. It could be done by flattening b_{ij}^Δ into a vector of scalars, however, this will result in information on the scalar features with respect to each individual component of b_{ij}^Δ rather than the complete tensor. Therefore, it was decided to do a cheap regression with degree 1, which will be explained in subsection 3.2.5, using all secondary features. From this regression, the important secondary features are extracted by observing the best-scoring models. These important features then form the set of secondary features used for the more expensive regressions.

3.2.5. Library Generation

Following the original paper by Schmelzer et al. [39], the library of non-linear functions which ultimately form the columns of C_Δ is generated using the FFX algorithm (Riolo et al. [59]). In this algorithm, the secondary and primary features are combined using (non-)linear functions to generate the basis functions that are regressed to the target data. In essence, the algorithm generates strings that described the basis functions which are evaluated on the training data set (after thinning and boxing). The strings have the following form:

$$f(s_i) \cdot g(s_j) \cdot h(s_k) \cdot \dots \cdot p_i \quad (3.40)$$

In which the $f(), g(), h(), \dots$ describe (non-)linear functions, s_i equals any secondary feature from the predefined set and p_i any primary feature from the predefined set. Note that only one primary feature is used to cohere to the structure of the original general effective viscosity hypothesis (Section 2.3), and to enforce unit consistency. The amount of secondary features

used is dependent on the user-defined library degree setting. If the library degree equals 1 then only univariate function strings (multiplied by a primary feature) are possible:

$$f(s_i) \cdot p_j \quad (3.41)$$

Yet if the degree is set to 2, more complex functional strings with up to 2 secondary features are generated, that is:

$$f(s_i) \cdot g(s_j) \cdot p_k \quad (3.42)$$

Note that, if for example, the degree equals 2 then the library also includes all degree 1 forms. The value of the library degree can be increased up to very complex levels, however, to keep the memory required for the regression within bounds the level has been set to 2 in this work.

For the functions $f()$ the following set has been used in this work:

$$\left[\tanh(x), |x|, \sqrt{|x|}, x^1, x^2 \right]$$

All basis functions are evaluated on the computational cells which result in the matrix C_Δ . Note that the features are scaled using their standard deviation which is determined using the Python library Numpy [60]. If any basis function results in a 'not-a-number', for example, because it tends to infinity, then the basis function is omitted from the matrix.

4

Results and Discussion

This chapter displays and discusses the various results of the various simulations performed in this work. First, in Section 4.1 a comparison is made between the baseline RANS simulation and the LES data to establish the fallacies in the baseline version of RANS. Secondly, in Section 4.2 the results of the k -corrective frozen approach are displayed. This section is followed by Section 4.3 in which the results of the propagation of the correction fields are shown. The sparse regression of the correction fields is discussed in Section 4.4. The propagation of the found models is discussed in Section 4.5. Finally, a discussion on several modifications made to the SpaRTA methodology for the junction flow is given in Section 4.6.

4.1. Baseline RANS Results and LES Data

Before diving into the machine learning it is essential to establish a baseline. Hence in this section the results of the baseline RANS will be displayed. Next to that, these results will be compared to the LES data by Alberts [23]. Note that all LES results shown in this work refer to the time-mean results. Although the LES data has minor discrepancies with respect to experimental data, due to the subgrid-scale model as shown by Ryu et al. [4], it is seen as the ground truth in this work. For the validation of the LES with experimental data please refer to Alberts [23].

4.1.1. Mean Flow Fields

Below in Figure 4.1 the x -velocity profile (U) is given in the symmetry plane at several locations upstream of the wing. In the most left profile ($x/T = -5.9$) it is shown that the boundary layer profile at the inlet of the domain of the RANS simulation matches the LES profile to a good extent. The latter is important as the shape and thickness of the boundary layer influence the formation of the vortices downstream, as described by the literature in Chapter 2. In the right three plots a significant discrepancy between the profiles and separation location is observed, this confirms the lack of prediction accuracy of RANS for the junction flow.

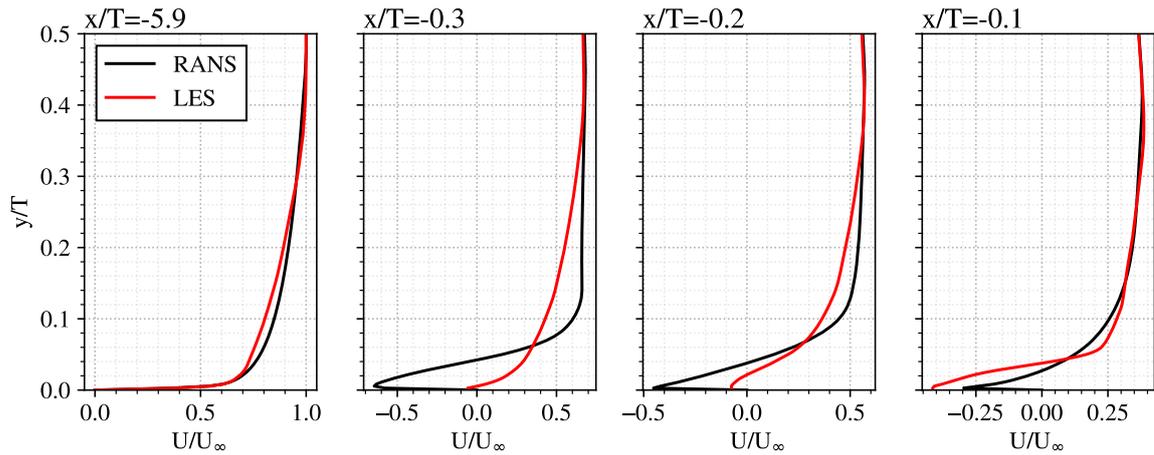


Figure 4.1: RANS and LES boundary layer profiles in the symmetry plane.

A better visualisation of the faulty separation prediction by the RANS is shown in Figure 4.2 in which the velocity magnitude ($|U|$) and velocity direction are shown in the symmetry plane upstream of the wing. The wrong prediction of the separation point causes the HSV to be misplaced by approximately $0.15T$.

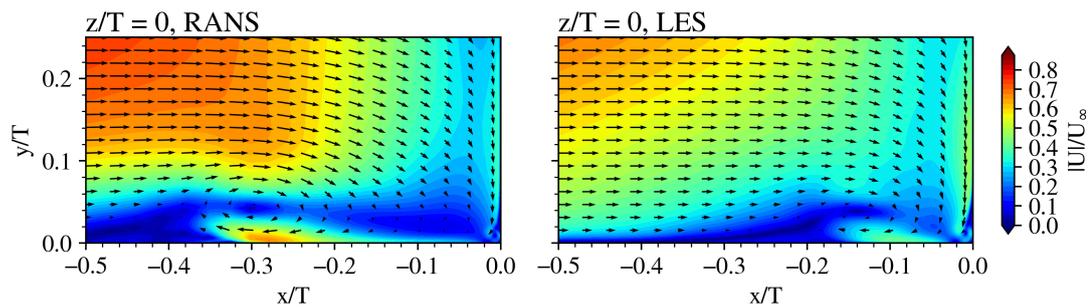


Figure 4.2: RANS and LES velocity magnitude in the symmetry plane upstream of the wing.

The location of the vortex core is easily recognised by observing the turbulent kinetic energy (k) in the symmetry plane, see Figure 4.3. In the LES data, it is noticed that the k profile shows the expected C-shape as described by Elahi et al. [11]. Besides the RANS not having the correct shape and location, it is also under-predicting the magnitude of k . Next to that, a discrepancy is observed in the corner region (near $x/T = 0.0$) where RANS is predicting zero k yet the LES is not.

In Figure 4.4 the vorticity in z -direction (Ω_z) is displayed in the symmetry plane. When comparing the RANS results with the LES data it is clearly visible that not only the location of the HSV is off, the shape is not representative either. The LES clearly shows the elliptical shape resulting from the time average of the bimodal behaviour of the HSV. This shape is not recognised in the RANS results. The latter could be due to the fact that the RANS equations, as described in Chapter 3, search for a steady-state solution without any information regarding the unsteady behaviour of the true flow. This means that although the LES shown in Figure 4.4 obeys the steady state RANS equations, as Alberts [23] showed that the simulation was statistically stationary, the elliptical shape is formed by averaging (over time) an unsteady phenomenon. This unsteady behaviour is unknown to the RANS solver and its closure model

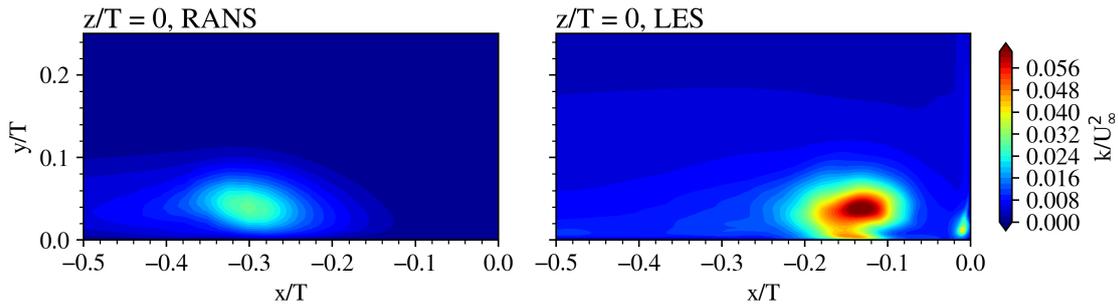


Figure 4.3: RANS and LES turbulent kinetic energy in the symmetry plane upstream of the wing.

which therefore could result in a different solution for the shape of the mean HSV. However, for the goal of this thesis, this does not pose a problem since the goal is not to find the correct instantaneous flow fields nor the correct transient behaviour of the HSV, but rather to find the correct mean field. And thus the goal is to adapt, or correct, this nonphysical steady-state solution such that it is similar to the time average of the true flow.

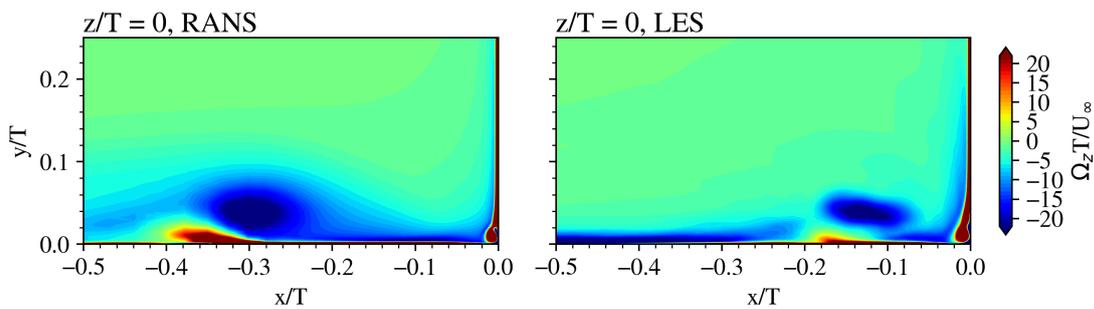


Figure 4.4: RANS and LES vorticity in z -direction in the symmetry plane upstream of the wing.

Although the HSV is dissimilar in the RANS versus LES, the corner vortex in the symmetry plane shows a similar shape and location. The corner vortex is recognised by the circular structure with positive Ω_z in Figure 4.5. The magnitude of the RANS is, however, lower compared the LES. Note that the colour scale has been adapted between Figure 4.4 and Figure 4.5 to clarify the shape of the corner vortex. The change in scale also indicates that the magnitude of the vorticity of the corner vortex core is significantly larger than the HSV.

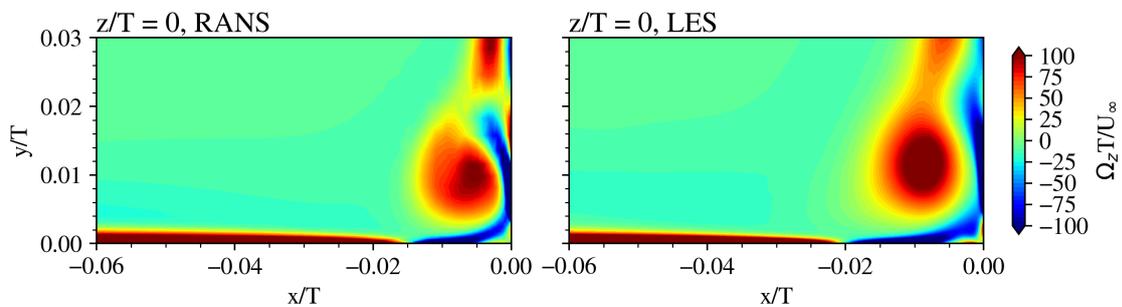


Figure 4.5: RANS and LES closeup of the z -vorticity in the symmetry plane upstream of the wing.

The behaviour of the HSV when convecting and stretching around the leading edge is shown in Figure 4.6. In this figure, the vorticity in x -direction (Ω_x) is shown at several y - z planes adjacent to the wing. A similar behaviour as in the symmetry plane is seen. Namely, the shape and especially the location of the HSV are dissimilar when comparing the RANS to the LES data. Next to that, it is also observed that the dissipation of the HSV in the RANS is weaker. Finally, the HSV in the RANS displaces further away from the wing when traversing downstream. It is plausible that this is the result of a less severe suction force imposed by the wing on the HSV, because the HSV is generated further away from the wing in the RANS.

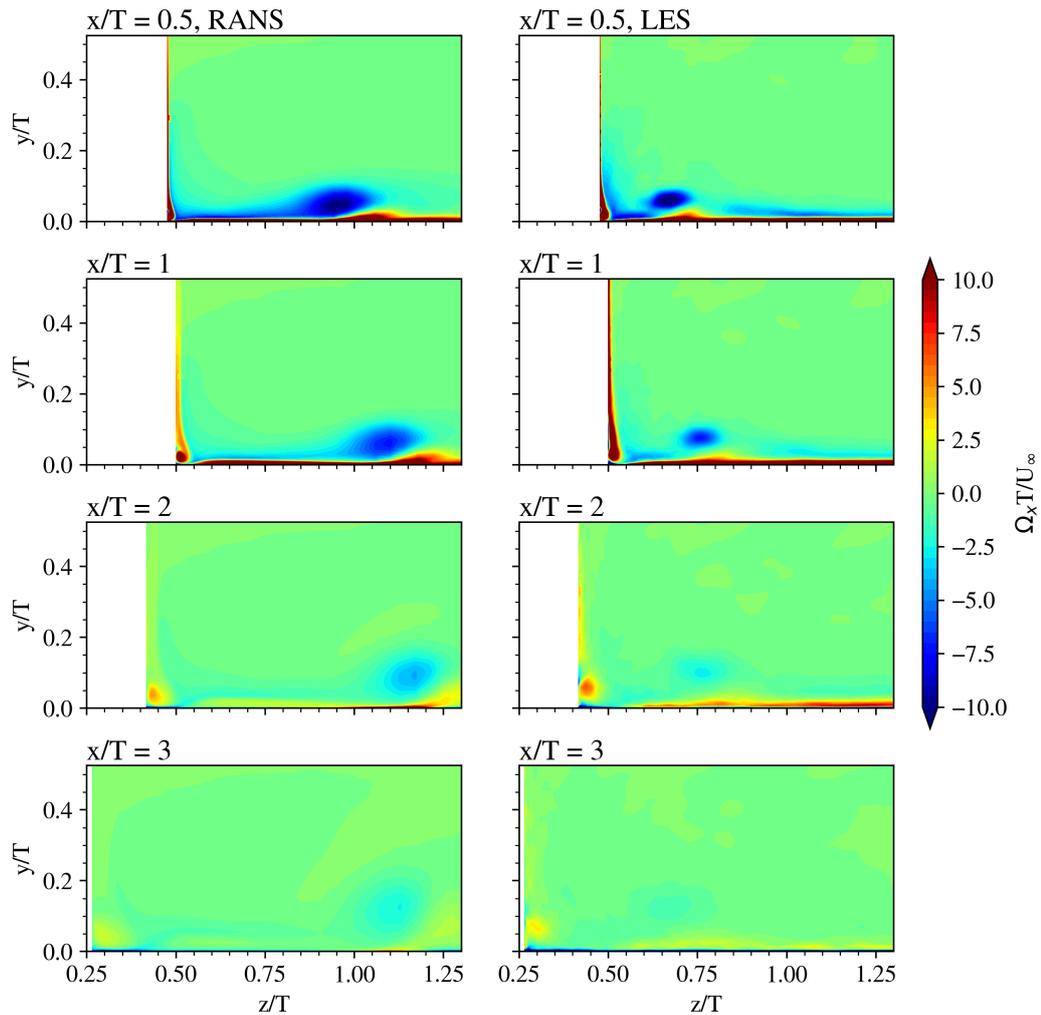


Figure 4.6: RANS (left) and LES (right) vorticity in free-stream direction (x) in planes normal to the flow (y - z planes).

An interesting region to investigate, for both the RANS and LES data, is the corner region in the y - z planes downstream of the leading edge. Figures 4.7, 4.8 and 4.9 show the Ω_x in this region at several planes, note that Figure 4.7 uses a different colour scale to clarify the structures. First of all, in the top row of Figure 4.7 ($x/T = 0.5$) it can be seen that although

the magnitude is slightly off the shape and different structures are similar, this was seen in Figure 4.5 as well. In the second row ($x/T = 1$), however, a negative Ω_x structure at $z/T = 0.50$ is present in the LES which is not predicted by the RANS. This is a good example of RANS not being able to predict these secondary vortex structures (or stress-induced vortices) as these depend on the gradients of the Reynolds stresses as explained in Chapter 2. Since two-equation models such as the $k-\omega$ SST model cannot predict the anisotropy of the Reynolds stress accurately it is expected that RANS with this turbulence model does not correctly predict secondary vortices. Note that the structure should be able to be represented by the RANS mesh since nine computational cells cover the dimension $z/T = [0.50, 0.51]$ and thirteen cells cover $y/T = [0.00, 0.01]$.

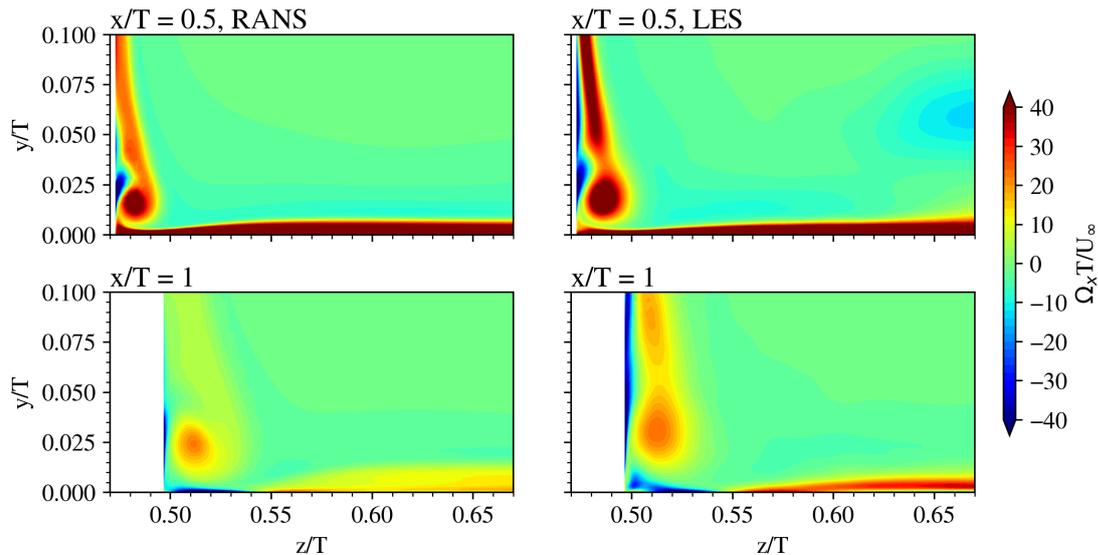


Figure 4.7: RANS (left) and LES (right) closeup of x -vorticity in y - z planes at two different x/T locations.

Another example of a secondary vortex is given in Figure 4.8 and Figure 4.9. Notice how the positive Ω_x structure in the LES data at $z/T = 0.265$ in Figure 4.8 and $z/T = 0.065$ in Figure 4.9 is not predicted by RANS. A possible result of not predicting these structures is the contribution to a corner separation in the RANS flow which will be shown later.

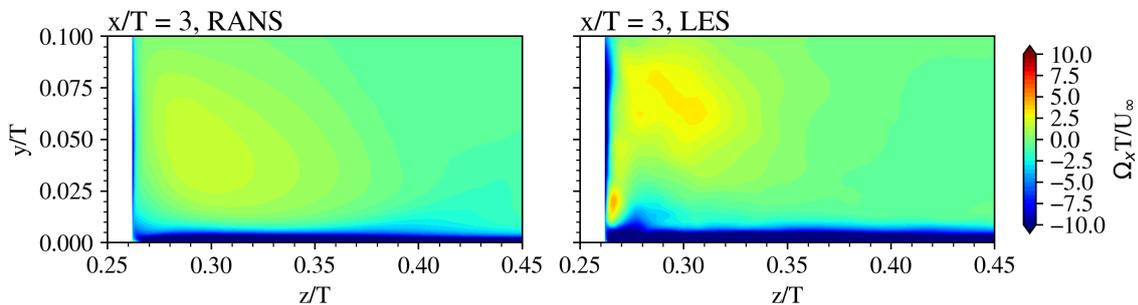


Figure 4.8: RANS and LES closeup of x -vorticity in the y - z plane at $x = 3T$.

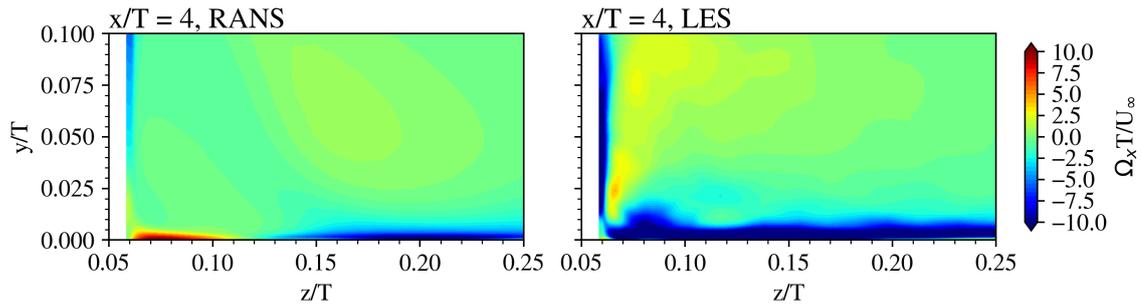


Figure 4.9: RANS and LES closeup of x -vorticity in the y - z plane at $x = 4T$.

In Figure 4.10 the U velocity (velocity in x -direction) is shown at several y - z planes downstream of the leading edge. The main discrepancy between the RANS and LES can be observed in the region just to the right of where the HSV exists. It can be seen that the HSV removes momentum in the free-stream direction to its right and adds momentum to the left (with respect to this side of the wing). This extraction right of the HSV appears to be more extreme in the RANS simulation close to the bottom wall compared to the LES. This can be linked to the larger (area-wise) HSV as seen in Figure 4.6. As a result of this an error in the drag prediction of the bottom wall will be induced since the wall shear stress is dependent on the velocity gradient at the wall (Pope [48]).

Next to the discrepancy caused by the HSV, is the difference seen at the wing wall in the corner region. In which the RANS, especially at $x/T = 3$ seems to predict lower velocities. However, above the bottom wall and corner region, that is $y/T \gtrsim 0.25$, the RANS and LES U -velocity fields match quite well.

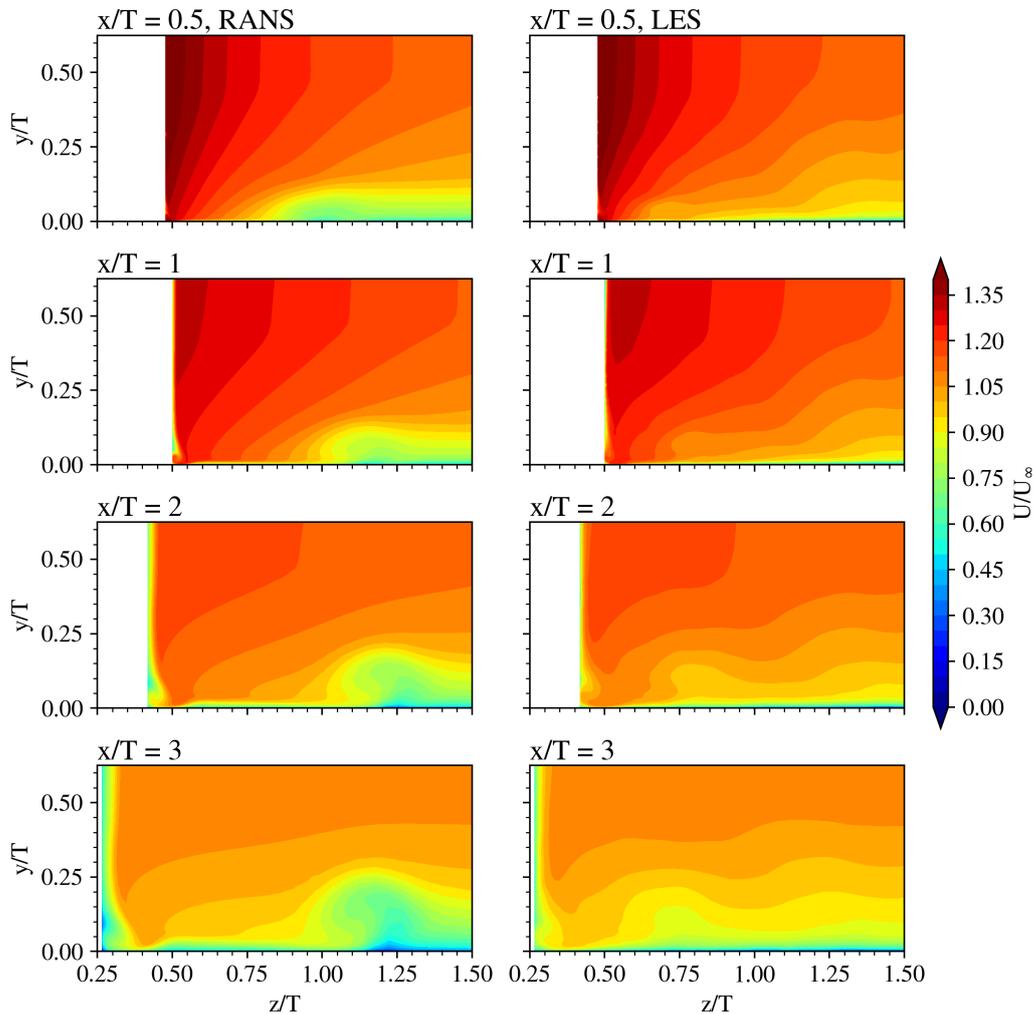


Figure 4.10: RANS (left) and LES (right) velocity in x direction for several y - z planes.

From the comparison done in this section a few key points are identified that describe the difference between RANS and LES on the wing body junction flow, and hence provide guidance for the improvements required by the new turbulence model. First the HSV, it is observed that the shape, location and k magnitude are different and hence need adaptation. Secondly, it was found that the corner vortex prediction is satisfactory, yet, the secondary vortices are not. And thus the anisotropy of the turbulence should be modified such that these secondary vortices are predicted. Finally, a laminar flow phenomenon difference was found which is described next in subsection 4.1.2.

4.1.2. Laminar Separation Bubble

As the LES data was investigated for the machine learning phase a laminar flow phenomenon was found which cannot be represented by the RANS equations as defined in Chapter 3. That is, a laminar separation bubble (LSB) was observed close to the leading edge of the airfoil, as visualised in Figure 4.11. This figure shows the U -velocity close to the airfoil wall at $y/T = 2$,

which is far away from the corner region. As can be seen in Figure 4.11 the LES data shows a bubble region of separated flow which is not present in the RANS result since the baseline $k-\omega$ SST model is not meant to predict flow transition.

The appearance of an LSB in the LES is not unexpected as the airfoil is relatively thick and the inflow above the boundary layer in the LES was fully laminar. This laminar flow separates in the adverse pressure region of the wing, transitions to turbulent flow, and consequently the flow re-attaches to the wing which causes the separation bubble. It should be noted that since the LES boundary layer over the bottom wall was turbulent, the corner region where the corner- and secondary vortices live does not have an LSB. However, above the corner region (approximately $y/T > 0.06$) the LSB exists on the wing and extends all the way up to the top of the domain. Note that this corner region with the turbulent flow is significantly smaller than the turbulent boundary layer thickness ($\delta_{inlet}/T \approx 0.44$). This is assumed to be caused by the HSV which pulls laminar flow from the upper region of the domain towards the bottom wall, as shown by the flow direction in Figure 4.2.

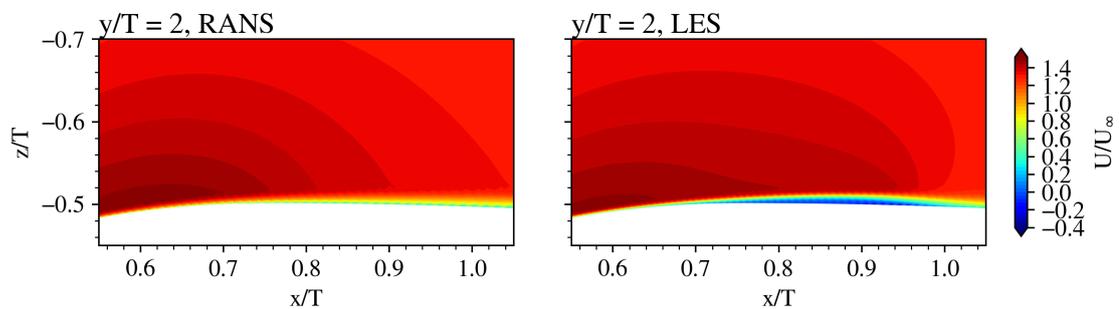


Figure 4.11: Presence of a laminar separation bubble on the surface of the airfoil in the LES solution, flow seen from above.

To further compare the pressure and viscous forces on the wing, the pressure and friction coefficient profiles (C_p and C_f) are displayed in Figure 4.12. Two different y -levels have been used, one in the turbulent corner region ($y/T = 0.025$) and one in the initially laminar region far away from the corner ($y/T = 2.0$). First of all, in the C_p plot it is observed that the RANS profile matches the LES data quite well and predicts the correct pressure peak and pressure recovery curve. Two discrepancies can be observed, that is the presence of the LSB in the LES data at $x/c = 0.2$ and the corner separation near the trailing edge in the RANS results.

This corner separation for $y/T = 0.025$ is especially visible in the C_f plot. In which it can be seen that for $y/T = 0.025$ the RANS results match the LES quite well except for the trailing edge region where the RANS result shows a (corner) separation. As discussed above a likely source for this separation is the fact that RANS cannot correctly predict the secondary vortices which causes the BL behaviour to be different, especially in the trailing edge region of the wing. The $y/T = 2.0$ curve is very different when comparing RANS to LES, which is due to the flow transition in the LES, the LSB can be clearly identified by the negative peak of C_f in the LES curve.

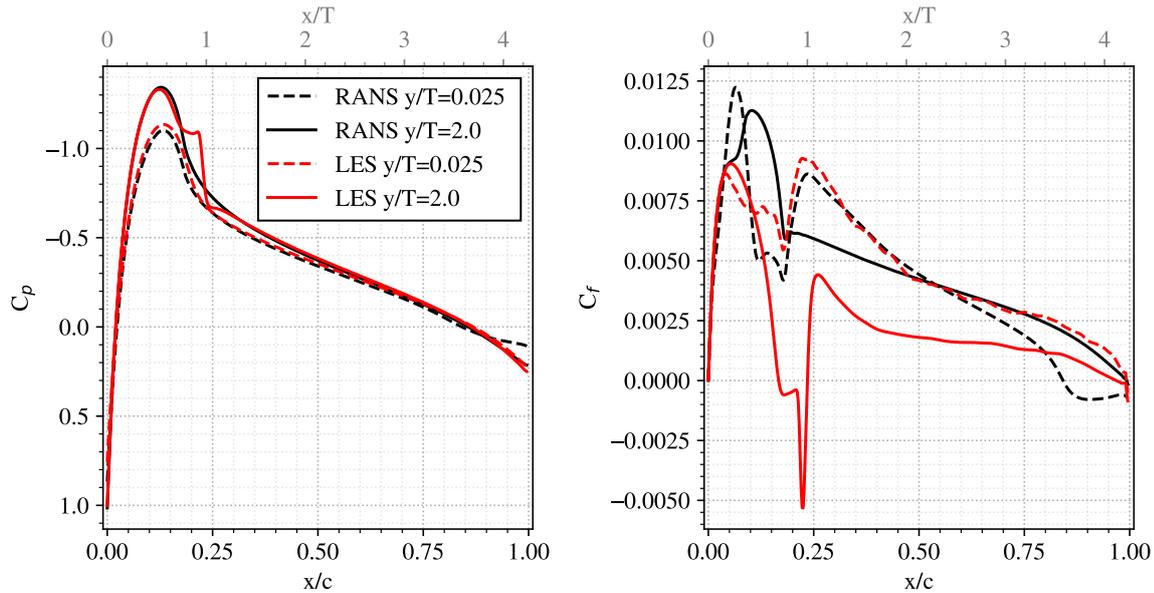


Figure 4.12: Pressure and skin-friction coefficient at two different wing span locations.

4.2. k-Corrective Frozen Results

From the comparison in the previous section, it is clear that corrections are required for the RANS turbulence model to decrease the discrepancies for the wing-body junction flow. In this section, the fields that define this correction are shown and discussed. To recall, the correction variables used in the SpaRTA method are the k -equation residual R and the turbulence anisotropy correction b_{ij}^Δ . These variables are found using the k -corrective frozen approach as described in Chapter 3.

4.2.1. Turbulent Kinetic Energy Correction

For the turbulent kinetic energy there exist two corrections, the direct correction via the residual R and the indirect correction of b_{ij}^Δ . The latter is through the production term in the k - and ω -equation, for reference revisit Equation 3.20. The production correction by b_{ij}^Δ is referred to as P_k^Δ ($= -2kb_{ij}^\Delta\partial_j U_i$). Although P_k^Δ is not regressed directly, since b_{ij}^Δ is regressed, it is still important to investigate the correction. This is because its significance or magnitude is similar to R . Below in Figure 4.13 the fields for R and P_k^Δ are shown in the symmetry plane upstream of the wing. As observed, the main corrections exist in the region where the HSV lives in the LES data, and in the corner region. A correction in these regions is expected when observing the difference between the k solutions in Figure 4.3. However, it has to be noted that the difference in k -fields (RANS vs. LES) cannot be directly linked to R and P_k^Δ since there is a mismatch in units.

