

Master of Science Thesis

---

# Development of a hybrid RANS/LES model for transitional boundary layers

J.D. Steenbeek

---

August 17, 2017





# **Development of a hybrid RANS/LES model for transitional boundary layers**

Master of Science Thesis

For obtaining the degree of Master of Science in Aerospace Engineering  
at Delft University of Technology

J.D. Steenbeek

August 17, 2017



**Delft University of Technology**

Copyright © Aerospace Engineering, Delft University of Technology  
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY  
DEPARTMENT OF AERODYNAMICS

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance the thesis entitled “**Development of a hybrid RAN-S/LES model for transitional boundary layers**” by **J.D. Steenbeek** in fulfillment of the requirements for the degree of **Master of Science**.

Dated: August 17, 2017

Supervisors:

---

Prof. dr. S. Hickel

---

Dr. ir. A.H. van Zuijlen

---

Dr. F. Avallone

---

Ir. J. Kok



---

# Acknowledgement

This report presents and describes the work of my thesis study and is the final part of obtaining a Master's degree in Aerodynamics at the faculty of Aerospace Engineering in Delft.

I would like to express my gratitude to my supervisors Prof. dr. Stefan Hickel, and Dr. ir Sander van Zuijlen for their supervision and constructive discussions. Your critical remarks and questions, have made me even more interested in aerodynamic research, thank you.

I would also like to thank my parents for their endless support and patience during my time in Delft. And special thanks to my girlfriend, Ellen for her pleasant distraction and for nominating my colourful fluid dynamics as art.



---

# Abstract

Hybrid RANS/LES models combine the accuracy of Large-Eddy Simulations (LES) with the cost efficiency of Reynolds-Averaged Navier-Stokes (RANS). However, these high-fidelity methods are unable to accurately predict the transition from laminar to turbulent boundary layer flows. Usually resulting in an incorrect prediction of boundary layer characteristics.

In this study, a hybrid RANS/LES model was developed for transitional boundary layer flows. The transitional  $kk_L - \omega$  (Walters and Cokljat, 2008) turbulence model was combined with the RANS/LES blending function from the framework of the Improved Delayed Detached Eddy Simulation (IDDES) (Shur et al., 2008). Initial simulations predicted a transition location that agreed with experimental data. However, development of turbulent structures were delayed. Therefore, velocity fluctuations were added to the flow at the location of transition. The intensity of the velocity fluctuations were based on the modelled kinetic energy that is present in the pretransitional boundary layer. The addition of the velocity fluctuations, accelerated the development of turbulence and reduced the classical *grey area* problem.

Flat-plate simulations with different free-stream turbulent intensities all predicted a transition location that agrees with experimental data. The predicted skin-friction in the transitional boundary layer improved with decreasing free-stream turbulence levels. Simulation of a flow over a NACA 0012 airfoil predicted a skin-friction distribution and trailing edge displacement thickness that was superior compared to an existing non-transitional hybrid RANS/LES model. Predicted skin-friction values of a DU91W250 wind turbine airfoil agreed with the  $kk_L - \omega$  RANS results.

Right after transition, the kinetic energy of the boundary layer was underpredicted by both simulations. This problem might be solved when the introduced velocity fluctuations more accurately represent the turbulent structures of the pretransitional boundary layer.





---

# Table of Contents

<b>Acknowledgement</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xxi</b>
<b>Nomenclature</b>	<b>xxiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Goal . . . . .	2
1.3 Project set-up . . . . .	3
<b>2 Flow Physics and Turbulence Modelling</b>	<b>5</b>
2.1 Boundary-layer concept . . . . .	5
2.2 Governing flow equations . . . . .	10
2.3 RANS . . . . .	11
2.3.1 SA model . . . . .	12
2.3.2 Physics based transitional $kk_L - \omega$ model . . . . .	13
2.4 LES . . . . .	17
2.5 Hybrid RANS/LES methods . . . . .	18
2.5.1 Spalart-Allmaras DES . . . . .	19
2.5.2 $kk_L - \omega$ SST-DES . . . . .	23

<b>3</b>	<b>Assessment of Hybrid RANS/LES Methods</b>	<b>25</b>
3.1	Simulation models . . . . .	25
3.1.1	Simulation 1: SA IDDES . . . . .	25
3.1.2	Simulation 2: modified SA IDDES . . . . .	26
3.1.3	Simulation 3: modified SA IDDES, 2D tripping . . . . .	26
3.1.4	Simulation 4: $kk_L - \omega$ SST-DES . . . . .	26
3.2	Flat-plate test case . . . . .	26
3.3	Results . . . . .	28
3.4	Interim summary . . . . .	30
<b>4</b>	<b>Assessment of <math>kk_L - \omega</math> RANS</b>	<b>31</b>
4.1	Simulation details . . . . .	31
4.2	Results . . . . .	32
<b>5</b>	<b>Proposed Transitional Hybrid RANS/LES Model</b>	<b>39</b>
5.1	Modification 1: turbulent length scale and RANS elevation function . . . . .	39
5.1.1	Selection of LES destruction term . . . . .	40
5.1.2	RANS/LES blending . . . . .	41
5.1.3	Near-wall elevation function . . . . .	42
5.1.4	Low-Reynolds-number correction . . . . .	43
5.1.5	Evaluation for flat-plate boundary layers . . . . .	44
5.2	Modification 2: background transition model . . . . .	45
5.3	Modification 3: LES limiter and damping term correction . . . . .	48
5.4	Modification 4: Addition of tripping mechanism . . . . .	51
5.5	Model summary . . . . .	56
<b>6</b>	<b>Verification and Validation</b>	<b>59</b>
6.1	Effect of tripping intensity . . . . .	59
6.2	RANS elevating-function . . . . .	62
6.3	Mesh sensitivity analysis . . . . .	63
6.4	Validation . . . . .	65
6.4.1	Simulation details . . . . .	65
6.4.2	Results . . . . .	65
<b>7</b>	<b>Test Case: NACA 0012 Airfoil</b>	<b>69</b>
7.1	Numerical grid . . . . .	69
7.2	Boundary conditions and solver settings . . . . .	70
7.3	Results . . . . .	71
7.3.1	Blending functions . . . . .	71

7.3.2	Pressure and skin-friction . . . . .	72
7.3.3	Boundary layer characteristics . . . . .	73
7.3.4	Turbulent kinetic energy profiles . . . . .	74
7.3.5	Vorticity contours . . . . .	77
7.3.6	Surface noise sources . . . . .	78
<b>8</b>	<b>Test Case: DU91W250 Airfoil</b>	<b>79</b>
8.1	Numerical grid . . . . .	79
8.2	Simulation details . . . . .	81
8.2.1	Boundary conditions . . . . .	81
8.2.2	Solver settings . . . . .	81
8.2.3	Computational resources . . . . .	82
8.3	Results . . . . .	83
8.3.1	Blending functions . . . . .	83
8.3.2	Pressure and skin-friction . . . . .	84
8.3.3	Boundary layer thickness . . . . .	86
8.3.4	Turbulent kinetic energy profiles . . . . .	87
<b>9</b>	<b>Conclusion and Recommendations</b>	<b>93</b>
9.1	Conclusion . . . . .	93
9.2	Recommendations . . . . .	96
	<b>Bibliography</b>	<b>97</b>
<b>A</b>	<b>Mesh generation procedure</b>	<b>1</b>
<b>B</b>	<b>Substitution of divergence free fluctuations</b>	<b>9</b>
<b>C</b>	<b>Vorticity contours DU91W250</b>	<b>11</b>
<b>D</b>	<b>Acoustics of DU91W250</b>	<b>13</b>
<b>E</b>	<b>Modified PISO solver</b>	<b>19</b>
<b>F</b>	<b><math>kk_L - \omega</math> IDDES OpenFOAM implementation</b>	<b>21</b>



---

## List of Figures

2.1	Vorticity contours (Q-iso=1e4) of a transitional boundary layer over a flat-plate. Coloured with the velocity magnitude. . . . .	5
2.2	Non-dimensional boundary layer profile for a zero-pressure gradient flat-plate. Graph from White by Lindgren(1965). . . . .	6
2.3	Turbulent intensities inside the turbulent boundary layer. Graph from Klebanoff (Klebanoff, 1954). . . . .	7
2.4	Energy spectrum inside a turbulent boundary layer at different locations from the wall (from White (2006) by Klebanoff (1954)). . . . .	7
2.5	Horsehoe vortex as described by Theodorsen (1952). . . . .	8
2.6	Schematic representation of the transitional boundary layer process (Figure from White White (2006)) . . . . .	9
2.7	Chordwise based Reynolds number at which transition occurs due to free-stream turbulence. Solid line: Andersson, Stars: van Driest model, Circle: experimental. Andersson et al. (1991). . . . .	10
2.8	Schematic representation of the <i>grey area</i> problem that can be found in hybrid RANS/LES simulations. <i>Left</i> : ideal case. <i>Right</i> : real case, <i>grey area</i> is marked orange. By Michel et al. (2010) . . . . .	19
2.9	Profiles of the RANS/LES blending function $f_B$ and the RANS elevating function $f_{e1}$ . . . . .	22
3.1	Side view of computational mesh. Left: inlet. Right: outlet. Top: free-stream. Bottom: symmetry plane and wall. . . . .	27
3.2	Wall friction coefficient. Blue line: SA. Circles: experimental. Black line: turbulent theory. Black dashed line: laminar theory. <i>Left</i> : original SA IDDES model. <i>Middle</i> : modified SA IDDES. <i>Right</i> : modified SA IDDES with 2D tripping. . . . .	28
3.3	Hybrid RANS/LES blending function from the IDDES. LES=0, RANS=1. White dashed line: 99 percent boundary layer thickness. . . . .	29

3.4	<i>Left</i> : Wall friction coefficient. Blue line: $kk_L - \omega$ SST-DES. Circles: experimental. Black line: turbulent theory (White, 2006). Black dashed line: laminar theory. <i>Right</i> : hybrid RANS/LES blending function from SST-DES. RANS=1, LES=0. White dashed line: 99 percent boundary layer thickness. . . . .	30
4.1	RANS wall friction coefficient along flat-plate for different free-stream turbulence intensities. Blue line: $kk_L - \omega$ . Circles: experimental (Coupland, 1990). Black solid line: turbulent theory (White, 2006). Black dashed line: Blasius theory. . . . .	32
4.2	Decomposition of the kinetic energy fields. <i>Top</i> : $Tu = 6\%$ . <i>Middle</i> : $Tu = 3\%$ . <i>Bottom</i> : $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness. . . . .	33
4.3	Turbulent kinetic energy damping functions. <i>Left</i> : turbulent length scale limiter. <i>Right</i> : shear sheltering limiter. <i>Top</i> : $Tu = 6\%$ . <i>Middle</i> : $Tu = 3\%$ . <i>Bottom</i> : $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness. . . . .	34
4.4	Decomposed large-scale eddy viscosity. <i>Left</i> : bypass production. <i>middle column</i> : natural production. <i>Right</i> : total. <i>Top</i> : $Tu = 6\%$ . <i>Middle row</i> : $Tu = 3\%$ . <i>Bottom</i> : $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness. . . . .	35
4.5	Transitional energy transfer from $k_L$ to $k_T$ . <i>Left</i> : bypass transition. <i>Right</i> : natural transition. <i>Top</i> : $Tu = 6\%$ . <i>Middle row</i> : $Tu = 3\%$ . <i>Bottom</i> : $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness. . . . .	36
4.6	Dissipation damping functions. <i>Left</i> : viscous wall effect. <i>Right</i> : intermittency function. <i>Top</i> : $Tu = 6\%$ . <i>Middle row</i> : $Tu = 3\%$ . <i>Bottom</i> : $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness. . . . .	37
4.7	Decomposition of large- and small-scale eddy viscosity. <i>Left</i> : large-scale eddy viscosity. <i>middle column</i> : small-scale eddy viscosity. <i>Right</i> : total eddy viscosity. <i>Top</i> : $Tu = 6\%$ . <i>Middle row</i> : $Tu = 3\%$ . <i>Bottom</i> : $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness. . . . .	38
5.1	Schematic decomposition of RANS/LES regions by $f_d$ , for a flat-plate simulation. RANS=yellow. LES=blue. White dashed line: 99% boundary layer thickness. . . . .	41
5.2	Elevating function $\Psi$ . Blue line: new elevating function. Circle's: data points from Spalart et al. (2006). . . . .	42
5.3	Wall skin-friction coefficient along the flat-plate. . . . .	44
5.4	Kinetic energy fields above the flat-plate. <i>Left</i> , laminar kinetic energy. <i>Middle</i> , turbulent kinetic energy. <i>Right</i> , total modelled kinetic energy. White dashed line: 99% boundary layer thickness. . . . .	45
5.5	The shear-sheltering function ( <i>left</i> ) returns a fully laminar boundary layer. Throughout the majority of the laminar boundary layer the LES branch is activated by the hybrid blending function ( <i>right</i> ), RANS=1 and LES=0. White dashed line: 99% boundary layer thickness. . . . .	45

5.6	Wall skin-friction coefficient along the flat-plate. . . . .	46
5.7	Vorticity contours (Q-iso=1e4) coloured with velocity magnitude. . . . .	47
5.8	The shear-sheltering function ( <i>left</i> ) indicates premature transition at $x=0.9$ [m] and the hybrid blending function ( <i>right</i> ) shows that the LES branch is active in the laminar boundary layer, RANS=1 and LES=0. White dashed line: 99% boundary layer thickness. . . . .	47
5.9	Excessive damping in the turbulent boundary layer is found in the viscous wall function ( <i>left</i> ) and intermittency function ( <i>right</i> ). White dashed line: 99% boundary layer thickness. . . . .	48
5.10	Schematic decomposition of the expected RANS/LES regions for a flat-plate simulation. RANS=yellow. LES=blue. . . . .	49
5.11	Wall skin-friction coefficient along the flat-plate. . . . .	50
5.12	Vorticity contours (Q-iso=1e4) coloured with velocity magnitude. . . . .	50
5.13	<i>Left</i> : shear-sheltering function. Laminar=0, turbulent=1. <i>Middle</i> : RANS/LES blending by shear-sheltering branch. <i>Right</i> : total RANS/LES blending. LES=0, RANS=1. White dashed line: 99% boundary layer thickness. . . . .	51
5.14	Schematic decomposition of RANS/LES regions for a flat-plate simulation. RANS=yellow. LES=blue. . . . .	53
5.15	<i>Left</i> : shear-sheltering function. Laminar=0, turbulent=1. <i>Middle</i> : RANS/LES blending by shear-sheltering branch. <i>Right</i> : total RANS/LES blending. LES=0, RANS=1. White dashed line: 99% boundary layer thickness. . . . .	54
5.16	Close-up of transition point. <i>Left</i> : RANS/LES blending by shear-sheltering branch determines the location for $u'$ . <i>Right</i> : substituted velocity fluctuation $\sqrt{u'^2}$ , located at the outflow edge of $f_{SS,d}$ . White dashed line: 99% boundary layer thickness. . . . .	55
5.17	Wall skin-friction coefficient along the flat-plate. . . . .	55
5.18	Vorticity contours (Q-iso=1e4) coloured with velocity magnitude. Small structures in the region between $x \approx 1.2\text{m}$ and $x \approx 1.4\text{m}$ are the added velocity fluctuations of the tripping mechanism. . . . .	55
6.1	Wall friction coefficient with $Tu = 0.8\%$ for different values of the tripping intensity $\alpha$ . . . . .	60
6.2	Wall friction coefficient with $Tu = 3\%$ for different scaling factors (alpha) of the tripping intensity. . . . .	60
6.3	Vorticity contours (Q-iso=1e4) coloured with flow velocity. $Tu = 0.8\%$ . <i>Top</i> : $\alpha = 0.25$ . <i>Middle</i> : $\alpha = 1$ . <i>Bottom</i> : $\alpha = 4$ . . . . .	61

6.4	Boundary layer thickness ( $\delta_{99\%}$ ) along flat-plate for different intensities of the tripping scale $\alpha$ . . . . .	62
6.5	Effect of IDDES elevating function $f_e$ on the non-dimensional velocity profile at $x = 2.5\text{m}$ . . . . .	62
6.6	Effect of model coefficients on the non-dimensional velocity profile at $x = 2.5\text{m}$ . . . . .	63
6.7	Wall friction of transitional and turbulent boundary layer. $Tu = 0.8\%$ . . . . .	64
6.8	Wall friction of transitional and turbulent boundary layer. $Tu = 3\%$ . . . . .	64
6.9	RANS/LES blending function $f_d$ . RANS=1, LES=0. White dashed line: 99% boundary layer thickness. . . . .	66
6.10	Wall friction coefficient along flat-plate. Blue line: $kk_L - \omega$ -IDDES. Circles: experimental. Black line: turbulent theory White (2006). Black dashed line: laminar theory, Blasius. . . . .	66
6.11	Boundary layer thickness. Blue line: $kk_L - \omega$ -IDDES. Circles: experimental. . . . .	67
6.12	Kinetic energy profiles. <i>Left</i> : resolved turbulent energy. <i>Right</i> : modelled plus resolved turbulent kinetic energy. . . . .	67
7.1	Size view of the NACA 0012 simulation domain. . . . .	70
7.2	Predicted decay of turbulent parameters in the farfield. <i>Blue line</i> : decay of parameter. <i>Red circle</i> : desired ambient condition. . . . .	71
7.3	<i>Left</i> : shear sheltering function. Turbulent=1, laminar=0. <i>Right</i> : hybrid blending function. RANS=1, LES=0. . . . .	72
7.4	Free-stream turbulence decay from inlet to airfoil. . . . .	72
7.5	Mean chordwise wall friction of the NACA 0012. Tripping intensity $\alpha = 1$ . Pressure and suction side overlap. . . . .	73
7.6	Boundary layer profiles of upper side of NACA 0012. <i>Left</i> : boundary layer thickness. <i>Right</i> : displacement thickness. . . . .	73
7.7	Decomposition of the modelled and resolved kinetic energy profiles of the NACA 0012 boundary layer. . . . .	75
7.8	Kinetic energy profiles of the NACA 0012 boundary layer obtained from the IDDES and RANS simulation. RANS results have been decomposed in laminar and turbulent kinetic energy profiles. . . . .	75
7.9	Instantaneous profiles of the small-scale eddy viscosity $\nu_{t,s}$ , natural $\nu_{t,l_{NAT}}$ and bypass $\nu_{t,l_{BP}}$ eddy viscosity. . . . .	76
7.10	Magnitude of the introduced velocity fluctuations by the tripping mechanism. Upper side of NACA 0012. White dashed line: 99% boundary layer thickness. . . . .	76



7.11	Vorticity contours (Q-iso=2e-5) around the NACA 0012 by $kk_L - \omega$ IDDES. Coloured with the velocity magnitude. . . . .	77
7.12	Vorticity contours (Q-iso=2e-5) around the NACA 0012 with the modified SA IDDES. Coloured with the velocity magnitude. . . . .	77
7.13	. . . . .	78
8.1	Spanwise cell distribution for 10% span. Blue = 1 cell. Gray = 32 cells. Red = 128 cells in spanwise direction. . . . .	80
8.2	Courant number near the trailing edge of the DU91W250 airfoil for a time step of 1.8e-6 seconds. $AoA = 6^0$ . . . . .	82
8.3	Scaling on SuperMUC with OpenFOAM v4.1 . . . . .	82
8.4	Blending functions <i>Left column</i> : shear sheltering function. Turbulent=1, laminar=0. <i>Right column</i> : RANS/LES blending function. RANS=1, LES=0. <i>Top row</i> : $AoA = 6^0$ and tripping intensity $\alpha = 1$ . <i>Bottom row</i> : $AoA = -2.85^0$ and tripping intensity $\alpha = 1$ . . . . .	83
8.5	Free-stream turbulence decay from inlet to airfoil, $AoA = 6^0$ . . . . .	84
8.6	Mean pressure distribution along surface chord. <i>Left</i> : Tripping intensity $\alpha = 2$ . <i>Right</i> : Tripping intensity $\alpha = 1$ . . . . .	84
8.7	Mean pressure distribution near the transition location. <i>Left</i> : $\alpha = 2$ . <i>Right</i> : $\alpha = 1$ . . . . .	85
8.8	Span-averaged skin-friction coefficient on suction and pressure sides of DU91W250, $AoA = 6^0$ . Tripping intensity alpha = 2. . . . .	85
8.9	Span-averaged skin-friction coefficient on suction and pressure sides of DU91W250, $AoA = -2.85^0$ . Tripping intensity $\alpha = 1$ . . . . .	86
8.10	Thickness of the boundary layer on the upper side. <i>Left</i> : $AoA = 6^0$ and tripping intensity $\alpha = 2$ . <i>Right</i> : $AoA = -2.85^0$ and tripping intensity $\alpha = 1$ . . . . .	86
8.11	Kinetic energy profiles on suction side of DU91, $AoA = 6^0$ . <i>Top</i> : decomposition of the modelled and resolved energy. <i>Bottom</i> : IDDES and RANS comparison. RANS results have been decomposed in their laminar and turbulent parts. . . . .	87
8.12	Kinetic energy profiles on pressure side of DU91, $AoA = 6^0$ . <i>Top</i> : decomposition of the modelled and resolved energy. <i>Bottom</i> : IDDES and RANS comparison. RANS results have been decomposed in their laminar and turbulent parts. . . . .	88
8.13	Kinetic energy profiles on upper side of DU91, $AoA = -2.85^0$ . <i>Top</i> : decomposition of the modelled and resolved energy. <i>Bottom</i> : IDDES and RANS comparison. RANS results have been decomposed in their laminar and turbulent parts. . . . .	88