From Figure 4.13 it is observed that the structures of the R field, in contrast to P_k^Δ , are quite complex and maybe even a bit noisy. From the comparison of the k -fields in the RANS versus LES comparison it was noticed that the k -field of the LES is more complex with respect to the RANS solution (recall the C-shape in Figure 4.3). Hence, using this more complex k -field to compute the value of R , it can be expected to consist of multiple (noisy) structures. Yet it should not be forgotten that the value of R is also influenced by the LES velocity field and the value of b_{ij}^Δ (via P_k^Δ). Thus it is difficult to pinpoint where this complex system of structures at the HSV location exactly comes from. Yet it should be kept in mind since the noisy structure

could introduce instabilities in upcoming simulations.

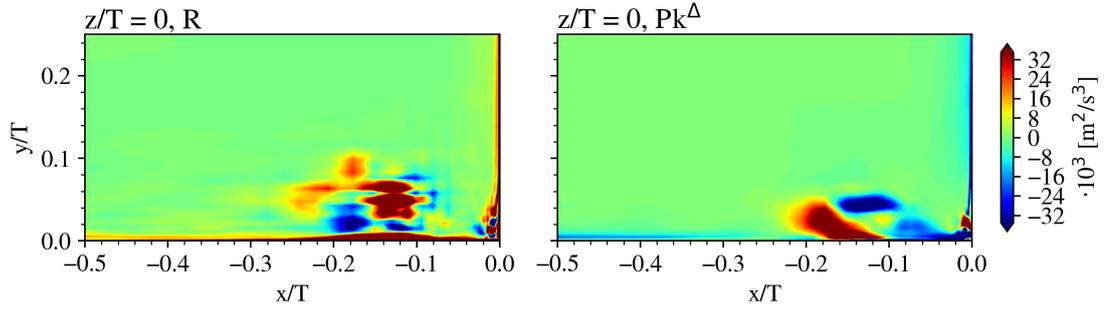


Figure 4.13: R and P_k^Δ in the symmetry plane upstream of the wing.

Similar to the symmetry plane, the results in the y - z planes shown in Figure 4.14 (R) and Figure 4.15 (P_k^Δ) also show two correction regions. That is again the HSV region and the corner region. It is noticed, however, that the R correction seems to be more significant in the HSV region compared to P_k^Δ . Comparing $x/T = 0.5$ and $x/T = 1$ it is observed that at the wing wall the correction is increasing, this is due to the LSB as shown in Section 4.1. The reason this correction is present is due to the fact that the LES solution is transitioning in this region from laminar flow to turbulent flow. Hence, to reproduce this transition behaviour in the RANS a significant k -correction is applied. Note that although it looks like R and P_k^Δ seem to cancel each other out on the wing wall this is not the case as will be shown.

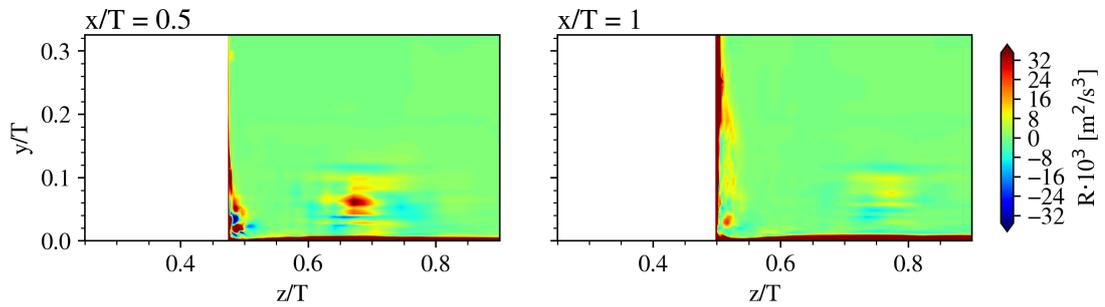


Figure 4.14: R in y - z planes at two x locations.

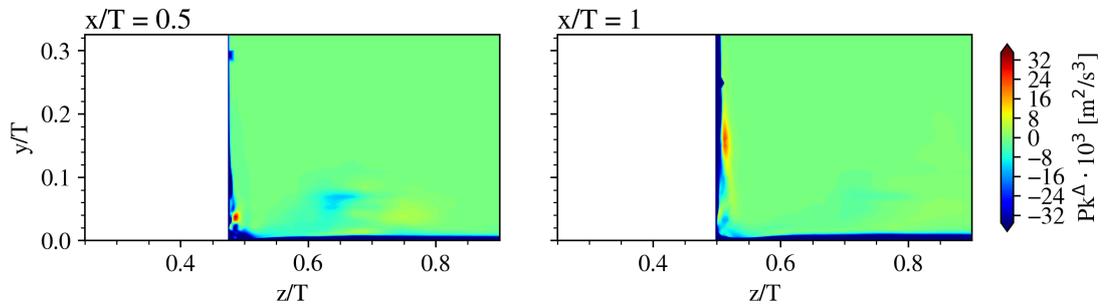


Figure 4.15: P_k^Δ in y - z planes at two x locations.

Adding R and P_k^Δ together results in the total correction to the k -equation, the summation of the two is shown in Figure 4.16. In this figure the planes $x/T = 2$ and 3 are also shown, however, as observed only the regions very close to the wall corrections at these locations. The top row figures still show the two previously discussed significant corrections in the corner and HSV region.

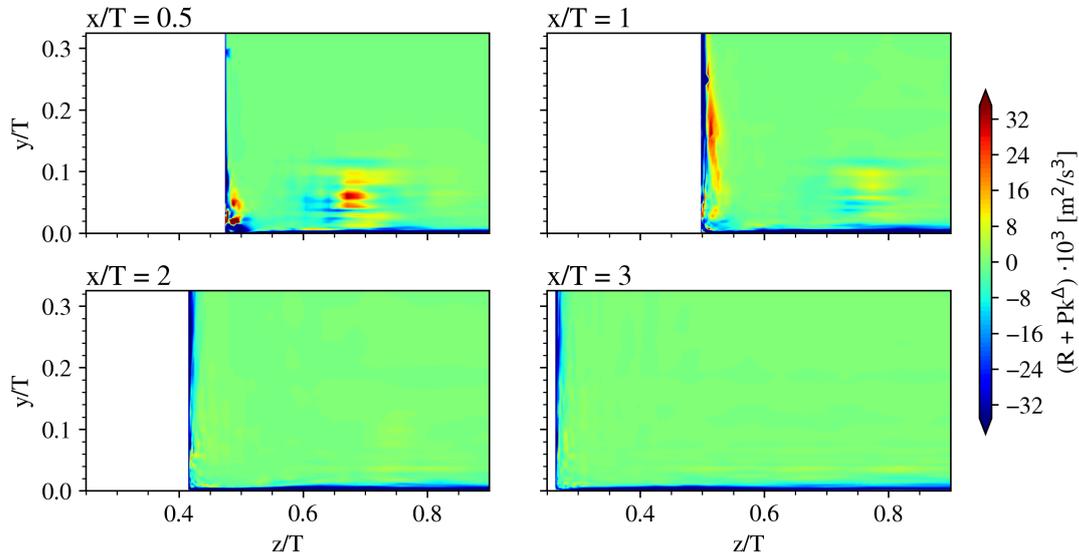


Figure 4.16: Sum of R and P_k^Δ in y - z planes at several x locations.

Next to the corrections for the HSV and corner region, a significant region of correction is found close to the bottom wall downstream of the leading edge. This correction has been visualised in Figure 4.17 for several z locations outside the range shown in Figure 4.16 at $x/T = 1$. From the curves, it is noticed that at $z/T = 2.6$ the correction is close to 0. This indicates that the correction is not just due to a slight mismatch in inlet boundary layer profiles between the RANS simulation and LES data since the domain extends in z/T from $[-6.3, 6.3]$. Therefore the correction is likely related to the separation and/or modification of the boundary layer by the vortex system close to the wall. Otherwise, a correction would be expected to extend all the way to $z/T = 6.3$.

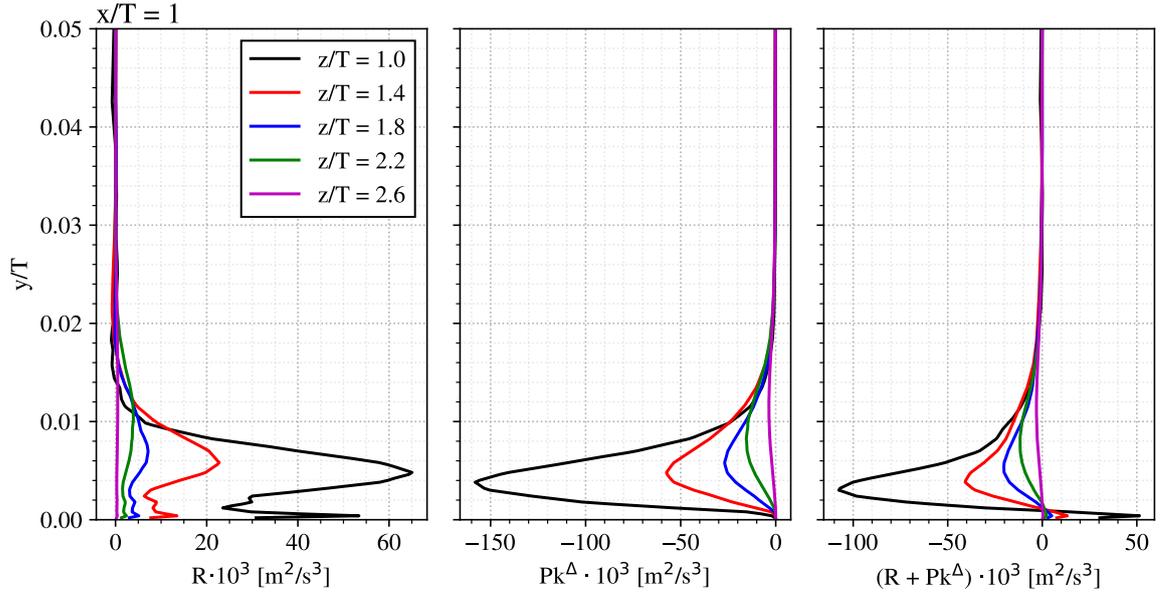


Figure 4.17: Profiles of R , P_k^Δ and $(R + P_k^\Delta)$ at $x/T = 1$ and multiple outboard z locations.

In Figure 4.14 it was observed that a correction close to the wing wall is present, the latter is to reproduce the transitional behaviour of the flow over the LSB. A better visualisation of this correction is present in Figure 4.18, in which the flow seen from above is shown at a distance far away from the corner region. Especially when comparing this figure to Figure 4.11, it can be seen that a significant k correction is present at the aft location of the LSB. Figure 4.18 also shows that the negative correction by P_k^Δ does not cancel the small bubble of positive R .

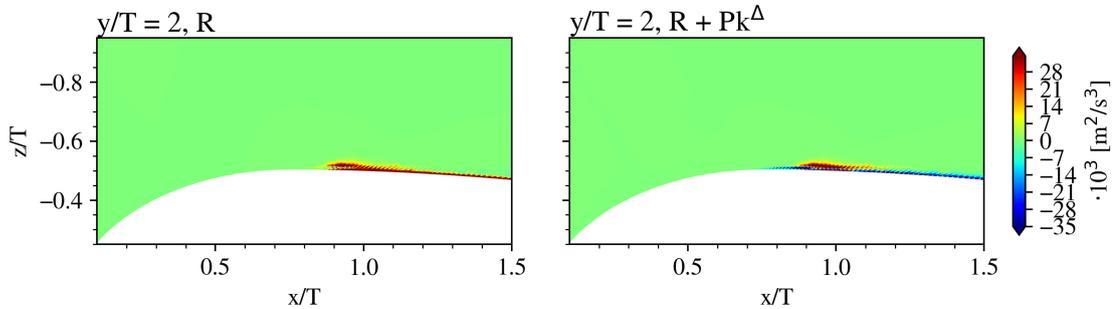


Figure 4.18: k -correction close to the wing wall far from the corner region to show the effect of the LSB, flow seen from above.

4.2.2. Anisotropy Correction

The second correction field is the Reynolds stress anisotropy correction tensor b_{ij}^Δ . This term aims at correcting the Reynolds stress tensor such that the anisotropy of the turbulence in the junction flow is more accurate. The resulting b_{ij}^Δ components from the frozen approach in the symmetry plane are shown below in Figure 4.19. In this figure corrections in the HSV region are found in the yy , zz and xy components. Note that b_{ij}^Δ , or b_{ij} in general, is multiplied by k in the RANS equations. Therefore, the corrections close to the wing wall and in the free stream region for xx and yy appear significant, yet the value of k is close to zero in those regions (see Figure 4.3), thus the effect is limited. The corrections to the xz and yz components are close

to zero and thus insignificant.

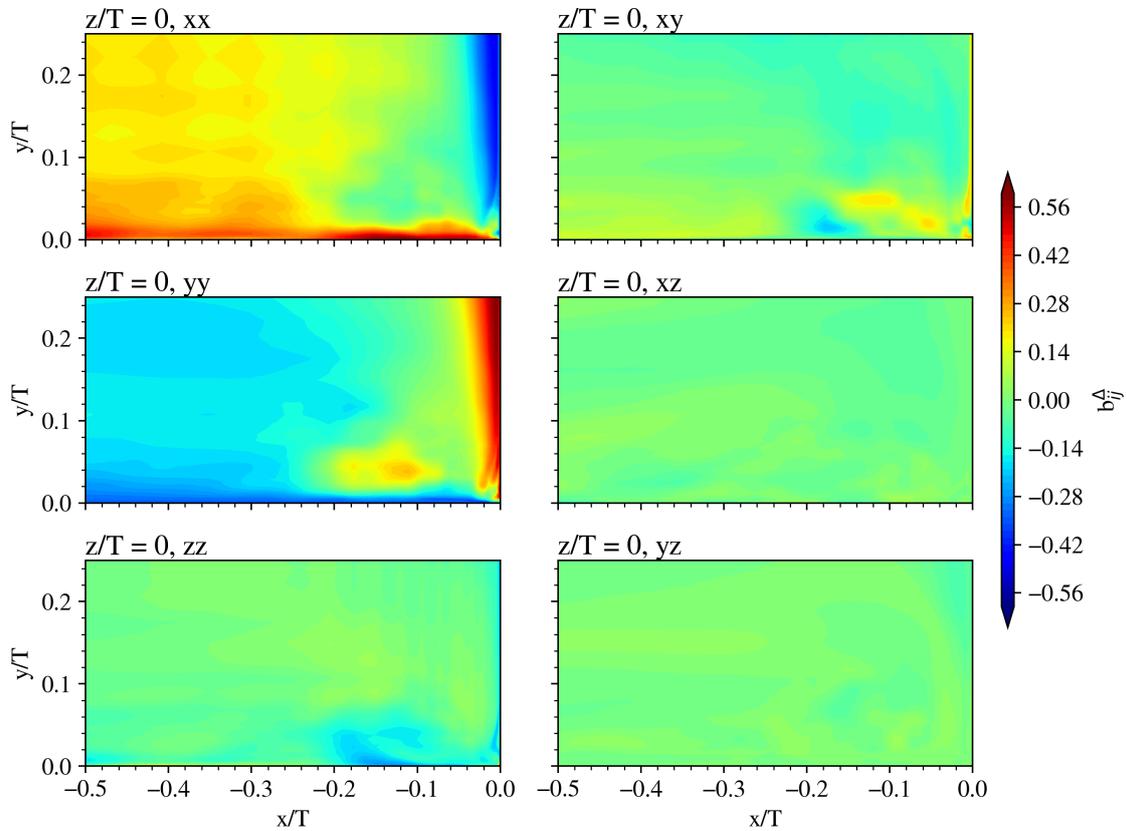


Figure 4.19: Components of b_{ij}^{Δ} in the symmetry plane upstream of the wing.

To continue the discussion of b_{ij}^{Δ} in the y - z planes it is important to visualize the value of k . This is because of the dependence of the effect of b_{ij}^{Δ} on k . Therefore, below in Figure 4.20 the value of k from the LES data is displayed for two y - z planes. This figure shows that, as expected, the value of k is high in the HSV region and the corner region as well as close to the bottom wall. For $x/T = 3$ the value of k is also higher at the wing wall since the flow has transitioned to turbulent flow in the LES (flow transitions is just upstream of $x/T = 1$ see Figure 4.12). This knowledge allows the continuation of the analysis of b_{ij}^{Δ} .

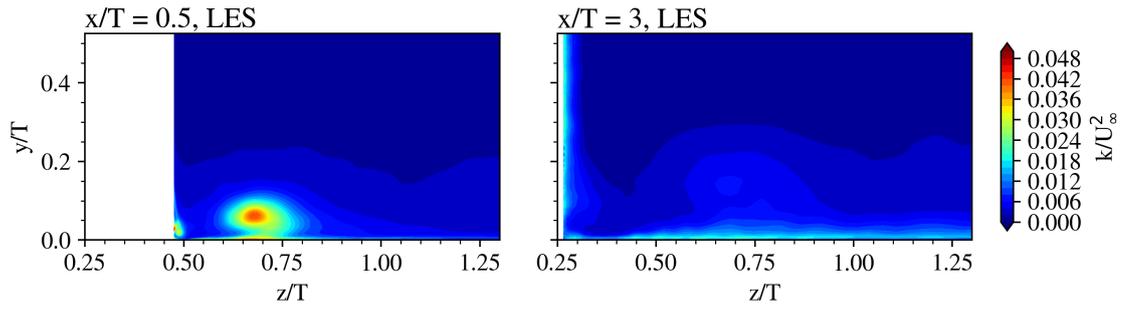


Figure 4.20: The k fields in y - z planes from the LES data at two x locations.

First b_{ij}^Δ at $x/T = 0.5$, see Figure 4.21, in this figure some corrections near the flat plate are seen in the yy and zz components. Next to that, there is a visible yy correction in the HSV region. Again the yy and zz corrections close to the wing wall are less critical due to the low value of k .

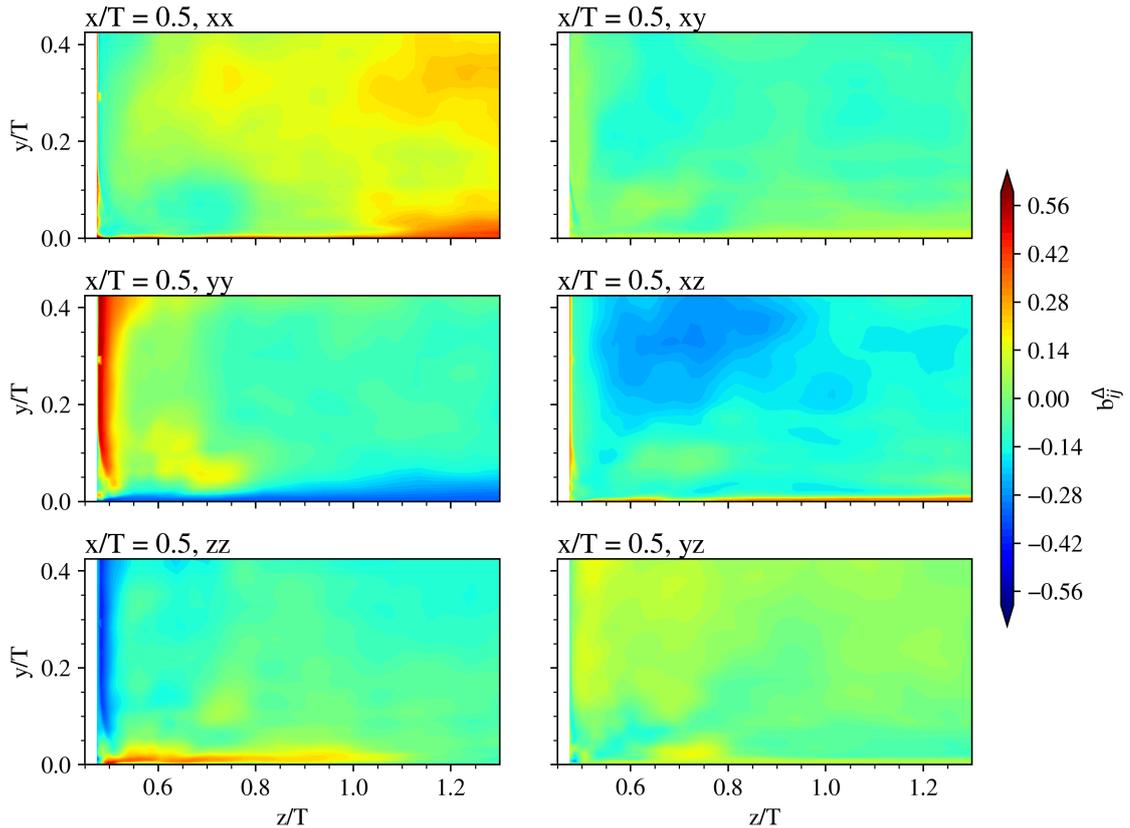


Figure 4.21: Components of b_{ij}^Δ in the y - z plane at $x/T = 0.5$.

A more exciting correction is found when zooming in on the corner region, which is shown in Figure 4.22. This figure shows anisotropy corrections in the region where the corner vortex

and the secondary vortices live. These corrections are seen in the yy , zz and yz components. Such corrections are expected since it was shown in Section 4.1 that the RANS simulation does not predict the secondary vortices well. And as discussed before, the prediction of these vortices is influenced by the prediction of the anisotropy in the Reynolds stress tensor. It is, therefore, seemingly not a coincidence that the corrections found in the corner region line up with the $P3$ term, which is a function of $v'v'$ (i.e. yy) and $w'w'$ (zz), and the $P4$ term, function of $v'w'$ (yz), from the vorticity equation in the free stream direction (Equation 2.3).

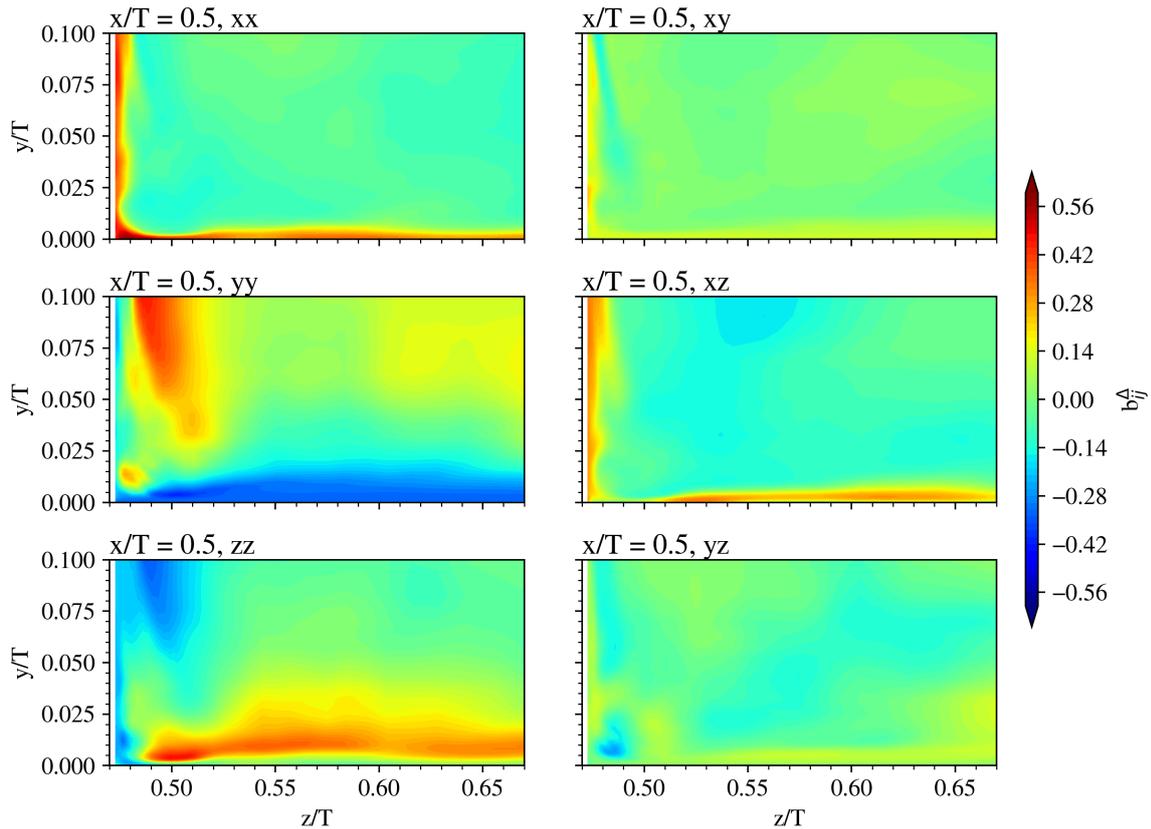


Figure 4.22: Components of b_{ij}^{Δ} in the corner region in the $y-z$ plane at $x/T = 0.5$.

Finally, the corrections at $x/T = 3$ are shown in Figure 4.23. From this figure, it is observed that only significant corrections are found in the diagonal components. For which the zz component only seems to correct the wing wall and not the bottom plate. Besides, the xx and yy components appear to exchange with one another. That is, the positive xx correction at the walls roughly cancels the combined negative corrections of yy and zz components indicating a redistribution of the turbulent components. The off-diagonal components do not show any significant corrections.

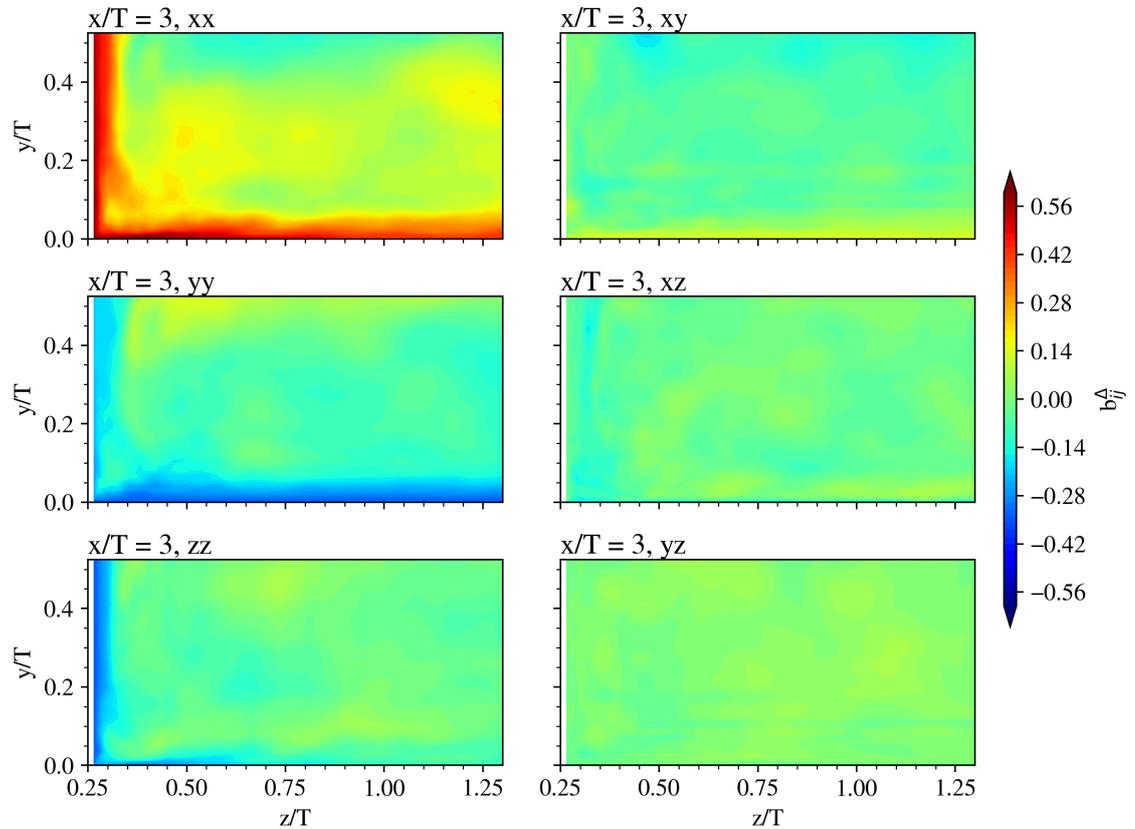


Figure 4.23: Components of b_{ij}^{Δ} in the $y-z$ plane at $x/T = 3$.

4.3. Correction Field Propagation

The propagation of the correction fields discovered in Section 4.2 is an important step in the SpaRTA process. This is because the propagation shows the theoretical best result achievable with respect to the discrepancy between the target LES data and the modified RANS. It is thus expected that the flowfields generated by the propagation closely match the LES data. As if this were not the case then the correction fields and or correction variables are insufficient for this flow problem. The propagation can therefore be seen as a verification step of the corrections in the SpaRTA process. This section displays the results of the propagation fields as found in Section 4.2, using the equations and spatial blending functions described in Chapter 3.

In Figure 4.24 the boundary layers of both the propagation and the LES data are shown in the symmetry plane upstream of the wing. From this figure, it is identified that the propagation matches the LES data very well. This means that the separation point upstream of the leading edge is determined correctly.

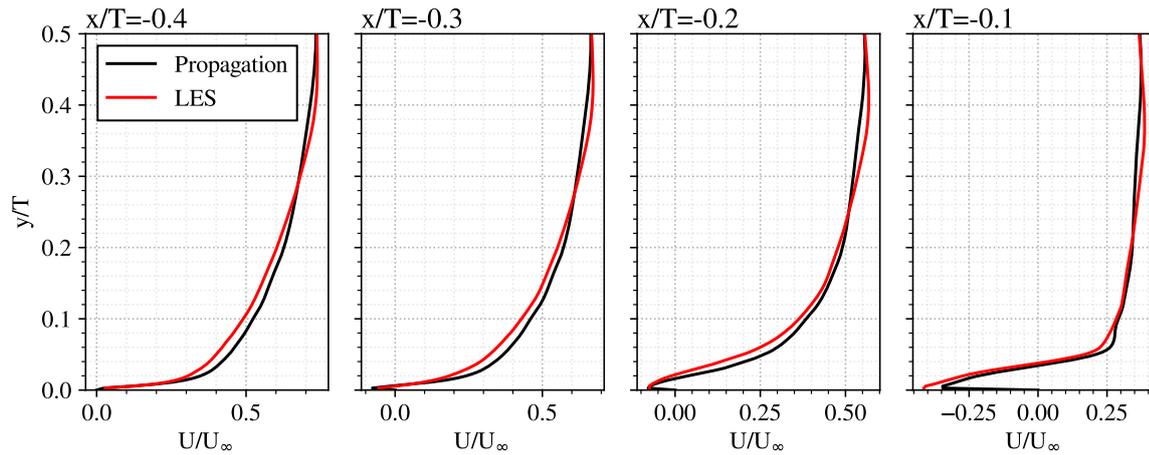


Figure 4.24: Propagation boundary layer profiles in the symmetry plane upstream of the leading edge.

The correct location of the separation point is also visible in Figure 4.25 in which the z -vorticity is displayed in the symmetry plane. Next to the separation point, the propagation displays the correct shape and magnitude of the HSV as well. This is a welcome result since the LES HSV shape is the result of time averaging an unsteady phenomenon. This propagation thus shows that with the correction fields determined by the frozen approach, the time average result of the HSV can be found using a steady-state CFD solver.

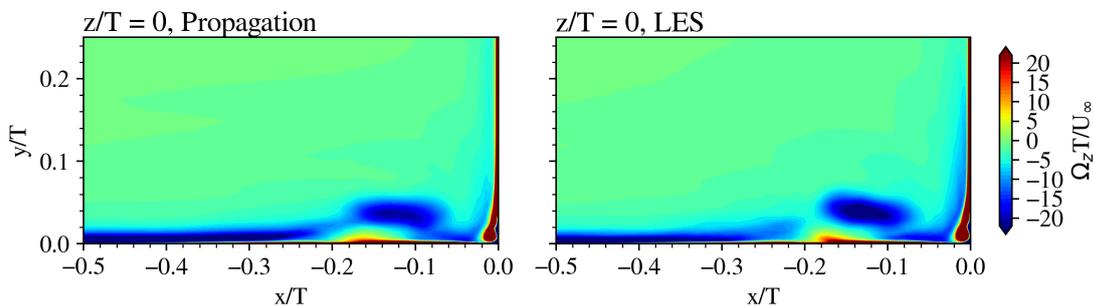


Figure 4.25: Propagation and LES z -vorticity in the symmetry plane upstream of the leading edge.

The comparison of k in the symmetry plane, presented in Figure 4.26, shows that the faulty k shape from the baseline RANS (Figure 4.3) is corrected in the propagation using the correction fields. Not only is the C-shape visible but also the magnitude of k in the corner region.

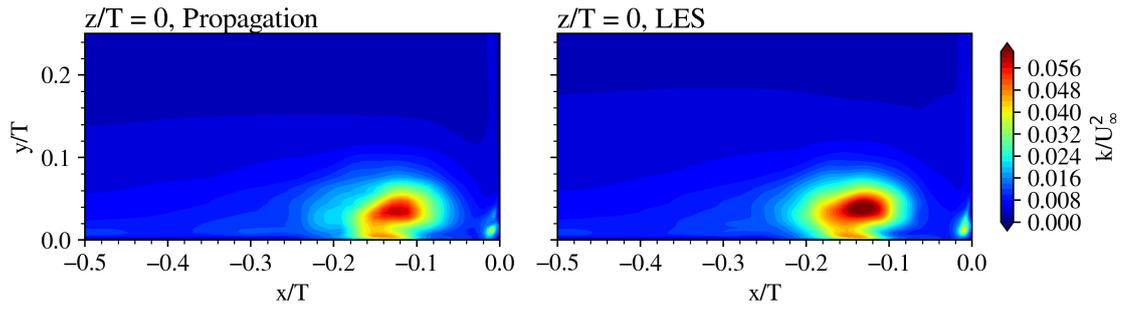


Figure 4.26: Turbulent kinetic energy in the symmetry plane upstream of the leading edge for both propagation and LES.

Next to the investigation in the symmetry plane is the comparison of the propagation with the LES in the y - z planes downstream of the leading edge, these are shown next. First in Figure 4.27 the vorticity in the x -direction is displayed. The main take-away from this figure is the fact that the shape, position and magnitude of the HSV are correctly shown in the propagation simulation.

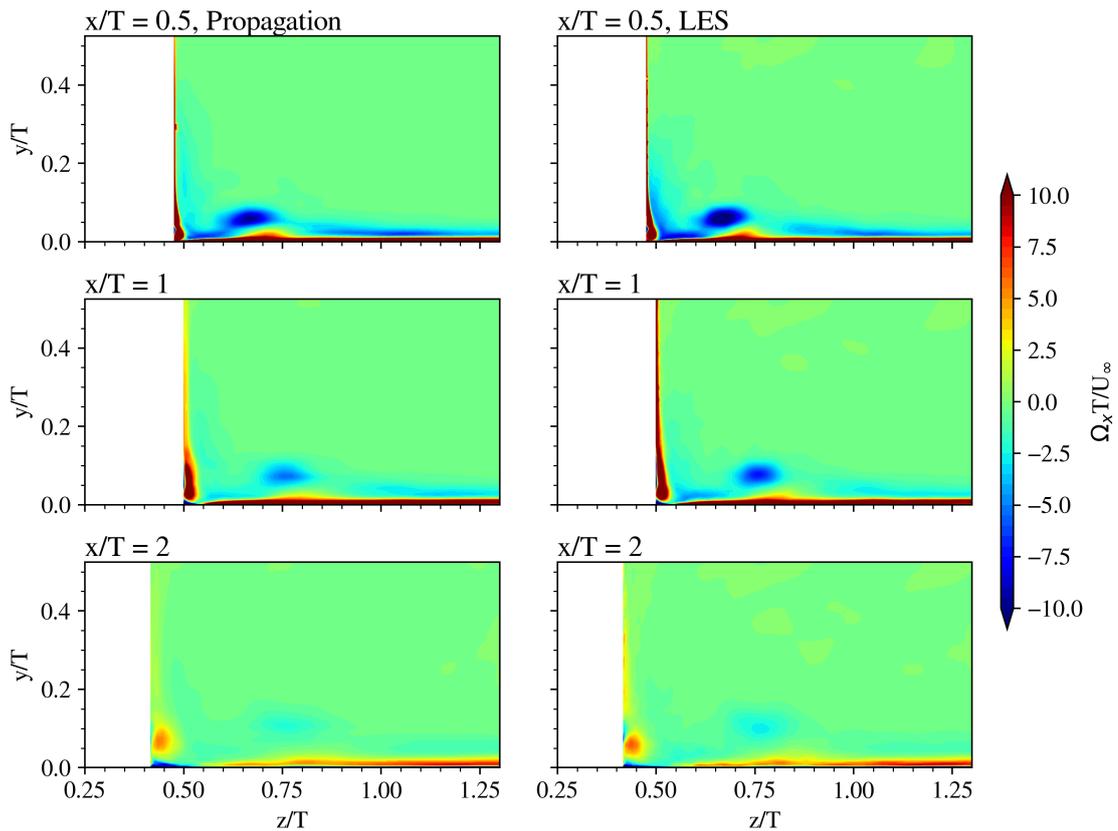


Figure 4.27: Vorticity in x -direction in y - z planes at several x/T locations.

However, more interesting is whether the propagation determines the correct secondary vortices in the corner region. To recall, the correct vortex topology was missing in the baseline RANS. Pleasantly, as presented in Figure 4.28 and Figure 4.29 the secondary vortices are in fact present in the propagation results. This includes both the small vortex with negative magnitude at $z/T \approx 0.5$ in Figure 4.28 and the positive structure at $z/T \approx 0.265$ in Figure 4.29. It is noted, however, that in there exists a discrepancy in the magnitude of the corner vortex as well as the negative vorticity close to the bottom wall in Figure 4.29. Yet these discrepancies are negligible compared to the improvements with respect to the baseline RANS.

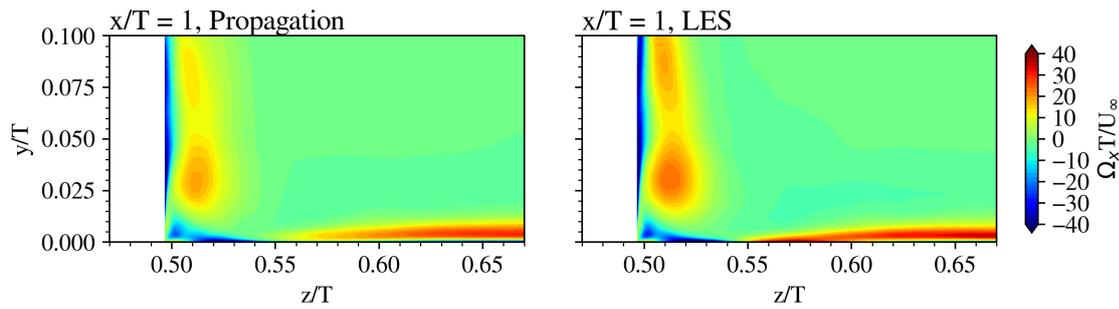


Figure 4.28: Vorticity in x -direction in the corner region, note the change in colour-scale.

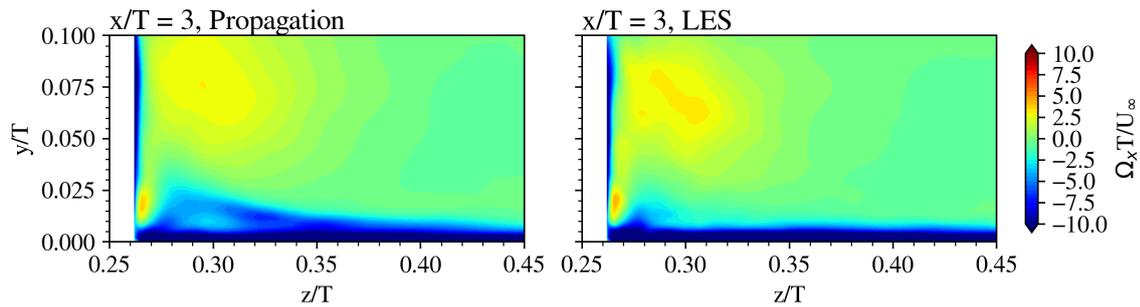


Figure 4.29: Vorticity in x -direction in the corner region.

With the correct vorticity, it is expected that the U velocity is accurate as well. And indeed as shown in Figure 4.30 the U velocity downstream of the leading edge matches the LES well. Only minor discrepancies are present on the wing wall, this is due to the spatial blending function used on the wing wall to filter the LSB. As shown in Chapter 3, this blending function linearly blends out the correction fields in very near proximity to the wing wall above the corner region.

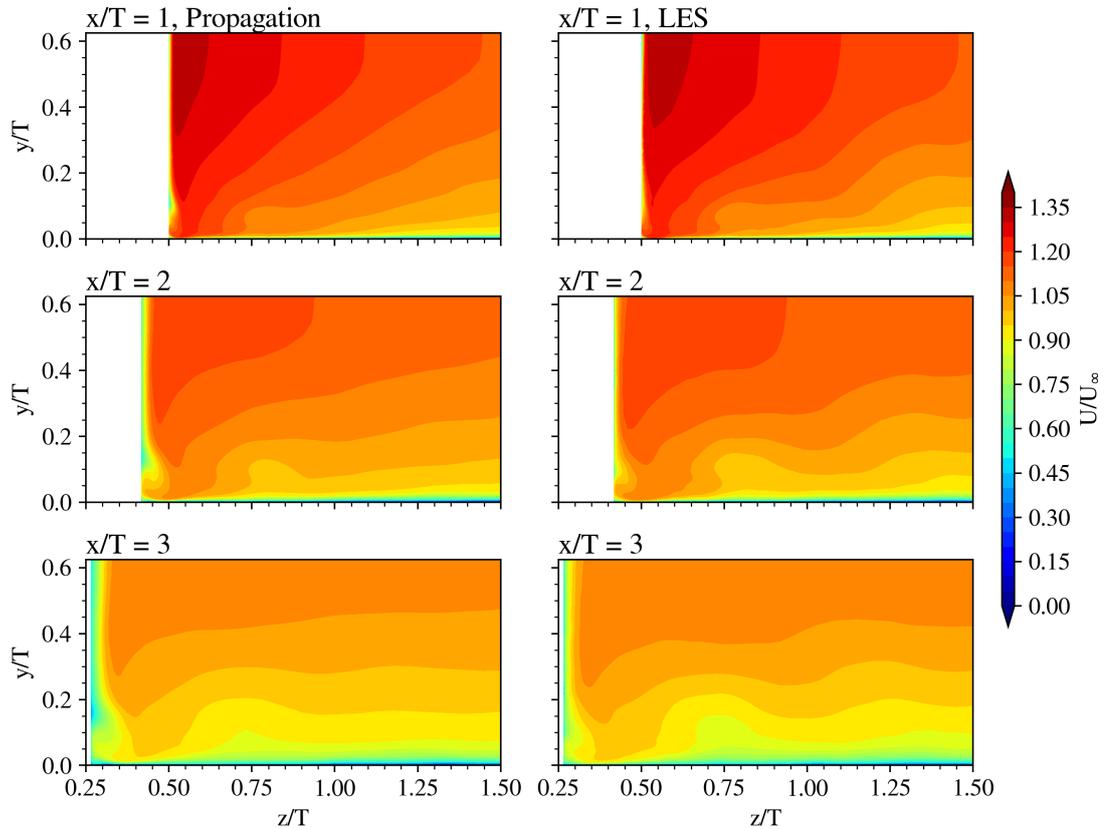


Figure 4.30: U velocity for several y - z planes for both propagation and LES.

The main effect of using the spatial blending function around the wing is that the wing forces, especially the viscous forces as shown in Section 4.1, will be dissimilar to the LES in the blended region. This is because instead of an LSB as present in the LES, the baseline RANS solution will approximately be present in this blended region. Note that this blending has been applied for the numerical stability of the later discussed model propagation and that it has been confirmed that indeed the corrections found by the frozen approach will induce a correct LSB in the propagation if used without this blending at the wing. The latter propagation is, however, not shown in this work as it is irrelevant to the discussion since the propagation is linked to the model propagation for which the blending is required. The discussion on why the spatial blending is required will be covered in Section 4.6.

Nevertheless, for the region below the spatial blending, i.e. the corner region, the wall forces can be compared. Below in Figure 4.31 the skin friction coefficient on the wing based on the wall shear stress in x -direction is displayed for various y levels. Note that the C_p profiles are omitted since as already shown in the baseline RANS these match very well in the corner region. From Figure 4.31 it can be seen that the propagation is matching the LES to a good extent. However, more interestingly is the removal of the corner separation present in the baseline RANS, which is likely due to the correct determination of the vortex topology in close proximity to the corner.

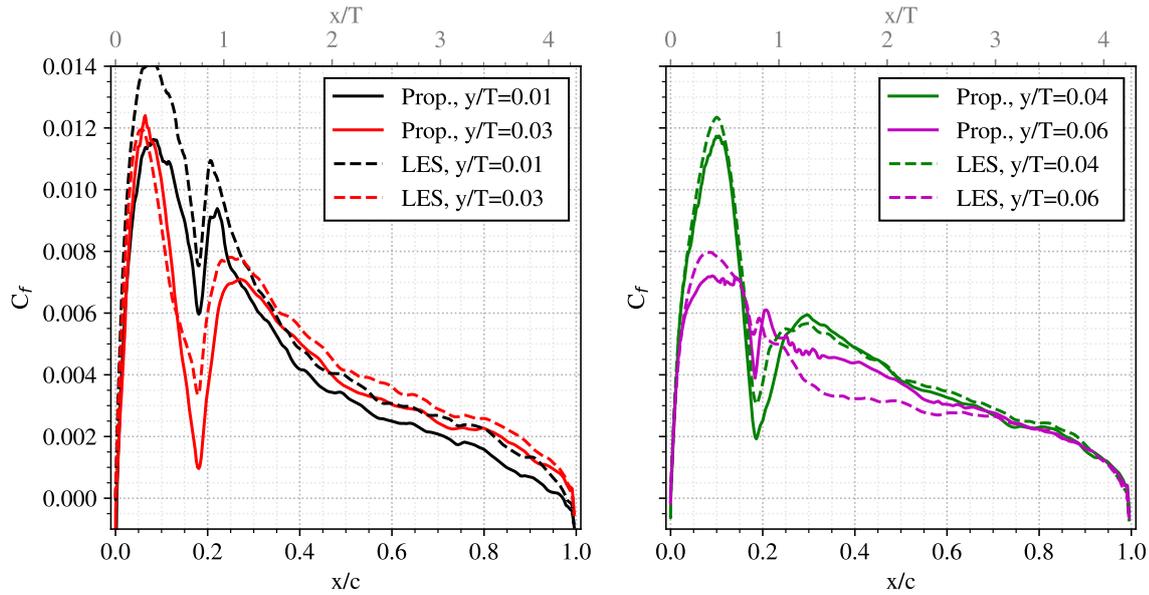


Figure 4.31: Skin friction coefficient profiles of the wing in the corner region for both propagation and LES.

At the start of this section, it was described that the propagation serves as a verification of the correction fields. The results showed that the correction fields were verified successfully since the flow fields match the LES data with only minor discrepancies.

4.4. Model Training Results

This section covers the results from the sparse regression according to the method described in Chapter 3. First, the regression of the turbulent kinetic energy residual R will be discussed. Secondly, the results regarding the regression of the turbulence anisotropy correction are displayed.

4.4.1. Turbulent Kinetic Energy Residual

As mentioned in Chapter 3 the first step in the regression of R is to choose a subset of the secondary features to reduce the computational memory requirement for the regression. For the regression of R , this subset is chosen based on the mutual information (MI) of each secondary feature with R . The resulting MI values are displayed below in Figure 4.32 for four different values of n . The value of n indicates the number of neighbours used by the numerical method to estimate the MI ($n = k$ in Equation 3.37). Only the 25 features with the highest MI score out of all secondary features are displayed and ordered from high to low based on the result of $n = 3$. Note that the variables in Figure 4.32 are written using alphabetic letters as used in the SpaRTA code, e.g. $W2_s = \{\Omega_s^2\} = II_s$, for the full list of the SpaRTA nomenclature please refer to Appendix A.

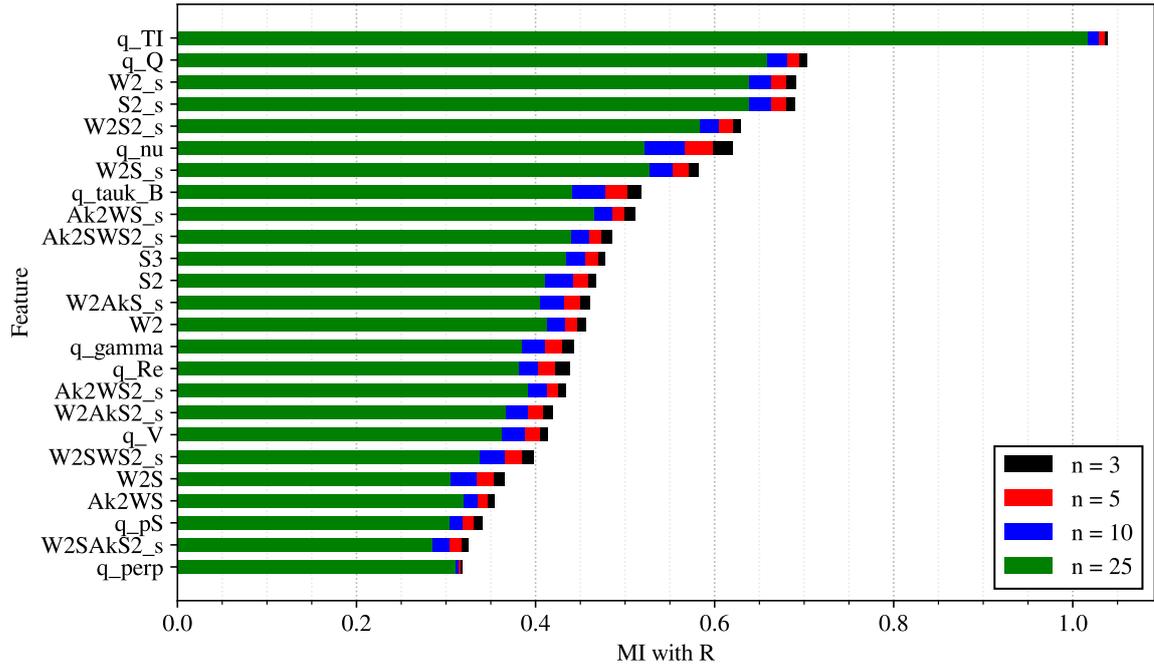


Figure 4.32: Mutual information of the different features with R using the propagation flow field, n equals the number of neighbours used in the numerical approximation.

In this figure, it can be seen that for this particular problem, apart from $n = 25$ which is outside the recommended range (see Kraskov et al. [57]), the relative score between different secondary features remains fairly constant for the different values of n . However, with increasing n the magnitude of the scores decrease. From this figure, the subset of secondary features is decided on, yet not all features are selected purely based on their MI score. This is because MI only indicates the level of dependency of feature i on R , but that does not necessarily mean that it forms a numerically stable model in an actual CFD simulation. Therefore, based on experience, it was decided to not include the invariants that rely on ∇k (Ak in Figure 4.32). It was reasoned that these features could be added later in the process if for any reason the regression did not fit the data well enough. Similarly, q_{TI} was omitted due to the feature not being Galilean invariant. Besides, it was decided to add features q_V and q_{Re} . The latter since, first, the vortex stretching in the junction flow is significant, therefore q_V could provide information. And second, it was observed that a significant amount of R is located near the wall for which q_{Re} could provide important information since it is based on wall distance. These arguments, combined with the MI, result in the following subset of secondary features used for the regression of R :

$$\text{Secondary features} = [I_s, II_s, IV_s, V_s, I, II, III, q_Q, q_\nu, q_{\tau k B}, q_\perp, q_V, q_{Re}] \quad (4.1)$$

An observant reader might wonder why q_{perp} is in the list of secondary features since it has a low MI score. This is because when the MI was initially estimated the flow was thinned and boxed (see Chapter 3) which resulted in a relatively good MI score for q_{perp} . However, as shown by Goderie [58] the more data points are available the more accurate the MI estimate becomes. Hence the MI setup was rerun using only boxing (no thinning) after the regression was performed which resulted in the scores shown in this section. The outcome was not significantly different, apart from the score of q_{perp} . This does, however, not pose a problem since the features that had a high MI score initially, also have a high score in the rerun, as expected.

For the primary features, the complete set is used, as defined in Equation 3.31. These two sets then allow generating the feature library as described in Chapter 3. The features are scaled by their standard deviation and the features are generated on cells randomly selected within a predefined box (see Chapter 3). Yet this order could be either way, that is, first select the cells within the box and then scale the variables using the selected cells to determine the standard deviation. Or, alternatively, scale the variables first using their standard deviation for the entire domain and then select the cells in the box. At first, it was thought that this would not result in significant differences in the regression, yet it did as the results will show.

The results of the regression are shown in Figure 4.33 and Figure 4.34 which show the value of the regression coefficients of each basis function (Y -axis) in each model (X -axis). Figure 4.33 shows the regression result for which the variables are scaled using the selected cells to calculate the standard deviation, while Figure 4.34 shows the result using the complete domain for scaling. To better explain the visualisation of the regression in Figure 4.33 and 4.34, it should be noted that for each combination of λ and ρ the elastic net regression regresses a model (M_i) which is of the form:

$$M_i/std(\Delta) = c_1 \cdot [f(s_j/std(s_j)) \cdot f(s_k/std(s_k)) \cdot \dots \cdot p_l] + c_2 \cdot [f(s_m/std(s_m)) \cdot \dots \cdot p_n] + c_3 \cdot [\dots] + \dots \quad (4.2)$$

In which f represents a (non-)linear function, Δ the target, s_i any secondary feature, p_i any primary feature and c_i the regression coefficients. The coefficients c_i in the equation above give an indication of the importance of one feature over another. The value of each c_i for each basis function is displayed in Figure 4.33 and 4.34. If a rectangle is blank then this means the basis function is not part of the specific model M_i . Although the value of the coefficients gives a good indication of the dominance within a model it has to be kept in mind that the features are based on physical quantities which can be more active in one region in the flow than in another. Meaning that, even though c_i might be relatively high if the actual feature is close to zero in a particular flow region then c_i is insignificant in that region. Again, for the features the SpaRTA nomenclature is used, please refer to Appendix A, and note that the Python language is used for the math operations, i.e. $x^{**2} = x^2$.

When comparing Figure 4.33 with Figure 4.34 the resulting models seem to be similar, yet there is a noticeable difference in the primary features. The models in Figure 4.33 use both primary features ϵ and $G_s^{(1)}$ from complexity 4 and onwards ($M2, M3, \dots$). In contrast, the models in Figure 4.34 only use the primary feature ϵ . The first $G_s^{(1)}$ feature appears in $M11$ which has complexity 12. Besides this difference in primary features, the regressions, as expected, also show similarities. For both regressions, the models rely heavily on the ϵ -element as well as the $[\sqrt{|TV_s|} \cdot \epsilon]$ -element. Although the coefficient of the latter is relatively small.

To further analyse the performance of the regression two models from the coefficient matrices, one from Figure 4.33 and one from Figure 4.34, are picked. In order to justify this decision the mean square error (MSE) and R^2 -score of each model are displayed in Figure 4.35 and Figure 4.36, note the model naming (Y -axis) corresponds to the naming in the coefficient matrices.

First regarding the models that were generated using the standard deviation in the boxed region, Figure 4.35. In this figure, it can be seen that the MSE globally reduces with the complexity of the model except for the outlier of $M5$. Note that the models are ordered based on their complexity, i.e. the complexity of $M1$ is lower than the one of $M2$, etc. Although the MSE reduces, the R^2 -score stays fairly constant. Since the models with lower complexity are preferred, as these will avoid overfitting the data and are likely to be more stable in an *a posteriori* CFD simulation, model 3 was chosen as the model to test from this regression.

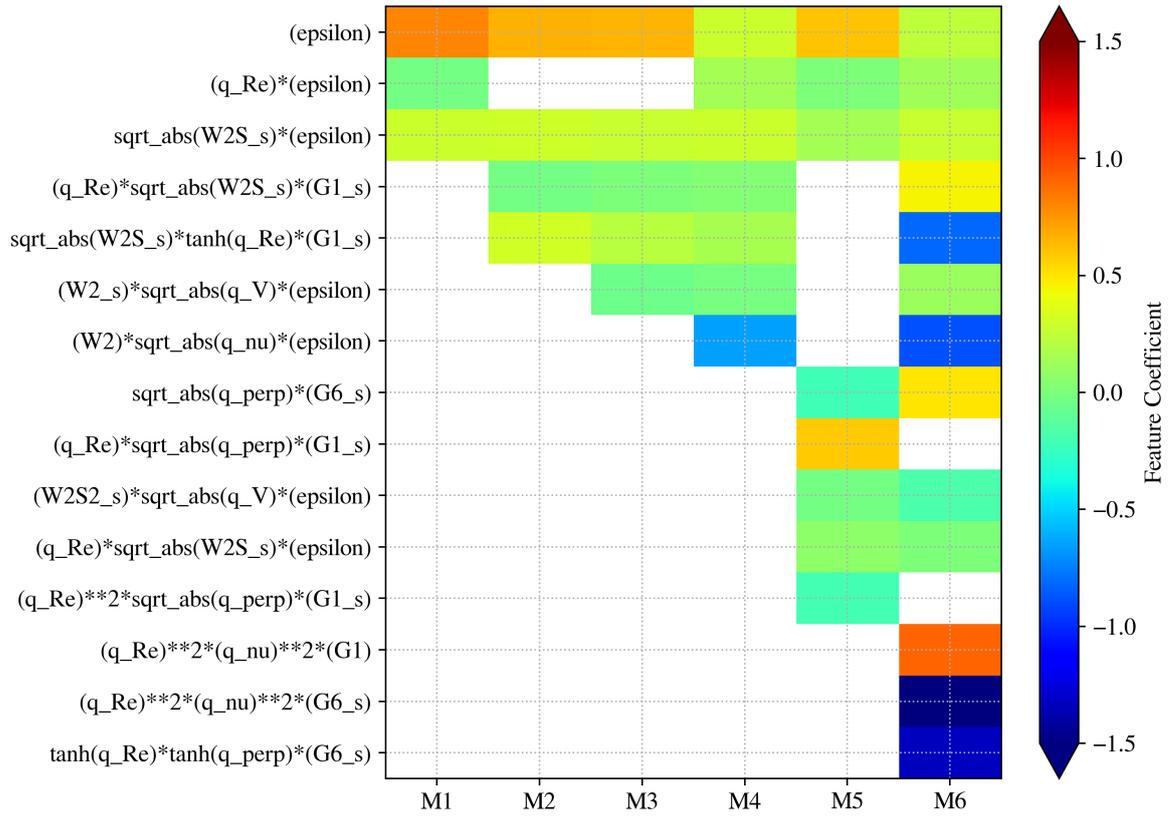


Figure 4.33: Feature coefficient matrix of the R models for which the features are scaled using the selected cells.

Since it still has a relatively low complexity of 5 terms while having a R^2 -score of 0.91 and a slightly lower MSE than model 2 which has the same R^2 -score.

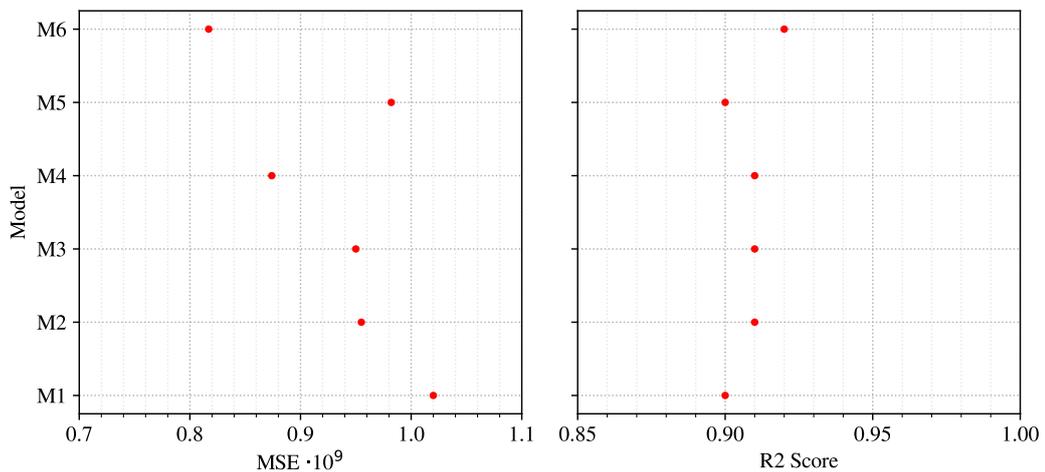


Figure 4.35: Mean squared error in $[m^4/s^6]$ and R^2 -score of the models for which the features are scaled using the selected cells.

Similarly, in Figure 4.36, the MSE and R^2 -score for the models displayed in Figure 4.34 are displayed. Again with increasing complexity, $M7 \rightarrow M16$, the MSE reduces, yet the R^2 -score

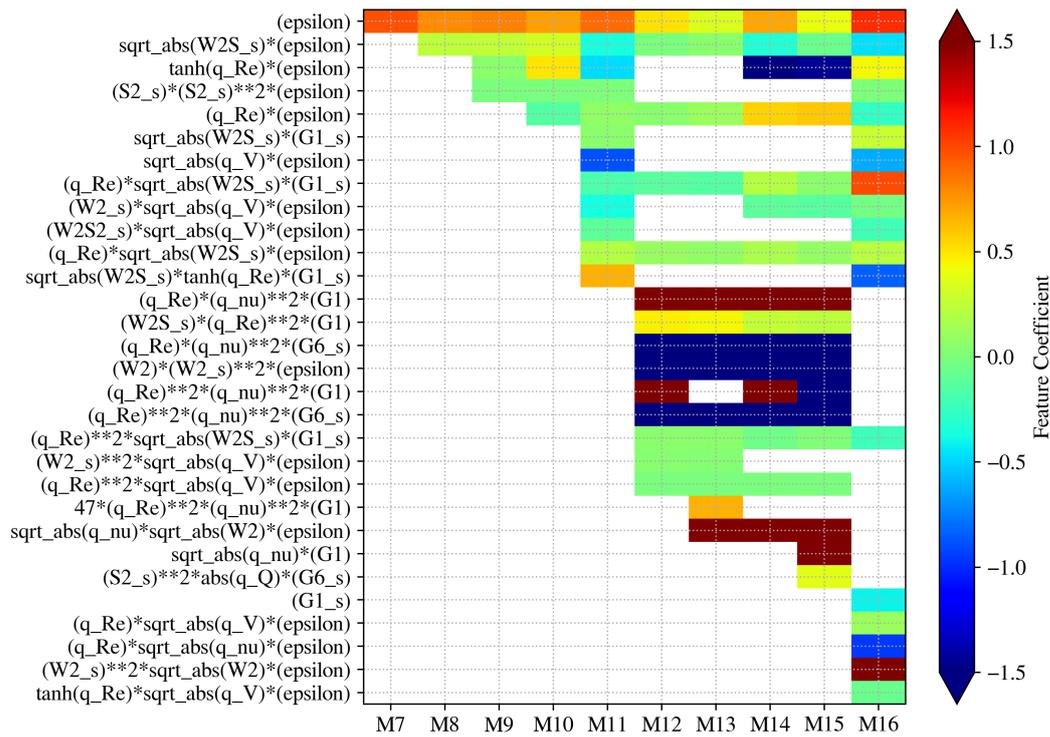


Figure 4.34: Feature coefficient matrix of the R models for which the features are scaled using the entire domain.

stays relatively constant for the lower complexity models. Eventually, it was decided to choose model 9 which has complexity 4. This is because model 9 has the same R^2 -score as $M10$ but is only dependent on the primary feature ϵ , which allows for comparison with model 3 which relies on ϵ and $G_s^{(1)}$. Note that for $M11 \rightarrow M16$ the MSE is decreasing and the R^2 -score is increasing, however, the complexity is relatively high for these models which cause overfitting and likely unstable CFD simulations. Therefore, these models are not considered.

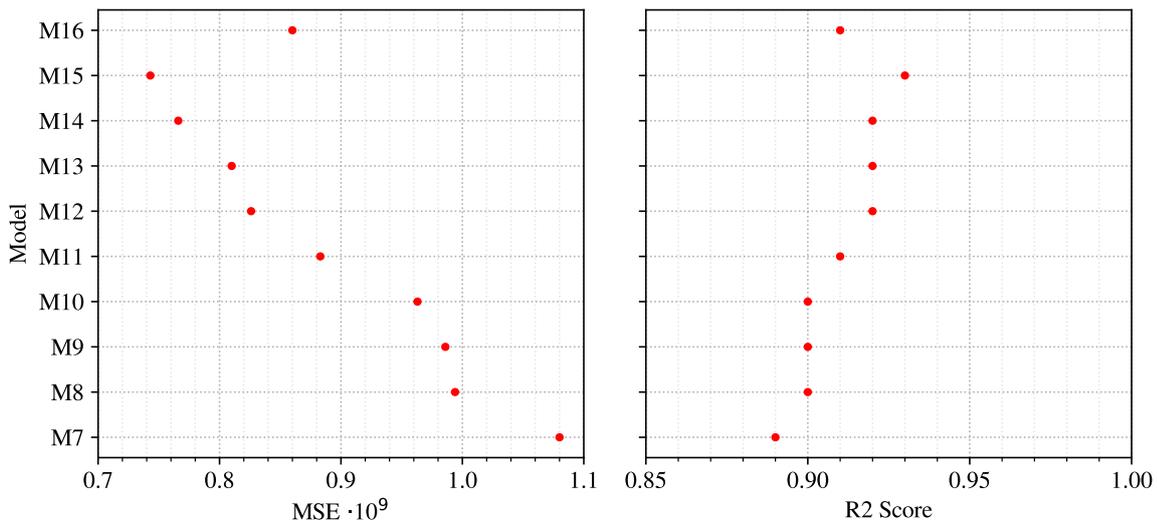


Figure 4.36: Mean squared error in $[m^4/s^6]$ and R^2 -score of the models or which the features are scaled using the entire domain.

Thus, the two R -models that will be further investigated are displayed below. Note that the complete model is given including the scaling factor of each feature.

$$\begin{aligned}
 M3_R = & [0.6505 \cdot (\epsilon/1198213.34) \\
 & + 0.2751 \cdot \sqrt{|(IV_s/0.026)|} \cdot (\epsilon/1198213.34) \\
 & + 0.009679 \cdot (q_{Re}/0.73) \cdot \sqrt{|(IV_s/0.026)|} \cdot (G_s^{(1)}/312067.87) \\
 & - 0.04692 \cdot (II_s/0.22) \cdot \sqrt{|(q_V/0.28)|} \cdot (\epsilon/1198213.34) \\
 & + 0.2163 \cdot \sqrt{|(IV_s/0.026)|} \cdot \tanh((q_{Re}/0.73)) \cdot (G_s^{(1)}/312067.87)] \cdot 101329.75 \quad (4.3) \\
 M9_R = & [0.8129 \cdot (\epsilon/479961.68) \\
 & + 0.05006 \cdot \tanh((q_{Re}/0.67)) \cdot (\epsilon/479961.68) \\
 & - 0.009438 \cdot (I_s/0.24) \cdot (I_s/0.24)^2 \cdot (\epsilon/479961.68) \\
 & + 0.2535 \cdot \sqrt{|(IV_s/0.017)|} \cdot (\epsilon/479961.68)] \cdot 39143.24
 \end{aligned}$$

An initial check on the 'correctness' of the regressed models is to generate scatter plots. Below in Figure 4.37 the scatter plots for both models from Equation 4.3 are displayed. These plots show the prediction of the model versus the target value, for a model with a perfect fit all points would line up with the black diagonal line. To generate the scatter plots the propagation field from Section 4.3 has been used to generate the features. Next to that, all cells within the boxed domain described in Chapter 3 are plotted.

Even though the models use different primary features, the scatter plots are nearly identical. That is except for the upper right corner where $M9$ seems to fit the target slightly better. However, as can be seen, both models fail to predict the negative values of R which is likely due to the vast amount of cells with positive values of R . Therefore, a natural bias to only predict positive values is created. Next to that, in the region Prediction R $[500, 1000] \cdot 10^3$, there seems to be a mismatch visible as well. But overall the majority of the points lie in close proximity of the diagonal indicating a relatively good fit. The latter was also indicated by the R^2 -score of both models.

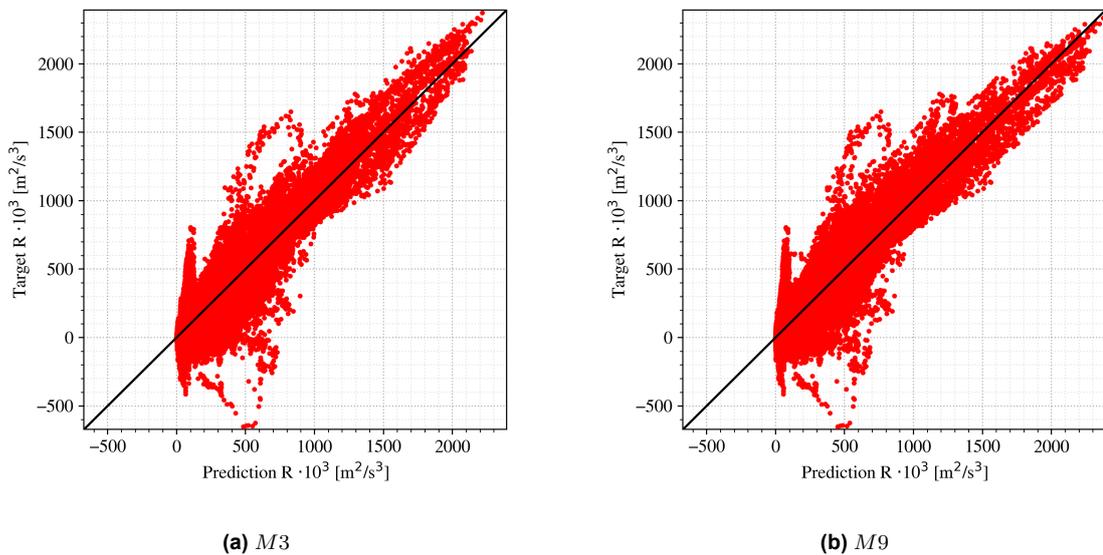


Figure 4.37: Scatter plots of two different R -models using all cells inside the boxed domain.