8.14	Kinetic energy profiles on lower side of DU91W250, $AoA = -2.85^\circ$ . <i>Top</i> : decomposition of the modelled and resolved energy. <i>Bottom</i> : IDDES and RANS comparison. RANS results have been decomposed in their laminar and turbulent parts. . . . .	89
8.15	Instantaneous profiles of the small-scale $\nu_{t,s}$ , natural $\nu_{t,l_{NAT}}$ and bypass $\nu_{t,l_{BP}}$ eddy viscosity near the laminar boundary. . . . .	90
8.16	Magnitude of the introduced velocity fluctuations by the tripping mechanism. <i>Top</i> : tripping intensity $\alpha = 2$ . <i>Bottom</i> : tripping intensity $\alpha = 2$ . White dashed line: 99% boundary layer thickness. . . . .	91
A.1	DU91W250 outline. Dots indicate the zones, based on the expected transition location(A.1). . . . .	2
A.2	stepsize . . . . .	2
A.3	Normal extrude smoothing parameters. . . . .	2
A.4	Third radial segment, covers the boundary layer and will have a high mesh density. . . . .	3
A.5	Second radial segment, acts a the transition between the thrid and first radials segment. . . . .	3
A.6	First radial segment will have a low mesh density. . . . .	3
A.7	MakeCyclicPlane . . . . .	4
A.8	Bounary conditions. . . . .	5
A.9	Volume conditions. . . . .	5
A.10	Solver attributes. . . . .	5
A.11	Correct representation of cyclic patch . . . . .	6
B.1	Magnitude of initial random velocity fluctuation $u'$ . . . . .	10
B.2	Magnitude of the divergence free velocity fluctuation $u'_{div.-free}$ , lacking randomness in the fluctuation magnitude and location. . . . .	10
C.1	Vorticity contours ( $Q - iso = 3e6$ ) on upper side of DU91W250. $AoA = 6^\circ$ . <i>Top</i> : $kk_L - \omega$ IDDES. <i>Bottom</i> : modified SA IDDES. . . . .	11
C.2	Vorticity contours ( $Q - iso = 3e6$ ) on lower side of DU91W250. $AoA = 6^\circ$ . <i>Top</i> : $kk_L - \omega$ IDDES. <i>Bottom</i> : modified SA IDDES. . . . .	12
C.3	Vorticity contours ( $Q - iso = 3e6$ ) on upper side of DU91W250. $AoA = -2.85^\circ$ . <i>Top</i> : $kk_L - \omega$ IDDES. <i>Bottom</i> : modified SA IDDES. . . . .	12

---

C.4	Vorticity contours ( $Q-iso = 3e6$ ) on lower side of DU91W250. $AoA = -2.85^\circ$ . <i>Top: <math>kk_L - \omega</math> IDDES .Bottom: modified SA IDDES.</i> . . . . .	12
D.1	SPL with 1/3 Octave band filter. $AoA = 6^\circ$ . . . . .	15
D.2	Directivity polar of different frequencies. $AoA = 6^\circ$ . . . . .	16
D.3	SPL for different frequencies. . . . .	16
D.4	Directivity polar of different SPL frequencies. . . . .	17



---

# List of Tables

2.1	Model coefficients of the SA model. . . . .	13
2.2	Blending coefficients of the $kk_L - \omega$ SST-DES model. . . . .	23
3.1	Boundary conditions applied to the RANS and LES case. . . . .	27
4.1	Inflow conditions for the three different simulations. Conditions correspond to experimental simulation from ERCOFTAC by Coupland (1990). . . . .	31
5.1	Turbulent length scale and destruction term for the hybrid $kk_L - \omega$ IDDES model. . . . .	42
6.1	New RANS elevating constants . . . . .	63
6.2	Overview of the turbulent inlet boundary conditions for the three simulations. . . . .	65
7.1	Boundary conditions for the $kk_L - \omega$ SA simulations. $Tu = 0.161\%$ . . . . .	70
7.2	Displacement thickness at the trailing edge of NACA 0012. . . . .	74
8.1	Mesh summary for DU91W250 mesh after refinement, $Re$ of 3.2 Million, span = 10% of chord, $\alpha = -2.85^\circ$ and $6^\circ$ . . . . .	80
8.2	Boundary conditions applied to the RANS and IDDES case. . . . .	81
A.1	Predicted transition locations from XFOIL for DU91W250, $n_{crit} = 9$ . . . . .	1
A.2	Mesh quality. . . . .	7



---

# Nomenclature

## Acronyms

CFD	computational fluid dynamics
DES	detached-eddy simulation
DDES	delayed detached-eddy simulation
DNS	direct numerical simulation
ERCOFTAC	European research community on flow, turbulence and combustion
IDDES	improved delayed detached-eddy simulation
LES	large-eddy simulation
LLM	log-layer mismatch
NACA	national advisory committee for aeronautics
PISO	pressure-implicit with splitting of operators
RANS	Reynolds-Averaged Navier-Stokes
SA	Spalart-Allmaras
SGS	sub-grid scale
SIMPLE	semi-implicit method for pressure linked equations
SST	shear stress transport
TS	Tollmein-Schlichting
WMLES	wall-modelled LES

## Roman Symbols

$C_P$	pressure coefficient
$C_f$	skin-friction coefficient
$C_\mu$	turbulent viscosity coefficient
$C_{DES}$	sub-grid scale coefficient of DES

$C_S$	Smagorinsky sub-grid scale constant	
$D_L$	large-scale anisotropic (near-wall) dissipation term	
$D_T$	small-scale anisotropic (near-wall) dissipation term	
$d_w$	shortest distance to a wall	$m$
$d$	distance	$m$
$f$	force	$N$
$f_\nu$	viscous near-wall damping	
$f_\omega$	boundary layer wake damping	
$f_{\tau,l}$	natural production damping function	
$F_1$	RANS/LES limiting function	
$f_B$	RANS/LES blending by WMLES branch	
$f_{dt}$	RANS/LES blending by DDES branch	
$f_d$	RANS/LES blending function	
$F_{DDES}$	RANS/LES blending function	
$f_e$	RANS elevating function	
$f_{e1}$	grid dependent RANS elevating function	
$f_{e2}$	RANS elevating intensity function	
$f_l$	laminar RANS elevating function	
$f_{NAT,crit}$	natural transition threshold function	
$f_{SS,d}$	RANS/LES blending by shear-sheltering branch	
$f_{INT}$	intermittency function	
$f_{SS}$	shear sheltering function	
$f_t$	turbulent RANS elevating function	
$f_W$	turbulent dissipation damping function	
$h_{max}$	maximum cell size	$m$
$h_{wn}$	wall-normal cell size	$m$
$k$	fluctuating kinetic energy	$m^2/s^2$
$k_L$	laminar kinetic energy	$m^2/s^2$
$k_T$	turbulent kinetic energy	$m^2/s^2$
$k_{T,l}$	effective large-scale turbulence	$m^2/s^2$
$k_{T,s}$	effective small-scale turbulence	$m^2/s^2$
$L$	length	$m$
$l_{DDES}$	DDES length scale	$m$
$l_{hyb}$	hybrid length scale	$m$
$l_{LES}$	LES length scale	$m$
$l_{RANS}$	RANS length scale	$m$



$n_{crit}$		
$p$	pressure	$N/m^2$
$P_{KL}$	production of laminar kinetic energy	$m^2/s^3$
$P_{KT}$	production of turbulent kinetic energy	$m^2/s^3$
$q$	energy flux	$J^1 m^{-2} s^{-1}$
$Q$	balance between $S$ and $\Omega$	$s^{-1}$
$R_{BP}$	bypass transition energy	$m^2/s^3$
$R_{NAT}$	natural transition energy	$m^2/s^3$
$Re$	Reynolds number	
$Re_{\Omega}$	natural transition Reynolds number	
$Re_T$	turbulence Reynolds number	
$Re_x$	Reynolds number based on distance	
$Re_{x,tr}$	Reynolds number at transition	
$Re_{\tau}$	Reynolds number based on friction velocity	
$S$	magnitude of strain-rate	$s^{-1}$
$\tilde{S}$	modified magnitude of strain-rate	$s^{-1}$
$S_{ij}^*$	strain rate tensor	
$t$	time	$s$
$Tu$	turbulence intensity	%
$U$	velocity magnitude	$ms^{-1}$
$u'$	magnitude of fluctuating velocity component	$ms^1$
$u$	velocity x-component	$ms^1$
$v$	velocity y-component	$ms^1$
$V_{outflow}$	edge of laminar region	
$V_{SS,d}$	outflow edge of laminar region	
$w$	velocity z-component	$ms^1$
$x$	distance in streamwise-direction	$m$
$x_{SS,d}$	region limiter based on $f_{SS,d}$ gradient	
$y^+$	normalised wall distance in wall units	
$y$	distance in y-diction	$m$
$z$	distance in spanwise-direction	$m$

## Greek Symbols

$\alpha_T$	Effective turbulent diffusivity	
$\alpha$	scaling for tripping intensity	
$\beta_{BP}$	bypass transition threshold function	
$\beta_{NAT}$	natural transition threshold function	
	SGS filter width	$m$
$\delta$	99% boundary layer thickness	$m$
$\delta^*$	displacement thickness	$m$
$\epsilon$	turbulent dissipation rate	$m^2/s^{-3}$
$\lambda_{eff}$	effective wall-limited turbulent length-scale	$m$
$\lambda_T$	effective turbulent length-scale	$m$
$\mu$	dynamic viscosity	$m^2s^{-1}$
$\nabla$	gradient operator	
$\nu$	kinematic viscosity	$m^2s^{-1}$
$\nu_{SGS}$	sub-grid scale eddy viscosity	$m^2s^{-1}$
$\nu_t$	eddy viscosity	$m^2s^{-1}$
$\nu_{t,l}$	large-scale eddy viscosity	$m^2s^{-1}$
$\nu_{t,s}$	small-scale eddy viscosity	$m^2s^{-1}$
$\tilde{\nu}$	transported eddy viscosity	$m^2s^{-1}$
$\omega$	specific turbulence dissipation rate	$s^{-1}$
$\Omega$	rotation magnitude	$s^{-1}$
$\phi_{BP}$	bypass transition model function	
$\phi_{NAT}$	natural transition model function	
$\Psi$	low-Reynolds number correction term	
$\rho$	material density	$kg^1m^{-3}$
$\tau'$	Reynolds stress tensor	$m^2s^{-2}$
$\tau_{ij}$	sub-grid scale tensor	$m^2s^{-2}$
$\chi$	viscosity ratio	

---

# Chapter 1

---

## Introduction

### 1.1 Motivation

In the last decades, hybrid RANS/LES methods have gained significant attention by the research community. The classical hybrid RANS/LES method employs a Reynolds-Averaged Navier-Stokes (RANS) model to the near-wall regions, while solving separated flows regions using sub-grid scale Large-Eddy Simulations (LES). This method provides a compromise between the accuracy of LES and cost efficiency of RANS.

The Improved Delayed Detached Eddy Simulation (IDDES) (Shur et al., 2008) is a hybrid RANS/LES model that can be used for wall-bounded flows. Not only the separated flow is solved using LES, but also a large part of the turbulent boundary layer. Examples of engineering application are noise predictions of wind turbine blades, heat exchange in axial turbomachinery and stall predictions of helicopter blades. Most of the hybrid RANS/LES methods make use of the Spalart-Allmaras (SA) turbulence model. Even though this turbulence model is developed for flows over aerodynamic bodies, it is incapable of predicting the transitional behaviour of the boundary layer (Castelli et al., 2012), usually predicting in a premature transition location. Various simulations have been performed in which the transition location is not specified in the numerical domain or by a turbulence model (Marsden et al. (2005), Verhoeven (2006) and Greschner et al. (2010)), which result in incorrect predictions of the boundary layer characteristics. There are solutions that force transition at a prescribed location (Wolf and Lele (2011a), Wolf and Lele (2011b), Nebenfuhr (2012) and Winkler and Moreau (2008)). Such methods include suction and blowing through at the transition location or by introducing an obstacle in the laminar boundary layer which generates instabilities and causes the boundary layer to become turbulent. However, this tripping of the flow through modification of the numerical domain have two disadvantages: the transition locations is required in advance and additional time is required to set up a numerical mesh which contains the tripping geometry. Current hybrid RANS/LES methods usually predict incorrect transition locations, which can result in discrepancies in the development of boundary layer, such as thickness, turbulent energy levels and heat exchange.

Development of models that can predict the process of transitional boundary layers are studied for over a century. Although significant progress is made, it remains challenging to construct one single model that can be used for simulations over various geometries and is valid for the all different types of flow states. Recently, Computation Fluid Dynamics (CFD) models ( $\gamma - Re_{\Theta}$  and  $kk_L - \omega$ ) have been proposed that can accurately model both the laminar and turbulent flow states, as well as the transitional process.

A solution for the inability of hybrid RANS/LES models to predict a correct location of transitional flow, could be found in turbulence model that can predict the transitional process of laminar to turbulent flow.

## 1.2 Goal

Currently, no hybrid RANS/LES method exists that can accurately predict the transitional process of a laminar to turbulent boundary layer. Therefore, the goal of this thesis is to explore the possibilities and limitations of combining the existing  $kk_L - \omega$  transition model and the RANS/LES blending framework of IDDES, into a new hybrid RANS/LES model that is able to predict laminar to turbulent transition.

The following specific objectives are defined:

1. Focus on existing hybrid RANS/LES models and the  $kk_L - \omega$  turbulence model: what are their limitations, how do they behave and what are general approaches for combining RANS and LES models?
2. Perform flat-plate simulation with existing hybrid RANS/LES model to assess their ability to predict transitional flows.
3. Perform flat-plate simulation with  $kk_L - \omega$  to provide insight to the modelled transition process.
4. Formulate and implement the new transitional hybrid RANS/LES model in OpenFOAM.
5. Verify that the new model converges with increasing mesh resolution.
6. Validate the model through a series of flat-plate simulations with different turbulent inflow conditions.
7. Demonstrate the model's ability to simulate transitional flows over aerodynamic objects which are relevant for the aero-acoustic community.

## 1.3 Project set-up

The outline of this report is as follows. First of all, a good understanding of boundary layer physics and the transitional process of a boundary layer is required. Therefore, Chapter 2 presents the characteristics of the laminar and turbulent boundary. Also, the natural and bypass mechanism, which cause transition, are discussed. In the remainder of this Chapter, the governing equations are presented. This is followed by the approach of turbulence modelling and a discussion on various hybrid RANS/LES models.

An assessment of the capabilities of current hybrid RANS/LES models to simulate transitional boundary layers is presented in Chapter 3.

The physics based transitional turbulence model  $kk_L - \omega$  is used for flat-plate simulation. The results are analysed and are used to provide more insight into the behaviour of the modelled transitional process. These results can be found in Chapter 4.

Based on the knowledge provided by the results from the hybrid RANS/LES simulation and the  $kk_L - \omega$  simulations, a new transitional hybrid RANS/LES model is suggested in Chapter 5. Various modifications are required before a satisfactory model is obtained.

After the final modifications are presented, a verification and validation is performed by means of a zero-pressure gradient flat-plate simulations with different turbulent inflow conditions. These can be found in Chapter 6.

After validation, two test cases are performed. The first test case is performed on NACA 0012 airfoil at a Reynolds number of 1 million and a zero degree angle of attack. Results are presented in Chapter 7. Chapter 8 presents a second, more challenging test case at a Reynolds number of 3.2 million, performed on a DU91W250 inboard wind-turbine airfoil section.

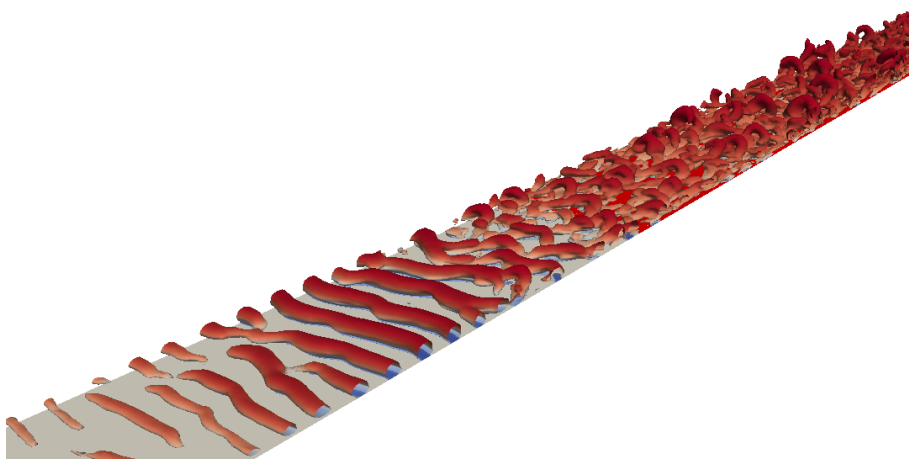
Finally, conclusions drawn from this study and a list of recommendations for future developments of transitional hybrid RANS/LES models, are presented in Chapter 9.



# Flow Physics and Turbulence Modelling

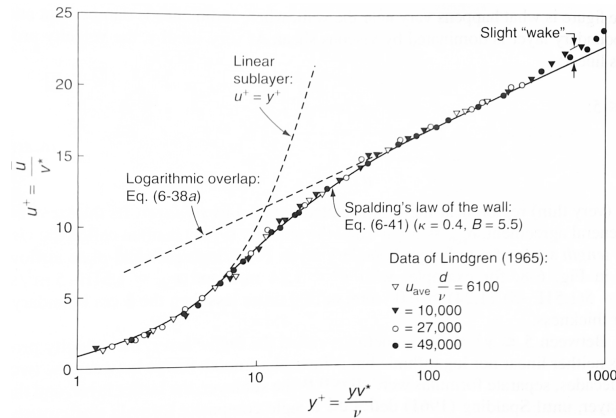
## 2.1 Boundary-layer concept

The thin region of flow adjacent to a surface, where the flow is slowed down by the influence of friction between a surface and the fluid, is referred to as the boundary layer. For flow over aerodynamics bodies, the boundary layer is initially laminar, meaning that the fluid particles are steady and follow the contours of the surface. Vortical structures can form inside the laminar boundary layer. Tollmein and Schlichting (TS) first discovered the formation of these two-dimensional structures (Schlichting, 1979). TS waves are generated by the steady shear force, present in the laminar boundary layer. The length-scale is relative large compared to structures present in turbulent flows. Furthermore, TS waves have very low energy dissipation.



**Figure 2.1:** Vorticity contours ( $Q\text{-iso}=1e4$ ) of a transitional boundary layer over a flat-plate. Coloured with the velocity magnitude.

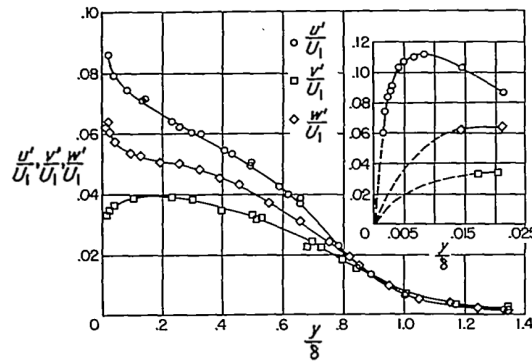
Formation of TS waves are visualized in Figure 2.1, the almost two-dimensional structures are easily recognized. Downstream of the TS waves, the formation of more turbulent structures are present. This turbulent boundary layer is characterized by unsteady fluid motion, forming various coherent structures. The turbulent boundary layer contains different distinct regions which can be visualized when the velocity profile is made dimensionless. Figure 2.2 shows a non-dimensional velocity profile. Very close to the wall ( $0 < y^+ < 5$ ), called the *viscous sublayer*, the flow is dominated by viscous shear forces and the turbulence in this region is damped. The region between the viscous layer and the log-layer is often referred to as the *buffer layer* ( $5 < y^+ < 30$ ). The linear velocity profile merges with that of the logarithmic part of the boundary layer. The outer region of the boundary layer is called the *wake region* and covers the largest part of the boundary layer. In 1969 Spalding deduced a single formula which describes the entire boundary layer profile for a zero pressure gradient flow.



**Figure 2.2:** Non-dimensional boundary layer profile for a zero-pressure gradient flat-plate. Graph from White by Lindgren(1965).

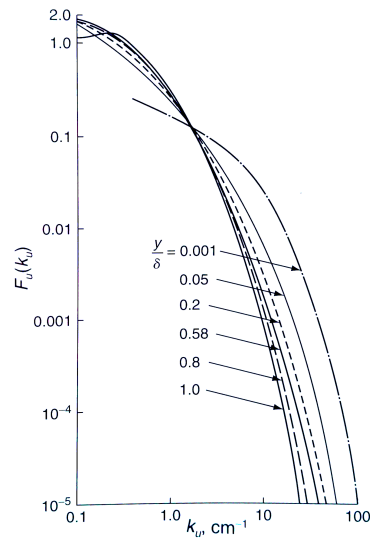
**Klebanoff (1954)** was able to capture the turbulent intensities inside a boundary layer over a flat-plate (Figure 2.3). The strongest fluctuations are found close to the surface ( $y/\delta \approx 0.01$ ), where  $\delta$  is the thickness of the boundary layer. The streamwise velocity fluctuations ( $u'$ ) are the largest, followed by the lateral fluctuations ( $w'$ ). These results clearly show the three-dimensional motion of turbulent flows. Any CFD simulation, in which one wishes to resolve the unsteady characteristic, should therefore span a three-dimensional domain.





**Figure 2.3:** Turbulent intensities inside the turbulent boundary layer. Graph from Klebanoff (Klebanoff, 1954).

Klebanoff also constructed a wave-number spectrum from the turbulent boundary layer, see Figure 2.4. The energy of the higher frequencies increases when approaching the wall, while at the same time the energy of the lower frequencies reduces. The frequency of the turbulence is directly related to the size of the turbulent structures, so when approaching the wall, the size of the turbulent structures decreases. It is noted that the scales of Figure 2.4 is logarithmic, and that the turbulent structures at  $\frac{y}{\delta} = 0.001$  is many times smaller than those in the logarithmic part of the boundary layer.

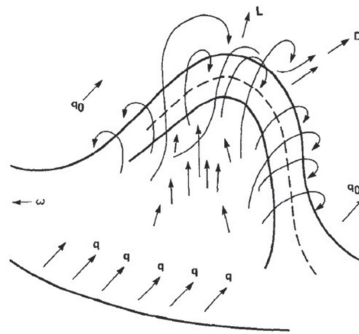


**Figure 2.4:** Energy spectrum inside a turbulent boundary layer at different locations from the wall (from White (2006) by Klebanoff (1954)).

From an numerical aeroacoustic point of view, the highest accuracy can be obtained when all of the turbulent fluctuations are captured. However, when only the most important turbulent structures would be captured, one could model the remaining fluid structures. These type of methods are addressed in Section 2.5. One of the first discovered coherent turbulent structures, are low-speed streaks (Dennis, 2015). The streaks grow in size along the streamwise

direction. When the streak is lifted up, away from the surface it starts to oscillate in the buffer layer and finally breaks up in the region  $10 < y^+ < 30$ . This *ejection* of fluid results in an unstable behaviour which causes a turbulent burst. An opposite event, called *sweeps* (or *intrushes*), is characterized by high speed fluid, approaching the wall. These sweeps also contribute to the local shear stress, but are more significant near the surface ( $y^+ < 15$ ). Corino [Corino and Brodkey \(1968\)](#) considered the ejection and sweeps of primary importance in the production and maintenance of turbulence. Corino also concluded that '*The most important feature of the all region is the ejection of fluid elements which occurs there ... The interaction of these elements with the mean flow creates turbulence*'.

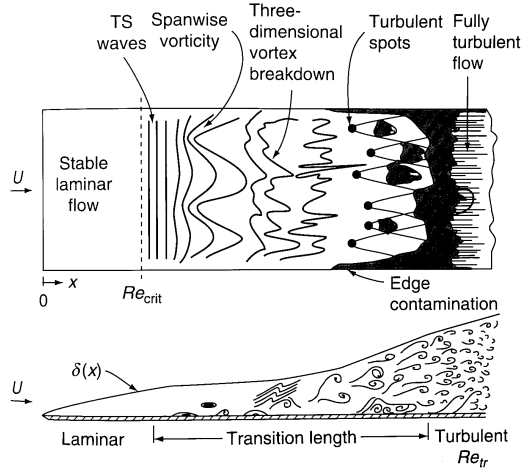
Hairpin vortices were first described by Theodorsen [Theodorsen \(1952\)](#) as vortical structures present in the turbulent boundary layer. Theodorsen's model was of an inclined horseshoe vortex moving away from the wall under an inclination of 45 degrees. A schematic description is given in Figure 2.5



**Figure 2.5:** Horseshoe vortex as described by [Theodorsen \(1952\)](#).

The vortical structures predicted by a CFD simulation using a method (SA IDDES, Section 2.5.1) which solves a large part of the unsteady fluid motion, is depicted in Figure 2.1. Typical TS waves can be recognized by the spanwise vortical structure. Downstream of the TS waves, the flow becomes turbulent. Although the turbulent flow might look chaotic, it actually consists (partially) of coherent hairpin vortices.

The process by which the TS waves break up into turbulent structures is referred to as laminar to turbulent transition. Figure 2.6 shows a schematic transition process of a flow on a flat plate. The typical transition process starts with a pretransitional laminar flow with developing TS waves. When secondary instabilities in the TS waves become strong enough, turbulent spots (or turbulent bursts) turn the flow into a full turbulent flow. Transition can be classified into two processes, *natural* transition and *bypass* transition.

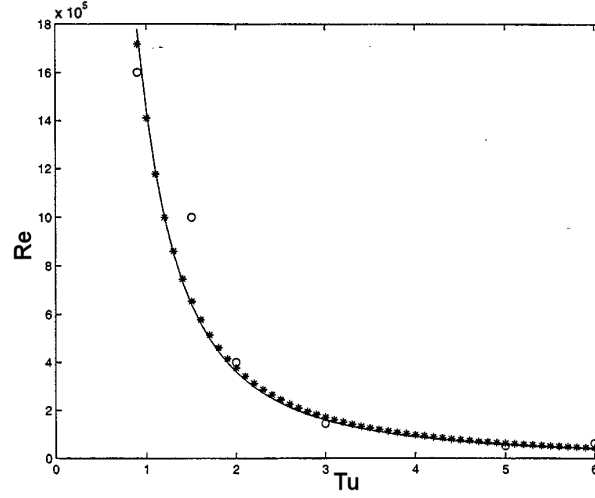


**Figure 2.6:** Schematic representation of the transitional boundary layer process (Figure from White [White \(2006\)](#))

In the case of natural transition, the TS waves grow and infinitely small perturbations are being amplified and finally form secondary instabilities, these are marked in Figure 2.6 as *Spanwise vorticity*. A relation between the transition location of a flow over a flat-plate and the ratio of inertial forces over viscous forces is given by

$$Re_{x,tr} \approx 3e6 \approx \frac{\rho U_{\infty} x}{\mu}, \quad (2.1)$$

where  $\rho$  is the fluid density,  $U_{\infty}$ , the free-stream velocity,  $x$  the distance to the leading edge and  $\mu$  the viscosity. Increasing the free-stream velocity will therefore shift the transition location  $x$  upstream. When transition takes place before natural transition is expected, the process is referred to as *Bypass* transition. Bypass transition can be caused by free-stream turbulence or by surface irregularities. Often the free-stream contains turbulence which introduces small velocity fluctuations inside the laminar boundary layer. The relation between the free-stream turbulent intensity ( $Tu$ ) and the transitional Reynolds number  $Re_{x,tr}$  is shown in Figure 2.7. When the free-stream turbulence intensity is increased the transition point will move upstream.



**Figure 2.7:** Chordwise based Reynolds number at which transition occurs due to free-stream turbulence. Solid line: Andersson, Stars: van Driest model, Circle: experimental. Andersson et al. (1991).

## 2.2 Governing flow equations

The Navier-Stokes (NS) equations have been recognised as the the governing equations for linear (Newtonina) viscous fluid flows. These equations can describe any fluid motion, in space and time. Equation 2.2 describes the conservation of momentum, one equation is used for each direction. The momentum equation is supplement by the conversation of mass and the conservation of energy:

$$\frac{D\rho\mathbf{u}}{Dt} = \frac{\partial\rho\mathbf{u}}{\partial t} + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot (\mu\nabla\mathbf{u}) + f \quad (2.2)$$

$$\frac{D\rho}{Dt} = \frac{\partial\rho}{\partial t} + \nabla \cdot \rho\mathbf{u} = 0 \quad (2.3)$$

$$\frac{D\rho e}{Dt} = \frac{\partial\rho e}{\partial t} + \nabla \cdot (\rho\mathbf{u}e) = -\nabla p\mathbf{u} + \nabla \cdot (\mu\mathbf{u}\nabla\mathbf{u}) - \nabla q \quad (2.4)$$

In this study, only low Mach number flows will be considered. Therefore, the flow field can be assumed to have an homogeneous density though the domain. Also, external forces, such as body forces and gravity, will be outside the scope of this study. Furthermore, it is expected that no temperature variation will be present (isothermal conditions). Hence, the fluid viscosity is assumed to be constant and the energy equation can be omitted. These simplification result in the following form of the NS equations

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla \frac{p}{\rho} + \nabla \cdot (\nu\nabla\mathbf{u}) \quad (2.5)$$

and

$$\nabla \cdot \mathbf{u} = 0, \quad (2.6)$$

where  $\nabla \cdot (\mathbf{u}\mathbf{u})$  is the Reynolds stress tensor  $\tau'$  and  $\nu = \frac{\mu}{\rho}$  is the kinematic viscosity.

## 2.3 RANS

The Reynolds-Averaged Navier-Stokes (RANS) equations provides time averaged flow solutions. The RANS equations are obtained by separating  $\mathbf{u}$  into a mean and fluctuating part:

$$\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}'. \quad (2.7)$$

Substitution of  $\bar{\mathbf{u}} + \mathbf{u}'$  into the NS equations gives

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}}\bar{\mathbf{u}}) = -\nabla \frac{\bar{p}}{\rho} + \nabla \cdot (\nu \nabla \bar{\mathbf{u}}) - \nabla \cdot (\overline{\mathbf{u}'\mathbf{u}'}), \quad (2.8)$$

where component  $\overline{\mathbf{u}'\mathbf{u}'}$  is usually referred to as the Reynolds stress tensor. In 1877, Boussinesq proposed (White (2006)) that the turbulent stresses, of incompressible flows can be modelled by the Reynolds stress tensor

$$\tau'_{ij} = 2\nu_t \bar{S}_{ij}^*, \quad (2.9)$$

where  $\nu_t$  is the eddy viscosity and  $\bar{S}_{ij}^*$ , is the mean strain-rate tensor,

$$\bar{S}_{ij}^* = \frac{1}{2} \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right). \quad (2.10)$$

Substitution of the Reynolds stress tensor in the moment equations and removing the time dependence of the left hand side terms, finally gives

$$\nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla \frac{p}{\rho} + \nabla \cdot ([\nu_t + \nu] \nabla \mathbf{u}) \quad (2.11)$$

and

$$\nabla \cdot \bar{\mathbf{u}} = 0. \quad (2.12)$$

The eddy viscosity in the second term on the right hand side, now accounts for the turbulent viscous effects. Different models have been developed to determine the eddy viscosity. The models that have been used in this study, are presented in the following sections.

### 2.3.1 SA model

The Spalart-Allmaras (SA), one equation, turbulence model (Spalart, 1992) was designed to facilitate a wider range of simulations in terms of grid complexity and flow. The effect of turbulence is modelled through the eddy viscosity transport equation

$$\frac{D\tilde{\nu}}{Dt} = c_{b1}[1 - f_{t12}]\tilde{S}\tilde{\nu} + \frac{1}{\sigma}[\nabla((\nu + \tilde{\nu})\nabla\tilde{\nu}) + c_{b2}(\nabla\tilde{\nu})^2] - [c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}]\left[\frac{\tilde{\nu}}{d}\right]^2 + f_{t1}\Delta U^2, \quad (2.13)$$

where,  $d$  is the distance to the nearest wall and  $\tilde{\nu}$  the transported eddy viscosity. The turbulent eddy viscosity is related to the transport equation by

$$\nu_t = \tilde{\nu} \cdot f_{v1}, \text{ where } f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \text{ and } \chi = \frac{\tilde{\nu}}{\nu}. \quad (2.14)$$

The production term scales with a modified value of the strain-rate magnitude

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad (2.15)$$

where,

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad (2.16)$$

and  $S$  corresponds to the magnitude of the vorticity. The function  $f_w$ , in the destruction term is given by

$$f_w = g \left[ \frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{1/6}, \quad (2.17)$$

where,

$$g = r + c_{w2}(r^6 - r) \text{ and } r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}. \quad (2.18)$$

It is noted that the SA model is able to simulate laminar, transitional and turbulent flows through the functions  $f_{t12}$  and  $f_{t1}$ . However, as stated by Spalart and Allmaras; "On no account should the model be trusted to predict the transition location. The responsibility of choosing transition points rest with the user .." Spalart (1992). This implies that the transition location should be known beforehand. This might be the reason that OpenFOAM does not include the tripping terms in their turbulence model. Rumsey (2007) noted that most users of the SA model do not make use of the tripping term at all but rather simulate in fully turbulent mode. Simulations of laminar to turbulent flow with the SA model are possible when the transition location is known, because it is then possible to postponing transition by zeroing the production term in the laminar region. The SA model constants are summarized in Table 2.1.

**Table 2.1:** Model coefficients of the SA model.

$c_{b1} = 0.1355$	$c_{w1} = c_{b1}/\kappa^2 + (1 + c_{b2}/\sigma)$
$c_{b2} = 0.622$	$\kappa = 0.41$
$c_{w2} = 0.3$	$c_{v1} = 7.1$
$c_{w3} = 2$	$\sigma = 2/3$

### 2.3.2 Physics based transitional $kk_L - \omega$ model

A transition model based on the  $k - \omega$  (Menter, 1993) framework was developed by Walters and Cokljat (2008) and includes an additional laminar kinetic energy ( $k_L$ ) transport equation. The transport equation represents the low-frequency velocity fluctuation in the pretransitional boundary layer. These velocity fluctuations are identified as the mechanism which initiates transition and are classified as laminar kinetic energy ( $k_L$ ). The transport equations are based on phenomenological relations that distinguish between the different flow states of a transitional boundary layer, instead of using purely empirical relations (such as SA,  $k - \omega$  and  $v^2 - f$ ). Walters model is therefore also referred to as a physics based RANS model. Flat-plate simulations performed by Walters and Cokljat (2008) indicate that the model yields a qualitative correct transition of the boundary layer for a wide range of flow conditions. The solver has also proven to be able to accurately simulate the transitional flow over an outboard wind-turbine DU96 airfoil (Zhang et al., 2013) for various Reynolds numbers. Fürst (2013) reproduced the simulation of Walters, and obtained the same results. Medina and Early (2016) concluded that the model predicts transition of a flat plate with reasonable accuracy. Based on these results, the  $kk_L - \omega$  model makes a good candidate as the turbulent transport model for the new hybrid RANS/LES model. The original paper from Walters contained error's in the model equations, Fürst (2013) has published a paper with the correct equations, which are presented in this section.

The three transport equations are

$$\frac{Dk_T}{Dt} = P_{k_T} + P_{BP} + P_{NAT} - \omega k_T - D_T + \frac{\partial}{\partial x_j} \left[ \left( \nu + \frac{\alpha_T}{\sigma_k} \right) \frac{\partial k_T}{\partial x_j} \right], \quad (2.19)$$

$$\frac{Dk_L}{Dt} = P_{k_L} - P_{BP} - P_{NAT} - D_L + \frac{\partial}{\partial x_j} \left[ \nu \frac{\partial k_L}{\partial x_j} \right] \quad (2.20)$$

and

$$\begin{aligned} \frac{D\omega}{Dt} = C_{w1} \frac{\omega}{k_T} P_{k_T} + \left( \frac{C_{\omega R}}{f} - 1 \right) \frac{\omega}{k_T} (R_{BP} + R_{NAT}) - C_{\omega 2} \omega^2 + \\ C_{\omega 3} f \omega \alpha_T f_W^2 \frac{\sqrt{k_T}}{d^3} + \frac{\partial}{\partial x_j} \left[ \left( \nu + \frac{\alpha_T}{\sigma_\omega} \right) \frac{\partial \omega}{\partial x_j} \right]. \end{aligned} \quad (2.21)$$

They represent the transport of turbulent kinetic energy, laminar kinetic energy and the scale-determining variable. Turbulent and laminar production is defined as:

$$P_{k_T} = \nu_{T,s} S^2 \quad (2.22)$$

and

$$P_{k_L} = \nu_{T,l} S^2, \quad (2.23)$$

where  $S$  is the strain-rate magnitude. The dynamics of laminar energy production is described by large scale, low amplitude fluctuations. These large scale fluctuations are distinguished from the small scales by a so called 'splat mechanism'. This mechanism divides the production of energy into wall-limited (large-scales) and non-wall-limited (small-scales). The large scales will contribute to the production of laminar energy while the small scales increase the turbulent production term. The large- and small-scale turbulent fluctuations were already discussed in Section 2.1. The small-scale eddy viscosity is defined as:

$$\nu_{T,s} = f_\nu f_{INT} C_\mu \sqrt{k_{T,s}} \lambda_{eff}. \quad (2.24)$$

The small scale turbulent kinetic energy  $k_{T,S}$  is given by

$$k_{T,s} = f_{SS} f_W k_T. \quad (2.25)$$

The turbulent length scale is given by,

$$\lambda_T = \frac{\sqrt{k_T}}{\omega}. \quad (2.26)$$

The effective turbulent length scale  $\lambda_{eff}$ , representing the turbulent fluctuations (which contain dissipation), is determined by

$$\lambda_{eff} = \min(C_\lambda d, \lambda_T). \quad (2.27)$$

where  $d$  is the shortest distance to the wall. The region with turbulent dissipation is marked by the damping function  $f_W$ ,

$$f_W = \left( \frac{\lambda_{eff}}{\lambda_T} \right)^{2/3}. \quad (2.28)$$

The term  $f_\nu$  is the viscous damping function and accounts for the viscous wall-effect and reads

$$f_\nu = 1 - \exp\left(-\frac{\sqrt{Re_T}}{A_\nu}\right), \quad (2.29)$$

where

$$Re_T = \frac{f_W^2 k_T}{\nu \omega}, \quad (2.30)$$



and  $A_\nu$  is a model constant. The intermittency function

$$f_{INT} = \min\left(\frac{k_T}{C_{INT}k_{TOT}}, 1\right), \quad (2.31)$$

indicates the time averaged fraction during which the flow is turbulent. The intermittency, together with the shear sheltering function

$$f_{SS} = \exp\left[-\left(\frac{C_{SS}\nu\Omega}{k_T}\right)^2\right], \quad (2.32)$$

makes the distinction between the laminar and turbulent flow states. In Equation 2.32,  $\Omega$  is the rotation-rate magnitude and  $C_{SS}$  is a model constant. Turbulent production is scaled with the realisability constrain  $C_\mu$

$$C_\mu = \frac{1}{A_0 + A_s\left(\frac{S}{\omega}\right)} \quad (2.33)$$

The production of laminar kinetic energy is than what remains of the turbulent kinetic energy minus the small-scale turbulent kinetic energy

$$k_L = k_T - k_{T,s} \quad (2.34)$$

This quantity is then used to compute the large-scale eddy viscosity

$$\nu_{t,l} = \min\left[f_{\tau,l}C_{11}\left(\frac{\Omega\lambda_{eff}^2}{\nu}\right)\sqrt{k_{T,l}}\lambda_{eff} + \beta_{TS}C_{12}Re_\Omega d^2\Omega, \frac{0.5(k_L + k_{T,l})}{S}\right] \quad (2.35)$$

The first term addresses the production of bypass transition, the second term represent the production of TS waves. The third term limits the production of eddy viscosity to ensure satisfaction of the realisability constrain for the total Reynolds stress contribution in stagnation points. The damping function  $f_{\tau,l}$  is defined as:

$$f_{\tau,l} = 1 - \exp\left[-C_{\tau,l}\frac{k_{T,l}}{\lambda_{eff}^2\Omega^2}\right]. \quad (2.36)$$

The natural transition parameter  $Re_\Omega$ , is based on wall distance, rotation rate and kinematic viscosity

$$Re_\Omega = \frac{d^2\Omega}{\nu}. \quad (2.37)$$

Breakdown to turbulence, due to instabilities is included through a production term for natural transition

$$\beta_{TS} = 1 - \exp\left[-\frac{\max(Re_\Omega - C_{TS,crit}, 0)^2}{A_{TS}}\right]. \quad (2.38)$$

The total eddy viscosity is

$$\nu_t = \nu_{t,l} + \nu_{t,s}. \quad (2.39)$$

A common form of the anisotropic (near-wall) dissipation for  $k_T$  and  $k_L$  are

$$D_T = \nu \frac{\partial \sqrt{k_T}}{\partial x_j} \frac{\partial \sqrt{k_T}}{\partial x_j} \quad (2.40)$$

and

$$D_L = \nu \frac{\partial \sqrt{k_L}}{\partial x_j} \frac{\partial \sqrt{k_L}}{\partial x_j}. \quad (2.41)$$

The effective diffusivity in the turbulent transport terms  $k_T$  and  $\omega$ , is defined as:

$$\alpha_T = f_\nu C_{\mu, std} \sqrt{k_{T,s}} \lambda_{eff}. \quad (2.42)$$

A damping term is included for the boundary layer production term in the  $\omega$  field of the form

$$f_\omega = 1 - \exp \left[ -0.41 \left( \frac{\lambda_{eff}}{\lambda_T} \right)^4 \right]. \quad (2.43)$$

### Transition terms

The following terms are related to the laminar turbulent transition mechanism. This works as follows: laminar energy is transferred to the turbulent region when specific threshold values of the laminar and turbulent kinetic energies are reached. The terms that transfer the energy are divided into a bypass transition term and a natural transition term:

$$R_{BP} = C_R \beta_{BP} k_L \omega / f_\omega, \quad (2.44)$$

$$R_{NAT} = C_{R,NAT} \beta_{NAT} k_L \Omega. \quad (2.45)$$

Both transition terms contain a related threshold function

$$\beta_{BP} = 1 - \exp \left( - \frac{\phi_{BP}}{A_{BP}} \right), \quad (2.46)$$

$$\phi_{BP} = \max \left[ \left( \frac{k_T}{\nu \Omega} - C_{BP, crit} \right), 0 \right], \quad (2.47)$$

$$\beta_{NAT} = 1 - \exp \left( - \frac{\phi_{NAT}}{A_{NAT}} \right), \quad (2.48)$$

$$\phi_{NAT} = \max \left[ \left( Re_{\Omega} - \frac{C_{NAT}}{f_{NAT,crit}} \right), 0 \right] \quad (2.49)$$

and

$$f_{NAT,crit} = 1 - \exp \left( - C_{NC} \frac{\sqrt{k_L} d}{\nu} \right). \quad (2.50)$$

A critical note is made with respect to the various damping terms ( $f_{SS}, f_{INT}, f_{\nu}$ ). These terms are based on size of  $k_T$ , which could pose a problem when operating in LES mode. In LES mode, the modelled turbulent energy is reduced by a filter width and only the sub-grid energy remains (Section 2.4 gives an overview of LES). This means that the damping terms could unintentionally become active inside the turbulent boundary, resulting in a too large suppression of  $k_T$ . The results presented by Walter indicate qualitatively correct transition locations for various flow conditions. The solver has also proven to be able to accurately simulate the transitional flow over an outboard wind-turbine DU96 airfoil [Zhang et al. \(2013\)](#) for various Reynolds numbers. The  $kk_L - \omega$  turbulence model would therefore be a good candidate to replace the SA model in IDDES framework.

## 2.4 LES

The concept of Large Eddy Simulations (LES) was proposed by [Smagorinsky \(1963\)](#) and was further pioneered by [Deardorff \(1970\)](#) in a channel flow simulation. The model resolved the important, large-scale turbulent structures, while modelling the less energetic turbulence. This is possible because LES accounts for the effect of the unresolved sub-grid scale turbulence by locally increasing the eddy viscosity. Opposite to the RANS equations, in which all turbulence is modelled through the Reynolds stress term, LES has a term for the resolved stresses and the sub-grid (Reynolds) stresses. The filtered NS equations are

$$\nabla \cdot \bar{\mathbf{u}} = 0 \quad (2.51)$$

and

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}}\bar{\mathbf{u}}) = -\nabla \frac{\bar{p}}{\rho} + \nabla \cdot (\nu \nabla \bar{\mathbf{u}}) - \nabla \cdot (\overline{\mathbf{u}'\mathbf{u}'}). \quad (2.52)$$

In the context of LES, the term  $(\overline{\mathbf{u}'\mathbf{u}'})$  is referred to as the sub-grid scale stress tensor

$$\tau_{ij} = 2\nu_{sgs} \bar{S}_{ij}^*, \quad (2.53)$$

where  $\nu_{sgs}$  is the sub-grid scale eddy viscosity, and is valid for incompressible flow. [Smagorinsky \(1963\)](#) suggested that the subgrid-scale stress, for isotropic turbulence, should scale with the filter width  $\Delta$  and the magnitude of the strain rate tensor  $S$

$$\nu_{sgs} = (C_s \Delta)^2 |S|, \quad (2.54)$$

where  $C_S$  is the sub-grid scaling constant and is calibrated for specific flow states. In the Smagorinsky (1963) model, the sub-grid filter width scales with the volume of the grid cells. However, other scaling laws have been suggested, the IDDES model for example (Section 2.5.1). This implies that an increased mesh resolution will result in a DNS resolution while a coarse mesh will return a unsteady RANS like solution. Since non-isotropic turbulence is present in turbulent boundary layer, discrepancies arise. van Driest (1956) substituted the constant  $C_S$  a damping function that corrects the near-wall sub-grid scale dissipation. Another solution to the static value of the Smagorinsky constant is the adjustment from Germano Germano et al. (1991). The dynamic sub-grid model replaces the constant scaling  $C_S$  by a field that is derived each time iteration and is dependent on the local flow state. This method allows a zero eddy viscosity in laminar region, improving the prediction of the transition point.