Similar to the scatter plots, the resulting R correction fields for both models can be plotted as well. For model 3 this is displayed in Figure 4.38. This figure shows R in the symmetry

plane, note again that the propagation fields are used to generate R . From this figure the observation from the scatter plot is confirmed, the model indeed fails to predict any negative values of R . Yet the locations of the corrections appear to be in the right place, namely at the HSV core location and in the corner region near the corner vortex. The structure is, however, far from the target. This could be expected since the target field for R had many complex structures, it can therefore not be expected, and arguably unwanted, that a regressed model with relatively low complexity can represent such shapes.

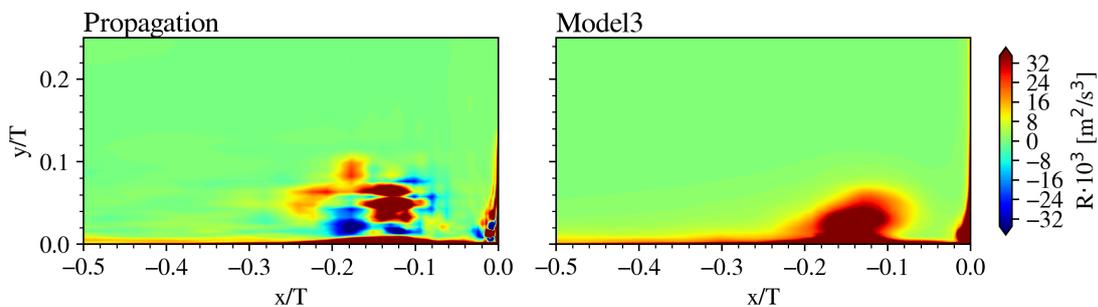


Figure 4.38: Comparison of Model3 with the target (Frozen data) in the symmetry plane.

In Figure 4.39 two y - z planes are shown downstream of the leading edge. The same observations as made in the symmetry plane hold. Namely the location and magnitude are similar to the target data, yet the model fails to predict the complex shapes and negative spots. Next to that, the model also predicts a significant correction at the wing wall, which is not present in the propagation data due to the spatial blending functions to filter the LSB (see Section 4.3). And even though the regression was trained on the propagation data it still predicts a correction for the LSB, this means that the spatial blending also has to be applied in the model propagation.

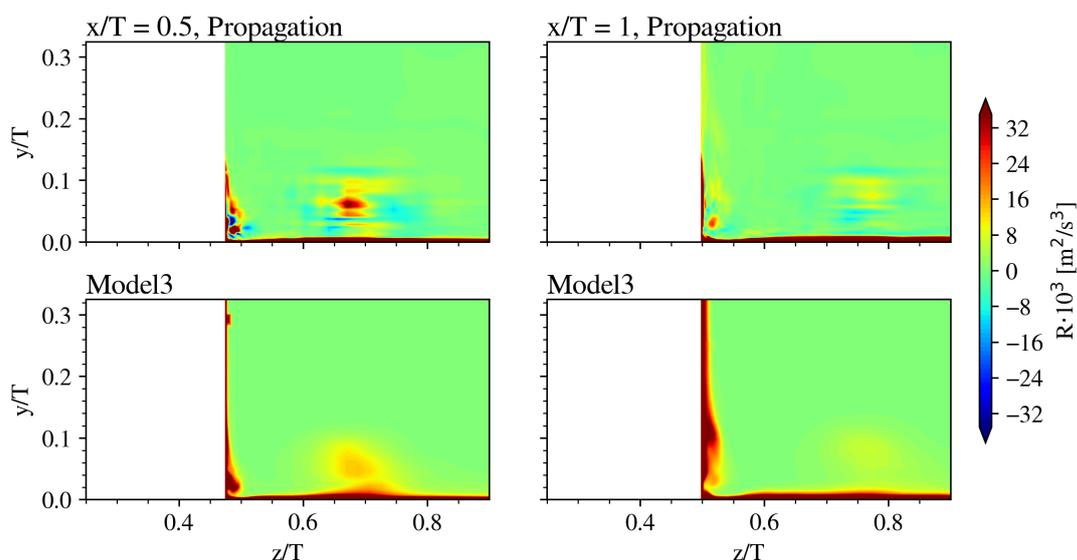


Figure 4.39: Comparison of Model3 with the target in two y - z planes.

For model 9 the flowfields are displayed in Figure 4.40 and Figure 4.41. The same discov-

eries as made for model 3 are present when comparing model 9 with the target. However, in contradiction to what is seen in the scatter plots, there exists a noticeable difference between the fields generated by model 3 and the fields generated by model 9. This difference is present in the region connecting the wall region and the HSV core location, model 9 shows a more distinct disconnection between the two regions.

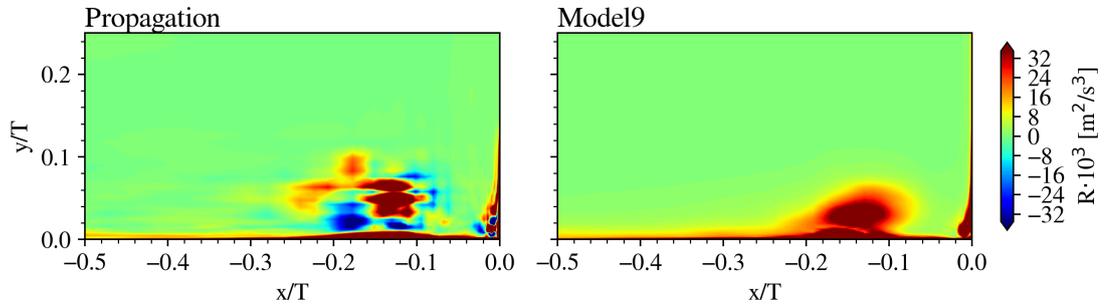


Figure 4.40: Comparison of Model9 with the target in the symmetry plane.

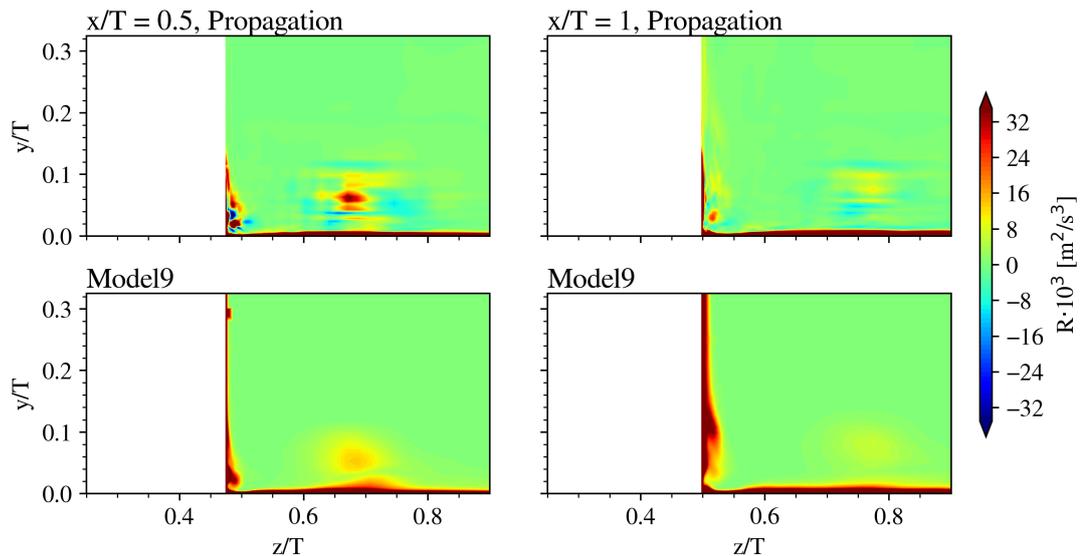


Figure 4.41: Comparison of Model3 with the target in two y - z planes.

To better visualise the difference in the peak magnitude of R between the target and the models in the symmetry plane, two lines have been extracted in this plane and displayed in Figure 4.42. As can be seen very close to the bottom wall the value of the models and the target are very similar. Yet for $y/T \gtrsim 0.02$ both models show a more moderate magnitude compared to the target. Next to that, this figure again highlights the fact that the models are not capable of reproducing the negative values of R , as was seen in the scatter plots. Although the models do not show the correct peak magnitudes, they do produce a smoother distribution of R . The latter is likely to reduce the accuracy of the flow solution in an *a posteriori* simulation, however, it will enhance the stability of the simulation. This increased stability is preferable as long as the loss in accuracy is not too big.

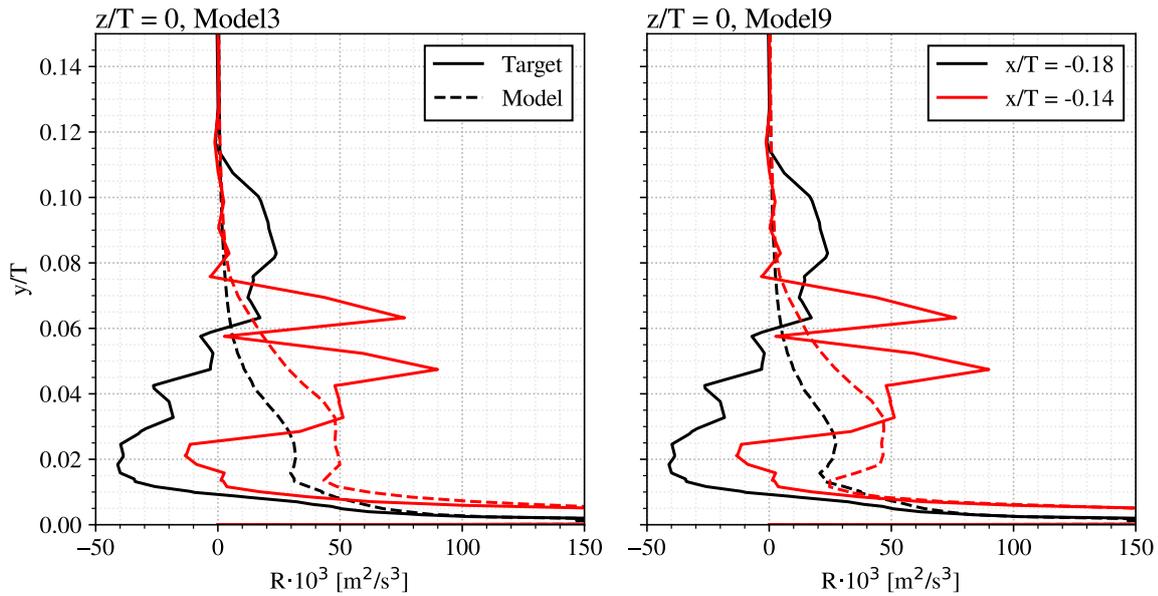


Figure 4.42: Profiles of R for both the target and the models in the symmetry plane at two locations upstream of the wing.

In the analysis of k -corrective frozen results, it was found that the value of R significantly far away from the wing goes to zero in the regions close to the wall. Therefore, to check whether this behaviour is also present in the fields generated by the models, Figure 4.17 has been recreated in Figure 4.43 including the profiles as determined by the models using the propagation field. And indeed both model 3 (left) and model 9 (right) show a similar trend of R going to zero when moving further outboard at $x/T = 1$.

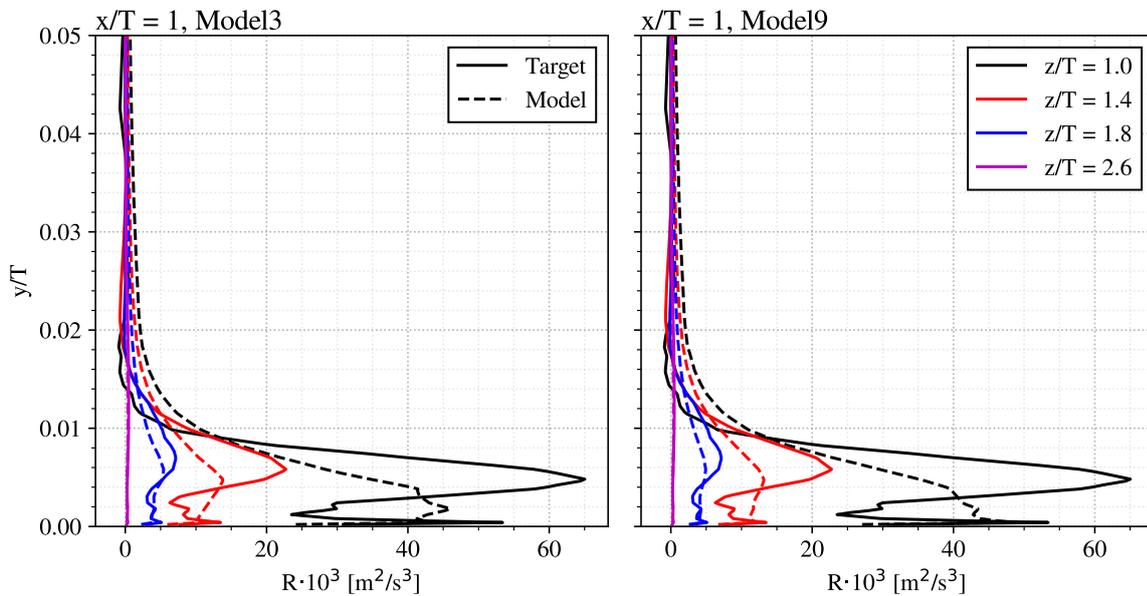


Figure 4.43: Profiles of R for both the target and the models at $x/T = 1$ at multiple outboard z locations.

Although there appears to be a very slight difference in the flowfields generated by model

3 versus model 9, it can be concluded that these models virtually generate equal corrections. However, judging from the MSE and the R^2 -score model 3 would be the better option due to the lower MSE and higher R^2 -score. Yet these improvements are very small.

An important analysis point not discussed in this subsection is, however, to test the generalisability of the models. Since the scatter and flow field plots are generated using the data on which the models have been trained, i.e. the propagation data. For a more complete comparison, the models should also be tested on a different test case, for example, a significantly different Reynolds number or a slight alteration of the geometry. As indicated in the flowchart in Chapter 3 the plan is to test generalisability *a posteriori* using the anti-fairing geometry. However, to decide which model to use in this *a posteriori* testing, it would be recommended to perform an *a priori* generalisability test as well to allow for a more complete comparison of the regressed models. This is beyond the scope of this work.

4.4.2. Turbulence Anisotropy Correction

Similar to the regression of R , for the regression of b_{ij}^{Δ} first a subset of the secondary variables should be established. This is because using all secondary features would require a too high amount of computational memory. This is especially true for the b_{ij}^{Δ} regression since it is a symmetric tensor, thus for each primary feature for each cell in the mesh, 6 components need to be stored instead of 1. As explained in Chapter 3 mutual information is not used to determine the subset due to the fact that b_{ij}^{Δ} is not a scalar and the secondary features are. Therefore, a regression with library degree 1 is performed using all secondary features. From this regression, the prominent features are extracted for the final 2-degree run. This resulted in the following (sub)set of secondary features:

$$\text{Secondary features} = [I, II, I_s, II_s, q_{\gamma}, q_{\nu}, q_Q, q_{Re}] \quad (4.4)$$

Using this set combined with Pope's tensor basis as the primary features resulted in the regression displayed by the coefficient matrix in Figure 4.44. Similar to R , two regressions were tested, one for which the features were scaled using the entire domain, and one for which the features were scaled using the boxed domain. Yet this did not result in a very different result with respect to which features were present, the coefficients were different, as expected, because the scaling was different. The figure below shows the result for which the features are scaled using the entire domain.

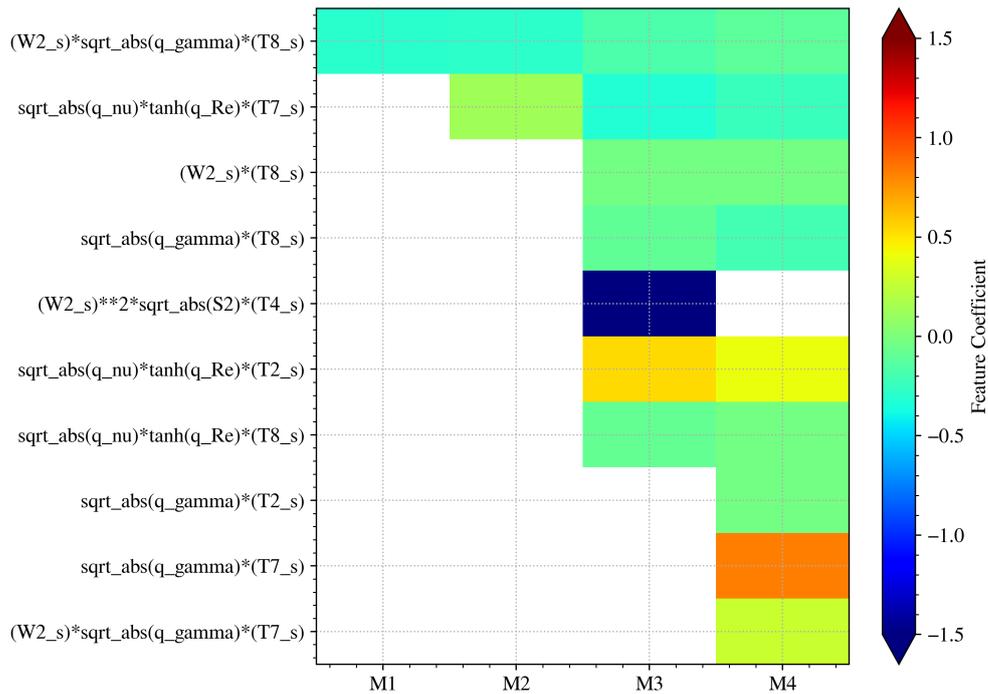


Figure 4.44: Feature coefficient matrix for b_{ij}^{Δ} .

To assess the fit of the regressed models to the target the MSE and R^2 -score are displayed below in Figure 4.45. Especially from the R^2 -score it can be seen that the fit for b_{ij}^{Δ} even for the models with higher complexity is poor.

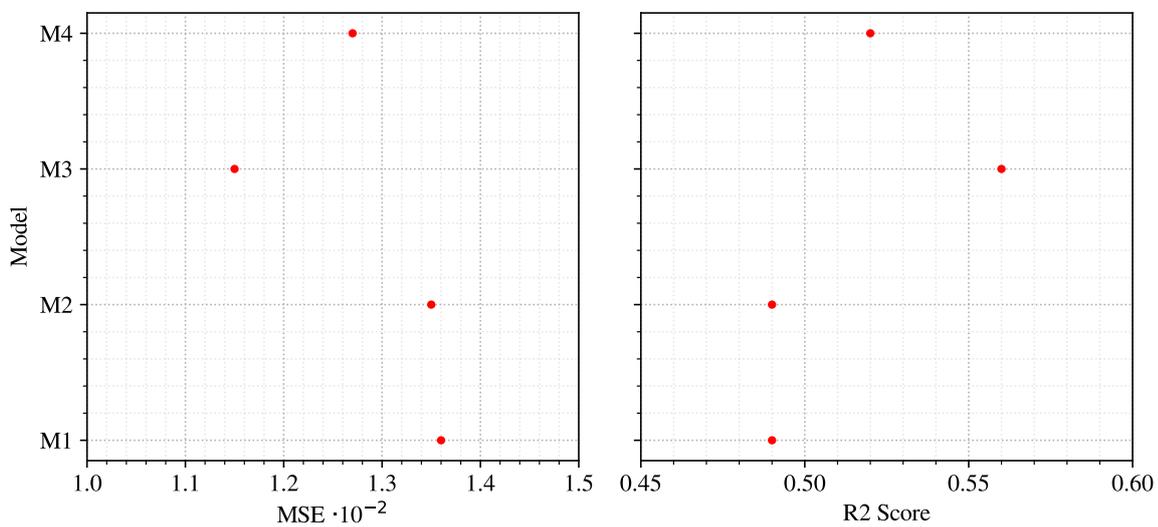


Figure 4.45: Mean squared error and R^2 -score of the regressed b_{ij}^{Δ} models.

This poor fit is better visualised using the scatter plots to display each component of the regressed model. The scatter plots for model 1 ($M1$) and model 3 ($M3$) are displayed in Figure 4.46 and Figure 4.47 respectively. These scatter plots were generated using the propagation data on which the models have been trained. It can be seen in both figures that the

spread of the model prediction is significant. And, therefore, the models do not represent the target. Even for model 3, which has a 14% better R^2 -score compared to model 1 the fit is still inaccurate. The complete equations for model 1 and model 3 are given below (including scaling).

$$\begin{aligned}
 M1_{b_{ij}^{\Delta}} &= -0.2973 \cdot (II_s/0.24) \cdot \sqrt{|q_{\gamma}/12.73|} \cdot (T_s^{(8)}/0.054) \\
 M3_{b_{ij}^{\Delta}} &= -0.03384 \cdot (II_s/0.24) \cdot (T_s^{(8)}/0.054) \\
 &\quad - 0.1003 \cdot \sqrt{|q_{\gamma}/12.73|} \cdot (T_s^{(8)}/0.054) \\
 &\quad - 10.75 \cdot (II_s/0.24)^2 \cdot \sqrt{|I/246615.19|} \cdot (T_s^{(4)}/0.061) \\
 &\quad - 0.1692 \cdot (II_s/0.24) \cdot \sqrt{|q_{\gamma}/12.73|} \cdot (T_s^{(8)}/0.054) \\
 &\quad + 0.5298 \cdot \sqrt{|q_{\nu}/2.55|} \cdot \tanh(q_{Re}/0.67) \cdot (T_s^{(2)}/0.21) \\
 &\quad - 0.3185 \cdot \sqrt{|q_{\nu}/2.55|} \cdot \tanh(q_{Re}/0.67) \cdot (T_s^{(7)}/0.051) \\
 &\quad - 0.08877 \cdot \sqrt{|q_{\nu}/2.55|} \cdot \tanh(q_{Re}/0.67) \cdot (T_s^{(8)}/0.054)
 \end{aligned} \tag{4.5}$$

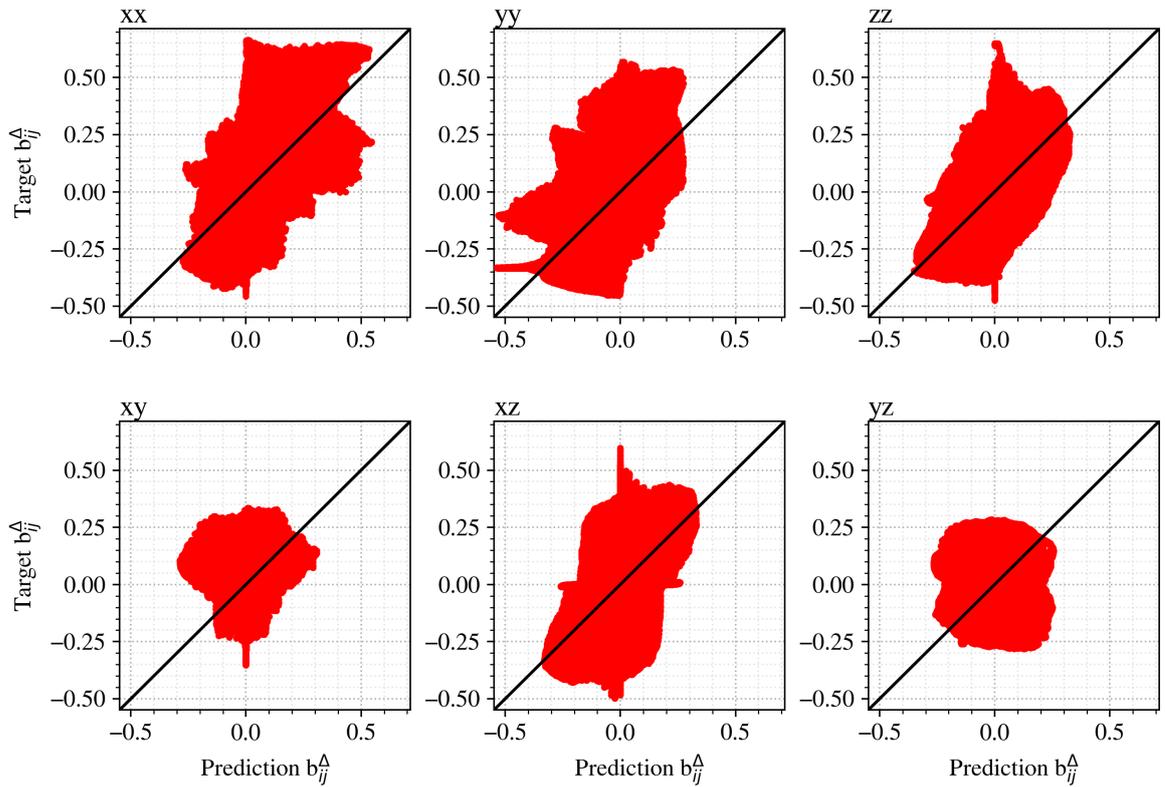


Figure 4.46: Scatter plots of b_{ij}^{Δ} Model1.

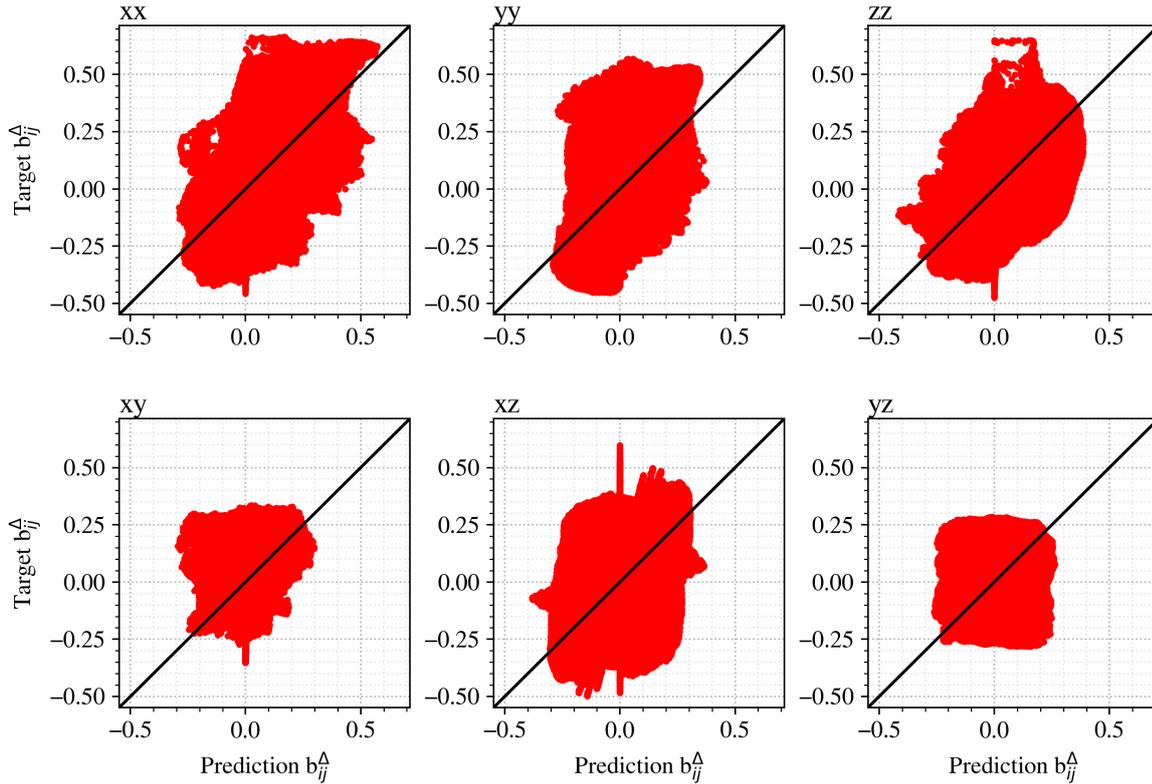


Figure 4.47: Scatter plots of b_{ij}^{Δ} Model3.

Unfortunately, in *a posteriori* testing model 3 did not produce a converged solution. Therefore, in this section, only the flow fields of model 1 are displayed. This model has a similar MSE and R^2 -score as model 2 yet with less complexity. Therefore there is no need to analyse model 2 as well. Model 4 is omitted from the discussion due to its relatively high complexity.

First, the normal stresses in the symmetry plane of model 1 are compared to the target in Figure 4.48, the plots of model 1 are created using the propagation as input. From this figure, it is observed that the model is only properly predicting the correction close to the bottom wall for $x/T < -0.2$. All other corrections are not captured by the model, for example, the correction at the HSV in the yy component is of the opposite magnitude. Similarly, the xx component of the model shows a correction at the HSV which is not present in the target. Finally, the model does also not determine the zz correction at the place where flow is ejected under the HSV. These flow fields, thus, confirm the observation made in the scatter plots.

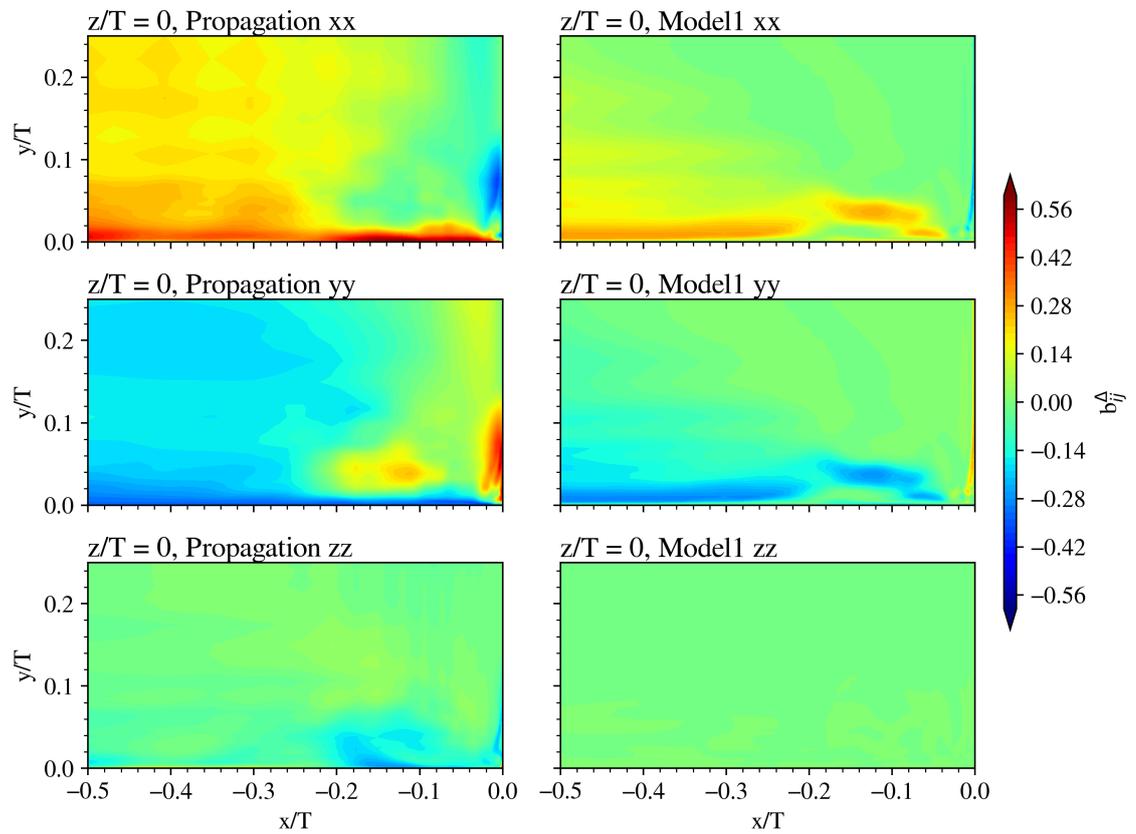


Figure 4.48: Comparison of Model1 with the target for the diagonal components in the symmetry plane.

The shear stresses in the symmetry plane are displayed in Figure 4.49. A positive note from this figure is that for the xz and yz components, for which hardly any correction is required, the model does not predict any correction either. Unfortunately, this cannot be said for the xy component, for which the model is predicting a correction that is not present in the target.

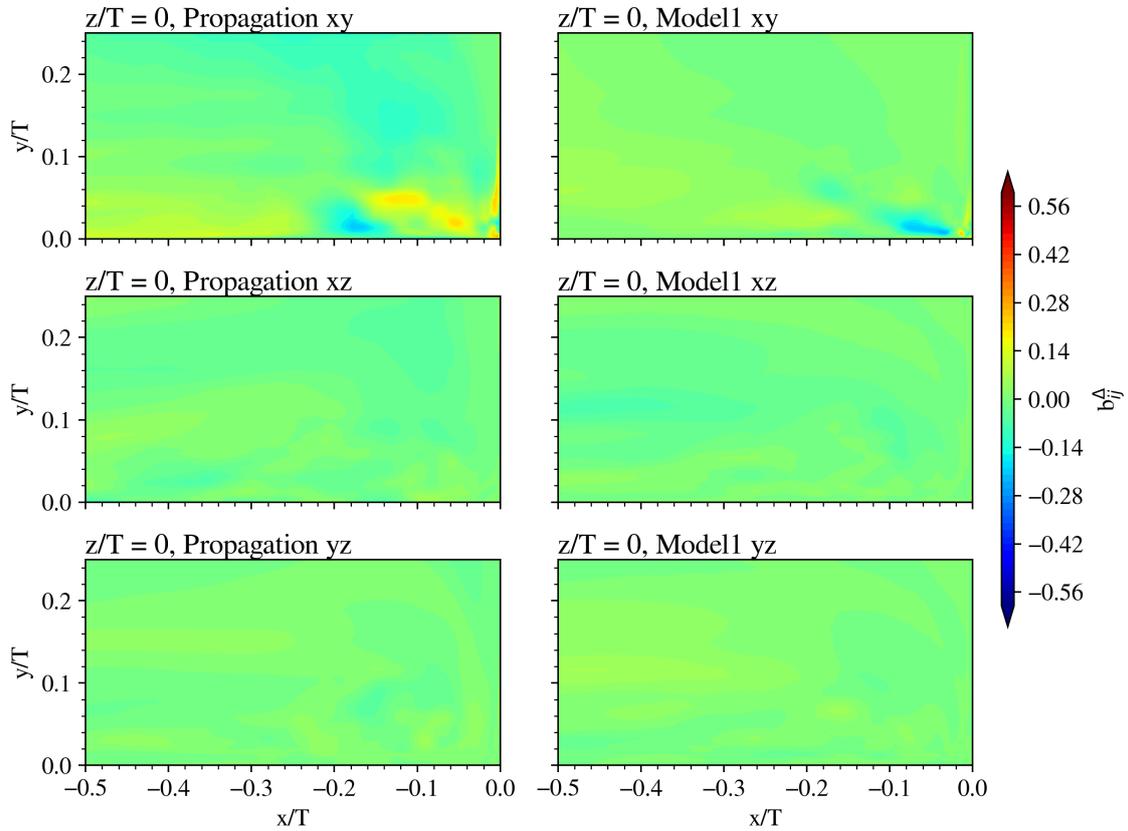


Figure 4.49: Comparison of Model1 with the target for the off-diagonal components in the symmetry plane.