## 2.5 Hybrid RANS/LES methods

LES simulation provide good accuracy but comes with high computational costs. For high Reynolds number flows the turbulent scales in the viscous layer can become so small that affordable LES is not possible. The hybrid RANS/LES method is developed to overcome this issue by resolving large scale eddies while modelling the near wall small-scale turbulence. The hybrid RANS/LES method makes use of a single turbulence model which blends between the RANS and LES method. The models reviews here all employ eddy viscosity LES models, meaning that the sub-grid scale destruction term scales with a mesh dependent filter width. When the LES branch extends into the turbulent boundary layer, the bulk of the turbulent structures will be resolved. This is referred to as wall-modelled LES (WMLES). WMLES at high Reynolds number flows has proven to be very challenging (Spalart et al., 2006). According to Piomelli (2008) it is unreasonable to expect that WMLES may become as accurate as fully resolved LES. In the inner boundary layer region, most turbulent production takes place, by *sweeps* and *ejection* (Section 2.1) upwards into the outer layer of the boundary layer. WMLES has to model this unsteady motion and will therefore not excite the boundary layer as a full LES simulations would. It is therefore unrealistic to expect the same accuracy as simulation which do include these important physical phenomena. Another problem with hybrid RANS/LES models is that when a flow exits the RANS region and enters the LES region it does not contain turbulent structures, even if the flow did contain modelled turbulence in the RANS region. The region between fully modelled turbulence and fully developed turbulence is referred to as the *gray area*. The left sketch of Figure 2.8 an ideal situations is depicted, the attached flow is solved using RANS and as soon as the flow detaches from the surface LES becomes active and all turbulent structures are resolved. However, in most simulations, the flow from the steady RANS solution requires more time to develop turbulent structures, this is visualized by the right sketch.



**Figure 2.8:** Schematic representation of the *grey area* problem that can be found in hybrid RANS/LES simulations. *Left:* ideal case. *Right:* real case, *grey area* is marked orange. By Michel et al. (2010)

### 2.5.1 Spalart-Allmaras DES

#### DES model

The first hybrid RANS/LES model was the detached eddy simulation (DES), designed by Spalart et al. (1997). The turbulent model used in DES is the Spalart-Allmaras (SA) model. In the SA model, the eddy viscosity is proportional to the inverse of the wall distance. It is this length-scale that is replaced by filter width. The eddy viscosity is reduced in the LES region and reduces towards zero for DNS like meshes. The LES branch of DES is designed to be active in the region containing eddies, outside the boundary layer (detached). The right sketch in Figure 2.8 displays a typical DES simulations, the flow that is detached from the surface is solved using LES while the (steady) attached flow is solved using RANS.

However, when grid density is fine enough, the LES branch would penetrate into the turbulent boundary layer and a *WMLES* is performed. This poses a problem as the boundary layer has a reduced eddy viscosity but does not contain resolved turbulent content. The turbulent content in the LES branch may be missing due insufficient mesh density and because of the delay of generation of instabilities (Spalart et al., 2006).

At high Reynolds number, a Log-Layer Mismatch (LLM) was observed by Nikitin (2000). The velocity profile of the modelled and resolved log layer do not line up as the near-wall fluctuations are too weak and elongated. The LLM also caused the skin-friction to be 10-15 percent too low, which is marked as a too large discrepancy.

Keating and Piomelli (2006) introduced a dynamic forcing method for WMLES at the interface of the RANS/LES region. The magnitude of the force is based on the assumption that modelled and resolved Reynolds shear stresses should be equal in the RANS/LES region. The LLM was successfully removed due to the improved prediction of velocity fluctuations. This shows that with extra effort, DES can perform WMLES without modifying the DES equations.

#### DDES model

The problem in which the LES branch of DES penetrates the boundary layer is solved in the Delayed Detached Eddy Simulation (DDES) by Spalart et al. (2006). A limiter function is introduced, which makes sure that the LES branch gradually turns to RANS inside the turbulent boundary layer.

The blending is based on the model length scale  $r_d$  by

$$f_d = 1 - \tanh[(8r_d)^3], \quad (2.55)$$

The model length  $r_d$  is defined as:

$$r_d = \frac{\nu + \nu_t}{\kappa^2 d_w^2 \cdot \max[(\sum_{ij} (\partial u_i / \partial x_j)^2)^{1/2}, 10^{-10}]}, \quad (2.56)$$

where,  $\partial u_i / \partial x_j$  is the velocity gradient,  $\kappa$  the von Karman constant,  $d_w$  the shortest distance to the wall and  $\nu_t$  and  $\nu$  the eddy viscosity and viscosity. The new hybrid length scale becomes

$$\tilde{d} = d - f_d \max[0, d - C_{DES} \Delta] \quad (2.57)$$

When  $f_d$  equals zero the models operates in RANS mode and the length scales returns to  $\tilde{d} = d$ . In case the LES branch is active  $d = \max[0, d - C_{DES} \Delta]$ . Spalart et al. (2006) also introduced a Low-Reynolds Number (LRN) correction term that should be applied for WMLES. The correction term is introduced through the LES length scale

$$\tilde{d} = d - f_d \max[0, d - C_{DES} \Psi \Delta], \quad (2.58)$$

where  $C_{DES} = 0.65$  and filter width  $\Delta$  is defined as the maximum local grid spacing

$$\Delta = \max[\Delta_x, \Delta_y, \Delta_z]. \quad (2.59)$$

Spalart et al. (2006) obtained the expression for  $\Psi$  under the assumption that at 'equilibrium' the modified subgrid model driven by  $C_{DES} \Delta \psi$  should reduce to a Smagorinsky-like model  $(C_S \Delta)^2 |S|$ . The correction term reads:

$$\psi^2 = \frac{\frac{f_w}{f_w^*} - \frac{c_b 1}{c_{w1} \kappa^2 f_w^*} [f_{t2} + (1 - f_{t2}) f_{v2}]}{f_{v1} (1 - f_{t2})} \quad (2.60)$$

## IDDES model

Shur et al. (2008) proposed a model that has the ability to perform wall-modelled simulations. The Improved Delayed Detached Eddy Simulation (IDDES) combines the DDES blending with a wall-modelled blending function. This model will return to a wall-modelled branch when inflow conditions contain turbulent fluctuations. When the inflow conditions are stable the model will return to the DDES branch. Shur et al. (2008) solved the problem of LLM by introducing an elevating function for the eddy viscosity for the near-wall RANS region as was typical for DES simulation. IDDES broadens the application purposes of DES and DDES by permitting the activation of LES and RANS for different flow types. It is questionable if IDDES can simulate transitional boundary layers as the model will only switch to the WMLES branch when turbulent inflow conditions are specified. The value of sub-grid scale eddy viscosity should scale with different flow regimes, this is made possible through a modified definition of the subgrid length scale length

$$\Delta = \min\{\max[C_w d_w, C_w h_{wn}], h_{max}\}, \quad (2.61)$$

where  $C_w = 0.15$ ,  $d_w$  is the shortest distance to the wall,  $h_{wn}$  is the grid step in the wall-normal direction. As mentioned before, the model has two branches, DDES and WMLES. The DDES branch was presented in Section 2.5.1. The hybrid length scale is repeated for convenience and the length scale parameters are replaced by the respective RANS and LES length scales

$$l_{DDES} = l_{RANS} - f_d \max[0, l_{RANS} - l_{LES}], \quad (2.62)$$

where the DDES blending function is depend on the flow solution. Blending of the WMLES branch is only a function of the numerical mesh

$$l_{WMLES} = f_B(1 + f_e)l_{RANS} + (1 - f_B)l_{LES}, \quad (2.63)$$

where  $f_B$  is given by Equation 2.64 and is dependent on the distance to the wall and the maximum local cell size  $h_{max}$  (not on the state of the flow)

$$f_B = \min[2\exp(-9\alpha^2), 1.0], \quad (2.64)$$

where  $\alpha = 0.25 - \frac{d_w}{h_{max}}$ . This results in a rapid switch between RANS and LES within the distance to wall

$$0.5h_{max} < d_w < h_{max}. \quad (2.65)$$

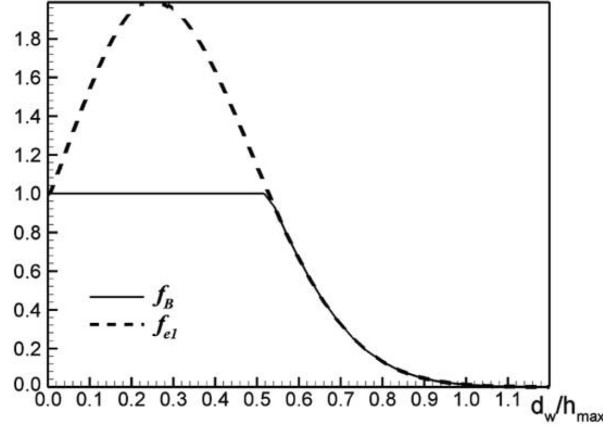
In the first term on the right hand side of Equation 2.64, the empirical function

$$f_e = \max[(f_{e1} - 1), 0] \Psi \cdot f_{e2}, \quad (2.66)$$

is included to compensate the excessive reduction of eddy viscosity in the near-wall RANS/LES interface. Increasing the eddy viscosity removes the log-layer mismatch which was found in simulations using DDES. In Equation 2.66,  $f_{e1}$  is defined as:

$$f_{e1}(d_w/h_{max}) = \begin{cases} 2\exp(-11.09^2) & \text{if } \alpha \geq 0 \\ 2\exp(-9.0^2) & \text{if } \alpha < 0 \end{cases}. \quad (2.67)$$

where  $\alpha$  was already defined to be a function of the cell size and distance to the wall. The elevation of  $f_{e1}$ , is therefore only dependent on the grid, but not on the solution. The elevating function is shown in Figure 2.9 by a dashed line and follows the profile of the RANS/LES blending function  $f_B$  when it is smaller than one. But when approaching the wall (smaller  $d_w$ ) the elevating function becomes active and the turbulent length-scale is increased.



**Figure 2.9:** Profiles of the RANS/LES blending function  $f_B$  and the RANS elevating function  $f_{e1}$ .

The strength of the elevating function  $f_{e1}$  scales with  $f_{e2}$

$$f_{e2} = 1 - \max\{f_t, f_l\}, \quad (2.68)$$

where  $f_t$  and  $f_l$  control the intensities of the turbulent and laminar sublayer of the boundary layer, respectively. The two functions are defined as:

$$f_t = \tanh[(c_t^2 r_{dt})^3] \quad (2.69)$$

and

$$f_l = \tanh[(c_l^2 r_{dl})^{10}], \quad (2.70)$$

where  $r_{dt}$  and  $r_{dl}$  are the turbulent and laminar sublayer relations

$$r_{dt} = \frac{\nu_t}{\kappa^2 d_w^2 \cdot \max[(\sum_{ij} (\partial u_i / \partial x_j)^2)^{1/2}, 10^{-10}]} \quad (2.71)$$

and

$$r_{dl} = \frac{\nu}{\kappa^2 d_w^2 \cdot \max[(\sum_{ij} (\partial u_i / \partial x_j)^2)^{1/2}, 10^{-10}]} \quad (2.72)$$

and  $c_t$  and  $c_l$  are model constants that depend on the turbulence model, which are 3.55 and 1.63, respectively. They are calibrated such that  $f_{e2}$  is virtually zero when either  $r_{dt}$  or  $r_{dl}$  is close to one. Opposite to the function  $f_{e1}$ , which is only dependent on the grid,  $f_{e2}$  is also a function of the flow solution. The final RANS/LES blending is defined by

$$f_d = \max\{(1 - f_{dt}), f_B\}. \quad (2.73)$$

The two right hand side terms are the DDES and WMLES branch, respectively. If the term  $1 - f_{dt} \ll 1$ , then the WMLES branch of IDDES will be selected. If both terms,  $1 - f_{dt}$  and  $f_B$  are close to one both branches will be active. The blending of the RANS-LES method can be deduced from Equation 2.74. When  $f_d$  equals zero, then the will  $l_{hyb}$  equal  $l_{LES}$  and the flow will be solved using LES. If  $f_d$  equals one, the flow will be solved using RANS. The blending function  $f_d$  is then used for the blending of the turbulent length-scale

$$l_{hyb} = f_d(1 + f_e)l_{RANS} + (1 - f_d)l_{LES}, \quad (2.74)$$



### 2.5.2 $kk_L - \omega$ SST-DES

Beehook (2016) constructed a DES and DDES model with the  $kk_L - \omega$  RANS model. The blending between RANS and LES is performed with the same blending function ( $F_{DDES}$ ) as is used in the  $k - \omega$  SST model from Menter (1993). Beehook (2016) made the following modifications to the  $kk_L - \omega$  model. In order to obtain a Smagorinsky like LES model, the destruction term of the turbulent kinetic energy equation is modified. The sub-grid scale destruction term of the turbulent kinetic energy field

$$\frac{Dk_T}{Dt} = P_{k_T} + P_{BP} + P_{NAT} - F_{DDES}\omega k_T - D_T + \frac{\partial}{\partial x_j} \left[ \left( \nu + \frac{\alpha_T}{\sigma_k} \right) \frac{\partial k_T}{\partial x_j} \right], \quad (2.75)$$

is allowed to increase in magnitude, resulting in a reduction of the eddy viscosity. The function  $F_{DDES}$  defines the RANS/LES blending

$$F_{DDES} = \max \left( \frac{\sqrt{k_T}}{C_{DES}\omega\Delta} (1 - F_1), 1 \right), \quad (2.76)$$

where  $C_{DES} = 0.61$ . The filter width is chose. to be a function of maximum grid size

$$\Delta = \max\{\Delta_x, \Delta_y, \Delta_z\}. \quad (2.77)$$

When the cell size is chosen small enough,  $F_{DDES}$  becomes larger than one and the model operates in LES mode. The limiting function

$$F_1 = \tanh \left[ \left( \min \left[ \max \left( \frac{\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega} \right), \frac{4\sigma_{\omega 2}k}{CD_{k\omega}y^2} \right] \right)^4 \right], \quad (2.78)$$

was introduced by Menter (1993) and removes the possibility that the LES region penetrates the boundary layer too deep. The possibility of grid induced flow separation is thereby prevented. In Equation 2.76 the term  $CD_{k\omega}$  is given by,

The model constants are listed in Table 2.2. Beehook performed two flat-plate simulation on a two dimensional domain. In the first simulation, the limiting function  $F_1$  was omitted. The simulation returned a fully laminar boundary layer, indicating that the DES model was unable to capture the transition process. The behaviour was assigned to the models high sensitivity to the computational grid. However, it is doubtful whether this could be the true reason, as flow separation on flat-plate simulations is unlikely to occur. Also, no conclusive results were shown to backup the conclusion. The skin-friction results of the second flat-plate simulations with the  $F_1$  term showed excellent agreement with experimental data. Unfortunately, no information of the LES region, nor the presence of developed turbulence were reported.

**Table 2.2:** Blending coefficients of the  $kk_L - \omega$  SST-DES model.

$\sigma_{\omega 2} = 0.856$	$\beta^* = 0.09$
-----------------------------	------------------



---

## Chapter 3

---

# Assessment of Hybrid RANS/LES Methods

In this Chapter two existing hybrid RANS/LES models are assessed on their ability to predict transitional boundary layer flow. Through several flat-plate simulations, a clear picture of the limitations and behaviour is established. The first model uses the Spalart-Allmaras (SA) turbulence model with the IDDES blending function(Section 2.5.1). In total three simulations are performed with the SA IDDES model of which two with a modification of the transported turbulence field. A second model which is examined makes use of the  $kk_L - \omega$  turbulence model and defines the RANS/LES blending based on the SST model (Section 2.5.2). For the assessment of the model performance a flow over a flat-plate with 0.8% turbulent kinetic energy inflow is simulated and compared with experimental data from ERCOFTAC.

### 3.1 Simulation models

#### 3.1.1 Simulation 1: SA IDDES

Although not intended for transitional boundary layer flow nor for steady free-stream inlet conditions, the SA IDDES model has been used in the past for simulation on aerodynamic bodies (Section 2.5.1). According to [Shur et al. \(2008\)](#) and [Spalart \(1992\)](#), several problems can be expected for simulation with low turbulence inflow conditions. 1: without resolved turbulent inflow conditions the IDDES model should remain in the RANS branch. 2: flow which enters an LES region will need time to develop turbulent structures. 3: the location of transition can not be predicted accurately by the SA turbulence model.

### 3.1.2 Simulation 2: modified SA IDDES

In Section 2.3.1 it was noted the SA model can not be trusted to predict the transition point accurately. A possible solution is to deactivate the turbulent viscosity  $\tilde{\nu}$  in the laminar boundary layer. This means that the transition onset should be known beforehand. With the transition point known, a numerical mesh can be constructed which consists of two parts, marking the laminar and turbulent region using the field variable  $\sigma_{\tilde{\nu}}$ . The turbulence solver will then deactivate the production of  $\tilde{\nu}$  in the laminar region by,

### 3.1.3 Simulation 3: modified SA IDDES, 2D tripping

In order to accelerate the development of turbulent structures, reducing the length of the *gray area*, a tripping ridge is located at the transition location. The presents of the obstacle in the flow generates a local jump in the pressure and velocity field. Any instabilities in the flow will be augmented, promoting the development of turbulent structures.

### 3.1.4 Simulation 4: $kk_L - \omega$ SST-DES

According to Beehook (2016), the  $kk_L - \omega$  turbulence model with the SST blending function (Section 2.5.2) should return an accurate wall friction. Since Beehook (2016) only performed DES simulation on a two dimensional flat-plate domain (one cell in the spanwise direction), it remains uncertain how the model performs when simulating on a three dimensional flat-plate. Questions also arise regarding the RANS/LES blending function. The SST blending function was designed to switch in the outer region of a boundary layer as was discussed in Section 2.5.2. It is therefore uncertain whether wall-modelled LES is supported by this model. Since the  $kk_L - \omega$  SST-DES model is not available in OpenFOAM, a model is constructed from the equation provided by Beehook (2016).

## 3.2 Flat-plate test case

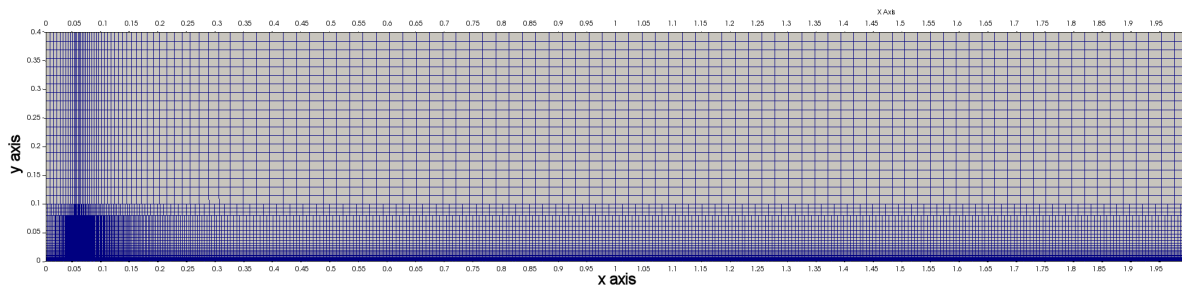
Experimental data from the ERCOFTAC researcher group Coupland (1990) is used for validate the simulation results. The validation case selected here is the T3A-. This is the experiment with the lowest upstream turbulence intensity available. Therefore this case will be challenging for turbulence models to predict a correct transition location. The length of the flat-plate is 2.9 meters. The maximum expected boundary layer thickness is 0.03 meters, the domain span is therefore selected to be 0.09 meters such that the span is three times wider than the boundary layer thickness. This should be enough to allow for the development of uncorrelated turbulent structures. The inlet plane is located 0.05 meters in-front of the flat-plate leading edge. A symmetry plane is located between the flat-plate leading edge and the inflow plane. The top plane has a zero gradient boundary conditions for all flow quantities.

The boundary conditions of the inlet, outlet and wall surface are listed in Table 3.1. The dynamic viscosity  $\mu$  is set to  $1.513\text{e-}5$ .

**Table 3.1:** Boundary conditions applied to the RANS and LES case.

	Inlet	Outlet	wall
$p$ [ $Nm^{-2}$ ]	$\frac{\partial p}{\partial n}$	0	$\frac{\partial p}{\partial n}$
$U$ [ $ms^{-1}$ ]	(19.8 0 0)	$\frac{\partial U}{\partial n}$	(0 0 0)
$\nu_t$ [ $(Ns)m^{-2}$ ]	1E-10	$\frac{\partial \nu_t}{\partial n}$	calculated
$\tilde{\nu}$ [ $(Ns)m^{-2}$ ]	1E-10	$\frac{\partial \tilde{\nu}}{\partial n}$	calculated
$k_T$ [ $m^2s^{-2}$ ]	0.04857	$\frac{\partial k_T}{\partial n}$	0
$k_L$ [ $m^2s^{-2}$ ]	0	$\frac{\partial k_L}{\partial n}$	0
$\omega$ [ $s^{-2}$ ]	23.8	$\frac{\partial \omega}{\partial n}$	$\frac{\partial \omega}{\partial n}$

The transition location based on the ERCOFTAC results is at 1.3 meter from the leading edge, this corresponds to  $Re_x = 1.5\text{e}6$ . The inlet values for  $\nu_t$  and  $\tilde{\nu}$  are almost set to zero because of the low inflow turbulence level and to avoid possible division by zero. Outside the boundary layer a less high mesh resolution is required. The domain is therefore divided into three layers. In each layer away from the wall the number of cells is halved in all three directions. A side view of the mesh is shown in Figure 3.1. At the wall, 1200 cells are used in the chordwise direction and 36 cells in the spanwise direction. The maximum non-dimensional  $\Delta x^+$  and  $\Delta z^+$  value is 140 which is a typical size for hybrid RANS/LES simulations. The height of the first wall-normal cells is chosen to be below  $y^+ < 1$ .



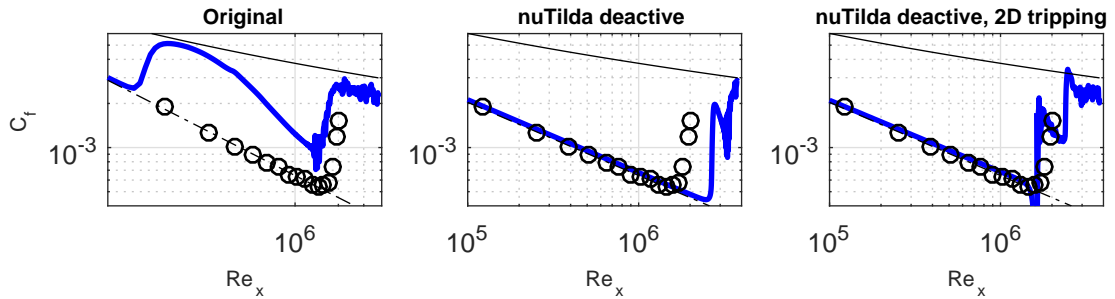
**Figure 3.1:** Side view of computational mesh. Left: inlet. Right: outlet. Top: free-stream. Bottom: symmetry plane and wall.

The following numerical schemes are used. For the pressure computations a PCG solver with a DIC preconditioned is selected. The velocity and turbulence fields are solved with a GaussSeidel solver. Thanks to the structured grid the number of outer correctors could be limited to 3 with 1 non-orthogonal corrector. The second order backward method is used for

the time integration. For the velocity divergence scheme a Gauss limitedLinearV scheme with a coefficient of 1 is used.

### 3.3 Results

The instantaneous spanwise-averaged wall friction of the three simulation performed with the SA turbulence model are shown in Figure 3.2. The original SA IDDES model predicts modelled transition at  $Re_x \approx 1 \cdot 10^5$ , which is at a chordwise location of 6% of the laminar boundary layer from the reference data. After the modelled transition the skin-friction slowly reduces as a consequence of the active LES branch. The TS waves break up at  $Re_x \approx 1.2 \cdot 10^6$  and turbulence develops. The region between the modelled transition point and the resolved transition location is the *grey area* and has a length of  $Re_x \approx 1.1 \cdot 10^6$ . The premature modelled transition of the SA IDDES can be removed by deactivating  $\tilde{\nu}$ , as can be seen in the wall friction distribution of the middle graph in Figure 3.2. Although  $\tilde{\nu}$  is only inactive up to  $Re_x \approx 1.5 \cdot 10^6$ , modelled transition takes place at  $Re_x \approx 2.5 \cdot 10^6$ . Resolved transition follows at  $Re_x \approx 3.1 \cdot 10^6$ , reducing the *grey area* to  $6 \cdot 10^5$ . Including the tripping edge at  $Re_x \approx 1.5 \cdot 10^6$  results in an accurate transition location. However, negative friction and a sharp increase in the wall friction are now present. Unfortunately the two dimensional shape of the tripping edge does not results in a rapid development of 3 dimensional turbulent structures and the *grey area* still has a length of  $Re_x \approx 10^5$ .



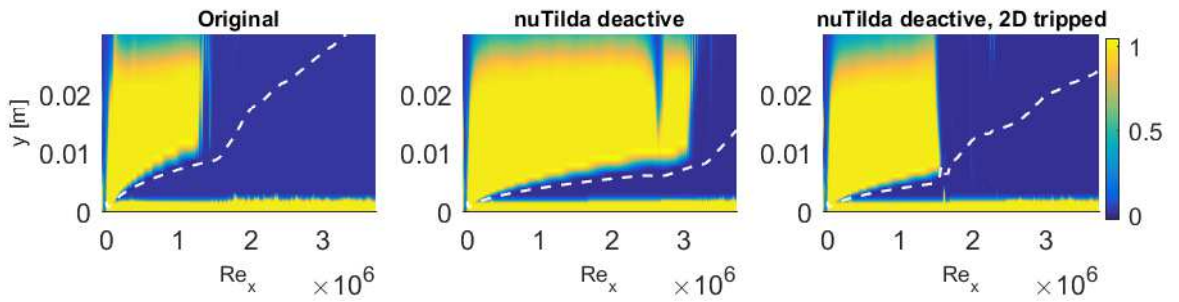
**Figure 3.2:** Wall friction coefficient. Blue line: SA. Circles: experimental. Black line: turbulent theory. Black dashed line: laminar theory. *Left:* original SA IDDES model. *Middle:* modified SA IDDES. *Right:* modified SA IDDES with 2D tripping.

The behaviour of the SA IDDES model and its response on the wall friction development is strongly related to the activation of the RANS and LES regions. Figure 3.3 shows the hybrid blending function  $f_d$  of all three simulations. The white dashed lines indicate the 99 percent boundary layer thickness

$$\delta_{99\%} = d_w, \text{ where } \frac{u}{U} = 0.99. \quad (3.1)$$

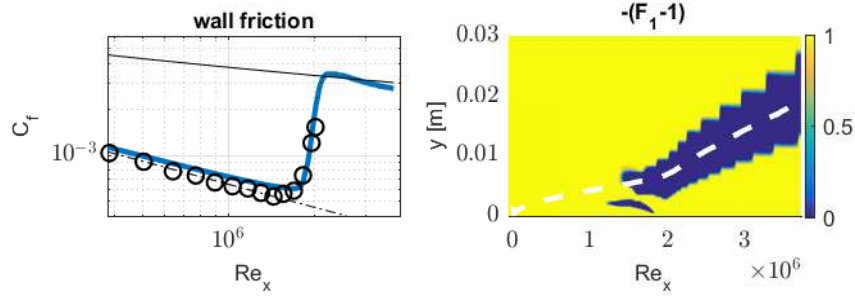
For all three simulations the LES region is active in the log layer while inside the viscous and buffer layer the RANS branch is active. In the SA IDDES simulation (left graph of Figure 3.3), the near wall RANS layer is the initiator of the modelled transition. However, as the

laminar boundary layer grows, a large portion is solved in the LES branch which reduces the eddy viscosity and the boundary layer becomes more or less laminar again. But only after the TS waves break and the skin friction increases, the boundary layer thickness increases. It is noted that the location of the resolved turbulence corresponds quite good with the reference data. However, this is a coincidence and case specific. When  $\tilde{\nu}$  is deactivated in the pretransitional boundary layer, which corresponds to the middle graph of Figure 3.3 the near wall RANS branch is in fact also a LES region. This explains why the wall friction for this case does not show premature transition. The tripping edge does not have a significant effect on the RANS/LES blending, left graph of Figure 3.3. The main difference can be observed in the earlier onset of the LES branch near the tripping edge.



**Figure 3.3:** Hybrid RANS/LES blending function from the IDDES. LES=0, RANS=1. White dashed line: 99 percent boundary layer thickness.

The  $kk_L - \omega$  SST-DES model returns a very accurate but steady instantiations skin friction as can be seen in the left graph of Figure 3.3. Both the transition onset and the slope of the transitional boundary layer show good agreement with the experimental data (Coupland, 1990). However, the steady wall friction already indicates that not much unsteady phenomena take place inside the boundary layer. The reason for this steady result can be found in the right graph of Figure 3.3, which shows the RANS/LES blending. There are two LES regions. One small near wall region located around the transition point and a large region around the 99 percent boundary layer thickness. All other boundary layer regions are solved using RANS. The SST blending function returns a typical DDES behaviour (Section 2.5.1); the LES region does not penetrate the turbulent boundary layer.



**Figure 3.4:** *Left:* Wall friction coefficient. Blue line:  $kk_L - \omega$  SST-DES. Circles: experimental. Black line: turbulent theory (White, 2006). Black dashed line: laminar theory. *Right:* hybrid RANS/LES blending function from SST-DES. RANS=1, LES=0. White dashed line: 99 percent boundary layer thickness.

### 3.4 Interim summary

The *original* SA IDDES model prematurely predicts modelled transition and is followed by premature resolved transition.

The *modified* SA IDDES model, which deactivates the eddy viscosity in the laminar region, removes the premature transition but does result in a slight delayed modelled transition location. After the modelled transition, turbulence developed too slowly, resulting in a drop in the skin-friction. Besides, the model does require the transition location to be known beforehand, since it requires a numerical mesh which differentiates between the laminar and turbulent region. The introduction of a two dimensional tripping edge results in a correct prediction of the transition location. However, due to the two-dimensional shape of the tripping edge, production of turbulence remained to be too low.

The  $kk_L - \omega$  SST-DES model did return a very accurate modelled transition location but failed to develop any turbulence. The SST RANS/LES blending function only allows the LES branch to be active in the outer region of the boundary layer, but probably due to the absence of resolved free-stream turbulence and/or locally too low strain rates, the development of turbulent structures did not take place.

A combination of the  $kk_L - \omega$  turbulence model and the IDDES blending function might therefore be able to predict a correct modelled transition location and at the same time support wall-modelled LES.



---

## Chapter 4

---

# Assessment of $kk_L - \omega$ RANS

Due to the complexity of the  $kk_L - \omega$  model the results of a flat-plate RANS simulation are analysed before proceeding with the construction of a hybrid RANS/LES model. This analysis will provide information on the transition process which will support the modification required to construct a working hybrid RANS/LES model.

### 4.1 Simulation details

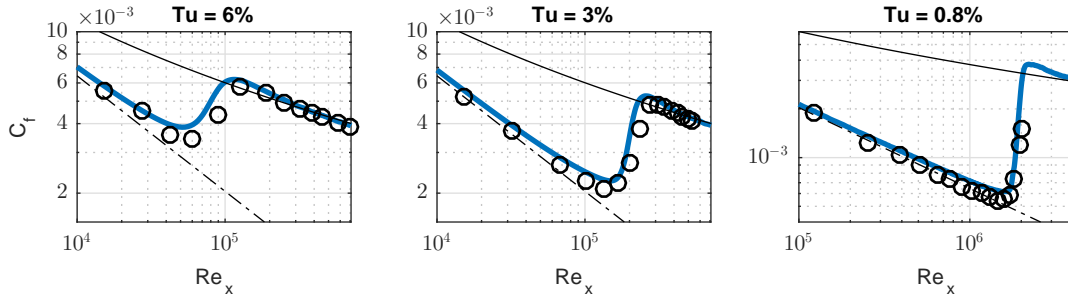
Three simulations with different inflow conditions are performed, the inflow conditions are listed in Table 4.1. Conditions used for the remaining boundaries correspond to those that were already stated in Chapter 3.

**Table 4.1:** Inflow conditions for the three different simulations. Conditions correspond to experimental simulation from ERCOFTAC by [Coupland \(1990\)](#).

	$Tu = 6\%$ (T3B)	$Tu = 3\%$ (T3A)	$Tu = 0.8\%$ (T3A-)
$p$ [ $Nm^{-2}$ ]	$\frac{\partial p}{\partial n} = 0$	$\frac{\partial p}{\partial n} = 0$	$\frac{\partial p}{\partial n} = 0$
$U$ [ $ms^{-1}$ ]	(9.2 0 0)	(5.4 0 0)	(19.8 0 0)
$k_L$ [ $m^2s^{-2}$ ]	0	0	0
$k_T$ [ $m^2s^{-2}$ ]	0.5386	0.04763	0.04857
$\omega$ [ $s^{-2}$ ]	56.8	23.8	23.8

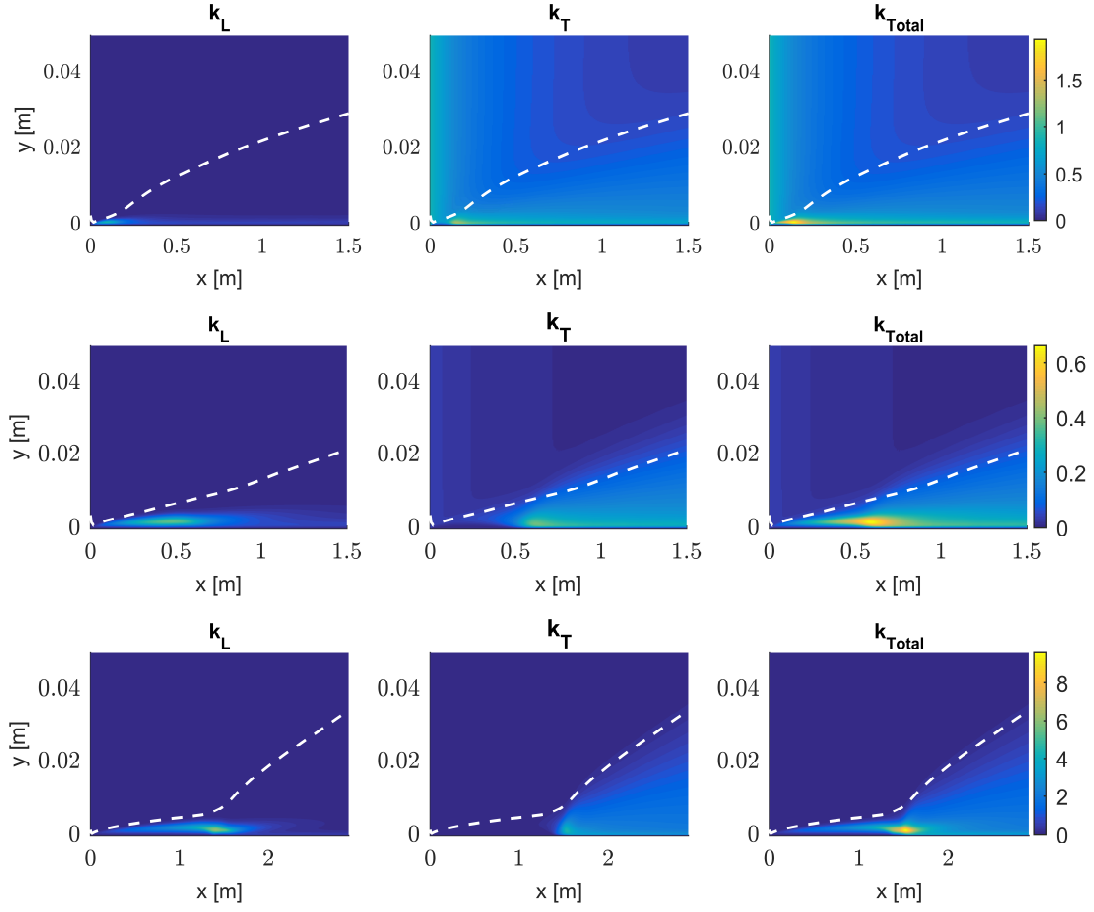
## 4.2 Results

The wall-friction coefficient along the flat-plate for the three simulations are visualized in Figure 4.1. The predicted skin-friction along the plate shows the characteristic of a transitional boundary layer and all three cases correspond with the experimental data (Coupland, 1990). The boundary is initially laminar and the skin-friction reduces downstream of the plate. The increase in skin-friction represents the transition from a laminar to a turbulent boundary layer. After the transition, the skin-friction reduces again in size. The skin-friction in the laminar boundary layer is for all three cases slightly overpredicted when compared to the experimental results. Also, the increase in skin-friction in the transitional region has a large gradient than those observed from the experimental data.



**Figure 4.1:** RANS wall friction coefficient along flat-plate for different free-stream turbulence intensities. Blue line:  $kk_L - \omega$ . Circles: experimental (Coupland, 1990). Black solid line: turbulent theory (White, 2006). Black dashed line: Blasius theory.

To aid the understanding of the working of this turbulence model, a closer examination of the modelled physics is made. It is noted that some of the remaining figures in this chapter represent data on a logarithmic scale. Figure 4.2 shows the laminar, turbulent and total kinetic energy fields of all three simulation cases. A clear distinction between the laminar and turbulent boundary layer can be observed from the laminar kinetic energy field (left column). It can also be observed that with decreasing turbulent inflow, the laminar boundary becomes relatively longer compared to the boundary layer thickness. The results from the turbulent kinetic energy field are less coherent. For the case with the highest turbulent inflow (upper row), the energy in the turbulent boundary is nearly as large as the free-stream energy. While for the case with the lowest turbulent inflow (bottom row) the turbulent boundary layer can clearly be distinguished by its high energy content. Combining the laminar and turbulent energy (right column), shows that around the transition point the kinetic energy is relatively high. For acoustic simulation it could therefore be preferred to solve the transition process using LES such that the acoustic sources can be predicted.

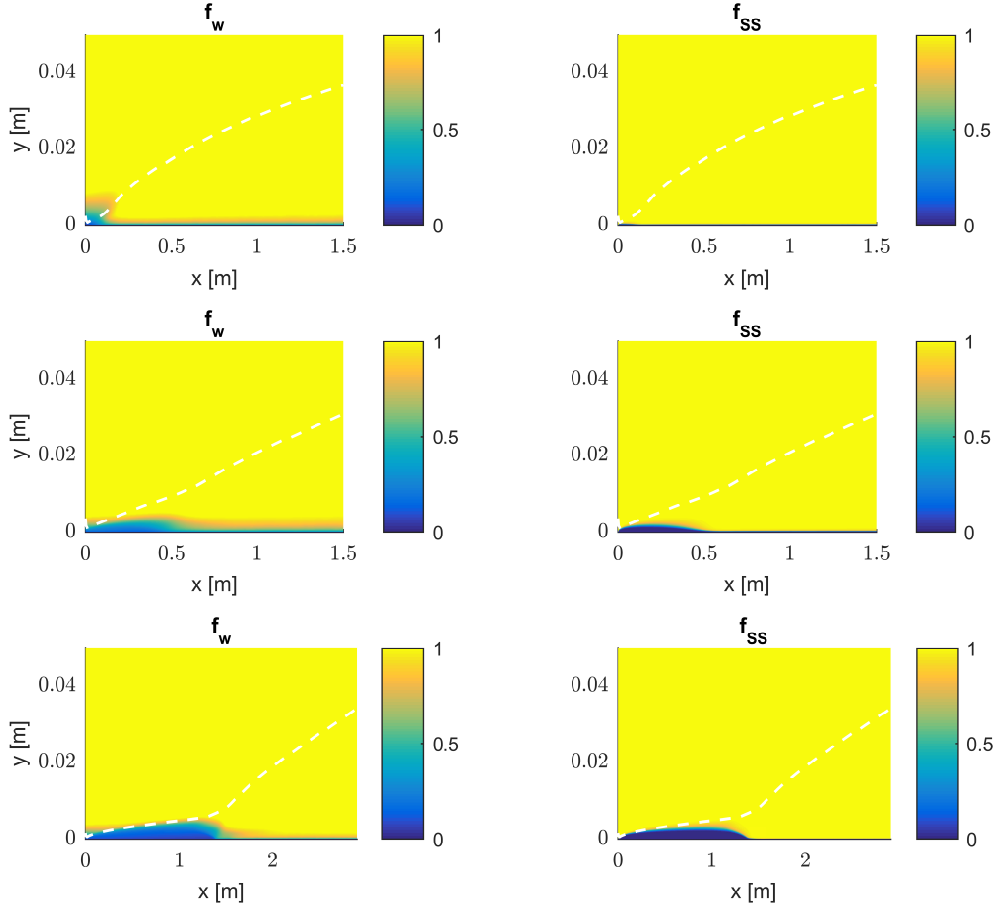


**Figure 4.2:** Decomposition of the kinetic energy fields. *Top:*  $Tu = 6\%$ . *Middle:*  $Tu = 3\%$ . *Bottom:*  $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness.

The process in which the total kinetic energy is separated in the laminar and turbulent part will be investigated. Turbulent kinetic energy is first divided into small-scale and large-scale turbulence. This is done through the shear sheltering  $f_{SS}$  and the damping function  $f_W$ ,

$$k_{T,s} = f_{SS} f_W k_T. \quad (4.1)$$

These damping function are depicted in Figure 4.3. The shear sheltering function  $f_{SS}$  returns a value of zero near the laminar boundary layer. The damping function  $f_W$  does not only return a value of zero around the laminar boundary layer but also around the viscous and buffer layer of the turbulent boundary layer. When the free-stream turbulence level is decreased, the activation of the shear sheltering function in the near wall region is delayed. The function  $f_W$  becomes active based on the size of the turbulent length scale and shows the same trend as  $f_{SS}$ .



**Figure 4.3:** Turbulent kinetic energy damping functions. *Left:* turbulent length scale limiter. *Right:* shear sheltering limiter. *Top:*  $Tu = 6\%$ . *Middle:*  $Tu = 3\%$ . *Bottom:*  $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness.