Figure 4.50 and Figure 4.51 show the normal-stress and shear-stress corrections in the y - z plane at $x/T = 0.5$ for both the target and model 1. From this comparison, an equal conclusion is drawn as for the symmetry plane. That is the near wall correction away from the HSV is somewhat correct. However, any correction near the HSV and corner vortices are incorrect or not even present.

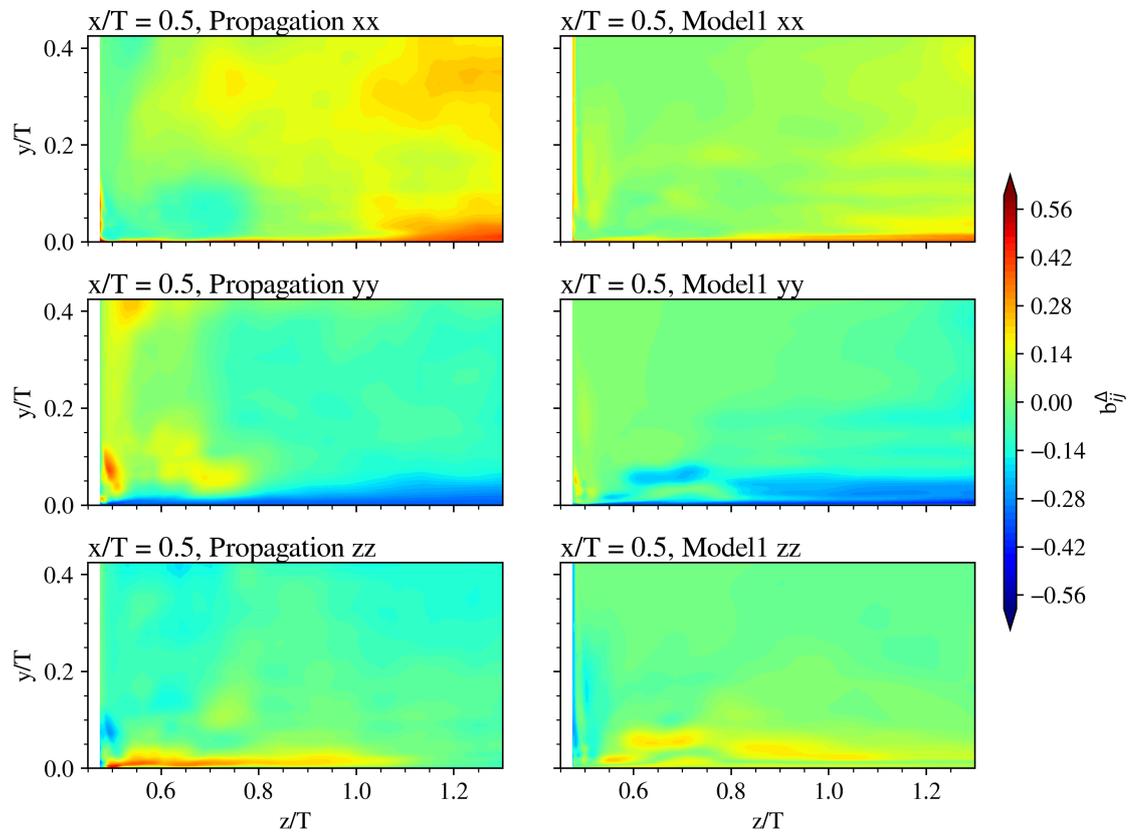


Figure 4.50: Comparison of Model1 with the target for the diagonal components at $x/T = 0.5$.

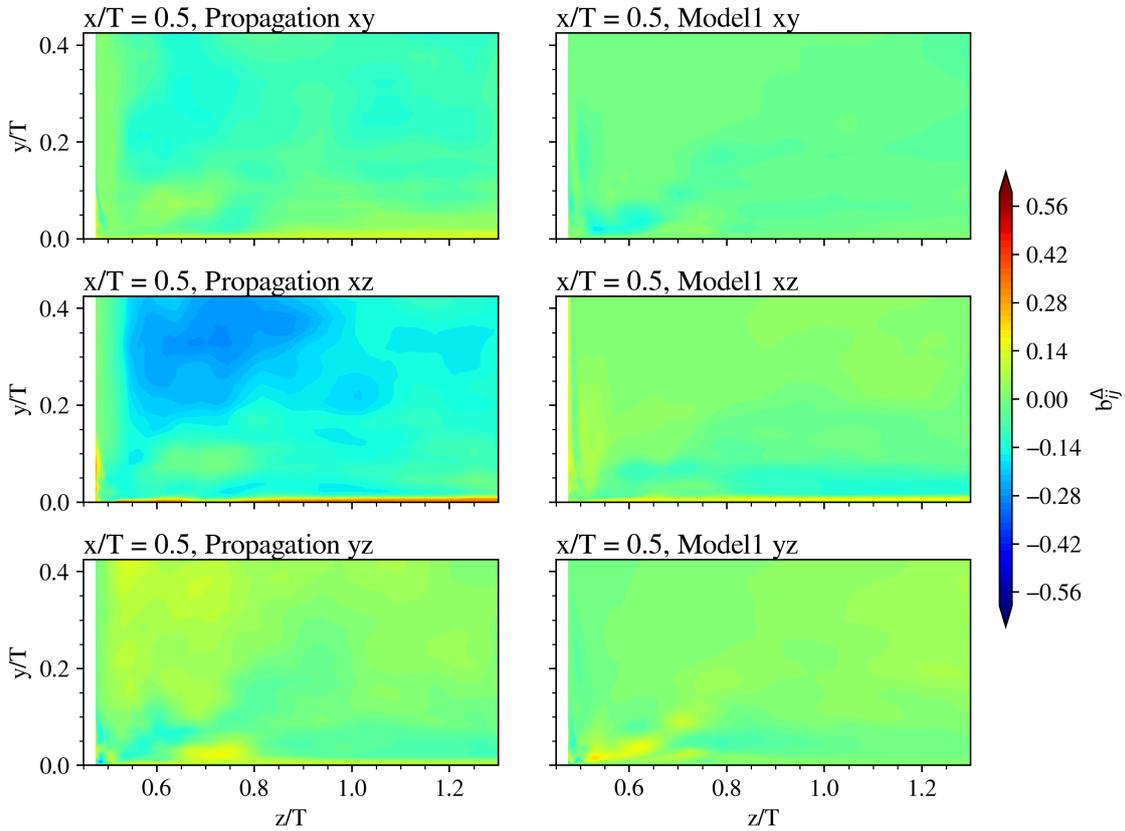


Figure 4.51: Comparison of Model1 with the target for the off-diagonal components at $x/T = 0.5$.

A close-up of the corner region of Figure 4.50 and Figure 4.51 is given in Figure 4.52 and Figure 4.53 respectively. As described in the frozen approach results, it is important to correct the anisotropy of the turbulence in the corner region to predict the correct secondary vortices. Yet unfortunately, due to the poor fit of b_{ij}^{Δ} the anisotropy corrections are not captured well by model 1. Therefore, as will be analysed in Section 4.5, the model propagation is not expected to have the correct vortex topology in the corner region. It should be noted, however, that the corrections close to the walls of the xx and xz components are captured quite well by model 1. The latter was, especially for xx , not easy to observe in Figure 4.50 and Figure 4.51.

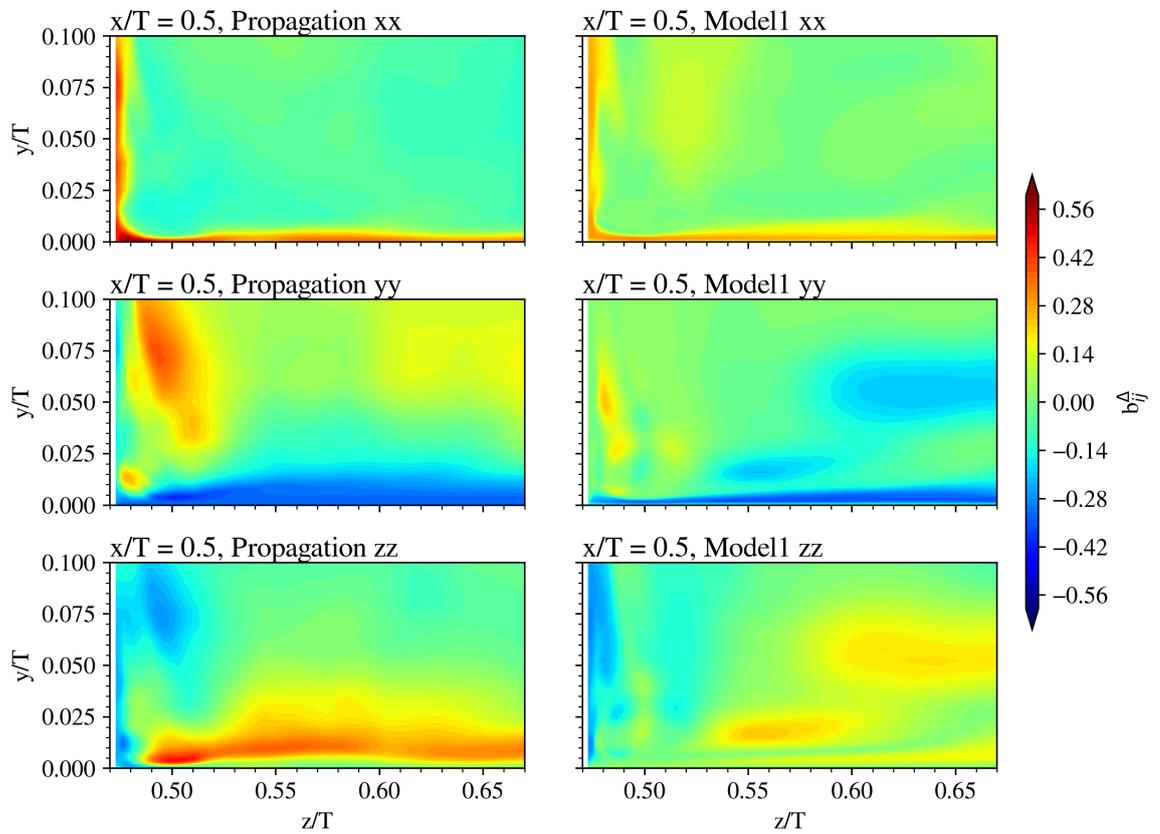


Figure 4.52: Comparison of Model1 with the target for the diagonal components at $x/T = 0.5$ in the corner region.

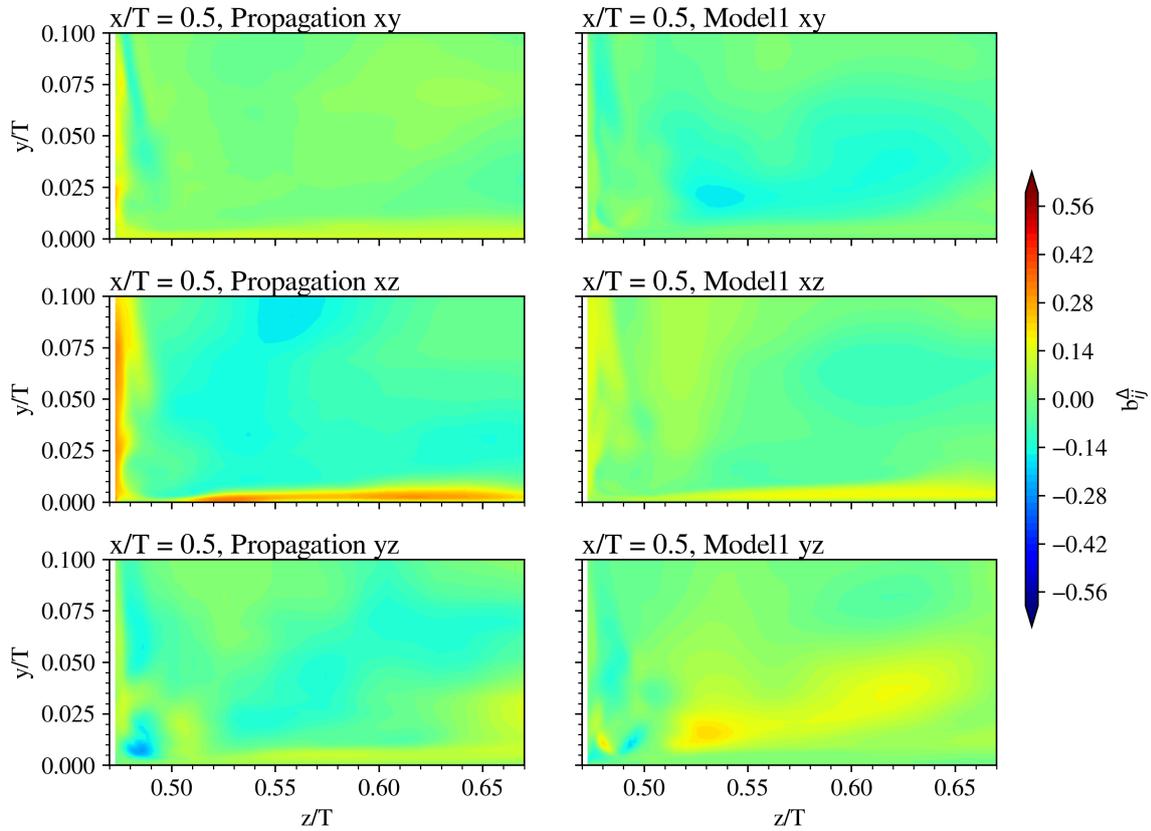


Figure 4.53: Comparison of Model1 with the target for the off-diagonal components at $x/T = 0.5$ in the corner region.

Finally, it is important to also check the flow fields of P_k^Δ which is a function of b_{ij}^Δ . This is because it is shown in Section 4.2 that the significance of P_k^Δ with respect to R cannot be neglected. Yet this poses a big problem for model 1 because it is only based on $T_s^{(8)}$, and since:

$$T_{s,ij}^{(8)} \frac{\partial U_i}{\partial x_j} = 0 \quad \text{for incompressible flows} \quad (4.6)$$

the value of P_k^Δ by model 1 throughout the entire flowfield is zero. Which is not representative of the target P_k^Δ at all (see Figure 4.13). The consequence of this will be displayed and discussed in Section 4.5. Note that the same holds for model 2 or 4 as Equation 4.6 also applies to $T^{(2)}$, $T^{(5)}$ and $T^{(7)}$. As an ad-hoc solution, a third correction field was defined, namely P_k^Δ , for which a scalar model was regressed using the same method as for R . The quality of the regression was similar to the regression of R , yet in *a posteriori* testing the P_k^Δ model would destabilize the simulation. Therefore, this methodology was not further pursued. That is beside the fact that such a model would be required because the fit of b_{ij}^Δ is poor. So rather than trying to reduce the error of a poor fit by introducing a new model, it would be better to improve the original fit. Note that this method is also only possible because the double dot product of ∇U with $T^{(8)}$ (as well as $T^{(2)}$, $T^{(5)}$ and $T^{(7)}$) equal zero for incompressible flows and thus P_k^Δ by these tensors equal zero. Finally, it is also worth mentioning that even if P_k^Δ would have

been correct then the anisotropy of the Reynolds stress tensor is still off which will influence the results.

Improving the Fit

Naturally one tries to improve the fit of b_{ij}^{Δ} using various numerical tricks and methodology alterations. Unfortunately, in the time frame of this Master thesis, an improved (and stable) fit has not been achieved. However, for the sake of increasing the common knowledge regarding the regression of anisotropy corrections, some noteworthy attempts are listed below:

- Non-linear curve fitting; as one might have noticed model 2 consists of a hyperbolic tangent function, by redefining this function to $\tanh(a \cdot feat_i + b)$ and fitting the a and b coefficients in the model using the non-linear curve fitting method from the Scipy library [50] an (unsuccessful) effort was made to improve the fit of model 2. The idea behind this method was that it would allow for a more complex equation for the regression and that it, therefore, would increase the R^2 -score.
- Using the QCR model as an extra primary feature; due to the success of the non-linear QCR correction for the SA turbulence model (see Chapter 2), an attempt was made to include the QCR correction (Equation 2.4) in the list of primary features for b_{ij}^{Δ} . Although it did show up in the regressed models, the fit was not better compared to regressions without the QCR correction as a feature.
- Using a different time-scale for Ω_{ij}^* ; Ω_{ij}^* in the standard version of SpaRTA is either non-dimensionalised by $1/\omega$ or by $1/\|\nabla U\|$, yet it can also be non-dimensionalised using $1/\|\Omega_{ij}\|$, as done by Wu et al. [56]. This did, however, also not improve the fit, which could be expected due to the similarity of $1/\|\nabla U\|$ and $1/\|\Omega_{ij}\|$.
- Using extra non-linear functions; inspired by the complexity of the EARSMs by [35] and [36] extra non-linear functions were added to the library generation step such as $\exp(x)$, $\log(x)$ and $1/x$. Unfortunately, this would often, as can be expected, result in models that are very unstable in *a posteriori* testing due to the asymptotic nature of these functions.

So the question that remains is whether it is possible to regress the anisotropy correction imposed by the junction flow using the current SpaRTA methodology. This question is asked with respect to the junction flow since it is shown that for other, so to say, more simple flows the methodology seems to provide satisfactory results (Schmelzer et al. [39], Huijing [40]). One possible reason for the poor performance could be the fact that in order to regress the correction tensor it is flattened to a vector of six components per cell. This has the consequence that the regression algorithm is trying to match each individual component of the tensor rather than for example the directionality or magnitude of the complete tensor. For the junction flow it could be interesting to see if one can focus the regression on the important Reynolds stresses in the vorticity equation rather than the complete tensor. That is to try to regress $(\overline{v'v'} - \overline{w'w'})$ and $\overline{v'w'}$ since these play an important role in the vorticity equation as defined in the x -direction using the reference frame as described in this work (Equation 2.3). The latter is important since focusing on a few components of interest in a defined reference frame does reduce the generalisability of the model. Yet for the sake of design optimisation, in which the reference frame is not altered from design iteration to design iteration, this does not pose a problem.

Another idea is to increase the complexity of the elements generated by the library generator. Since, at the time of writing, the SpaRTA library generation code is only able to generate elements such as:

$$f(s_i) \cdot g(s_j) \cdot h(s_k) \cdot \dots \cdot p_l \quad (4.7)$$

In which f, g and h represent (non-)linear functions and s_i, p_i any secondary and primary features. It would be interesting to see if combining features within one (non-)linear function

can improve the regression results:

$$f(s_i, s_j, \dots) \cdot g(s_k, \dots) \cdot \dots \cdot p_l \quad (4.8)$$

As well as adding, for example, constants within the functions. Yet it is not guaranteed that increasing the complexity of the features or library elements will improve the fit. It is a possibility that it will result in unstable models when testing *a posteriori*. However, some promising work utilising a more complicated regression algorithm to fit b_{ij}^Δ has been published by Hoefnagel [61].

4.5. Model Propagation

In this section, the models found in the regression are propagated into the CFD solver to correct the turbulent kinetic energy via a model for R (M_R) and the Reynolds stress tensor via a model for b_{ij}^Δ ($M_{b_{ij}^\Delta}$). The two models that are propagated are displayed below:

$$\begin{aligned} M_R = & \left(0.6505 \cdot (\epsilon/1198213.34) + 0.2751 \cdot \sqrt{|(IV_s/0.026)|} \cdot (\epsilon/1198213.34) \right. \\ & + 0.009679 \cdot (q_{Re}/0.73) \cdot \sqrt{|(IV_s/0.026)|} \cdot (G_s^{(1)}/312067.87) \\ & - 0.04692 \cdot (II_s/0.22) \cdot \sqrt{|(q_V/0.28)|} \cdot (\epsilon/1198213.34) \\ & \left. + 0.2163 \cdot \sqrt{|(IV_s/0.026)|} \cdot \tanh(q_{Re}/0.73) \cdot (G_s^{(1)}/312067.87) \right) \cdot 101329.75 \end{aligned} \quad (4.9)$$

$$M_{b_{ij}^\Delta} = \left(-0.2973 \cdot (II_s/0.24) \cdot \sqrt{|q_\gamma/12.73|} \cdot (T_s^{(8)}/0.054) \right) \cdot 1.0 \quad (4.10)$$

these models are the direct result of the regression in Section 4.4.

The model propagations displayed in this section are using the same geometry and flow conditions as used in the training setup. Therefore, an unseen prediction of the models given by Equation 4.9 and Equation 4.10 is missing in this work. The latter poses a problem regarding testing the generalisability of the models. Note that this is next to the also missing generalisability test in the *a priori* environment. A logical recommendation therefore would be to propagate the models on the anti-faring geometry (see Chapter 2) to check the generalisability. The latter would be a sufficient test in the view of design optimisation. In such an optimisation problem, the geometry would namely only alter slightly from one design iteration to another.

Although a generalisability test is missing, it is still important to check the effectiveness of the models on the original flow. To check this effectiveness the initial conditions of the model propagation, in fact also of the propagation, are not equal to the converged LES solution. Instead, only the turbulence fields are from the converged true solution while the mean fields (U and p) are set to zero except for the boundary conditions. A discussion on this particular initial condition setup is given in Section 4.6.

Each figure in the following discussion displays the results of four simulations, the four different simulations are listed below:

- the baseline *RANS* results
- the *propagation* results (Prop.)
- the *model propagation of R* results, which represents the simulation in which R is determined by M_R yet b_{ij}^Δ is equal to the field from the frozen approach (Section 4.2)
- the *model propagation* results, which show the results for which R is determined by M_R and b_{ij}^Δ by $M_{b_{ij}^\Delta}$, i.e. a complete model propagation

The idea behind the two different model propagations is to access the performance of M_R individually since it is shown in Section 4.4 that the fit of b_{ij}^Δ was lacking accuracy. Thus having a model propagation for which b_{ij}^Δ equals the exact target field allows to decouple the performance of M_R from the (likely) faulty performance of $M_{b_{ij}^\Delta}$, both in terms of anisotropy correction and turbulence production correction (P_k^Δ).

In order to achieve the best result the factors by which M_R and $M_{b_{ij}^\Delta}$ are incorporated into the solver were manually altered. The factors that showed the best qualitative flow fields were as follows:

- *model propagation of R*: $R = 0.75 \cdot M_R$ and $b_{ij}^\Delta = b_{ij}^\Delta$
- *model propagation*: $R = 0.5 \cdot M_R$ and $b_{ij}^\Delta = 1.0 \cdot M_{b_{ij}^\Delta}$

Below is Figure 4.54 the U -velocity profile upstream of the wing in the symmetry plane is displayed for the four aforementioned simulations. Note that the propagation resembles the truth (or target result), it has been shown that the propagation is very similar to the LES data. From Figure 4.54 it can be seen that both model propagations match the propagation in terms of profile and separation point fairly well. It is even observed that the model propagation outperforms the model propagation of R in matching the target for $y/T < 0.2$. Above 0.2 the model propagation shows a significant discrepancy in the most right plot. But overall both model propagations show a significant improvement with respect to the baseline RANS.

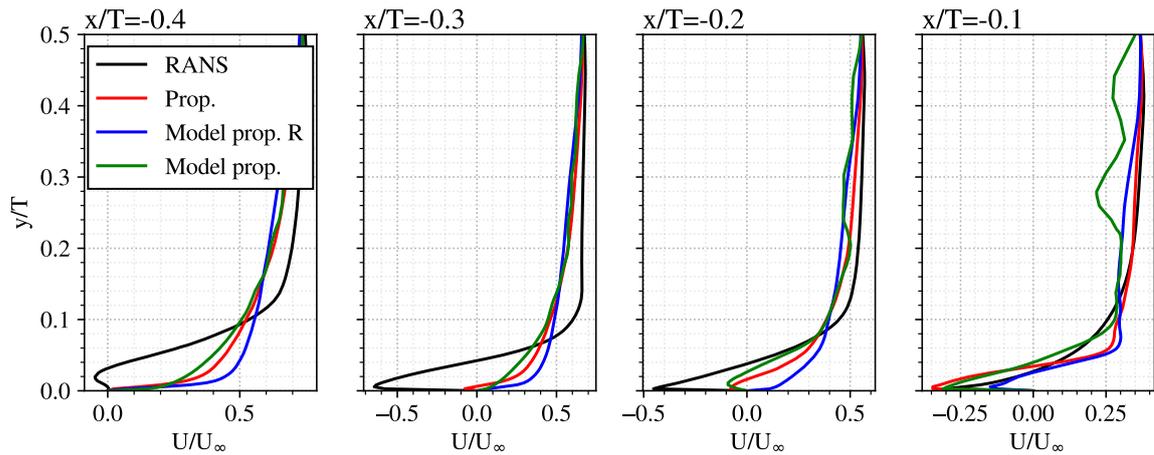


Figure 4.54: Boundary layer profiles upstream of the wing in the symmetry plane.

This improvement of the velocity field upstream of the wing by both model propagations is also displayed in Figure 4.55. In this figure the velocity magnitude as well as the velocity direction is displayed in the symmetry plane upstream of the wing. Not only is the prediction in the separation point and HSV location more accurate, but also the velocity magnitude further away from the HSV (upper left corner in the plots).

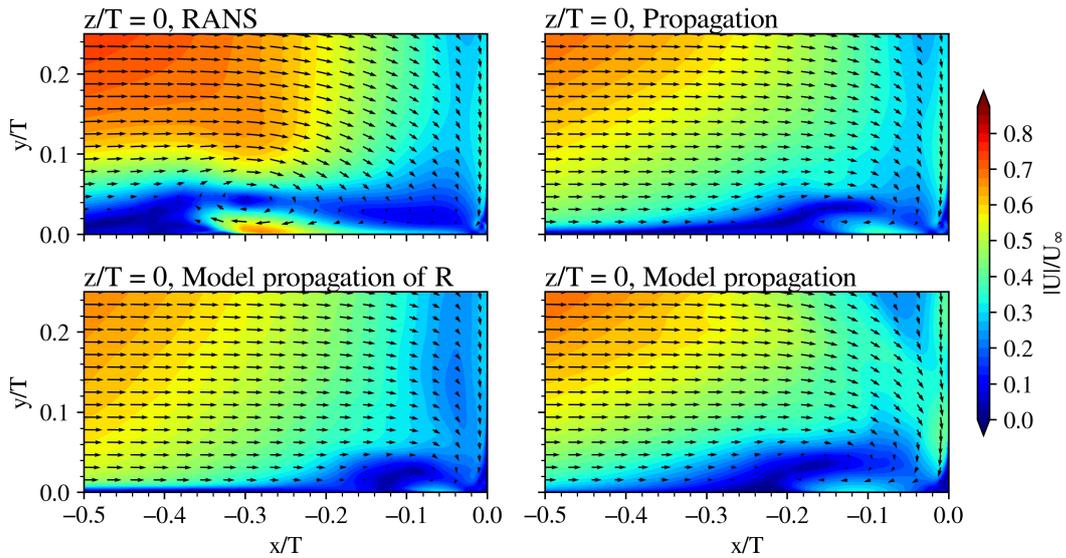


Figure 4.55: Velocity magnitude in the symmetry plane upstream of the wing for the different simulations.

To confirm the correct location of the HSV the vorticity in z -direction in the symmetry plane is displayed in Figure 4.56. In which, indeed, it is seen that both model propagations show an improved result in the position of the HSV. However, the shape of the HSV, especially in the model propagation of R , is dissimilar to the propagation result. It is likely that this is due to the fact that M_R was not able to represent all complex structures present in the target field of R (recall Figure 4.13). The shape of the HSV in the model propagation is slightly better but still shows discrepancies with respect to the propagation. The latter is surprising due to the fact that in the model propagation $P_k^\Delta = 0$, hence it was not expected to show a noticeable improvement with respect to the baseline RANS.

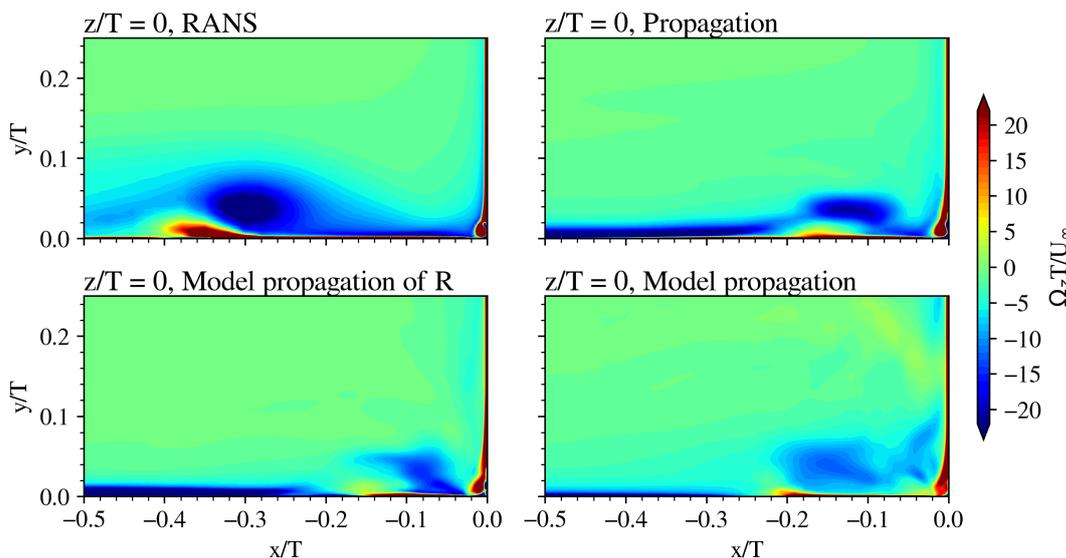


Figure 4.56: Vorticity in z -direction in the symmetry plane.

The turbulent kinetic energy in the symmetry plane is shown in Figure 4.57. This figure shows that both model propagations improve the location of the bulb of high k associated with

the HSV, however, the characteristic C-shape is not captured. Again, for the model propagation of R , this is likely due to M_R not being able to represent all complex structures that were present in the target data. For the complete model propagation this is a combination of the latter and the poor fit of $M_{b_{ij}^\Delta}$. However, a more concerning difference is found when observing the small area of high k in the corner region. Which is present in the model propagation of R yet not clearly present in the model propagation. This region, as will be shown, is also the region in which the largest discrepancy exists between the propagation and the model propagation. Because up to this point, the poor fit of $M_{b_{ij}^\Delta}$ is not really noticeable in the velocity and vorticity fields upstream of the wing, that is with respect to the results of the model propagation of R .

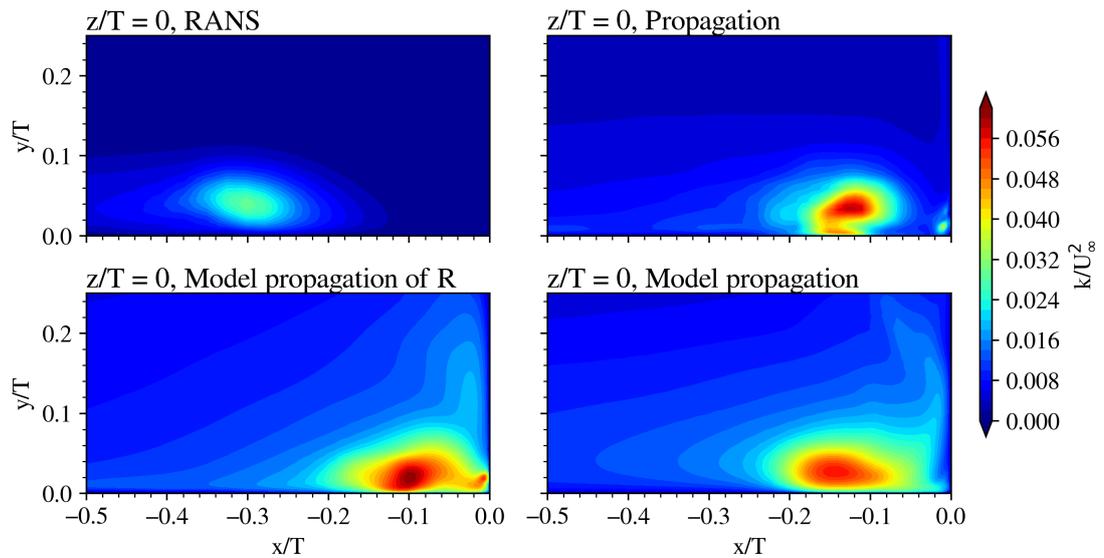


Figure 4.57: Turbulent kinetic energy in the symmetry plane for the different simulations.

However, when zooming in on the corner region it can be seen that the model propagation is not determining the corner vortex and the secondary vortices in an accurate manner. For example, below in Figure 4.58 the z -vorticity in the corner region in the symmetry plane is displayed. Note the change in colour scale from Figure 4.56 to Figure 4.58 to clarify the structures. Figure 4.58 shows that the model propagation is under-predicting the magnitude of the corner vortex significantly. The same holds for the negative vorticity structures surrounding the corner vortex. It is expected that this is due to the fact that P_k^Δ equals zero in the model propagation. This is because Section 4.2 showed that the value of P_k^Δ especially in the regions where the HSV and corner vortex live cannot be neglected. Therefore the fact that the model propagation shows fairly good results for the HSV upstream of the wing is surprising. It could, however, be that the error made in M_R cancels the effect of having no P_k^Δ .

Since the value of P_k^Δ is correct for the model propagation of R the resulting corner vortex is more accurate. However, compared to the baseline RANS it cannot be concluded that it is an improvement over the baseline RANS.