The kinetic energy which is limited by the  $f_W$  and  $f_{SS}$  makes the large-scale laminar kinetic energy,

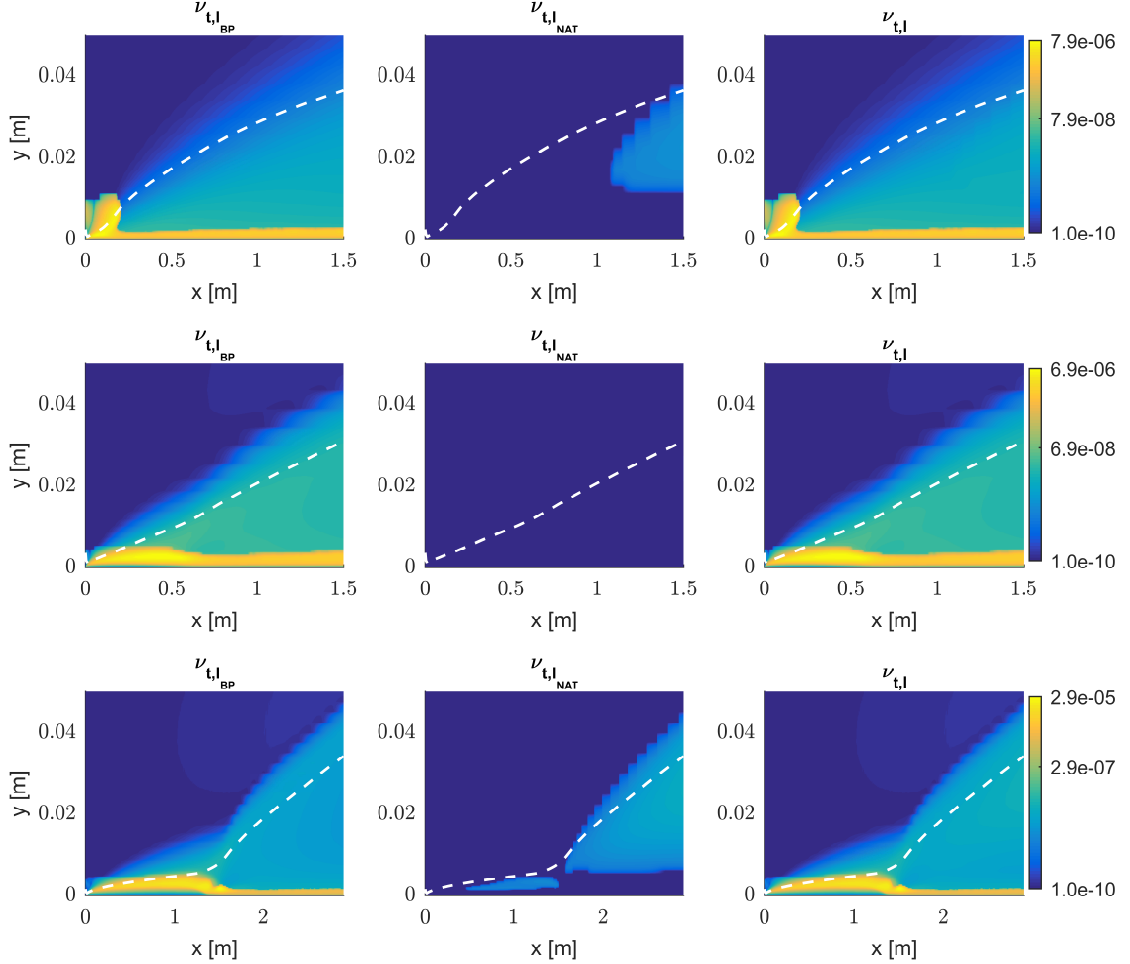
$$k_L = k_T - k_{T,s}. \quad (4.2)$$

This large-scale kinetic energy is then used to compute the large-scale eddy viscosity of the bypass mechanism (first term on RHS of Equation 4.3). Together with the eddy viscosity of the large-scale natural (TS waves) mechanism (second term on RHS of Equation 4.3), the total large-scale eddy viscosity is given by

$$\nu_{t,l} = \min \left[ f_{\tau,l} C_{11} \left( \frac{\Omega \lambda_{eff}^2}{\nu} \right) \sqrt{k_{T,l}} \lambda_{eff} + \beta_{TS} C_{12} Re_{\Omega} d^2 \Omega, \frac{0.5(k_L + k_{T,l})}{S} \right] \quad (4.3)$$

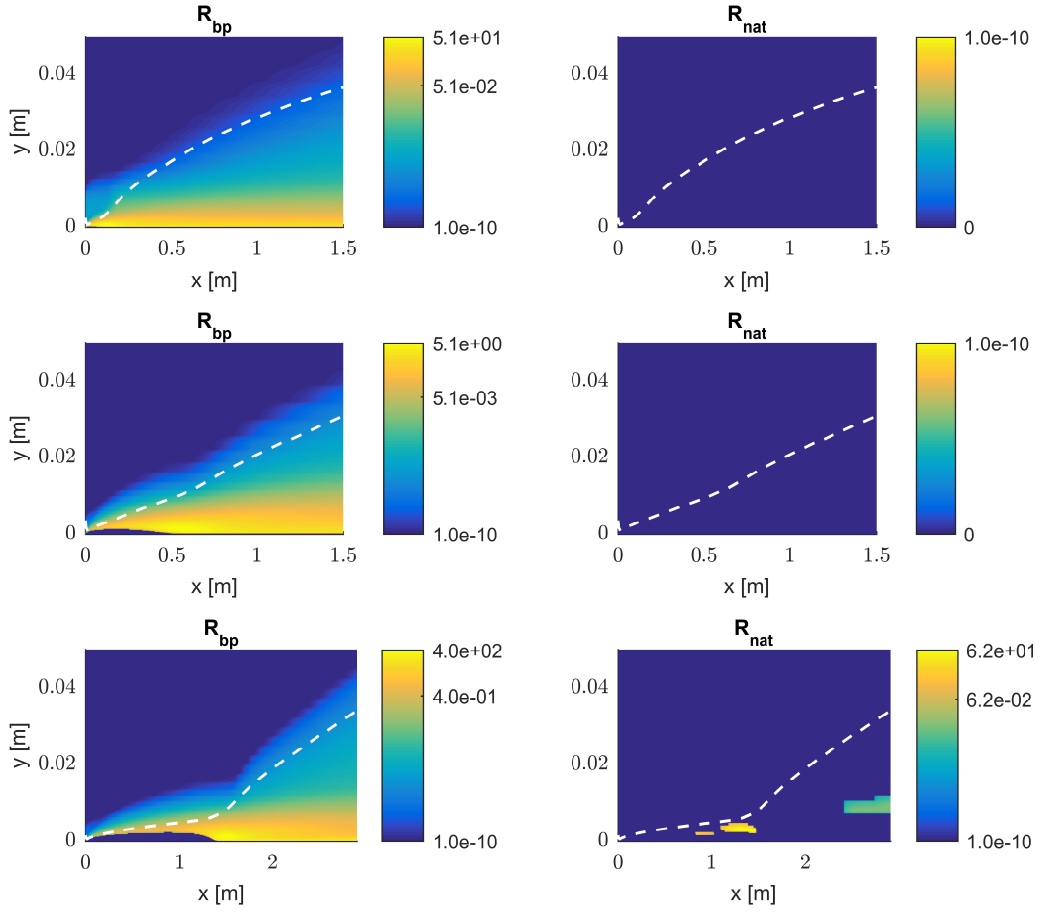
and is visualized in Figure 4.4. The left column contains the bypass eddy viscosity  $\nu_{t,l_{BP}}$ , the right column shows the natural eddy viscosity  $\nu_{t,l_{NAT}}$ . The bypass eddy viscosity is mostly concentrated in the laminar boundary layer and in the near wall region of the turbulent

boundary layer. The natural eddy viscosity is much smaller and is only present in the laminar boundary for case  $Tu = 0.8\%$  (bottom row). The natural eddy viscosity grows with the development of the laminar boundary layer thickness.



**Figure 4.4:** Decomposed large-scale eddy viscosity. *Left:* bypass production. *middle column:* natural production. *Right:* total. *Top:*  $Tu = 6\%$ . *Middle row:*  $Tu = 3\%$ . *Bottom:*  $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness.

When the laminar and turbulent kinetic energy reaches a threshold level, the function  $R_{BP}$  and  $R_{NAT}$  become active and reduces the laminar kinetic energy and increases the turbulent kinetic energy. This process in which the turbulent kinetic energy is increased finally results in the activation of the shear-sheltering function and the boundary layer becomes turbulent. The transition functions are depicted in Figure 4.5, where the left column shows the term responsible for bypass transition while the right column shows the natural transition term.

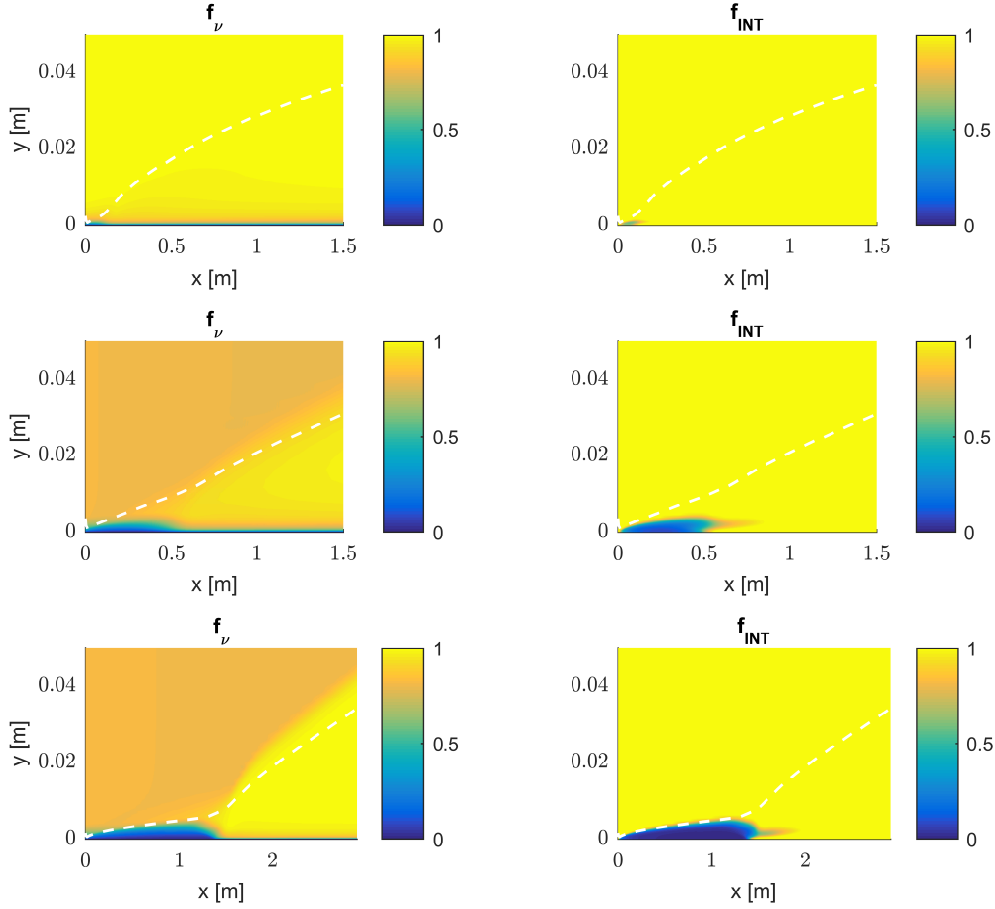


**Figure 4.5:** Transitional energy transfer from  $k_L$  to  $k_T$ . *Left:* bypass transition. *Right:* natural transition. *Top:*  $Tu = 6\%$ . *Middle row:*  $Tu = 3\%$ . *Bottom:*  $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness.

The small-scale turbulent kinetic energy is used to compute the small scale eddy viscosity,

$$\nu_{T,s} = f_\nu f_{INT} C_\mu \sqrt{k_{T,s}} \lambda_{eff}. \quad (4.4)$$

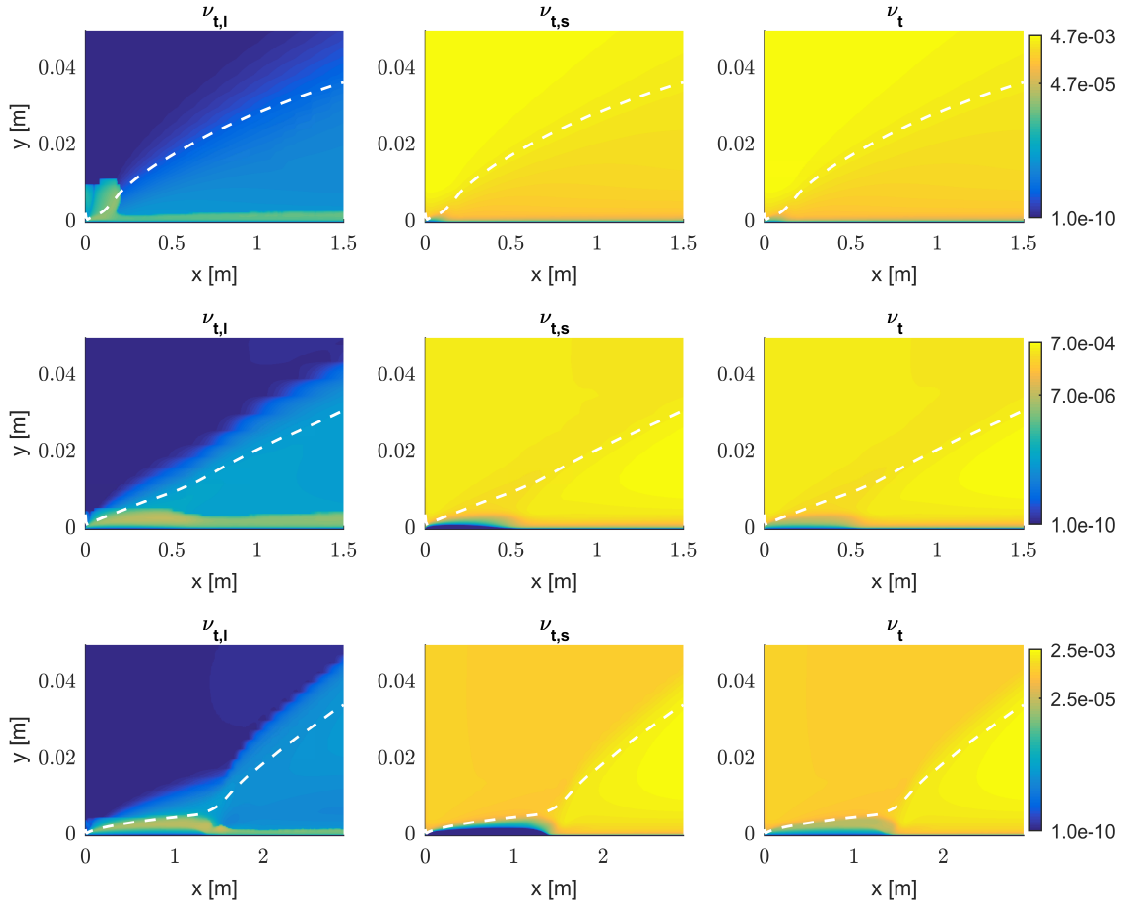
For this, two more damping functions are used. The first is the viscous wall damping function  $f_\nu$ , which together with the intermittency function are visualized in Figure 4.3. The viscous wall function,  $f_\nu$  of case  $Tu = 6\%$ , damps the production of eddy viscosity in the laminar boundary and in the turbulent viscous layer. For the other two cases,  $Tu = 3\%$  and  $Tu = 0.8\%$  the damping is also active in the free-stream flow. It is unclear why a *near wall* viscous damping function  $f_\nu$  returns another value than 1 for a *free-stream* flow. The second damping function is the intermittency  $f_{INT}$ , which indicates the time fraction during which the flow is in a turbulent state.



**Figure 4.6:** Dissipation damping functions. *Left:* viscous wall effect. *Right:* intermittency function. *Top:*  $Tu = 6\%$ . *Middle row:*  $Tu = 3\%$ . *Bottom:*  $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness.

The small-scale eddy viscosity  $\nu_{t,s}$ , the large-scale eddy viscosity  $\nu_{t,l}$  and the total eddy viscosity  $\nu_t$  are shown in Figure 4.7. The large-scale eddy viscosity clearly represents the diffusion in the region of the laminar boundary layer, while the small-scale eddy viscosity accounts for the free-stream turbulence and the turbulent boundary layer, but is many times smaller than  $\nu_{t,l}$ . The observation that the free-stream kinetic energy is almost as large as the kinetic energy in the turbulent boundary layer (Figure 4.2), is also reflected in the eddy viscosity. This can be seen in the right column, where the free-stream eddy viscosity of the  $Tu = 6\%$  case, is even larger than the eddy viscosity of the turbulent boundary layer. High values for the eddy viscosity in the free-stream might pose a problem for DES simulations in which the free-stream flow is modelled. In this case there will not be any excitation of the outer edge of the boundary layer, possible resulting in too slow development of turbulent motion. However, in this study the focus is on simulating flows over aerodynamic bodies that have a substantial amount of laminar flow, which is typically found for low free-stream turbulent values. The results presented in the bottom row are therefore more representative. A relative long laminar boundary layer is present with a low diffusion rate. Besides, the

turbulent boundary layer can be distinct from the rest of the flow by its high value of eddy viscosity.



**Figure 4.7:** Decomposition of large- and small-scale eddy viscosity. *Left:* large-scale eddy viscosity. *middle column:* small-scale eddy viscosity. *Right:* total eddy viscosity. *Top:*  $Tu = 6\%$ . *Middle row:*  $Tu = 3\%$ . *Bottom:*  $Tu = 0.8\%$ . White dashed line: 99% boundary layer thickness.

Although not all model equations are reviewed here, it is clear that the  $kk_L - \omega$  model consists of many limiting and damping functions, which are correlated to the turbulence fields. The modification required to construct a DES model might therefore also spread and pollute the behaviour of the damping and limiter functions.



---

## Chapter 5

---

# Proposed Transitional Hybrid RANS/LES Model

This chapter presents the steps taken that lead to a properly working laminar to turbulent transitional hybrid RANS/LES model. The new model is constructed from the  $kk_L - \omega$  model, which was presented in Section 2.3.2 and was analysed in Chapter 4. The framework from the Improved Delayed Detached Eddy Simulation (Shur et al., 2008) is used for the blending between RANS and LES.

Through a series of three modifications (Section 5.1, Section 5.2 and Section 5.3) a  $kk_L - \omega$  IDDES model is obtained that is able to predict the transition location at the same point as the  $kk_L - \omega$  RANS model does. However, when the steady flow from the RANS region enters the LES region, the development of turbulent structures take too long. Therefore, a last modification is made by introducing a tripping mechanism (Section 5.4). Each modified model is tested by means of a flat-plate simulation and results are analysed before proceeding to a next modification.

### 5.1 Modification 1: turbulent length scale and RANS elevation function

The first modifications, which are made to the  $kk_L - \omega$  model, are the implementation of the hybrid length scale and the inclusion of a low-Reynolds number limiter,  $\Psi$ .

### 5.1.1 Selection of LES destruction term

The classical modification for obtaining a DES model, requires a modification in the length scale of the transported turbulence field. In order to use the turbulence field  $k_T$  as the transported subgrid-scale variable in the LES branch of DES, it should behave as a Smagorinsky model. This implies that the subgrid-scale viscosity, for flows in equilibrium, should scale with the filter width  $\Delta$  and the strain rate  $S$ . The flow is said to be in equilibrium when the state of the flow does not change in time. For the turbulent kinetic energy equation (Equation 2.19) this will then results in

$$P_{k_T} = \omega k_T. \quad (5.1)$$

However, this equation does not yet scale with  $\Delta$  and  $S$ . Therefore,  $\Delta$  is introduced through the turbulent destruction term by replacing the scale-determining variable  $\omega$ . The following form is suggested for the length-scale  $\omega$  without destroying the dimensional form

$$\omega = \frac{\sqrt{k_T}}{C_{DES}\Delta}. \quad (5.2)$$

Substituting  $\omega$  in the destruction term ( $\omega k_T$ ) by  $\frac{\sqrt{k_T}}{C_{DES}\Delta}$  results in

$$P_{k_T} = k_T \frac{\sqrt{k_T}}{C_{DES}\Delta}. \quad (5.3)$$

The production term  $P_{k_T}$ , was already defined in Equation 2.22 as:

$$P_{k_T} = \nu_{T,s} S^2, \quad (5.4)$$

Substitution of  $\nu_{T,s} S^2$  in Equation 5.3 results in

$$\nu_{T,s} S^2 = k_T \frac{\sqrt{k_T}}{C_{DES}\Delta}. \quad (5.5)$$

In the above equation  $k_T$  should be expressed in terms of the sub-grid scale eddy viscosity and the filter width, only. This is accomplished by introducing the relation between the turbulent eddy viscosity and turbulent kinetic energy and yields

$$\nu_{T,s} \propto \frac{k_T}{\omega}. \quad (5.6)$$

Rewriting and replacing  $\omega$  in the above equation by  $\frac{\sqrt{k_T}}{C_{DES}\Delta}$  from Equation 5.2 gives

$$k_T \propto \frac{\nu_{T,s}^2}{(C_{DES}\Delta)^2}. \quad (5.7)$$

Substitutions of  $\frac{\nu_{T,s}^2}{(C_{DES}\Delta)^2}$  in Equation 5.5 results in

$$\nu_{T,s} S^2 \propto \frac{\nu_{T,s}^3}{(C_{DES}\Delta)^4}. \quad (5.8)$$

Rewriting and bringing the strain rate to the right hand side and the eddy viscosity to the left hand side gives

$$\nu_{T,s} \propto (C_{DES}\Delta)^2 |S|. \tag{5.9}$$

The above equation does behave as a Smagorinsky like relation. The proposed form for the LES destruction term is therefore correct. The destruction term should be able to blend between the two different destruction relations

$$\epsilon = \begin{cases} \omega k_T & \text{when in RANS mode} \\ k_T \frac{\sqrt{k_T}}{C_{DES}\Delta} & \text{when in LES mode} \end{cases}. \tag{5.10}$$

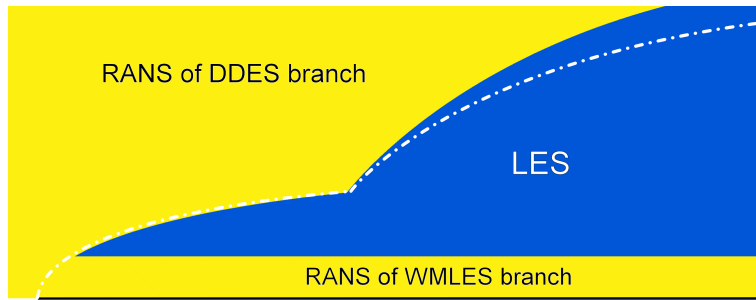
The blending between RANS and LES is treated in the following section.

### 5.1.2 RANS/LES blending

The blending function for the new model is taken from the IDDES model and differentiates between the RANS and LES turbulent length scales through the relation

$$l_{hyb} = f_d(1 + f_e)l_{RANS} + (1 - f_d)l_{LES}, \tag{5.11}$$

where  $l_{hyb}$  is the hybrid length-scale,  $f_d$  is the RANS/LES blending function,  $f_e$  is an elevating function for the near-wall RANS region and  $l_{RANS}$  and  $l_{LES}$  are the RANS and LES length scale, respectively. For clarification, a typical RANS/LES blending for a flat-plate simulation using the IDDES relation is shown in Figure 5.1. This figure indicates that the near-wall region is solved using the WMLES branch of IDDES, the thickness of the region decreases when the mesh resolution increases. The DDES branch is in RANS mode when no turbulent content is present in the farfield. The branch returns the LES mode when the strain rate increases or the eddy viscosity decreases. For more information of the DDES and WMLES branch see Section 2.5.1.



**Figure 5.1:** Schematic decomposition of RANS/LES regions by  $f_d$ , for a flat-plate simulation. RANS=yellow. LES=blue. White dashed line: 99% boundary layer thickness.

The hybrid length-scale  $l_{hyb}$ , will substitute the length scale parameter  $\omega$  in the destruction term of the turbulence equation, which yields

$$\frac{Dk_T}{Dt} = P_{k_T} + P_{BP} + P_{NAT} - k_T \frac{\sqrt{k_T}}{l_{hyb}} - D_T + \frac{\partial}{\partial x_j} \left[ \left( \nu + \frac{\alpha_T}{\sigma_k} \right) \frac{\partial k_T}{\partial x_j} \right]. \tag{5.12}$$

For  $l_{RANS}$  and  $l_{LES}$ , parameters should be chosen such that the requirements from Equation 5.10 are met. The RANS length-scale  $l_{RANS}$  will therefore take the form  $\frac{\sqrt{k_T}}{\omega}$  such that the destruction term returns to the relation  $\omega k_T$ . For the LES mode the length scale  $l_{LES}$  will return  $C_{DES}\Delta$  such that the destruction term returns  $k_T \frac{\sqrt{k_T}}{C_{DES}\Delta}$ . These relations are summarized in Table 5.1.

**Table 5.1:** Turbulent length scale and destruction term for the hybrid  $kk_L - \omega$  IDDES model.

Parameter	RANS	LES
length scale, $l_{hyb}$	$\frac{\sqrt{k_T}}{\omega}$	$C_{DES}\Delta$
destruction term, $\epsilon$	$\omega k_T$	$k_T \frac{\sqrt{k_T}}{C_{DES}\Delta}$

### 5.1.3 Near-wall elevation function

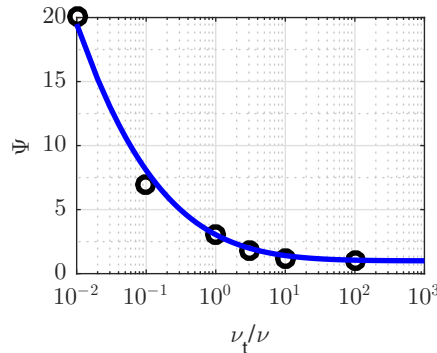
In the original IDDES model an elevating function  $f_e$  (Section 2.5.1)

$$f_e = \max[(f_{e1} - 1), 0] \Psi_{e2}, \quad (5.13)$$

was introduced to prevent the log-layer mismatch of the boundary layer velocity profile. In the definition of  $f_e$ , the function  $\Psi$  was introduced as a purely empirical function, which increases the effectiveness of the elevation function. Since  $\Psi$  is a function of  $\tilde{\nu}$ , a new relation is required for the new hybrid model. Based on the paper of Spalart et al. (2006), a rough estimation for  $\Psi$  as a function of  $\nu_t/\nu$  is made,

$$\Psi = 500 \exp \left[ -5.5 \left( \frac{\nu_t}{\nu} \right)^{1/9} \right] + 1. \quad (5.14)$$

The new and old function are both shown in Figure 5.2.



**Figure 5.2:** Elevating function  $\Psi$ . Blue line: new elevating function. Circle's: data points from Spalart et al. (2006).

Activation of  $f_e$  is not limited to the turbulent region and could therefore pollute the boundary layer characteristics of the laminar boundary layer. A function is required that can limit  $f_e$

to the turbulent region. The intermittency function meets these criteria. In the turbulent region the intermittency is active up to the viscous wall region. The new elevation function reads

$$f_e = \max[(f_{e1} - 1), 0] \Psi \cdot f_{e2} \cdot f_{INT}. \quad (5.15)$$

### 5.1.4 Low-Reynolds-number correction

In the  $kkL - \omega$  model, a Low-Reynolds Number (LRN) term is implemented through the viscous wall destruction terms,

$$D_T = \nu \frac{\partial \sqrt{k_T}}{\partial x_j} \frac{\partial \sqrt{k_T}}{\partial x_j} \quad (5.16)$$

and

$$D_L = \nu \frac{\partial \sqrt{k_L}}{\partial x_j} \frac{\partial \sqrt{k_L}}{\partial x_j}. \quad (5.17)$$

When this near-wall dissipation term is added to the equilibrium of production and destruction we obtain

$$P_{k_T} = k_T \frac{\sqrt{k_T}}{l_{hyb}} - D_T. \quad (5.18)$$

However, in LES mode, no LRN term should be applied. Hence, the last destruction term of the above equation should disappear when operating in LES mode. IDDES provides a useful blending parameters  $f_d$ , which will be used for the LRN limiter. The function  $f_d$  (Equations 5.11 and 2.73) is equal to 0 in LES and equal to 1 in RANS and will therefore limit the viscous wall destruction term to the RANS mode.

$$P_{k_T} = k_T \frac{\sqrt{k_T}}{l_{hyb}} - f_d D_T \quad (5.19)$$

The final form of the turbulent and laminar kinetic energy equations become

$$\frac{Dk_T}{Dt} = P_{k_T} + R_{BP} + R_{NAT} - k_T \frac{\sqrt{k_T}}{l_{hyb}} - f_d D_T + \frac{\partial}{\partial x_j} \left[ \left( \nu + \frac{\alpha_T}{\sigma_k} \right) \frac{\partial k_T}{\partial x_j} \right] \quad (5.20)$$

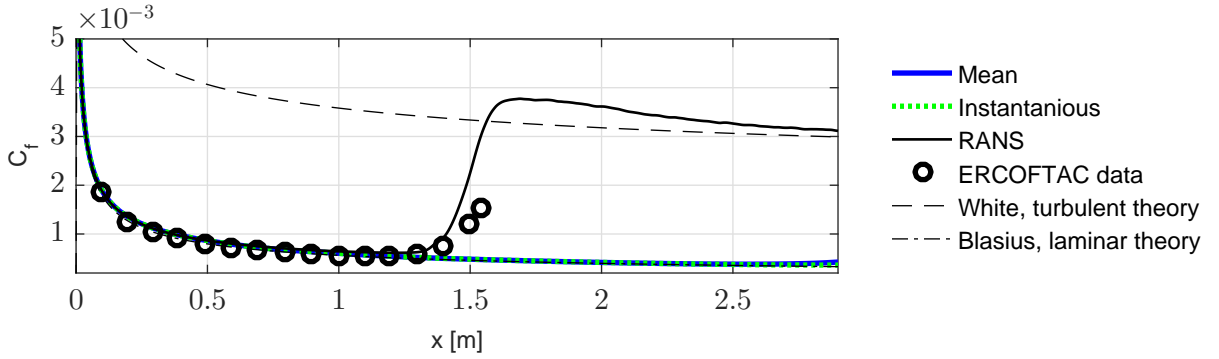
and

$$\frac{Dk_L}{Dt} = P_{k_L} - R_{BP} - R_{NAT} - f_d D_L + \frac{\partial}{\partial x_j} \left[ \nu \frac{\partial k_L}{\partial x_j} \right]. \quad (5.21)$$

All other model equations remain unaltered.

### 5.1.5 Evaluation for flat-plate boundary layers

Due to the modification made to the laminar and turbulent transport equation, the model was unable to predict transition from laminar to turbulent, this is concluded from the skin-friction distribution (Figure 5.3). The time averaged skin-friction and the instantaneous skin-friction both indicate that a fully laminar boundary layer is predicted. Inspection of the energy fields (Figure 5.4) confirms that laminar kinetic energy is present but turbulent kinetic energy is missing.

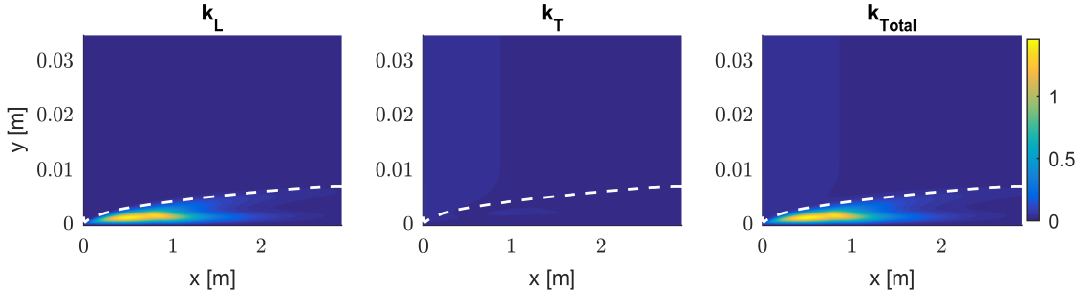


**Figure 5.3:** Wall skin-friction coefficient along the flat-plate.

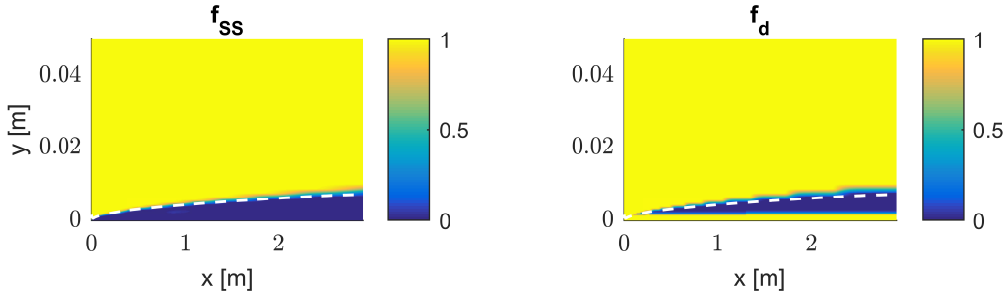
The reason that the turbulent kinetic energy field remains small can be found in Figure 5.5, which depicts the shear-sheltering function (left). The boundary layer is completely in the laminar state and does therefore not allow development of turbulent kinetic energy. The reason why the shear-sheltering function predicts a fully laminar boundary layer lays in the combination of the hybrid blending function  $f_d$  (Figure 5.5) and the shear-sheltering function

$$f_{SS} = \exp \left[ - \left( \frac{C_{SS} \nu \Omega}{k_T} \right)^2 \right]. \quad (5.22)$$

Since the LES branch is active in the boundary layer, it increases the destruction term (Equation 5.3) and  $k_T$  remains therefore small. In fact, the destruction term is so large that  $k_T$  never grows large enough to activate the shear-sheltering function.



**Figure 5.4:** Kinetic energy fields above the flat-plate. *Left*, laminar kinetic energy. *Middle*, turbulent kinetic energy. *Right*, total modelled kinetic energy. White dashed line: 99% boundary layer thickness.



**Figure 5.5:** The shear-sheltering function (*left*) returns a fully laminar boundary layer. Throughout the majority of the laminar boundary layer the LES branch is activated by the hybrid blending function (*right*), RANS=1 and LES=0. White dashed line: 99% boundary layer thickness.

## 5.2 Modification 2: background transition model

In order to circumvent the direct relation between the LES destruction term and the shear-sheltering function a *background* RANS model will simulate parallel to the hybrid method. To make this possible, three RANS fields are introduced:  $k_{L,RANS}$ ,  $k_{T,RANS}$  and  $\omega_{RANS}$  are computed using the original equation of the  $kk_L - \omega$  model but will make use of the same velocity and pressure field as the hybrid method. Not only the shear-sheltering function but also the intermittency and the threshold function of natural and bypass transition are now a function of the *background* RANS fields. The new equation, which make use of the *background* RANS fields read

$$f_{INT} = \min\left(\frac{k_{T,RANS}}{C_{INT}k_{TOT,RANS}}, 1\right), \quad (5.23)$$

$$f_{SS} = \exp\left[-\left(\frac{C_{SS}\nu\Omega}{k_{T,RANS}}\right)^2\right], \quad (5.24)$$

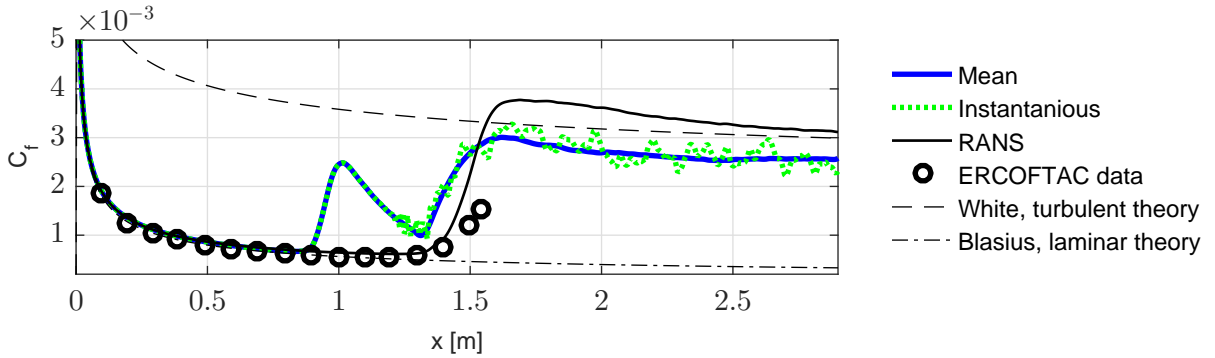
$$\phi_{BP} = \max \left[ \left( \frac{k_{T,RANS}}{\nu\Omega} - C_{BP,crit} \right), 0 \right] \quad (5.25)$$

and

$$f_{NAT,crit} = 1 - \exp \left( - C_{NC} \frac{\sqrt{k_{L,RANS}d}}{\nu} \right). \quad (5.26)$$

### Evaluation for flat-plate boundary layers

The background transition model makes it possible for the flow to turn turbulent, this is concluded from the chordwise skin-friction distribution in Figure 5.6. Transition takes place at  $x = 0.9m$ , but right after the transition peak a decrease in the skin-friction is observed. This decrease is present since turbulent structures have not yet been developed, this phenomenon is referred to as the *grey area* (Section 2.5.1). The *grey area* was also observed in the flat-plate simulation using the SA IDDES model, see Section 3.3.



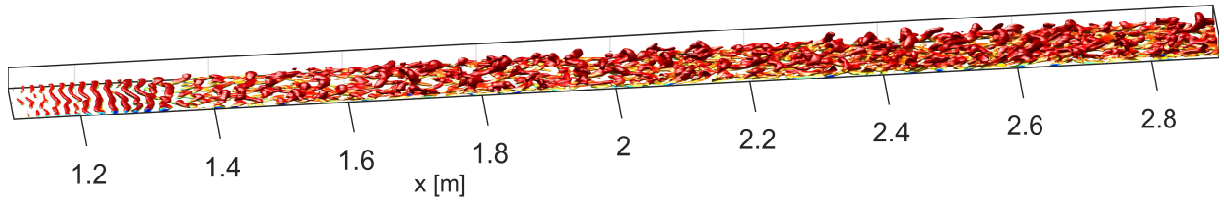
**Figure 5.6:** Wall skin-friction coefficient along the flat-plate.

In the *grey area* the LES branch is active and a reduced eddy viscosity allows for the formation of 2-dimensional vortical structures as can be seen in Figure 5.7. The vortical structures in the flow are visualized by plotting the iso-contours of

$$Q = \frac{1}{4}(\Omega^2 + 2S_{ij}S_{ij}), \quad (5.27)$$

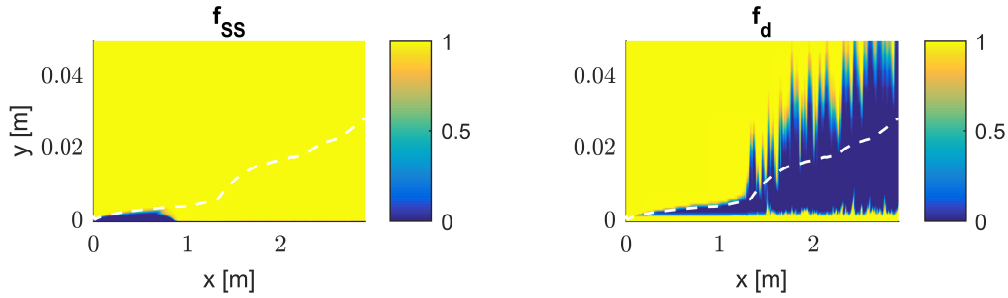
where  $S_{ij}$  is the strain rate and  $\Omega$  is the rotation rate. Only after  $x = 1.3m$ , resolved transition takes place: the resolved 2-dimensional vortical structures break up into 3-dimensional turbulence and the skin-friction increases again. It is noted that the location of the resolved transition location, almost coincides with the experimental data by ERCOFTAC, however this is expected to be a coincidence.





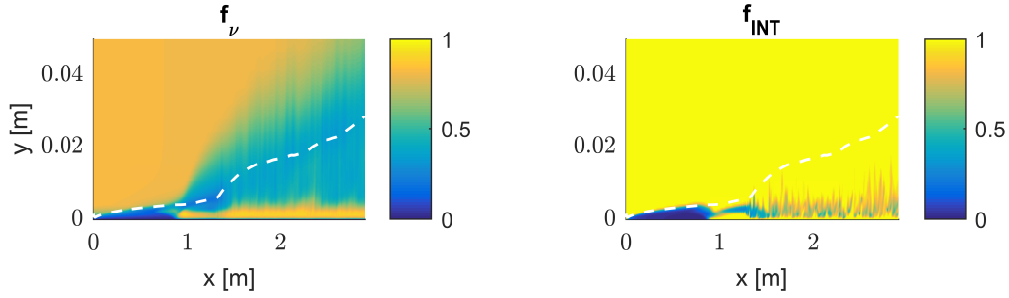
**Figure 5.7:** Vorticity contours ( $Q\text{-iso}=1e4$ ) coloured with velocity magnitude.

That the modelled transition takes place too soon corresponds with the behaviour of the shear-sheltering function, Figure 5.8. At the outer edge of the laminar boundary layer, the LES branch is active and reduces the quantity  $k_T$ . Due to a decrease in  $k_T$  the strain rate is reduced. Subsequently, the background RANS model observes the reduced strain rate and compensates for this by increasing  $k_{T,RANS}$ . Finally,  $k_{T,RANS}$  becomes so large that it activates the shear-sheltering transition threshold function and transition to turbulence takes place.



**Figure 5.8:** The shear-sheltering function (*left*) indicates premature transition at  $x=0.9$  [m] and the hybrid blending function (*right*) shows that the LES branch is active in the laminar boundary layer, RANS=1 and LES=0. White dashed line: 99% boundary layer thickness.

Another issue that is observed, is that inside the LES branch damping terms become unexpectedly active. The viscous wall damping term and the intermittency function are visualized in Figure 5.9. The value of the viscous wall term should be equal to 1 inside a turbulent boundary, however in the current simulation it is less than 0.5. Also the intermittency function should take the value 1 inside a turbulent boundary layer, but predicts a much lower value in the near-wall region the turbulent boundary layer. The same sort of behaviour was observed for the damping function  $f_W$ . Although their effects on  $f_W$  is very small, it may pose a problem in simulations of different flows.



**Figure 5.9:** Excessive damping in the turbulent boundary layer is found in the viscous wall function (*left*) and intermittency function (*right*). White dashed line: 99% boundary layer thickness.

A lower value for these damping and limiter functions will have an effect on the eddy viscosity and will therefore pollute the development of the boundary layer characteristics.

### 5.3 Modification 3: LES limiter and damping term correction

Two problems have been identified after the second modification: premature transition and the excessive damping of the turbulent eddy viscosity. First, the premature transition of the background RANS model can be prevented when the LES region can not extend inside the laminar boundary layer. This is accomplished by introducing a new blending function  $f_{SS,d}$  (the  $d$  stands for delayed), which makes use of the shear-sheltering function. All areas where  $f_{SS}$  is smaller or equal to 0.99 must operate in RANS mode

$$f_{SS,d} = \begin{cases} 1 & \text{if } f_{SS} > 0.99 \\ 0 & \text{if } f_{SS} \leq 0.99 \end{cases}. \quad (5.28)$$

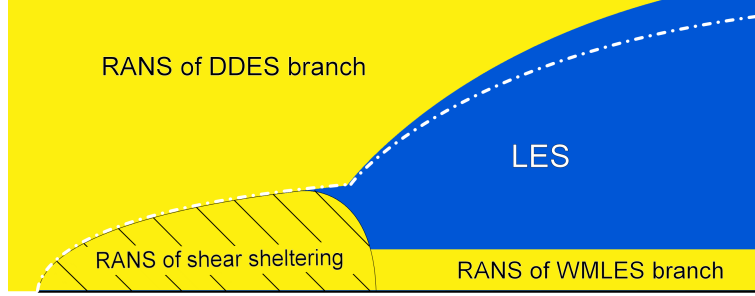
This is made possible using

$$f_{SS,d} = 1 - \exp(-10^4 \cdot \max[f_{SS} - 0.99, 0]). \quad (5.29)$$

The new hybrid blending function then becomes

$$f_d = \max(1 - f_{dt}, f_B, 1 - f_{SS,d}), \quad (5.30)$$

where  $1 - f_{dt}$  is the DDES blending function,  $f_B$  is the WMLES blending function and  $1 - f_{SS,d}$  is the new shear-sheltering blending function. The addition of  $1 - f_{SS,d}$  will therefore activate the RANS branch in the laminar boundary layer. A sketch of the expected blending behaviour is visualized in Figure 5.10. This figure includes the DDES branch ( $1 - f_{dt}$ ), the WMLES branch ( $f_B$ ) and the shear-sheltering branch ( $1 - f_{SS,d}$ ).



**Figure 5.10:** Schematic decomposition of the expected RANS/LES regions for a flat-plate simulation. RANS=yellow. LES=blue.