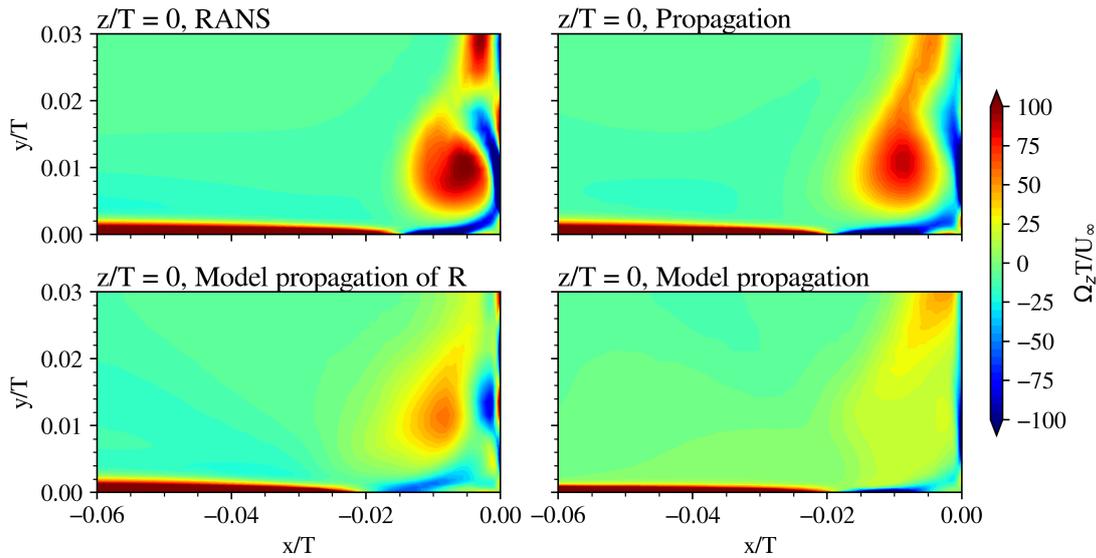


Figure 4.58: Close-up of the corner vortex in the symmetry plane, the vorticity displayed is in the z -direction.

To continue the analysis, the vorticity in the free-stream direction in the y - z planes at several locations downstream of the leading edge is discussed. Figure 4.59 shows the x -vorticity at $x/T = 0.5$, in contradiction to the symmetry plane the model propagation of R is predicting the shape and magnitude of the HSV better compared to the model propagation. It should be noted, however, that the location of the HSV in the model propagation is still significantly improved compared to the baseline RANS. Although it is noticed that the magnitude of the vorticity is lower compared to the propagation.

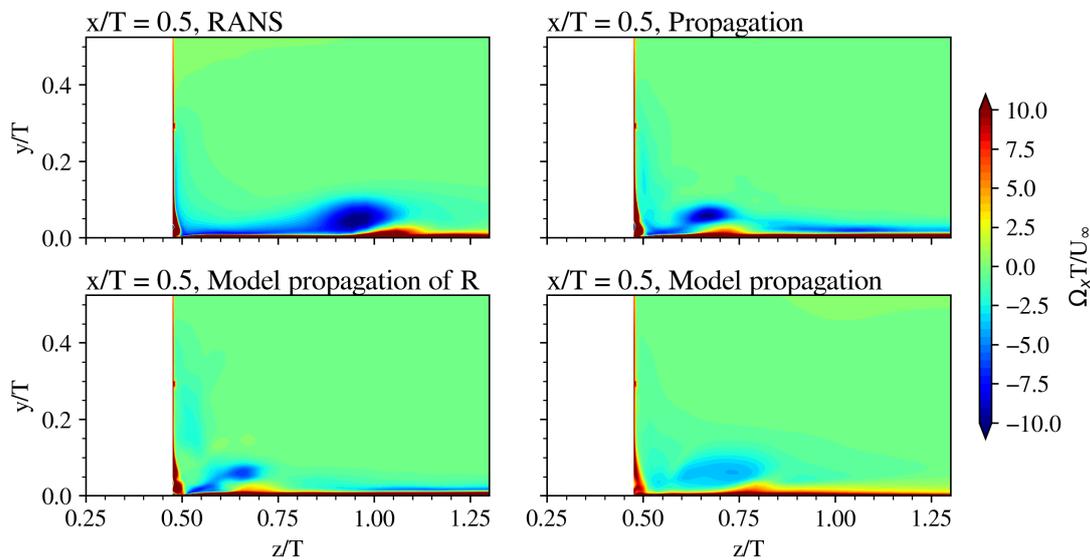


Figure 4.59: Vorticity in the x -direction in the y - z plane at $x/T = 0.5$.

Taking a closer look at the corner region, Figure 4.60, reveals that the model propagation is not predicting the magnitude of the corner vortex accurately, similar to Figure 4.58. Next to that, the negative vorticity structures close to the wing wall are not captured either by the model propagation. The model propagation of R predicts the vortex structures better when

comparing them to the propagation. Yet it cannot be left unnoticed that, in this particular plane, the baseline RANS result is very close to the propagation.

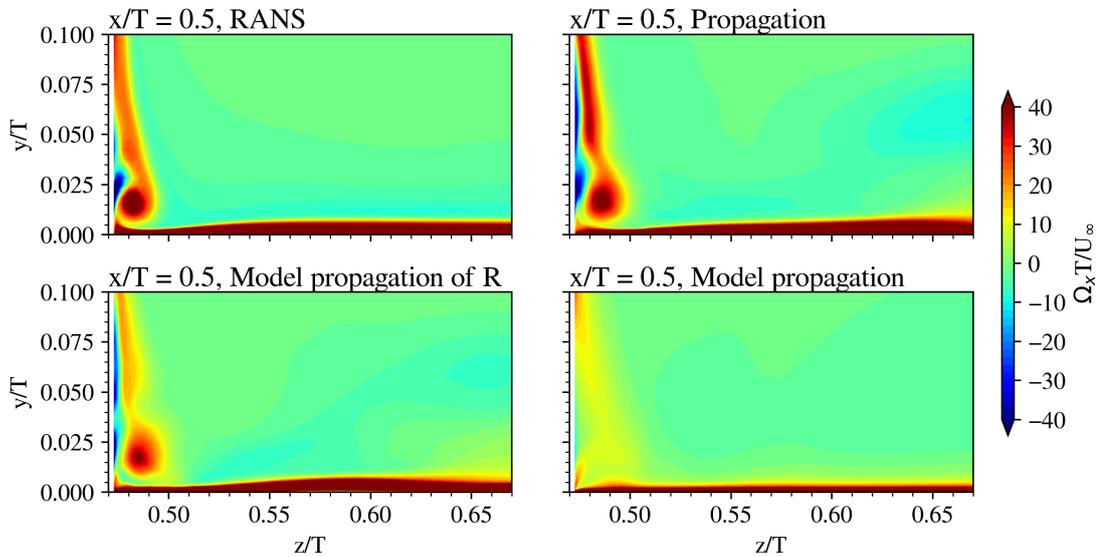


Figure 4.60: Close-up of the corner region displaying the vorticity in the x -direction in the y - z plane at $x/T = 0.5$.

Moving a half maximum wing thickness downstream results in Figure 4.61. Similarly to $x/T = 0.5$, the magnitude, shape and location of the HSV in the model propagation of R are quite similar to the propagation and therefore a significant improvement compared to the baseline RANS. Unfortunately, the model propagation is not determining the HSV well. Thus, even though upstream of the wing the location of the HSV by the model propagation is relatively good, when convecting downstream the result starts to differentiate from the propagation result.

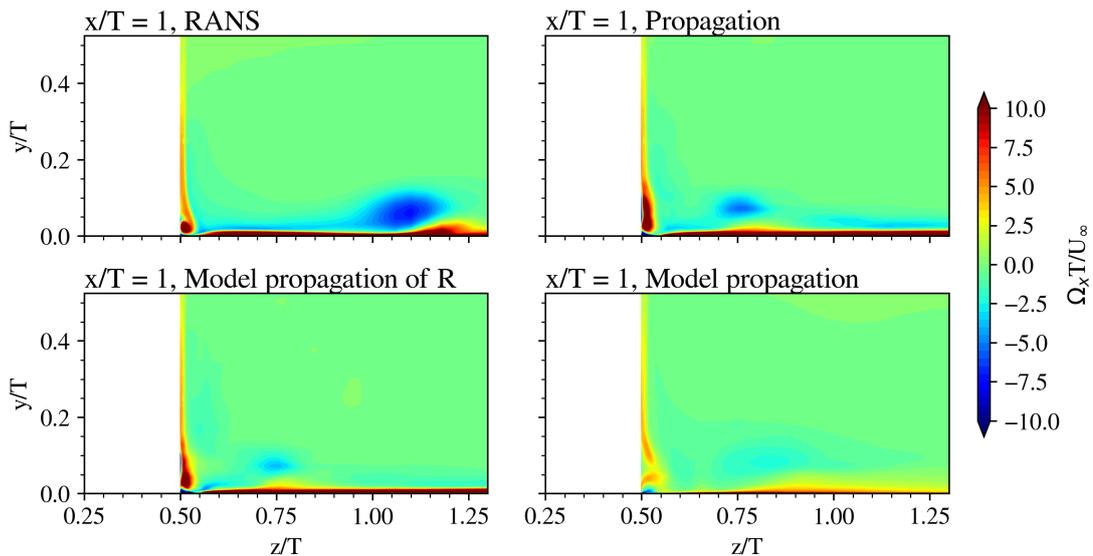


Figure 4.61: Vorticity in the x -direction in the y - z plane at $x/T = 1.0$.

Recall that in Section 4.1 it was discovered that a secondary vortex is present in the LES

data in the corner region of the y - z plane at $x/T = 1.0$ that was not present in the RANS solution. This was linked to the deficient turbulence anisotropy prediction by the baseline k - ω SST model. Therefore, one of the goals of the b_{ij}^Δ correction was to correct this anisotropy such that these secondary structures would be present in the RANS solution. However, as can be seen in Figure 4.62, the model prediction is not showing this negative vorticity structure near $z/T \approx 0.50$. Yet it is present in the model propagation of R , from which it, therefore, can be concluded that this is due to the poor fit of $M_{b_{ij}^\Delta}$.

In contradiction to the y - z plane at $x/T = 0.5$, improvements with respect to the baseline RANS of the model propagation of R are present in Figure 4.62. First of all, the corner vortex has an improved shape and magnitude. And secondly, the positive vorticity close to the bottom wall is also very similar to the propagation. The latter is, however, not only due to M_R since the value of P_k^Δ is also significant near the bottom wall (see Figure 4.15). As mentioned above the secondary vortex at $z/T \approx 0.50$ is due to the correct b_{ij}^Δ in that region and can therefore not be linked to M_R .

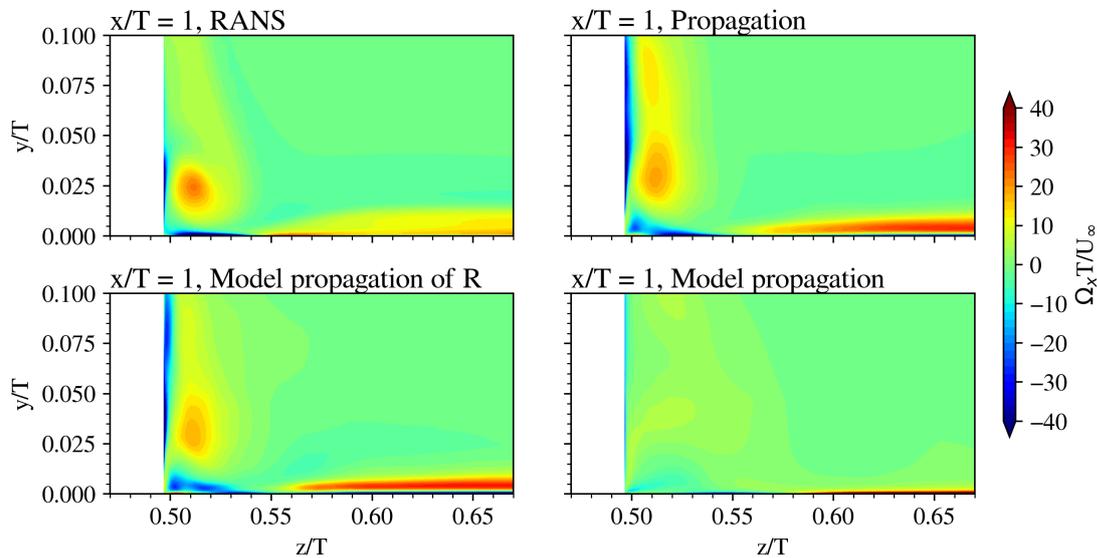


Figure 4.62: Close-up of the corner region displaying the vorticity in the x -direction in the y - z plane at $x/T = 1.0$.

Figure 4.63 displays the x -vorticity in the y - z plane at $x/T = 2$. Similar to the more upstream planes, the main improvement is present in the correct location of the HSV for the model propagation of R . The HSV in the model propagation has already dissipated while it is still slightly present in the propagation field. Next to that, it is again observed that the corner vortex is not present in the model propagation.

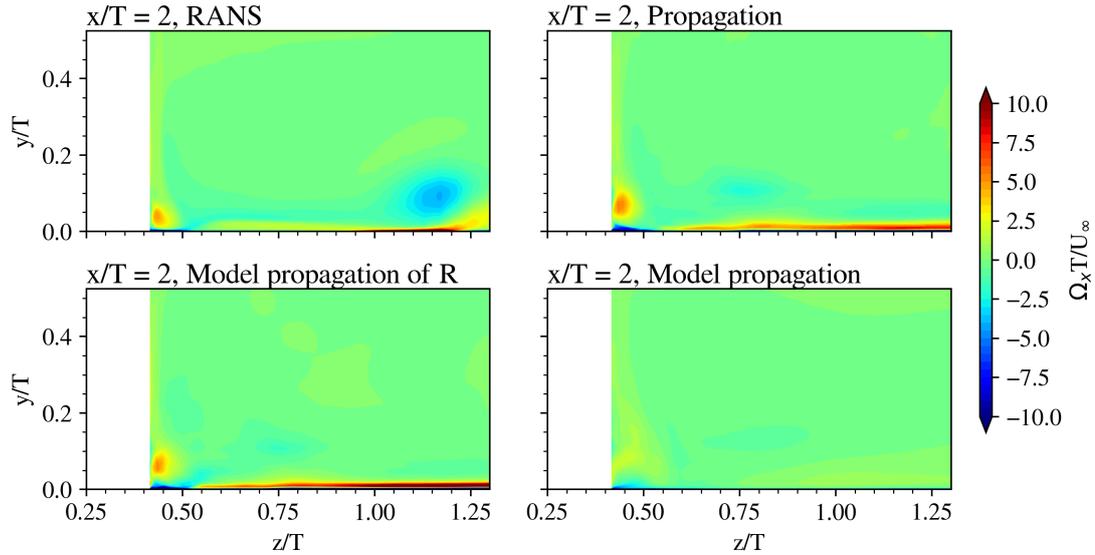


Figure 4.63: Vorticity in the x -direction in the y - z plane at $x/T = 2.0$.

To analyse the behaviour of the model propagations at the trailing edge the U -velocity fields are plotted at $x/T = 4$ in Figure 4.64. This figure basically confirms the behaviour seen in the vorticity plots. That is, first of all for the model propagation, the HSV has a smaller magnitude compared to the propagation when moving downstream. This causes the pull-down effect of higher velocity flow by the HSV towards the bottom wall to be less significant. Which has the effect that the velocity profile is less 'full' for the model propagation compared to the propagation as observed in Figure 4.64. Secondly, the significantly lower magnitude corner vortex and absence of secondary vortices (almost) cause flow separation in the corner for the model propagation which is less noticeable in the propagation data.

Regarding the model propagation of R , the resulting flow field is much closer to the propagation field. The only discrepancies found are close to the wing wall where the propagation predicts a lower velocity and at the bottom wall near $z/T = 0.63$. At this location, the propagation field shows a hump in the iso-lines which is not present in the model propagation of R . Yet compared to the baseline RANS the model propagation of R significantly improved the U flow field. This is also true, although to a lesser extent, for the model propagation.

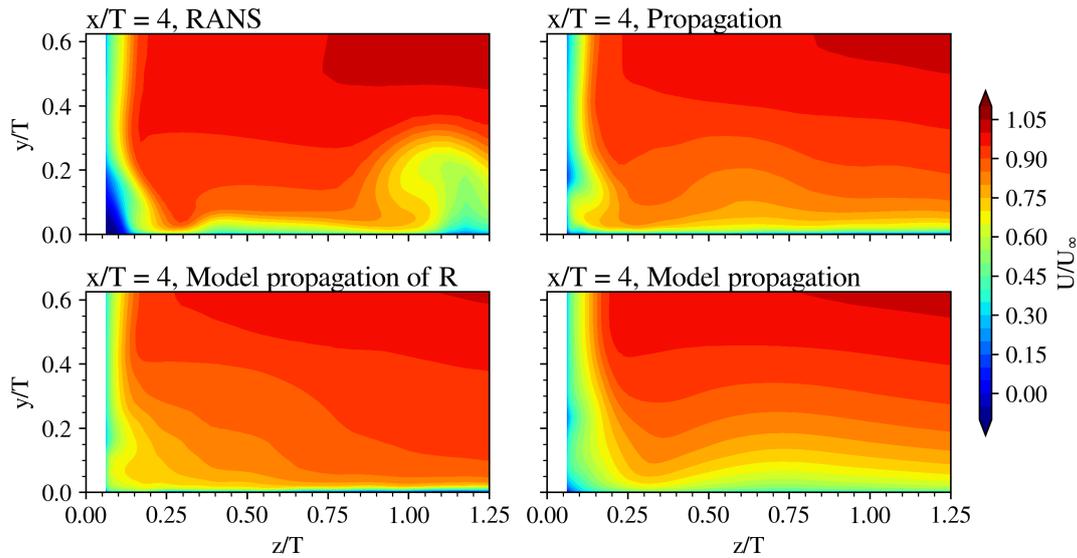


Figure 4.64: Velocity in x -direction in the y - z plane close to the trailing edge ($x/T = 4.0$).

The final part of the analysis for the model propagations is to check the aerodynamic forces on the wing. Figure 4.65 shows the friction coefficient of the wing in the corner region at two different heights (i.e. outside the spatial blending region). As already discussed in Section 4.1 the RANS results show a trailing edge separation (corner separation) which is not present in the LES and thus also not in the propagation. It is expected that this is due to the effect of the correct vortex topology around the wing as described in the literature in Chapter 2. Figure 4.65 shows that the model propagation of R is quite accurately following the trend of the propagation. This is due to the fact that the vortex system (HSV, corner vortex and secondary vortices) is determined correctly (to a certain extent) as shown in this section and hence it is expected that the wall shear stress in the corner region is correct. Regarding the model propagation, it is observed that apart from the friction peak at $x/c \approx 0.1$ the curve at $y/T = 0.01$ is well determined. For $y/T = 0.03$ the discrepancy is larger, yet the slope of the curve seems to be correct.

More surprisingly, the model propagation does not show a corner separation. This is surprising as it has been shown that the corner and secondary vortices are hardly present in the model propagation. The presence of these should suppress the corner separation. Hence with these vortices being absent, a separation would be expected. There are a couple of possible explanations for the not-existing corner separation in the model propagation. First of all, it could be that in this non-lifting wing configuration the effect of placing the HSV at the correct location, which is closer to the wing compared to baseline RANS, can already suppress the separation by a sufficient amount. Secondly, it could be due to the always positive k correction at the walls throughout the flow field. To nuance this, it should be recalled that the k -equation correction (R and P_k^Δ) is always positive because M_R is positive and $P_k^\Delta = 0$. This will likely increase the turbulence intensity of the boundary layers to be higher compared to the baseline RANS. Which suppresses the tendency to separate. Now since the k -equation correction should be negative at the walls, see Figure 4.16, it is possible that the increased turbulence cancels the need for the vortex structures in the corner to suppress the separation.

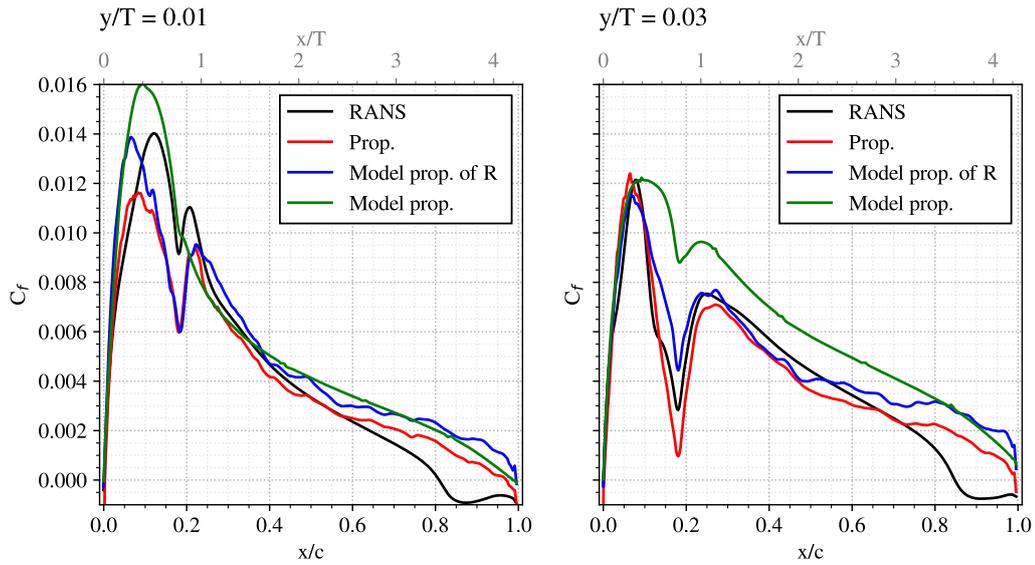


Figure 4.65: Wing friction coefficient profiles for the different simulations in close proximity to the corner.

Figure 4.66 displays the pressure coefficient profiles of the wing at two different heights, one in the corner region and one outside the corner region (thus in the blending region). The latter is done because the LSB did not have a very significant effect on the C_p plots, thus the spatial blending will not alter the profiles by a large amount. From both plots, it can be seen that both model propagations follow the propagation accurately. It is noticed, however, that the model propagation under-predicts the suction peak for $y/T = 0.3$. And both model propagations show an LSB-like behaviour just aft of the suction peak, which confirmed the need for the spatial blending function in the model propagation (not only in the propagation). Since the baseline RANS already had the correct C_p profile with respect to the propagation for which the LSB has been filtered out, there is not really an improvement required from the model propagations. Yet it is good to confirm that the models do not alter the profiles significantly.

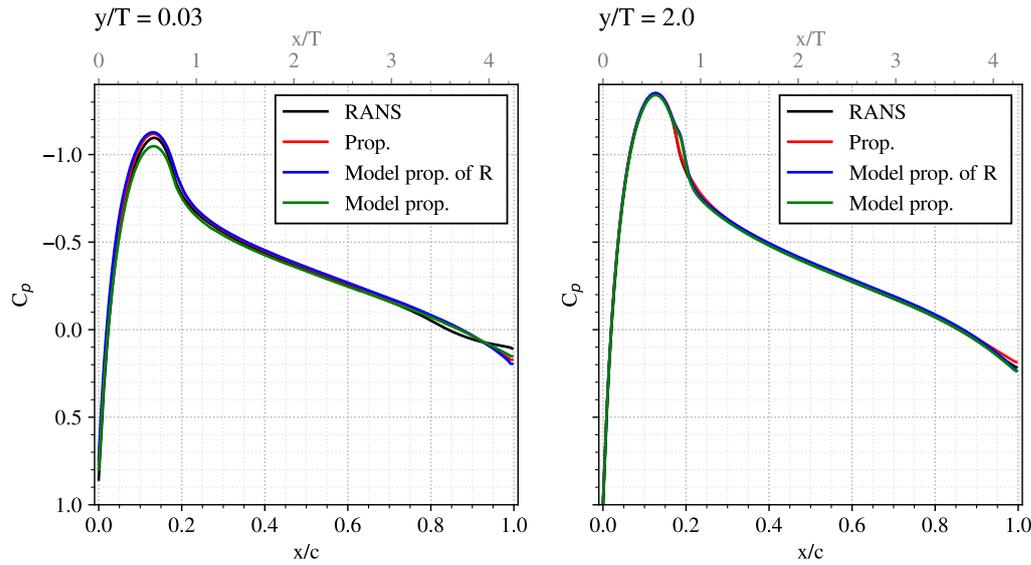


Figure 4.66: Wing pressure coefficient profiles for the different simulations, (left) in close proximity to the corner, (right) at half wing-span.

4.6. Discussion on SpaRTA Modifications

In this section, a discussion is given on the modifications of the SpaRTA methodology with respect to the original publication by Schmelzer et al. [39]. This includes the modification of the ω transport equation, the extra production limiter, the use of different initial conditions and the spatial blending functions applied to the correction terms.

4.6.1. Altering the Transport Equation

In Chapter 3 it is described that one of the correction fields within SpaRTA is the k residual R . In the original publication by Schmelzer et al. [39] the correction field R is present in both the ω transport equation and the k transport equation. However, in this work, R is omitted from the ω -equation (Equation 3.21). The basic reason for this is that with R in the ω -equation, the propagation simulation would diverge rapidly. Thus it was omitted to obtain a converged solution. Note that unsuccessful attempts were made on limiting R in the ω -equation using limiters based on the production limiter applied in the k - ω SST model.

What should be investigated is whether R is required in the ω -equation. Because of the almost perfect results in the propagation (Section 4.3) it can be deduced that for this particular junction flow, R is apparently not required in the ω -equation. Whether R is required depends on the interpretation of R . If it is really seen as just the residual of the k -equation then there is no need to include it in the ω -equation. However, if R is interpreted as a production correction in the k -equation, i.e. similar to P_k^Δ , then it is natural to include R in the ω -equation. This is because the production terms in the original transport equations are equal as well, see Wilcox [31].

The simple conclusion from this work is that based on the propagation results, R acts simply as an extra correction within the k -equation that is not necessarily connected to the production term. However, to confirm this, a thorough investigation using simple academic flows (period hill, square duct, etc.) would be required to verify that it is not just a coincidence for the junction flow.

4.6.2. Production Limiter

Next to the conventional production limiter in the $k-\omega$ SST model which limits the production term in the transport equations to be less than $10\beta^* \omega k$, another limiter has been added in this work. Namely that the total production is always larger or equal to zero, see Equation 3.20. This has been done similarly to all modifications in this section to enhance the stability of the (model) propagation simulations. It should be noted that without this extra limit, the solution would diverge rapidly. Since R is not seen as a production correction it is not included in the limiter.

From the propagation results (Section 4.3) it is seen that this limiter does not restrict the solution to be very different from the LES and thus that the limiter is safe to use in the model propagation. However, it is important to check the regions where the limiter is active. Therefore the active regions as well as the production term without the new limit have been displayed in Figure 4.67 in the symmetry plane. To produce the right image the frozen results have been used, this means the LES data for U and k and the frozen solution for b_{ij} . In the left image, the blue regions indicate where the limiter would be active. From Figure 4.67 it is observed that apart from the negative production structure near the bottom wall in the corner region the limiter does not really impose big constraints. Because in the other regions where the limiter is active the production is close to zero and thus less significant.

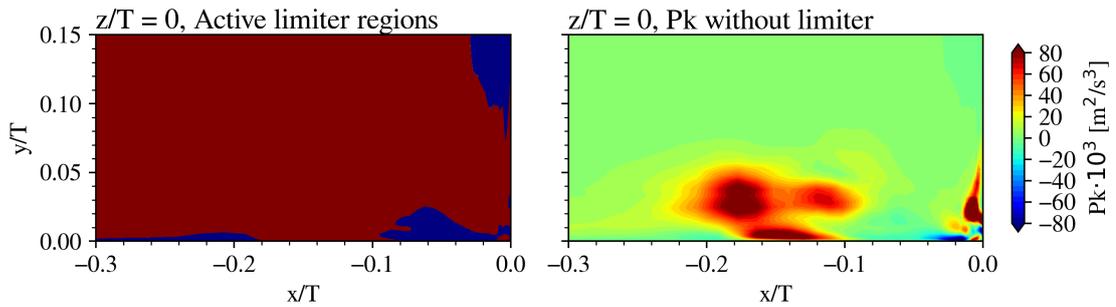


Figure 4.67: Visualisation of the production limiter, the left figure shows where the limiter is active (blue) and inactive (red), the right image shows the value of P_k without the limiter, figure from the frozen approach simulation at $z/T = 0$.

Figure 4.68 shows the active regions and production without limiter in the $y-z$ plane at $x/T = 0.5$. Again, for most regions where the limiter is active, it is not a problem since the production is close to zero. Only the small structure near $z/T = 0.5$ will be influenced by the limiter. However, even though the limiter does remove two relatively small, yet significant in magnitude, negative production regions it is concluded from the propagation that this does not pose a big problem. Since the solution matches the LES data very well, as shown in Section 4.3.

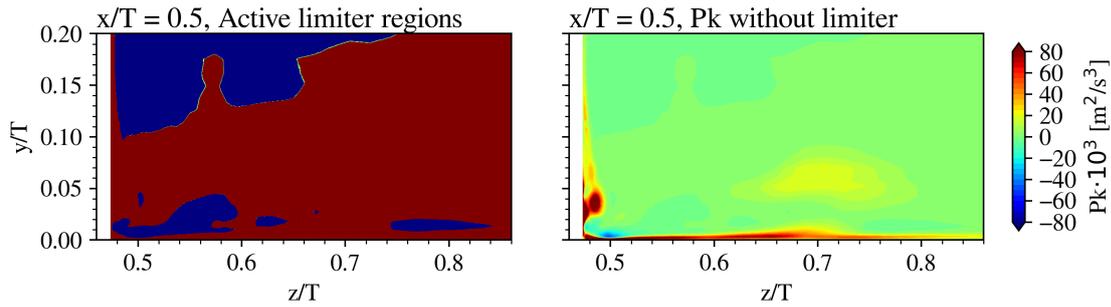


Figure 4.68: Visualisation of the production limiter, the left figure shows where the limiter is active (blue) and inactive (red), the right image shows the value of P_k without the limiter, figure from the frozen approach simulation at $x/T = 0.5$.

4.6.3. Initial Conditions

For both the propagation and the model propagation a decision has to be made on which initial condition (IC) to use. In essence, there are three options within the SpaRTA framework, which are:

- Using the initial conditions as used for the baseline RANS. For example for U that would be a uniform zero vector within the flow field and the BL profile specified at the inlet.
- Using the converged baseline RANS solution as the IC.
- Using the time-mean LES data as the IC for which ω is set to the field found by the k -corrective frozen approach.

Ideally speaking the preferred option is to use the converged baseline RANS solution as the IC. This is because this IC guarantees that any change in the resulting flow field is because of the correction models. And since the starting point is not an empty (or zero) field the number of iterations required to converge to a steady-state solution is likely lower than when using an IC field as one would use for a standard RANS solution. Using the time-mean LES solution as the IC is sub-optimal since, if for some reason the corrections by the models are insignificant, it could be that a seemingly good resulting flow field is just the IC that was stable enough not to change. And thus the effect of the models is not really tested.

When performing the propagation simulation it was found that using the converged baseline RANS solution as IC would result in an unsatisfactory flow field, namely two horseshoe vortices were present in the steady-state solution rather than the expected one. Using the LES as IC on the other hand would result in an almost perfect propagation. However, as already explained, using the LES as IC does not test the effectiveness of the corrections. Fortunately, a good propagation was found when using the LES fields as IC for k and ω and empty fields for U and p (i.e. U and p were equal to the IC of the baseline RANS). The advantage of this IC over using the LES for all variables is that the important vortex structures are not present yet at iteration zero since U is zero everywhere (except at the inlet boundary). And because the propagation results were satisfactory it confirmed the effectiveness of the correction fields since the solver had to compute the vortex structures from almost zero. Not exactly zero since the quantities for ω and k are active at the vortex locations. However, the LES solution was disrupted enough, by setting U and p to zero, that the effectiveness of the corrections is tested to a good amount.

The question that remains is whether using this setup for the IC is a problem within the goal of this thesis. Since it requires the 'correct' fields for k and ω . Fortunately, it is not a problem, this is because the eventual goal of this work is to perform design optimisation of the

wing-body junction flow. So if an LES is used as a starting point then every design iteration can use the solution of the previous iteration as the IC. And because the LES solution of the starting point is present since it is required for the data-driven approach this method is possible.

4.6.4. Propagation for Regression

In the original paper by Schmelzer et al. [39] the combination of the LES data and the ω field from the frozen approach were used to build the features for the sparse regression. In this work, however, the propagation flow fields were used as the basis for the regression features. This has been done since, as mentioned before, the propagation is the theoretically best result achievable with the RANS solver. That is, with respect to the discrepancy with the LES solution. Therefore, using the propagation flow fields to build the regression features is a more realistic scenario since it is nearly impossible to exactly match the time-mean LES solution by the RANS with corrections.

One can argue, however, that this approach is leaning towards an engineering solution. From a physics standpoint, using the LES solution for the regression could lead to a physical insight. This means that a physical understanding of the turbulence correction might be found rather than just an empirical function that appears to work well. This thought of line does however assume that the LES solution resembles the truth. Yet finding a physical understanding using the propagation is more uncertain as a small, but present, discrepancy exists between the propagation and the truth, as shown in Section 4.3.

4.6.5. Spatial Blending Functions

Although the use of spatial blending functions is not necessarily a modification to the SpaRTA method it is an important numerical trick that contributed to this work and thus its effects should be discussed. But first, the reason behind the use of the spatial blending functions that are displayed subsection 3.2.2. Regarding the blending at the wing wall, it was found that the regressed model for R would result in a very high injection of k correction near the LSB to simulate the transition in the model propagation. This injection would, however, result in numerical instabilities with divergence of the solution as a consequence. Initially, this was solved by limiting the model for R , that is instead of using M_R use $0.5 \cdot M_R$. However, this resulted in not enough k in the region aft of the LSB, which would cause premature separation from the wing. And thus a better solution was using a linear blending function very close to the wing that goes to zero if the wall distance goes to zero. Essentially, forcing the baseline RANS solution at the wing wall in a numerically stable manner. The only requirement was that this function excludes the region where the corner and secondary vortices live since it is desired to correct these as shown in the propagation (Section 4.3). The latter is not a problem since the corner region is turbulent, as a turbulent boundary layer is used, and thus in this corner region no laminar separation bubble is present.

As shortly touch upon in Section 4.3, the consequence of using the blending function at the wing wall is that the forces, especially the viscous forces, cannot be compared to any dataset. This is because it is not exactly RANS with k - ω SST as the blending distance is rather short, but it is also not comparable to the LES data since that includes an LSB. The pressure forces are less of a concern since these were not significantly affected by the LSB in the LES, however, the same argument as for the viscous forces still holds. Ultimately to solve this issue it is recommended to rerun the LES for which the boundary layer is tripped in close proximity to the leading edge.

Finally regarding the blending in the far-field, behind this blending there is no physical reasoning. It was observed that quite some noise was present in the correction fields towards the top and outlet of the domain. For b_{ij}^{Δ} this was likely due to the low value of k in the training

data in these regions. Therefore, using blending functions to blend the corrections to zero in the far-field allowed the (model) propagation simulation to be significantly more stable. And since this blending only occurs in the far-field, where no corrections are required, there are no consequences while doing this. That is as long as the portion of the bottom wall and wing near the outlet and top of the domain are disregarded in the calculation of the aerodynamic forces.

5

LES Drag Comparison Anti-Fairing

One of the remaining research questions in this work was to investigate whether the anti-fairing actually reduces the drag force on the wing body junction flow. In this chapter, the drag force of the two wall-resolved LES simulations by Alberts [23] for the baseline geometry and by Vigner [24] for the anti-fairing geometry are computed and compared. These LES simulations are virtually the same except for the geometry in the region connecting the bottom wall to the wing, for the anti-fairing geometry please refer to Chapter 2. This chapter is structured as follows, first, the method of computing the drag force is established in Section 5.1. This section is followed by Section 5.2 in which the drag results are displayed and discussed. Finally, a conclusion is given in Section 5.3.

5.1. Computational Setup

In this section, the numerical methodology for computing the drag force will be discussed. As defined in Anderson [1], the drag force is "the component of the aerodynamic force parallel to the relative wind". In this work the free-stream flow is parallel to the x -direction as defined in Chapter 3. Hence to compute the drag force (F_d) the viscous force and pressure force on both the bottom wall and on the wing should be computed in the x -direction, this means that the drag force for this particular geometry is formulated as:

$$F_d = \underbrace{F_{x,v,1} + F_{x,p,1}}_{\text{bottom wall}} + \underbrace{F_{x,v,2} + F_{x,p,2}}_{\text{wing}} \quad (5.1)$$

The viscous drag force ($F_{x,v}$) is obtained by integrating the wall shear stress in the x -direction ($\tau_{w,x}$) over the solid surface S (Belligoli [62]):

$$F_{x,visc} = \int_S -\tau_{w,x} dS \quad (5.2)$$

Note that the wall shear stress is provided by the OpenFOAM solver. The negative sign is present due to the sign convention used by OpenFOAM to export the wall shear stress. The pressure drag force, $F_{x,p}$, is obtained by integrating the x component of the vector that results from the product of the pressure and the wall normal vector over the solid surface S (Belligoli [62]):

$$F_{x,p} = \int_S ([1, 0, 0]^T \cdot (p \cdot \vec{n})) dS \quad (5.3)$$

Note that the normal vector \vec{n} is pointing into the solid.

To perform the integrations the post-processing software ParaView¹ is used. This software also allows calculating the wall normal vector required for the pressure force. Similar to the work by Srikumar [63] the drag force is computed on multiple geometries. This means that cuts are made in the domain to exclude irrelevant parts for the drag comparison. The different geometries and their naming are displayed in Figure 5.1 for the bottom wall and in Figure 5.2 for the wing. Note that for the anti-fairing geometry, the wing span is slightly longer compared to the baseline version due to the indents present in the bottom wall. This is because in the analysis the wing is cut at a certain y value and since the wing in the anti-fairing starts at a negative value of y instead of $y = 0$ (baseline geometry) the span is slightly longer.

The four geometries that will be used are labelled complete (complete bottom and wing wall), small, which uses a small portion of the bottom wall and half of the wing span and medium. The latter has two configurations one that uses half the wing span and one that uses a short span which equals 5% of the complete span (indicated by medium ss).

Finally, to non-dimensionalise the drag force the drag coefficient is computed as:

$$C_d = \frac{F_d}{\frac{1}{2}\rho U_\infty^2 \delta_{inlet} T} \quad (5.4)$$

Due to the changing wetted area, it is decided to use $\delta_{inlet} \cdot T$ as the reference area.

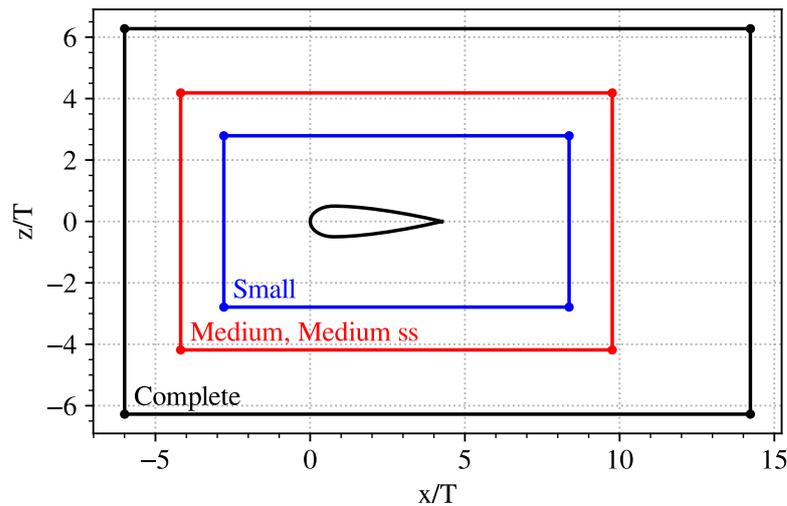


Figure 5.1: The four bottom wall geometries for the drag computation.

¹<https://www.paraview.org/>, Last accessed: 05-07-2023

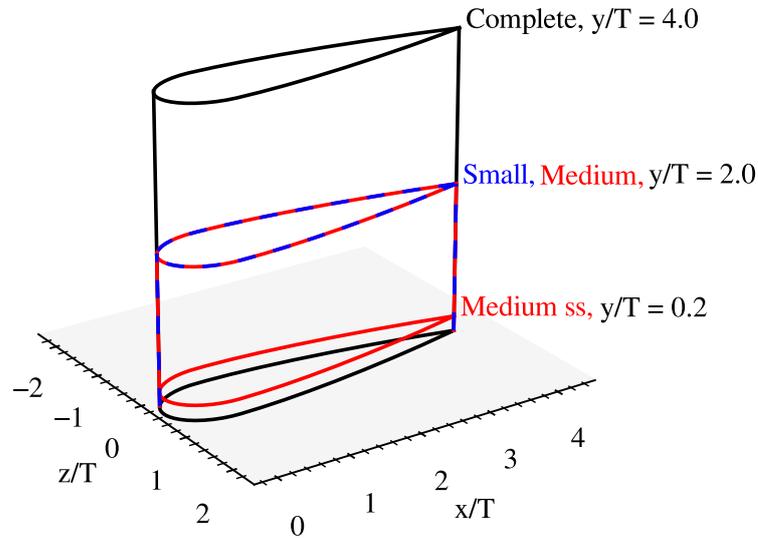


Figure 5.2: The four wing geometries used for the drag computation.

5.2. Drag Results

This section discusses the outcome of the drag computation as described above in Section 5.1. The drag coefficient of the previously mentioned geometries is displayed below in Figure 5.3. This figure shows that the anti-fairing geometry has a drag advantage for the *Medium ss* geometry with respect to the baseline geometry. In the right plot, it can be seen that the *Medium ss* geometry has the propulsive pressure force as described by Belligoli et al. [22]. According to Belligoli et al., this is due to the incomplete pressure recovery over the anti-fairing geometry resulting in a propulsive net force. For all other geometries, the drag computed over the anti-fairing is higher than for the baseline geometry.

Interestingly, the only contribution to the pressure drag on the baseline geometry is by the wing since the normal vector of the bottom wall is constant, namely $[0, 1, 0]^T$. Hence the pressure force by the bottom wall in the x -direction is always zero for the baseline geometry. Together with Figure 5.3 this confirms that the propulsive pressure force seen in the right plot by the anti-fairing for the *Medium ss* geometry is indeed due to a propulsive mechanism by the bottom wall, as the pressure force by the wing in x -direction is close to zero due to the short span. This pressure drag advantage seems to disappear, however, when increasing the wing span, see the *Medium* geometry.

To further investigate this drag advantage of the *Medium ss* geometry an analysis was performed for which the wing span was changed step-wise. The result of this analysis is displayed in Figure 5.4 in which the x -axis displays the wing span length. Note the *Medium* geometry is used for the bottom wall and remained constant. Besides, the drag force increases with y/T since the wetted surface area increases, recall that C_d is computed using $\delta_{inlet} \cdot T$ as the reference area. From Figure 5.4, it is more clear that the drag advantage of the anti-fairing geometry comes from the pressure drag difference. From the left plot, it is deduced that the anti-fairing has a lower drag force compared to the baseline geometry up to a wing span of approximately $0.8y/T$ which corresponds to 20% of the complete wing span used in this work.

The non-linear behaviour seen in the pressure drag curve explains why the propulsive

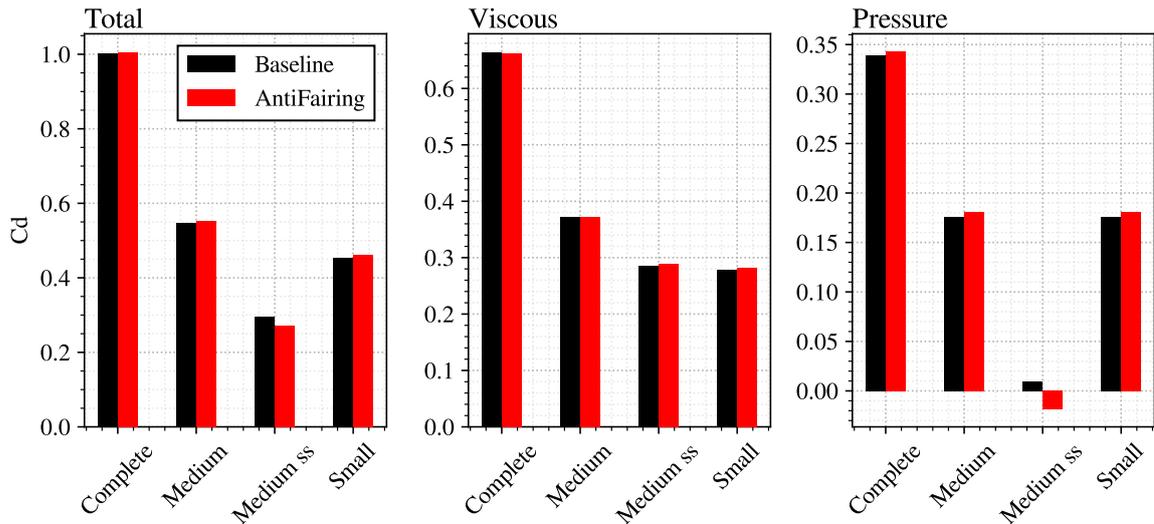


Figure 5.3: Bar chart of the drag components for the four geometries, (left) the complete drag coefficient, (middle) the drag coefficient based on the viscous drag force, (right) the pressure drag coefficient.

pressure force seen for the *Medium ss* geometry in Figure 5.3 does not translate to the *Medium* geometry. Since according to the right plot in Figure 5.4 the pressure drag shows a non-linear trend with wing span length with the added observation that the pressure drag increases more rapidly for the anti-fairing geometry.

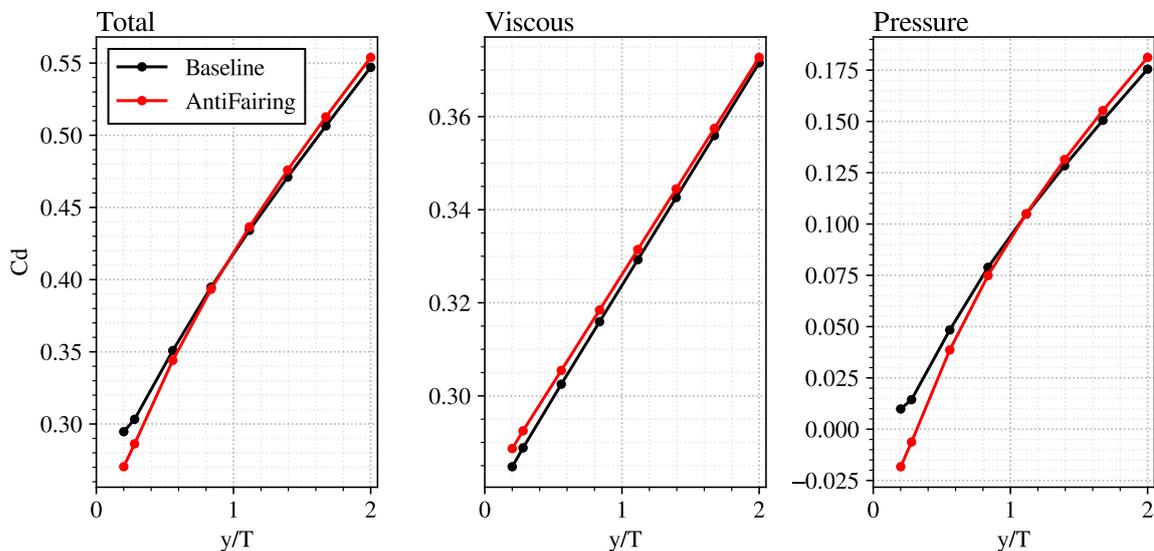


Figure 5.4: Drag coefficient curve showing the combination of the *Medium* bottom wall plus a portion of the wing for which the height is indicated by the y/T -axis. (left) Complete drag coefficient, (middle) viscous drag coefficient, (right) pressure drag coefficient.

A possible explanation for the dissimilar non-linear behaviour of the baseline geometry versus the anti-fairing geometry could lie in the presence of the laminar separation bubble (LSB). Because the presence of an LSB will influence the drag forces. It, namely, increases the pressure drag and marginally decreases the friction drag due to the flow reversal in the

bubble. Fortunately, an LSB was present in both the LES of the baseline geometry and the anti-fairing geometry which allows the LES simulations to be compared. Yet it must be noted that it is likely that the anti-fairing influences the behaviour of the LSB as it influences the HSV as shown in Vigner [24], which, in its turn, impacts the motion of the laminar flow. And thus it could impact the pressure drag behaviour of the anti-fairing simulation. Next to that, since the wing span of the anti-fairing is slightly longer, a larger portion of the LSB is present when computing the drag.

An investigation regarding the impact of the anti-fairing on the shape of the LSB is missing in this work and would be left as a recommendation if the impact of the anti-fairing on the drag force would be significant for more a realistic geometry (such as *Small* or *Medium*).

It should be noted that this trend of the anti-fairing only being effective for a portion of the complete geometry was also observed by Srikumar [63] who performed a wall-modelled LES of the anti-fairing geometry. However, the results by Srikumar were influenced by an unexpected trailing edge separation above the vorticity region. Next to that, it is unclear whether an LSB was present in the simulation by Srikumar [63].

5.3. Concluding Remarks

To conclude the comparison, it is seen that indeed the anti-fairing will reduce the drag of the wing-body junction flow by approximately 8% using a propulsive pressure mechanism over the bottom wall if observing a specific geometry, that is the *Medium ss* geometry. However, this specific geometry is rather unrealistic. If increasing the wing span length used to compute the drag force to a more realistic value, the drag force of the anti-fairing is approximately equal to the baseline geometry and can therefore, according to this LES comparison, not be labelled as a drag-reducing geometry.

As counterarguments to the latter conclusion, one could mention the negative effect of the LSB and the slightly longer wing span of the anti-fairing. To investigate the effect of the LSB another wall-resolved LES will be required for which the flow is tripped close to the leading edge. Yet it could be argued whether this is worth the extra computational effort as it is shown in the experimental study by Belligoli et al. [22] that the achievable drag reduction of the anti-fairing computed by the numerical methods seems to be too optimistic. Next to that, the extra wing span will increase the drag due to the extra wetted area of the wing compared to the wing of the baseline geometry. Yet in a realistic scenario, the anti-fairing would likely be retrofitted to an already existing commercial aircraft, for which it is not expected that the wing span will be changed due to the presence of a new wing fairing. Hence it is more realistic to compare the baseline geometry to the anti-fairing with the same span measuring from $y = 0$ and thus to take the extra wing area below $y = 0$ for granted.

6

Conclusion

This chapter gives the conclusion of the work performed in this thesis. For which the main research question was formulated as follows:

Can the discrepancy in the flow field of a standard RANS simulation with respect to a large eddy simulation for the wing-body junction flow be reduced using the SpaRTA machine learning algorithm?

Prior to addressing this main question, is to answer the sub-questions as defined in Chapter 2. Subsequently, the main question will be discussed followed by a set of recommendations for future work.

Which regions in the RANS solution require significant corrections? From the comparison of the baseline RANS simulation with the LES data as well as from the k -corrective frozen results, several discrepancies between the RANS and LES flow fields were found. First of all, is the difference in the location of flow separation upstream of the leading edge, which causes the horseshoe vortex (HSV) to be misplaced throughout the flow domain in the RANS solution. Secondly is the lack of secondary vortices in the RANS result. This was linked to the inaccurate prediction of anisotropy of the Reynolds stress tensor by the k - ω SST model for this flow. The lack of the secondary vortices also answers the sub-question: *is the vortex topology different between the LES and the baseline RANS?* And finally, a dissimilar skin friction profile on the wing was observed. This dissimilarity is due to the presence of a laminar separation bubble (LSB) on the surface of the wing in the LES solution and due to the corner separation present in the RANS solution. Thus to answer the question, the two regions that require significant corrections are the location where the time-mean HSV is formed, that is upstream of the leading edge. And the corner region wrapping the complete wing where the secondary vortices and corner vortex live. Recall that it has been decided to not correct the RANS to predict the LSB since the original k - ω SST model is not meant to incorporate transition prediction.

What are the discrepancies in the flow fields between the propagated correction fields and the high-fidelity simulation? The propagation results showed that the effectiveness of the correction fields on the baseline turbulence model was very good. To such an extent that there are hardly any significant discrepancies found between the propagation and the LES data. The differences that are notable are present in the peak magnitude of the vorticity of the HSV and the peak magnitude of the turbulent kinetic energy. However, these dissimilarities can be neglected compared to the major improvements with respect to the baseline RANS. Which

include the correct location, shape and topology of the system of vortices and a matching skin friction curve on the surface of the wing.

What is the error between the regressed model and the training data? From the regression it was found that the mean squared error (MSE) for the regression of R was equal to $0.95 \cdot 10^9$ [m^4/s^6] and $1.36 \cdot 10^{-2}$ for b_{ij}^Δ . However, these values cannot be compared to each other nor be solely used to quantify the prediction performance of the model. The latter is because the MSE was used as one of multiple tools to compare different models. Regarding the regression of the correction to the k -equation R , the fit of the found models with an R^2 -score of at least 0.9 was quite satisfactory. From the flow fields and scatter plots, it was concluded that the regressed models were able to correctly predict the location of the required corrections and to some extent the correct magnitude of the correction values. However, it was also noticed that the negative values present in the target data were not determined by the models.

The satisfactory result for R does unfortunately not hold for the regression of the anisotropy correction b_{ij}^Δ . From the flow field visualisation and the scatter plots, it was concluded that the fit was poor. This has the consequence that a correct prediction of the secondary vortices in any model propagation cannot be expected due to the incorrect anisotropy determination. Next to that is the fact that the turbulent production correction by the b_{ij}^Δ model was found to be zero throughout the flow domain. This is significant since it was found in the results of the k -corrective frozen method that the magnitude of this production correction cannot be neglected in comparison to R .

Do the regressed models run stable in a RANS simulation? From the various model propagations performed for this work, of which only the best is shown, it was found that both correction models ran stable in a CFD solver for the same flow case on which the models have been trained. The latter is an important note since testing for generalisability of the models, which will be discussed below, is missing in this work. However, it was found for b_{ij}^Δ that only the low complexity models would be stable, this is likely a side-effect of the poor regression fit.

What are the discrepancies in the flow fields between RANS with correction models and the high-fidelity simulation? and; is the vortex topology present in the LES solution predicted by the custom turbulence model? Based on the results obtained from the model propagation, it was demonstrated that both the complete model propagation and the model propagation of R improved the calculation of the HSV location. However, in the corner region, significant discrepancies were found in the complete model propagation. First of all the corner vortex was reduced in magnitude and secondly, the secondary vortices were lacking. The latter was expected due to the poor fit of the b_{ij}^Δ model. In the model propagation of R the vortex topology in the corner region was correct. However, regarding the secondary vortices, this correct determination is mainly due to the fact that b_{ij}^Δ is correct in this model propagation. Finally, the skin friction profile on the wing revealed that the complete model propagation did not have a corner separation, which was not expected due to the lack of (secondary) vortices in the corner region. It was deduced that this is likely due to the increased turbulence in the boundary layers which is a result of the production correction by b_{ij}^Δ being zero instead of the negative value as found in the frozen method.

Is the new turbulence model generally applicable? Will the new model also improve wing-body junction flows with different types of fairings and wing geometries? A big topic in the quest to find new turbulence models (and or techniques) is the question of whether these new models are generally applicable. Unfortunately, this question cannot be answered for the SpaRTA methodology on the junction flow by this work. Due to the limited time frame of a Master's thesis, generalisability tests, both *a priori* and *a posteriori*, are missing. A final

step would be to propagate the found models on the anti-fairing geometry, as this geometry could represent a design iteration in an aerodynamic shape optimisation problem. For which, originally, the objective of this work was meant. However, before performing this test it is recommended to first improve the machine-learning methodology for generating a b_{ij}^{Δ} model, but more on that later.

Before going to the general conclusion of this thesis a final sub-question which followed from the literature review is remaining: *What is the difference in drag force between the LES on the baseline wing-body geometry by Alberts [23] and the LES on the anti-fairing geometry by Vigner [24]?* The comparison of the two LES's showed that indeed the anti-fairing reduces the drag force by a propulsive pressure mechanism over the bottom wall. However, this reduction in total drag was only measured if a small portion of the complete wing span was included in the geometry over which the drag was computed, that is 5% to 20% of the original span. If this span was increased to larger values the drag reduction vanished and for some geometries even increased the total drag. However, it should be kept in mind that both LES simulations did include an LSB over a large portion of the wing, this very likely influenced the drag comparison since it is expected that the anti-fairing influences the location and form of the LSB with respect to the baseline geometry.

Finally, to answer the main question of this work on whether the discrepancy of a RANS simulation on the wing-body junction flow can be decreased using the SpaRTA methodology the answer would be no. This is mainly due to the methodology in its current state not being able to find a sufficiently good model for the anisotropy correction, which causes the vortex topology in the corner region to be more dissimilar with respect to the truth compared to a baseline RANS simulation. Yet, it cannot be overlooked that an improvement is found in the positioning of the HSV. Next to that, the propagation results, as well as the model propagation of R , showed that with the right determination of the correction fields, a steady RANS simulation can be very close to the time average of the true flow field.

Logically, design optimisation with the turbulence correction models found in this work is undesirable. Despite the improved positioning of the HSV, it is preferred to have the correct vortex topology since the drag force is dependent on this vortex system.

6.1. Recommendations

Naturally, from the conclusion flows a set of recommendations for future work. Although some have already been given throughout the various chapters they will be repeated in this section. First of all, is improving the methodology of finding a model for the anisotropy correction. Ideas that were established in Chapter 4 were:

- To focus on the important Reynolds stress components with respect to the secondary vortices instead of the complete tensor, that is to focus on $\overline{v'v'}$, $\overline{w'w'}$ and $\overline{v'w'}$ in the current reference system. This is, however, specific for the junction flow problem and not for the SpaRTA methodology overall. Besides, it should be checked whether the resulting correction does not deteriorate the other components of the stress tensor.
- Inspired by the complexity of some explicit ARSM ([35, 36]), to increase the complexity of the basis functions, for example, include multiple secondary variables in a single non-linear function.
- For more complex three-dimensional problems, such as the junction flow, to represent the b_{ij}^{Δ} correction by a more sophisticated machine learning approach such as a neural network (for example the TBNN by Ling et al. [41]). This will reduce the ability to understand the correction model, however, in the view of design optimisation for which the

model would only be used on a single flow problem this would not pose a big problem. Especially if a neural network could increase the accuracy prediction.

A second recommendation flowing from this work is to include the junction flow in the set of test cases for novel turbulence models and or novel turbulence model methodologies. This is because this flow includes, as shown, both a misprediction in the separation of a larger flow structure (the HSV) and a required Reynolds stress anisotropy correction to get the correct vortex topology. Naturally, one would only run this case after the novel technique has shown promising results on more academic flow cases, such as the square duct. To include the wing-body junction flow in the test cases it is, however, recommended to rerun the LES by Alberts [23] for which the flow is tripped close to the leading edge to prevent any (negative) influence on the results by the laminar separation bubble.

Thirdly, it is recommended to test the generalisability of the found models using the anti-fairing geometry (i.e. the LES by Vigner [24]). Yet, it is only recommended to do so if improved b_{ij}^{Δ} models have been found. An *a posteriori* test can be done by comparing the LES data to the model propagation results. Next to that, performing a *k*-corrective-frozen simulation on the anti-fairing LES data also allows to use the test case as an *a priori* test by using the propagation or LES data as input for the found models.

Finally, as shown in Chapter 3, in comparison to the original SpaRTA methodology the R term has been omitted from the ω equation in this work. It is thus recommended to more thoroughly investigate the effect of this by running small academic flow cases such as the periodic hill and comparing the results.

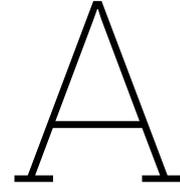
References

- [1] J. D. Anderson. *Introduction to Flight Eighth Edition*. International Student Edition. McGraw-Hill, 2016. ISBN: 978-981-4636-18-6.
- [2] R. L. Simpson. “JUNCTION FLOWS”. In: *Annual Review of Fluid Mechanics* 33.1 (2001), pp. 415–443. DOI: 10.1146/annurev.fluid.33.1.415.
- [3] W. J. Devenport et al. “Effects of a leading-edge fillet on the flow past an appendage-body junction”. In: *AIAA Journal* 30.9 (1992), pp. 2177–2183. DOI: 10.2514/3.11201.
- [4] S. Ryu et al. “Large-Eddy Simulation of a Wing–Body Junction Flow”. In: *AIAA Journal* 54.3 (2016), pp. 793–804. DOI: 10.2514/1.J054212.
- [5] Yu Zhang et al. “Customized data-driven RANS closures for bi-fidelity LES–RANS optimization”. In: *Journal of Computational Physics* 432 (2021), p. 110153. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2021.110153.
- [6] K. Duraisamy, G. Iaccarino, and H. Xiao. “Turbulence Modeling in the Age of Data”. In: *Annual Review of Fluid Mechanics* 51.1 (2019), pp. 357–377. DOI: 10.1146/annurev-fluid-010518-040547.
- [7] F. Gand, V. Brunet, and S. Deck. “A Combined Experimental, RANS and LES Investigation of a Wing Body Junction Flow”. In: *40th Fluid Dynamics Conference and Exhibit*. 2010. DOI: 10.2514/6.2010-4753.
- [8] F. Gand et al. “Investigation of a Nonlinear Reynolds-Averaged Navier–Stokes Closure for Corner Flows”. In: *AIAA Journal* 54.2 (2016), pp. 386–398. DOI: 10.2514/1.J054313.
- [9] F. Gand et al. “Flow dynamics past a simplified wing body junction”. In: *Physics of Fluids* 22.11 (2010), p. 115111. DOI: 10.1063/1.3500697.
- [10] W. J. Devenport and R. L. Simpson. “Time-dependent and time-averaged turbulence structure near the nose of a wing-body junction”. In: *Journal of Fluid Mechanics* 210 (1990), pp. 23–55. DOI: 10.1017/S0022112090001215.
- [11] S. S. Elahi, E. A. Lange, and S. P. Lynch. “Effect of Reynolds number on turbulent junction flow fluid dynamics and heat transfer”. In: *International Journal of Heat and Mass Transfer* 142 (2019), p. 118328. ISSN: 0017-9310. DOI: 10.1016/j.ijheatmasstransfer.2019.06.084.
- [12] E. P. Rood and D. G. Anthony. “Tail Profile Effects on Unsteady Large Scale Flow Structure in the Wing and Plate Junction”. In: *AMSE* (1985).
- [13] J. L. Fleming. “An Experimental Study of a Turbulent Wing-Body Junction and Wake Flow”. MA thesis. Blacksburg, Va.: Virginia Polytechnic Institute and State University, 1991.
- [14] J. L. Fleming et al. “An experimental study of a turbulent wing-body junction and wake flow”. In: *Experiments in Fluids* 14 (1993), pp. 366–378. DOI: 10.1007/BF00189496.
- [15] M. A. Kegerise et al. “An Experimental Investigation of a Wing-Fuselage Junction Model in the NASA Langley 14- by 22-Foot Subsonic Wind Tunnel”. In: *AIAA Scitech 2019 Forum*. DOI: 10.2514/6.2019-0077.

- [16] H. J. Perkins. "The formation of streamwise vorticity in turbulent flow". In: *Journal of Fluid Mechanics* 44.4 (1970), pp. 721–740. DOI: 10.1017/S0022112070002112.
- [17] C. L. Rumsey et al. "Improvements to the Quadratic Constitutive Relation Based on NASA Juncture Flow Data". In: *AIAA Journal* 58.10 (2020), pp. 4374–4384. DOI: 10.2514/1.J059683.
- [18] C. L. Rumsey, D. Neuhart, and M. A. Kegerise. "The NASA Juncture Flow Experiment: Goals, Progress, and Preliminary Testing (Invited)". In: *54th AIAA Aerospace Sciences Meeting*. 2016. DOI: 10.2514/6.2016-1557.
- [19] W. J. Devenport et al. "Effects of a fillet on the flow past a wing-body junction". In: *AIAA Journal* 28.12 (1990), pp. 2017–2024. DOI: 10.2514/3.10517.
- [20] Z. Belligoli et al. "Using an Anti-Fairing to Reduce Drag at Wing/Body Junctions". In: *AIAA Journal* 57.4 (2019), pp. 1468–1480. DOI: 10.2514/1.J057481.
- [21] J. Brezillon and R. P. Dwight. "Applications of a discrete viscous adjoint method for aerodynamic shape optimisation of 3D configurations". In: *CEAS Aeronautical Journal* 3.1 (2012), pp. 25–34. DOI: 10.1007/s13272-011-0038-0.
- [22] Z. Belligoli et al. "A Review of Numerical and Experimental Studies of the Anti-Fairing". In: *AIAA AVIATION 2020 FORUM*. DOI: 10.2514/6.2020-2703.
- [23] J. A. Alberts. "Wall-Resolved Large Eddy Simulation of a Wing-Body Junction". MA thesis. Delft University of Technology, 2021. URL: <http://resolver.tudelft.nl/uuid:e3b8dac2-600a-463d-a93e-0e503eb31d06>.
- [24] G. Vigner. "Data driven modeling of junction flows". MA thesis. Delft University of Technology, 2022. URL: <http://resolver.tudelft.nl/uuid:6644f3b4-818c-40a6-8927-e66ead02d4d2>.
- [25] D. D. Apsley and M. A. Leschziner. "Investigation of Advanced Turbulence Models for the Flow in a Generic Wing-Body Junction". In: *Flow, Turbulence and Combustion* 67 (2001), pp. 22–55. DOI: 10.1023/A:1013598401276.
- [26] G. Kalitzin, A. R. B. Gould, and J. J. Benton. "Application of two-equation turbulence models in aircraft design". In: *AIAA 34th Aerospace Sciences Meeting and Exhibit*. 1996. DOI: 10.2514/6.1996-327.
- [27] F. R. Menter. "Two-equation eddy-viscosity turbulence models for engineering applications". In: *AIAA Journal* 32.8 (1994), pp. 1598–1605. DOI: 10.2514/3.12149.
- [28] S. Hickel. *Lecture notes from CFD for Aerospace Engineers (course AE4202), Reynolds-Averaged Navier-Stokes : RANS*. Faculty of Aerospace Engineering, Delft University of Technology. 2020.
- [29] P. R. Spalart. "Strategies for turbulence modelling and simulations". In: *International Journal of Heat and Fluid Flow* 21.3 (2000), pp. 252–263. DOI: 10.1016/S0142-727X(00)00007-2.
- [30] P. R. Spalart and S. R. Allmaras. "A one-equation turbulence model for aerodynamic flows". In: *AIAA 30th Aerospace Sciences Meeting and Exhibit*. 1992. DOI: 10.2514/6.1992-439.
- [31] D. C. Wilcox. "Formulation of the k- ω Turbulence Model Revisited". In: *AIAA Journal* 46.11 (2008), pp. 2823–2838. DOI: 10.2514/1.36541.
- [32] S. Hickel. *Lecture notes from CFD for Aerospace Engineers (course AE4202), Large Eddy Simulation : LES*. Faculty of Aerospace Engineering, Delft University of Technology. 2020.

- [33] A. W. Vreman. “An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications”. In: *Physics of Fluids* 16.10 (2004), pp. 3670–3681. DOI: 10.1063/1.1785131.
- [34] S. B. Pope. “A more general effective-viscosity hypothesis”. In: *Journal of Fluid Mechanics* 72.2 (1975), pp. 331–340. DOI: 10.1017/S0022112075003382.
- [35] T. B. Gatski and C. G. Speziale. “On explicit algebraic stress models for complex turbulent flows”. In: *Journal of Fluid Mechanics* 254 (1993), pp. 59–78. DOI: 10.1017/S0022112093002034.
- [36] S. Wallin and A. V. Johansson. “An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows”. In: *Journal of Fluid Mechanics* 403 (2000), pp. 89–132. DOI: 10.1017/S0022112099007004.
- [37] J. Weatheritt and R. Sandberg. “A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship”. In: *Journal of Computational Physics* 325 (2016), pp. 22–37. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2016.08.015.
- [38] J. Weatheritt and R. Sandberg. “The development of algebraic stress models using a novel evolutionary algorithm”. In: *International Journal of Heat and Fluid Flow* 68 (2017), pp. 298–318. ISSN: 0142-727X. DOI: 10.1016/j.ijheatfluidflow.2017.09.017.
- [39] M. Schmelzer, R. P. Dwight, and P. Cinnella. “Discovery of Algebraic Reynolds-Stress Models Using Sparse Symbolic Regression”. In: *Flow, Turbulence and Combustion* 104 (2020), pp. 579–603. DOI: 10.1007/s10494-019-00089-x.
- [40] J. P. Huijting. “Data Driven Turbulence Modelling”. MA thesis. Delft University of Technology, 2020. URL: <http://resolver.tudelft.nl/uuid:713d549d-4026-4ecf-bf9e-c25c84b2b245>.
- [41] J. Ling, A. Kurzwski, and J. Templeton. “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance”. In: *Journal of Fluid Mechanics* 807 (2016), pp. 155–166. DOI: 10.1017/jfm.2016.615.
- [42] Z. J. Zhang and K. Duraisamy. “Machine Learning Methods for Data-Driven Turbulence Modeling”. In: *22nd AIAA Computational Fluid Dynamics Conference*. 2015. DOI: 10.2514/6.2015-2460.
- [43] J. Ling and A. Kurzwski. “Data-driven Adaptive Physics Modeling for Turbulence Simulations”. In: *23rd AIAA Computational Fluid Dynamics Conference*. 2017. DOI: 10.2514/6.2017-3627.
- [44] J. Ling and J. Templeton. “Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty”. In: *Physics of Fluids* 27.8 (2015), p. 085103. DOI: 10.1063/1.4927765.
- [45] J. Steiner, A. Viré, and R. P. Dwight. “Classifying Regions of High Model Error Within a Data-Driven RANS Closure: Application to Wind Turbine Wakes”. In: *Flow, Turbulence and Combustion* 109.3 (2022), pp. 545–570. DOI: 10.1007/s10494-022-00346-6.
- [46] C. L. Rumsey and G. N. Coleman. *NASA Symposium on Turbulence Modeling: Roadblocks, and the Potential for Machine Learning*. Tech. rep. National Aeronautics and Space Administration, 2022.
- [47] H. Xiao and P. Cinnella. “Quantification of model uncertainty in RANS simulations: A review”. In: *Progress in Aerospace Sciences* 108 (2019), pp. 1–31. ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2018.10.001.

- [48] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000. DOI: 10.1017/CB09780511840531.
- [49] H. Reichardt. "Vollständige Darstellung der turbulenten Geschwindigkeitsverteilung in glatten Leitungen". In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 31.7 (1951), pp. 208–219. DOI: 10.1002/zamm.19510310704.
- [50] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [51] C. Greenshields. *OpenFOAM v7 User Guide*. London, UK: The OpenFOAM Foundation, 2019. URL: <https://doc.cfd.direct/openfoam/user-guide-v7>.
- [52] L. S. Caretto et al. "Two calculation procedures for steady, three-dimensional flows with recirculation". In: *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*. Ed. by Henri Cabannes and Roger Temam. Springer Berlin Heidelberg, 1973.
- [53] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [54] J. O. Rawlings, S. G. Pantula, and D. A. Dickey. *Applied Regression Analysis A Research Tool*. Springer New York, NY, July 1998. ISBN: 978-0-387-98454-4.
- [55] P. R. Spalart. "Philosophies and fallacies in turbulence modeling". In: *Progress in Aerospace Sciences* 74 (2015), pp. 1–15. ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2014.12.004.
- [56] J. Wu, H. Xiao, and E. Paterson. "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework". In: *Phys. Rev. Fluids* 3 (2018). DOI: 10.1103/PhysRevFluids.3.074602.
- [57] A. Kraskov, H. Stögbauer, and P. Grassberger. "Estimating mutual information". In: *Phys. Rev. E* 69 (6 2004), p. 066138. DOI: 10.1103/PhysRevE.69.066138.
- [58] M. W. Goderie. "Enhancement of data-driven turbulence models for wind turbine wake applications". MA thesis. Delft University of Technology, 2020. URL: <http://resolver.tudelft.nl/uuid:5f62c19f-7cf0-4c0b-9cc0-561348e59417>.
- [59] R. Riolo, E. Vladislavleva, and J. H. Moore. *Genetic Programming Theory and Practice IX*. Springer New York, NY, 2011. DOI: 10.1007/978-1-4614-1770-5.
- [60] C. R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.
- [61] K. Hoefnagel. "Multi-Flow Generalization in Data-Driven Turbulence Modeling: An Exploratory Study". MA thesis. Delft University of Technology, 2023. URL: <http://resolver.tudelft.nl/uuid:324d4b2d-bf58-40a0-b60d-4e2e0b992797>.
- [62] Z. Belligoli. "Numerical optimization techniques for practical problems in Aerodynamics". PhD thesis. Delft University of Technology, 2021.
- [63] S. Srikumar. "LES of a novel wing/body junction: Anti-Fairing". MA thesis. Delft University of Technology, 2019. URL: <http://resolver.tudelft.nl/uuid:86621aa3-90a6-4929-9534-97459ef11f87>.



SpaRTA Nomenclature

In the table displayed below the translation of the nomenclature used within SpaRTA is displayed. This table is useful when reading the mutual information plot as well as the coefficient matrices.

SpaRTA	Math symbol	SpaRTA	Math symbol
S2_s	I_s	G1_s	$G_s^{(1)}$
W2_s	II_s	G1	$G^{(1)}$
W2S_s	IV_s	G6_s	$G_s^{(6)}$
W2S2_s	V_s	T2_s	$T_{ij,s}^{(2)}$
W2SWS2_s	VI_s	T4_s	$T_{ij,s}^{(4)}$
S2	I	T7_s	$T_{ij,s}^{(7)}$
W2	II	T8_s	$T_{ij,s}^{(8)}$
S3	III	q_Q	q_Q
W2S	IV	q_nu	q_ν
Ak2WS_s	$IX_{k,s}$	q_tauk_B	$q_{\tau k B}$
W2AkS_s	$VIII_{k,s}$	q_perp	q_\perp
Ak2WS2_s	$XII_{k,s}$	q_pS	q_{pS}
W2AkS2_s	$X_{k,s}$	q_gamma	q_γ
Ak2SWS2_s	$XIII_{k,s}$	q_V	q_V
W2SAkS2_s	$XI_{k,s}$	q_Re	q_{Re}
Ak2WS	IX_k	q_TI	q_{TI}
Ak2S2	III_k	sqrt_abs(x)	$\sqrt{ x }$
WAKS2	VII_k	tanh(x)	$\tanh(x)$
epsilon	ϵ	x**a	x^a

B

Mesh Study

In the methodology description (Chapter 3) it is described that in order to resolve the smaller vortex structures in the corner region a mesh with $y^+ \approx 1$ is required. This appendix discusses the small mesh study performed for this work which resulted in that conclusion. Three meshes were established for which the y^+ values are summarised in Table B.1. These meshes were established by refining or coarsening the mesh towards the wall in the boundary layer region over the bottom wall and the wing. Note that *Mesh 3* has been coarsened towards the outer parts of the domain to keep the total number of cells similar to *Mesh 2*.

Table B.1: Values of y^+ and the total number of cells for the different meshes.

	Wing			Bottom Wall			num. of cells
	min. y^+	max. y^+	mean y^+	min. y^+	max. y^+	mean y^+	
Mesh 1	1.91	55.62	28.98	1.47	68.91	30.55	$\sim 15.1\text{m}$
Mesh 2	0.56	25.32	14.15	0.62	27.58	12.22	$\sim 16.4\text{m}$
Mesh 3	0.035	1.86	0.96	0.019	2.54	0.94	$\sim 16.4\text{m}$

The mesh study was performed using the baseline RANS configuration, i.e. steady state RANS with the $k-\omega$ SST turbulence model. The resulting vorticity field in the symmetry plane upstream of the leading edge for the three meshes is given in Figure B.1. From this figure, it can be concluded that the HSV is not very sensitive to the mesh and that *Mesh 2* would have been sufficient. Although it must be argued that the mean y^+ value of *Mesh 2* is neither in the viscous sub-layer nor in the log-law layer, therefore using wall models can be difficult.

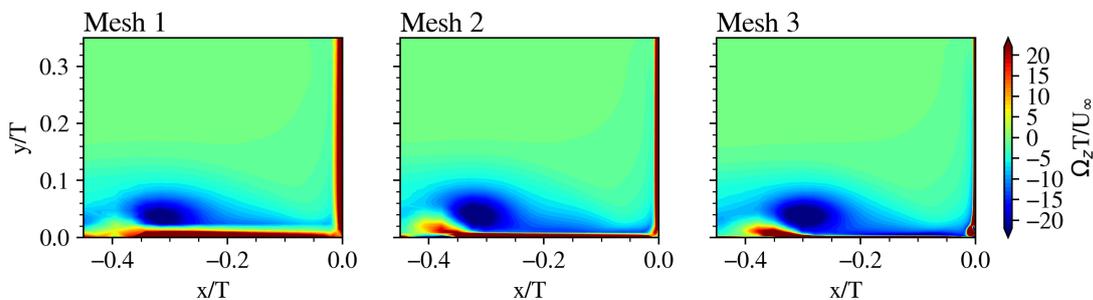


Figure B.1: Vorticity in z -direction in the symmetry plane upstream of the wing for the different meshes.

When closer examining the corner region of Figure B.1, that is Figure B.2, it becomes more clear why *Mesh 3* is required for this work. That is, to be able to resolve the smaller vortex structures, a fine mesh is required in the corner region. Since both *Mesh 1* and *2* do not show the corner vortex nor the smaller vortex structures. Therefore it was decided to use *Mesh 3* for all RANS and SpaRTA related simulations in this work.

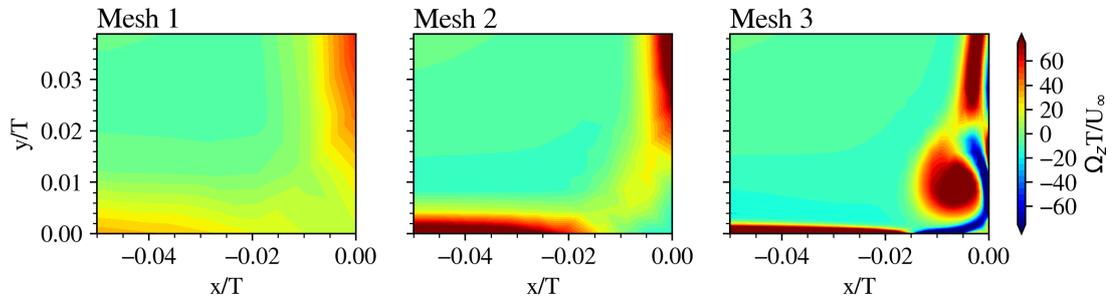


Figure B.2: Vorticity in the corner region in the symmetry plane for the different meshes.

For completeness the pressure and skin friction profiles (C_p and C_f) over the wing both close to and far away from the corner region are displayed in Figure B.3 and Figure B.4. In which it is seen that especially far away from the corner region the profiles match very well. However, *Mesh 1* does show the tendency to a laminar separation region, yet since C_f remains positive the flow stays attached. In the corner region ($y/T = 0.025$) the C_f profiles show a larger discrepancy with respect to each other.

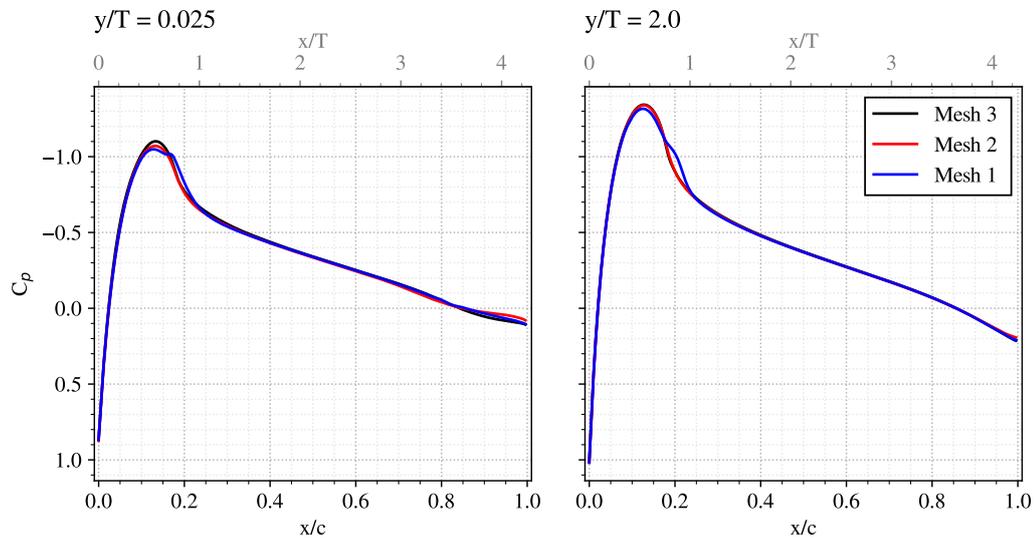


Figure B.3: Pressure coefficient profiles of the wing for the different meshes at two y -levels.

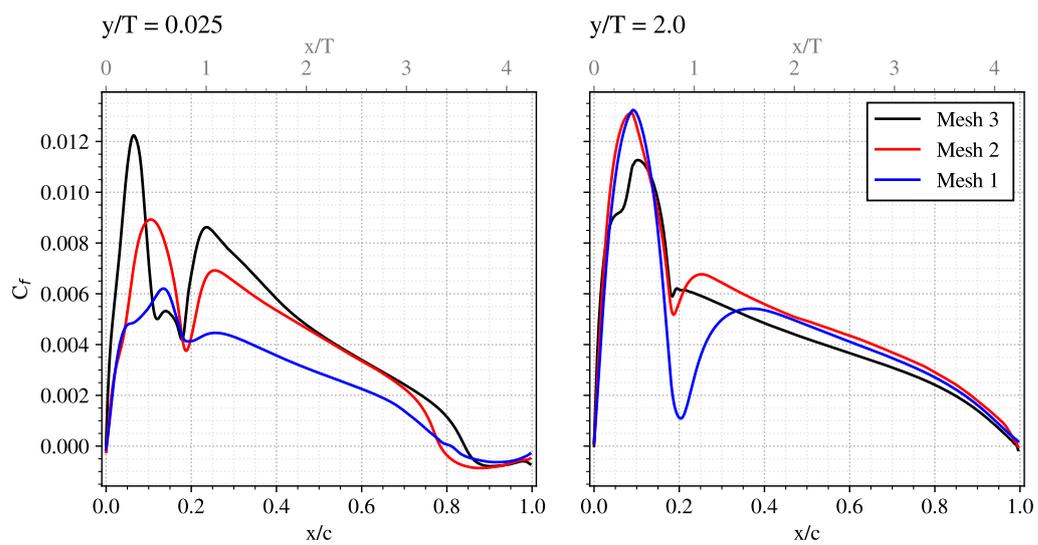


Figure B.4: Skin friction coefficient profiles of the wing for the different meshes at two y -levels.



Simulation Residual Plots

This appendix displays the residual plots of the simulations performed in this thesis.

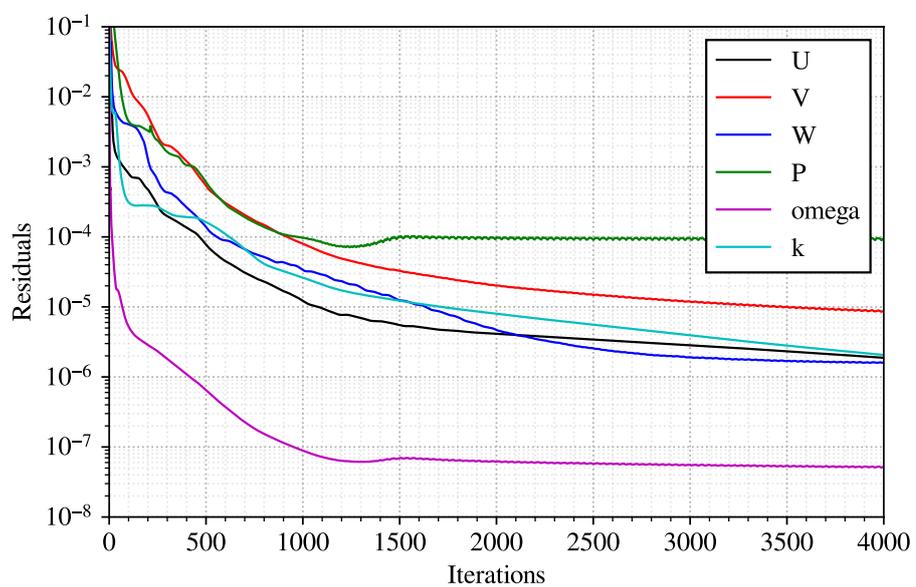


Figure C.1: Residuals of the baseline RANS simulation.

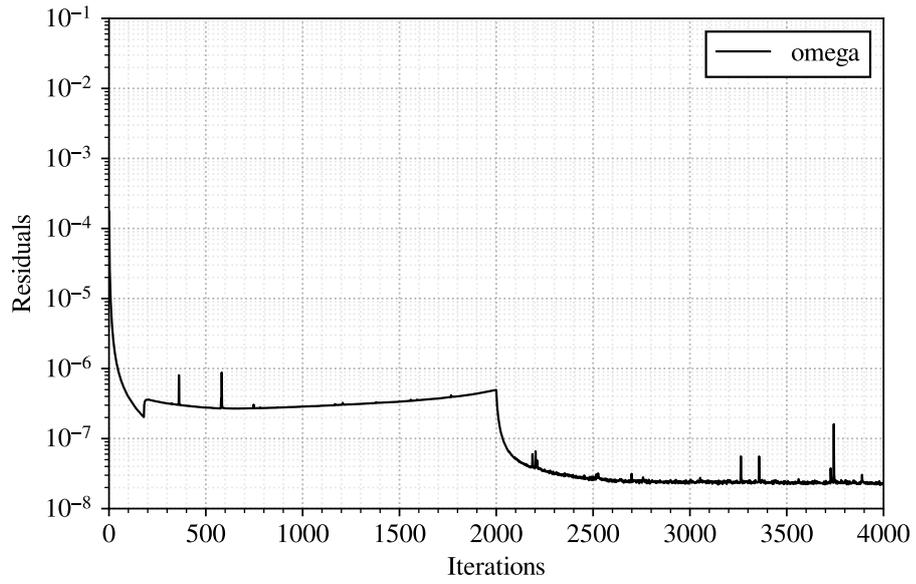


Figure C.2: Residual of the k -corrective frozen simulation.

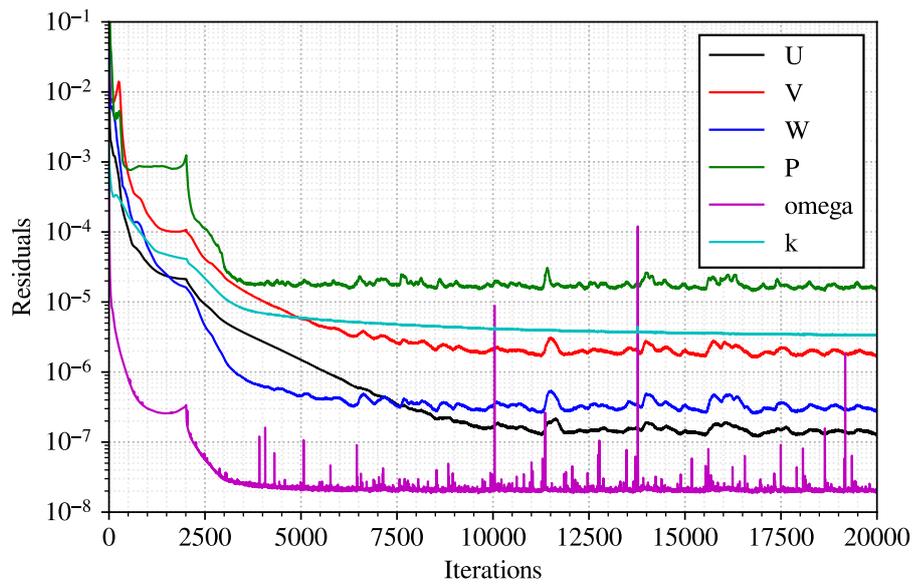


Figure C.3: Residuals of the propagation simulation.

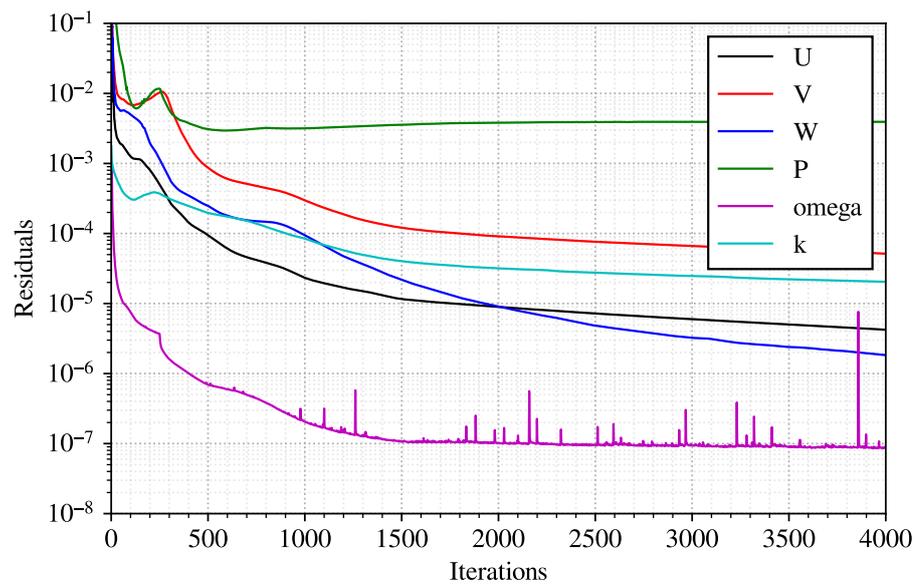


Figure C.4: Residuals of the model propagation of R simulation.

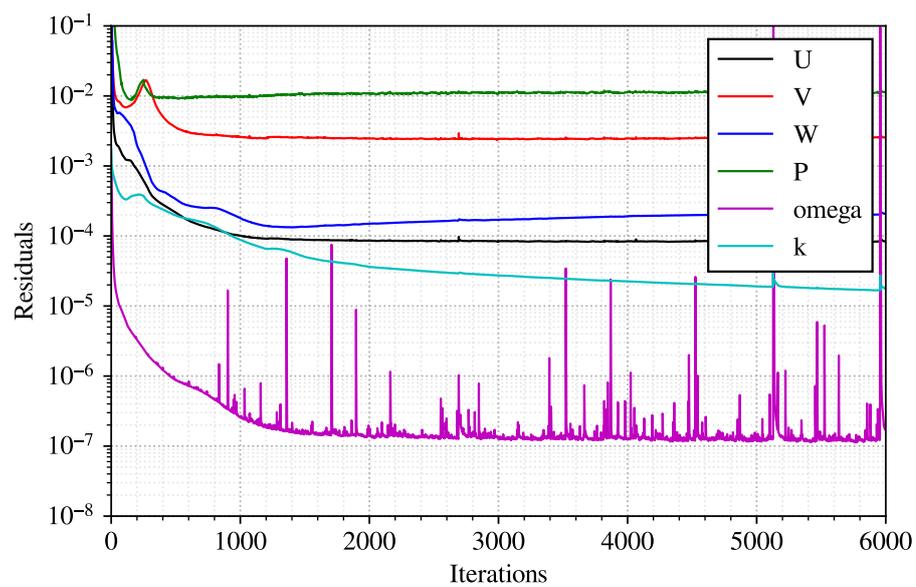


Figure C.5: Residuals of the model propagation simulation.

D

Code Implementations

The modification to the k - ω SST turbulence model for the model propagation to account for the newly introduced lower limit of the production term is displayed in the code snippet below. The complete model propagation code is available in Hoefnagel [61].

```
1 // ...
2 // Modification: the production term G2 will be always equal or larger than 0
3 volScalarField GbyNu(dev(twoSymm(tgradU())) && tgradU());
4 volScalarField::Internal G(this->GName(), nut()*GbyNu);
5 volScalarField G2
6 (
7     "G2",
8     max(0.0*nut*GbyNu, nut*GbyNu - xi_ * useRST_ * sigma_ * (2*(this->k_)*
9         bijDelta_ && tgradU()))
10 );
11 // ...
```

The removal of the R term in the ω -equation in the turbulence model for the model propagation:

```
1 // ...
2 // Modification: kDeficit term R is omitted from the omega-eq
3 // Turbulent frequency equation
4 tmp<fvScalarMatrix> omegaEqn
5 (
6     fvm::ddt(alpha, rho, omega_)
7     + fvm::div(alphaRhoPhi, omega_)
8     - fvm::laplacian(alpha*rho*this->DomegaEff(F1), omega_)
9     ==
10    alpha()*rho()*gamma
11    *min(
12        // Production modified due to RST correction
13        G2 / nut(),
14        (this->c1_/this->a1_)*this->betaStar_*omega_()
15        *max(this->a1_*omega_(), this->b1_*F23()*sqrt(S2()))
16    )
17    - fvm::SuSp((2.0/3.0)*alpha()*rho()*gamma*divU, omega_)
18    - fvm::Sp(alpha()*rho()*beta*omega_(), omega_)
19    - fvm::SuSp
20    (
21        alpha()*rho()*(F1() - scalar(1))*CDkOmega()/omega_(),
22        omega_
23    )
24    + this->Qsas(S2(), gamma, beta)
25    + this->omegaSource()
26    + fvOptions(alpha, rho, omega_)
```

```

27     );
28     // ...

```

Below the two Python functions that were used to perform the far-field spatial blending and the spatial blending close to the wing (`blending()` and `LSB_blending()` respectively). In the model propagation, the flowfield in each iteration is passed from the OpenFOAM solver to a Python interpreter that calculates the value of R and b_{ij}^{Δ} based on the flow field and the model equations (Hoefnagel [61]). Therefore, the functions shown below were inserted in the Python routine just before the arrays for R and b_{ij}^{Δ} were passed back to the OpenFOAM solver.

```

1  def blending(y, x, bijD=None, kD=None):
2      """
3          This function applies the spatial blending towards the top and outlet of
4          the domain
5
6          Args:
7              y (numpy array): Cell y values
8              x (numpy array): Cell x values
9              bijD (numpy array): bijDelta
10             kD (numpy array): kDeficit
11             """
12     y_start = 0.2 # hardcoded; y > y_start -> blending (top domain)
13     y_max = 0.281973
14
15     x_start = 0.9 # hardcoded; x > x_start -> blending (outlet)
16     x_max = 1.022
17
18     factor_y = (y - y_max) / (y_start - y_max)
19     factor_x = (x - x_max) / (x_start - x_max)
20     factor_x[factor_x < 0.0] = 0.0
21     mask_y = (y >= y_start)
22     mask_x = (x >= x_start)
23
24     # apply blending functions
25     if kD is not None:
26         kD[mask_y] *= factor_y[mask_y]
27         kD[mask_x] *= factor_x[mask_x]
28     if bijD is not None:
29         # Note!: bijD has shape (6, Npoints)!!!
30         bijD[:, mask_y] *= factor_y[mask_y][np.newaxis, :]
31         bijD[:, mask_x] *= factor_x[mask_x][np.newaxis, :]
32
33     # to clean memory space, gc.collect() (collection) occurs outside of this
34     function
35     del y_start, x_start, y_max, x_max, factor_y, factor_x, mask_y, mask_x
36
37     if kD is not None:
38         return kD
39     if bijD is not None:
40         return bijD
41
42     def LSB_blending(walldist, y, bijD=None, kD=None):
43         """
44             This function applies the spatial blending on the wing to filter the LSB
45
46             Args:
47                 walldist (numpy array): Cell walldistance
48                 y (numpy array): Cell y values
49                 bijD (numpy array): bijDelta
50                 kD (numpy array): kDeficit

```



```

python library by the research group of Dr. R. P. Dwight.
3 Author: Matthijs van Ede
4 """
5
6 ## Importing libraries
7 import sys
8 import json
9 import numpy as np
10 import sparta.util_plot as util_plot
11 from myfunc import boxing, print_settings
12 from sparta import util
13 from sparta.features import FlowFeatures
14 from sparta.discovery import SymbolicRegressionElasticNet
15 from sparta.library import CandidateLibraryScalarPrincipal
16 from sparta.transform import TransArray, TransformScaler, transform_db
17 from sparta.model import ModelSort, print_models, compute_score
18
19
20 def model_bijD(bijDeltaEq, _flow):
21     # This function evaluates the bijDeltaEq using _flow
22     featureDict = _flow.vv # Double check if its OK to use this before
23     transformScaler
24     featureDict['const'] = lambda : np.ones(_flow.vv["k"].shape) # Can use any
25     scalar feature for shape
26     funcDict = {'exp': np.exp, 'abs': np.abs, 'tanh': np.tanh, #Numpy
27     'rdiv': util.rdiv, 'rlog': util.rlog, 'sqrt_abs': util.
28     sqrt_abs} #Custom
29     bijD_model = eval(bijDeltaEq, funcDict, featureDict)
30     return bijD_model.T
31
32 def Pk_bijD(bijD, _flow):
33     # This function calculates Pk_bijD from bijDelta (bijD)
34     # from openfoam code:
35     # nut * GbyNu - xi_ * userST_ * sigma_ * (2*(this->k)*bijDelta_ && tgradU()
36     )
37     # from Smelzer et al:
38     # Pk_bijD = -2k * bijD_ij * (dU_i/dx_j)
39     _bijD = util.as_fulltensor(bijD)
40     _k = _flow.vv["k"]
41     _gradU = _flow.vv["gradU"]
42     return -2.0 * np.einsum("n,nij,nij->n", _k, _gradU, _bijD)
43
44 ## Settings =====
45 input_file_name = \
46     "/home/mvanede/scratch/MachineLearning/sparta_iM_yp1_extra/inputs_bijD.
47     json"
48
49 # Inputs are read from "inputs.json"
50 input_file = open(input_file_name)
51 inputs = json.load(input_file)
52 input_file.close()
53
54 froz_data_path = inputs["froz_data_path"] # Path to frozen data
55 thinning_factor = inputs["thinning_factor"] # 0.5 -> 50% thinning
56 box_coords = inputs["box_coord"] # Box coordinates
57 target_name = inputs["target_name"] # Name of the target data
58 lib_degree = inputs["lib_degree"] # Degree in generating library
59 system_mem = inputs["system_mem_gb"] # Available memory in gb
60 max_complex = inputs["max_complex"] # Maximum complexity

```

```

58 p_vars          = inputs["p_vars"]          # Principle variables
59 read_lib        = inputs["read_lib"]       # Read library (bool)
60 lib_path        = inputs["lib_path"]       # Path to library file
61
62 print_settings(inputs)
63 sys.stdout.flush()
64 # =====
65
66 ## Reading flow field
67 flow = FlowFeatures.from_openfoam(openfoam_casepath=froz_data_path,
68                                 verbose=True)
69
70 ## Thin and box the flow
71 print("Boxing and thinning flow befor setting up features")
72 flow = boxing(flow=flow, box_coord=box_coords) # Boxing the flow
73 flow.thin(factor=thinning_factor)           # thinning flow
74
75 # define func. for all features (lazily)
76 flow.setup_features(WallinTimeScale=True)
77
78 ## Selecting principle and secondary variables
79 # s_vars = flow.invariant_names + flow.scalarfeature_names
80 # These were found with Mutual Information:
81 s_vars = ['S2', 'W2', 'S2_s', 'W2_s', 'q_gamma', 'q_nu', 'q_Q', 'q_Re']
82
83 vars = p_vars + s_vars
84 print(vars)
85 sys.stdout.flush()
86
87 ## Determine scale factors of the features
88 vars_rm_list = []
89 for var in vars:
90     v = flow.vv[var]
91     std = np.std(v)
92     # checking if std and v are 'reasonable', if not remove them
93     if (std < 1.e-10) or (util.is_nan_or_inf(v)):
94         vars_rm_list.append(var)
95         transform_db[var] = TransformScaler(std)
96
97 if target_name == "kDeficit":
98     kDeficit_std = np.std(flow.vv[target_name])
99
100 ## Scaling the features
101 for var in vars:
102     v = flow.vv[var]
103     flow.vv[var] = TransArray(v, transform_db[var])
104
105 ## Define target
106 if (target_name == "kDeficit"):
107     target = TransArray(flow.vv[target_name],
108                        TransformScaler(kDeficit_std))
109 else: # -> bijDelta
110     target = TransArray(flow.vv[target_name],
111                        TransformScaler(1))
112
113
114 ## Removing the bad variables from p_vars and s_vars
115 print(f'Removing variables: {vars_rm_list} due to nan/inf or too small std')
116 p_vars = [pi for pi in p_vars if pi not in vars_rm_list]
117 s_vars = [si for si in s_vars if si not in vars_rm_list]
118 vars = p_vars + s_vars

```

```

119
120
121 ## Build the regression library
122 lib = CandidateLibraryScalarPrincipal(
123     flow=flow,
124     principal_names=p_vars,
125     secondary_names=s_vars,
126     target=target
127 )
128 # Generate, evaluate and filter library
129 if read_lib:
130     lib.read(
131         filename=lib_path
132     ).evaluate(
133         available_memory_gb=system_mem
134     ).filter(
135         available_memory_gb=system_mem
136     )
137 else:
138     lib.generate(
139         degree=lib_degree
140     ).evaluate(
141         available_memory_gb=system_mem/2
142     ).filter(
143         available_memory_gb=system_mem/2
144     )
145 # Write library to text file
146 lib.write(filename="library.txt")
147
148
149 ## Defining the regression matrices
150 if lib.Beval.ndim == 3: # Tensor regression -> bijDelta
151     num_functions, num_locations, num_components = lib.Beval.shape
152     X = lib.Beval.reshape(num_functions, -1).T # reshape to desired form
153     y = lib.target.q().reshape(-1) # reshape to desired form
154     if target_name == "bijDelta":
155         print("Weighing by k")
156         _weight = flow.vv["k"]
157         weight = (_weight[:,np.newaxis] * np.ones((1,6))).reshape(-1)
158 elif lib.Beval.ndim == 2: # Scalar regression -> kDeficit
159     X = lib.Beval.T # Beval is generated with transformed
160     y = lib.target.q() # q() func gives transformed array
161     print("check if lib.target == target:", np.sum(lib.target - target))
162     weight = None
163
164 if weight is not None:
165     print("Weighting is applied!")
166
167 ## Regression
168 regression = SymbolicRegressionElasticNet(
169     library=lib,
170     target_transform= target.op,
171     # max_iter= ...,
172     # l1_ratios= ...,
173     # alphas= ...,
174     verbose=True
175 )
176 # model discovery
177 models = regression.discover(
178     X=X,
179     y=y,

```

```
180     scaling=lib.B_scaling,
181     weight=weight
182 )
183
184 # Sort the discovered models
185 modelsort = ModelSort(models)
186 modelsort.cull_degenerate()
187 modelsort.cull_low_score(flow, lib.target, threshold=0.1)
188 modelsort.cull_complex(max_complex)
189 modelsort.uniqueify(flow, lib.target, sort='mse') # useless !, only usefull if
190     after refitting
191 modelsort.sort_complexity_mse(flow, lib.target)
192
193 # print and plot models before refitting
194 print_models(modelsort.models, [flow], [lib.target], ['Train'])
195 util_plot.scatter(modelsort.models, [flow], [lib.target], 'pre_scatter.png')
196 util_plot.model_matrix(modelsort.models, flow, lib.target, 'pre_matrix.png')
197
198 # Refit and print models
199 modelsort.refit(flow, lib.target)
200 print_models(modelsort.models, [flow], [lib.target], ['Train'])
201
202 ## Plotting
203 util_plot.scatter(modelsort.models, [flow], [lib.target], 'post_scatter.png')
204 util_plot.model_matrix(modelsort.models, flow, lib.target, 'post_matrix.png')
```