The excessive damping of the modelled turbulent region by  $f_\nu$ ,  $f_{INT}$  and  $f_W$  is undesired, as it destroys the Smagorinsky like LES behaviour. The solution is to correct the values of  $f_\nu$ ,  $f_{INT}$  and  $f_W$  by their default values (as intended by the  $kk_L - \omega$  model) inside the LES branch. This is done using the hybrid blending function  $f_d$ . The new equations become

$$f_\nu = \max \left[ 1 - \exp \left( - \frac{\sqrt{Re_T}}{A_\nu} \right), (1 - f_d) \right], \quad (5.31)$$

$$f_{INT} = \max \left[ \min \left( \frac{k_T}{C_{INT} k_{TOT}}, 1 \right), (1 - f_d) \right] \quad (5.32)$$

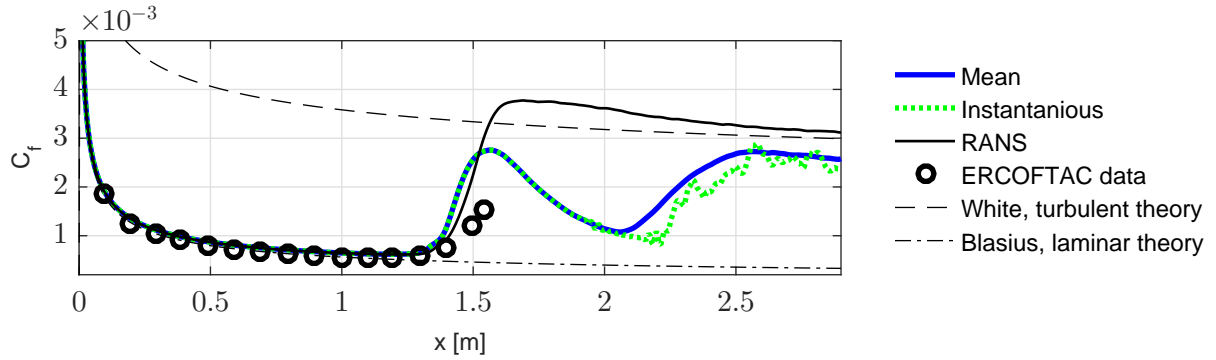
and

$$f_W = \max \left[ \left( \frac{\lambda_{eff}}{\lambda_T} \right)^{2/3}, (1 - f_d) \right]. \quad (5.33)$$

The first terms on the right hand side of the above equations are the original relations and were presented in Section 2.3.2.

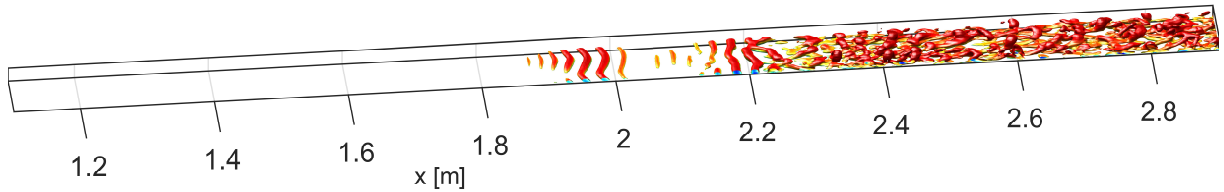
### Evaluation for flat-plate boundary layers

The  $kk_L - \omega$  IDDES model now performs as desired. Due to the inclusion of the shear-sheltering blending function  $f_{SS,d}$ , the transition location is now postponed and corresponds to the RANS simulation, as can be seen from the skin-friction distribution in Figure 5.13. As expected, the *grey area* between the modelled transition ( $x = 1.4\text{m}$ ) and resolved transition point ( $x = 2.2\text{m}$ ) remains present since the development of turbulence requires time.



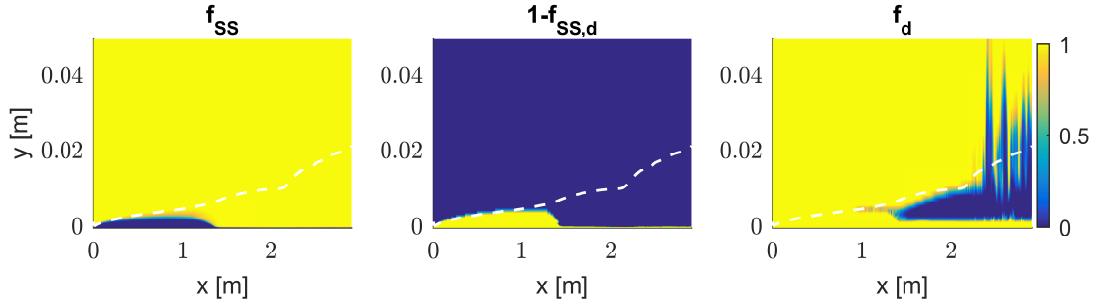
**Figure 5.11:** Wall skin-friction coefficient along the flat-plate.

After the modelled transition point, the eddy viscosity is reduced and 2-dimensional Tollmien-Schlichting waves are formed, (Figure 5.12). Due to the reduced eddy viscosity and the absence of turbulent structures, the wall shear stress is reduced. Instabilities in the flow finally result in the formation of turbulence around  $x = 2.2\text{m}$  and the wall shear stress increases again.



**Figure 5.12:** Vorticity contours ( $Q\text{-iso}=1e4$ ) coloured with velocity magnitude.

The shear-sheltering limiter function  $f_{SS,d}$  is shown in Figure 5.13, the RANS region covers the laminar boundary layer. Its effect on the blending function  $f_d$  is clear, the LES branch is, as intended, limited to the turbulent boundary layer only.



**Figure 5.13:** *Left:* shear-sheltering function. Laminar=0, turbulent=1. *Middle:* RANS/LES blending by shear-sheltering branch. *Right:* total RANS/LES blending. LES=0, RANS=1. White dashed line: 99% boundary layer thickness.

A working transitional hybrid RANS/LES model is constructed from the  $kk_L - \omega$  turbulence model and combined with the IDDES blending function. This model does suffer from the *grey area* just as other hybrid RANS/LES models and is therefore not a truly transitional hybrid RANS/LES model.

## 5.4 Modification 4: Addition of tripping mechanism

The transition process from the laminar RANS boundary layer to the fully resolved turbulent boundary layer takes place too slowly (Figure 5.12). This is a result of the steady inflow conditions of the free-stream flow and laminar boundary layer. To circumvent the delayed development of turbulence, a tripping mechanism is introduced. Turbulent fluctuations will be added to the flow near the transition location. The magnitude of the fluctuating velocities will scale with the laminar kinetic energy field since this field contains the pretransitional boundary layer energy. Since the transition location is partially solved using LES, the transition will be of a transient type. For this unsteady process, a model is required which can also introduce the velocity fluctuations in an unsteady flow. Furthermore, it is difficult to establish a two dimensional plane for the turbulent transition phenomena which is characterized by three-dimensional structures. A method which can substitute velocity fluctuations inside a 3-dimensional dynamic domain is designed and presented in this Section.

For this, the shear-sheltering function is used to define the location where the velocity fluctuations will be added to the flow. First, the gradient of the shear-sheltering region is taken to define a sharp edge between the laminar and turbulent region. Second, only when flow exits the laminar region, the tripping edge is active. At last, the intensity of the added fluctuations shall scale with the local laminar kinetic energy.

The LES limiter function  $f_{SS,d}$  has a sharp transition from 0 to 1 and can therefore provide a sharp edge that can mark the transitional region. To prevent the addition of velocity fluctuations in ambiguous grids, the RANS/LES blending function  $f_B$  of the WMLES branch,

limit the addition of velocity fluctuations in the near-wall region.

$$x_{SS,d} = \begin{cases} 1 & \text{if } \nabla f_{SS,d} > 0 \text{ and } (1 - f_B) \leq 0.5 \\ 0 & \text{if } \nabla f_{SS,d} = 0 \text{ and } (1 - f_B) > 0.5 \end{cases}. \quad (5.34)$$

This is made possible by using

$$x_{SS,d} = [1 - \exp(-\nabla f_{SS,d})] \cdot f_B. \quad (5.35)$$

Velocity fluctuations should only be substituted at the edge  $x_{SS,d}$  when the laminar flow enters the turbulent region. For this it is first determined whether flow enter or exits the laminar area. This is done by taking the gradient of a smooth function which defines the laminar and turbulent regions. For this a modified form of the intermittency function is used

$$f_{\nabla INT,mix} = \nabla \min \left( \frac{k_{T,RANS}}{C_{INT}(k_{T,RANS} + k_L)}, 1 \right). \quad (5.36)$$

This equation combines the turbulent kinetic energy field of the background RANS model and the laminar kinetic energy equation and assures that the intermittency in the turbulent region is 1. The fractional gradient component is determined by

$$f_{\nabla INT,mix(i,j,k)} = \frac{f_{\nabla INT,mix}}{\text{mag}(\nabla f_{\nabla INT,mix})}, \quad (5.37)$$

as well as the fractional velocity component

$$u_{(i,j,k)} = \frac{U}{\text{mag}(U)}. \quad (5.38)$$

Multiplying the fractional gradient component with velocity component will return negative values when flow enters the laminar region and returns positive values when the flow leaves the laminar region

$$V_{direction} = \begin{cases} > 0 & \text{if flow exists the laminar region} \\ < 0 & \text{if flow enters the laminar region} \end{cases}. \quad (5.39)$$

This is obtained by

$$V_{direction} = f_{\nabla INT,mix(i,j,k)} \cdot u_{(i,j,k)}. \quad (5.40)$$

The combinations of the volume field  $V_{direction}$ , and the sharp boundary  $x_{SS,d}$  (defined by the edge of the laminar region), is one where the laminar boundary layer becomes turbulent and zero everywhere else

$$V_{SS,d} = \begin{cases} 1 & \text{if } x_{SS,d} > 0 \text{ and } V_{direction} > 0 \\ 0 & \text{if } x_{SS,d} = 0 \text{ and } V_{direction} < 0 \end{cases}. \quad (5.41)$$

This is possible by removing all negative values and setting all positive values to 1,

$$V_{SS,d} = 1 - \exp(-10^{10} \cdot x_{SS,d} \cdot V_{boundary}). \quad (5.42)$$

The final fluctuations that are substituted to the flow, are determined using

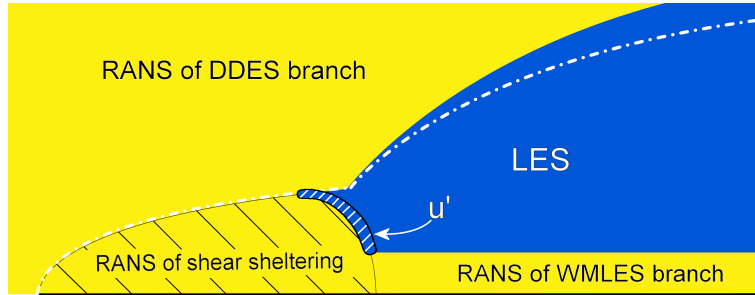
$$u' = \alpha \cdot V_{SS,d} \cdot RANDOM \cdot \sqrt{\frac{2}{3} k_L}, \quad (5.43)$$

where  $\alpha$  is a scaling function which can be used for calibration and *RANDOM* is a volume field with random values ranging from -0.5 to 0.5. The velocity fluctuations  $u'$  will be added to the velocity field at the end of every time step. Due to the addition of the non-divergence free fluctuations, the PISO solver will require additional pressure-velocity correction to make the velocity field divergence free. The modified PISO solver can be found in Appendix E.

To make sure that the unsteady velocity fluctuations are solved in the LES branch the blending function  $f_d$  is supplemented with an extra term that marks the region with velocity fluctuations

$$f_d = \max(1 - f_{dt}, f_B, 1 - f_{SS,d}) \cdot (1 - V_{SS,d}), \quad (5.44)$$

A schematic representation of the RANS/LES regions and the addition of the velocity fluctuations is visualized in Figure 5.14. The velocity fluctuations  $u'$  are located at the RANS/LES interface of the RANS shear-sheltering branch. Near the wall,  $u'$  is limited by the RANS WMLES branch. In the upstream direction  $u'$  is limited because flow enters the shear-sheltering branch, instead of leaving.



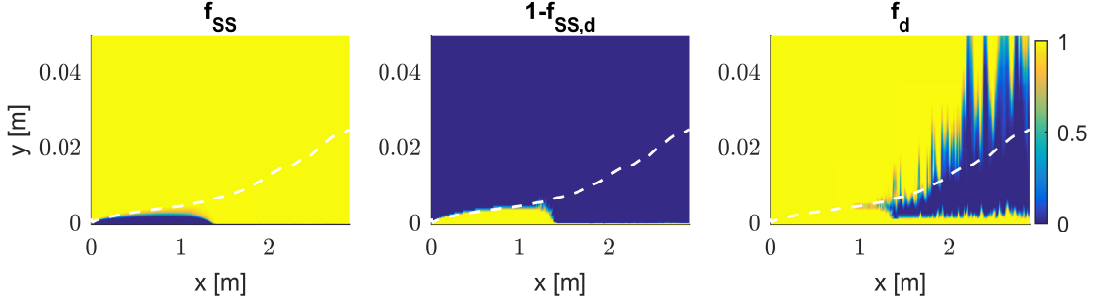
**Figure 5.14:** Schematic decomposition of RANS/LES regions for a flat-plate simulation. RANS=yellow. LES=blue.

An attempt was made to replace the non-divergence free fluctuation by divergence free fluctuations. Unfortunately, this attempt failed because of an unstable behaviour. The method is explained in Appendix B.

### Evaluation for flat-plate boundary layers

The shear-sheltering function  $f_{SS}$  and the shear-sheltering blending function  $f_{SS,d}$  are shown in the left and middle graph of Figure 5.15. Compared to the results from modification 3

(Figure 5.13), little changed in shape are present. The laminar and turbulent regions are therefore expected to be predicted correctly. The RANS/LES region, depicted in the right graph of Figure 5.15 shows that the LES region starts to grow in height right after the transition location, indicating that a thick turbulent boundary layer is forming.



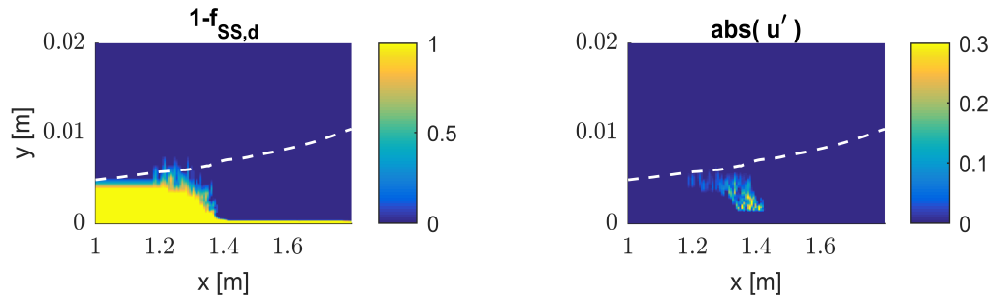
**Figure 5.15:** *Left:* shear-sheltering function. Laminar=0, turbulent=1. *Middle:* RANS/LES blending by shear-sheltering branch. *Right:* total RANS/LES blending. LES=0, RANS=1. White dashed line: 99% boundary layer thickness.

Addition of the time averaged velocity fluctuations is visualized in the right graph of Figure 5.16. The graph also indicates the 99 percent boundary layer thickness. It can be seen that the introduced fluctuations are limited to the laminar boundary thickness. In the near wall region the fluctuations do not exist, since they are limited by the WMLES branch  $f_B$ . In the stream-wise direction the fluctuations are active over a rather larger region than what was expected, from the gradient of  $f_{SS,d}$  (Figure 5.13). However, also in the left graph of Figure 5.16 a less sharp gradient is observed at the same region as where the fluctuations are substituted. It is concluded that the substituted fluctuations have a direct effect on the rotation rate of the local flow. Recall that the shear-sheltering function is based on the rotation rate tensor

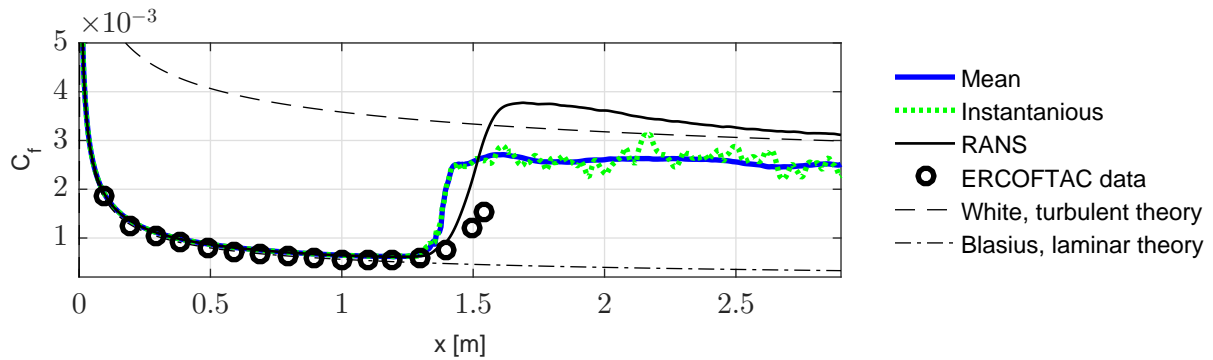
$$f_{SS} = \exp \left[ - \left( \frac{C_{SS} \nu \Omega}{k_T} \right)^2 \right]. \quad (5.45)$$

This suggests that there is a slight feedback loop between  $u'$  and  $k_L$ . This model is therefore not optimal. However, after the initial modelled transition location at  $x \approx 1.4\text{m}$  the skin-friction, Figure 5.17, stays almost constant in the turbulent region. It can therefore be concluded that the *grey area* problem is reduced by the addition of artificial turbulence (velocity fluctuations). At the transition point, the skin-friction gradient is larger than the one from the RANS result. This increase in gradient is expected to be an indirect result of the added velocity fluctuations. An increase in eddy viscosity was observed in the near-wall RANS region, right below the location with the added velocity fluctuations.



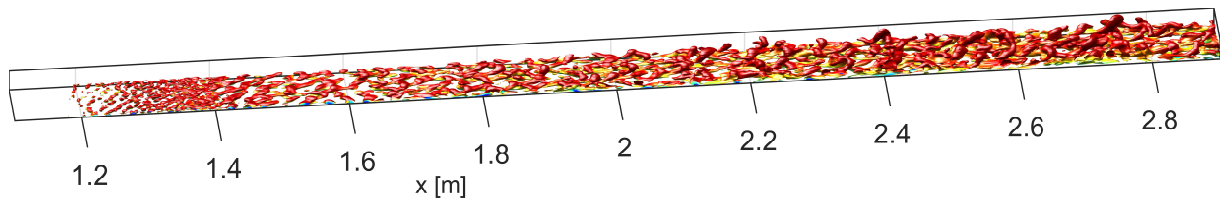


**Figure 5.16:** Close-up of transition point. *Left:* RANS/LES bleding by shear-sheltering branch determines the location for  $u'$ . *Right:* substituted velocity fluctuation  $\sqrt{u'^2}$ , located at the outflow edge of  $f_{SS,d}$ . White dashed line: 99% boundary layer thickness.



**Figure 5.17:** Wall skin-friction coefficient along the flat-plate.

Visualization of the turbulent flow structures can be found in Figure 5.18. After the velocity fluctuations are substituted ( $1.2\text{m} < x < 1.4\text{m}$ ) the downstream flow indicates that three-dimensional flow structures remain present and continue to develop.



**Figure 5.18:** Vorticity contours ( $Q\text{-iso}=1e4$ ) coloured with velocity magnitude. Small structures in the region between  $x \approx 1.2\text{m}$  and  $x \approx 1.4\text{m}$  are the added velocity fluctuations of the tripping mechanism.

These initial results are satisfactory, transition is initiated at the same location as the RANS

result and the experimental data predicts. The increase in skin-friction at transition is steeper than what would be expected from the RANS results and the skin-friction in the turbulent regime is underpredicted but the typical *grey area* of DES simulations is bypassed. This model may therefore be called a laminar-turbulent transitional hybrid RANS/LES model.

## 5.5 Model summary

Modifications and additions to the IDDES framework, the  $kk_L - \omega$  turbulence model and the PISO solver are listed below. The OpenFOAM solvers files `kkLOmegaIDDES.H` and `kkLOmegaIDDES.C` can be found in Appendices F.1 and F.2, respectively. Appendix E contains the modified PISO solver.

- Modifications to IDDES

The original RANS/LES blending function of IDDES is supplemented with two new branches, the shear-sheltering branch ( $f_{SS,d}$ ) and a velocity fluctuating branch ( $V_{SS,d}$ )

$$f_d = \max(1 - f_{dt}, f_B, 1 - f_{SS,d}) \cdot (1 - V_{SS,d}). \quad (5.46)$$

Modifications to the near-wall RANS elevation function

$$f_e = \max[(f_{e1} - 1), 0] \Psi \cdot f_{e2} \cdot f_{INT}, \quad (5.47)$$

are the addition of the intermittency function ( $f_{INT}$ ) and a new relation for  $\Psi$

$$\Psi = 500 \exp \left[ -5.5 \left( \frac{\nu_t}{\nu} \right)^{1/9} \right] + 1. \quad (5.48)$$

- Modifications to  $kk_L - \omega$

The new turbulent transport equation are

$$\frac{Dk_T}{Dt} = P_{k_T} + R_{BP} + R_{NAT} - k_T \frac{\sqrt{k_T}}{l_{hyb}} - f_d D_T + \frac{\partial}{\partial x_j} \left[ \left( \nu + \frac{\alpha_T}{\sigma_k} \right) \frac{\partial k_T}{\partial x_j} \right] \quad (5.49)$$

and

$$\frac{Dk_L}{Dt} = P_{k_L} - R_{BP} - R_{NAT} - f_d D_L + \frac{\partial}{\partial x_j} \left[ \nu \frac{\partial k_L}{\partial x_j} \right]. \quad (5.50)$$

Modelled transition did not take place when the laminar to turbulent transition parameters were coupled to the sub-grid scale destruction term. Therefore, the three original turbulent transport equations are included ( $k_{L,RANS}$ ,  $k_{T,RANS}$  and  $\omega_{RANS}$ ) to provide the RANS solution to the terms responsible for transition

$$f_{INT} = \min \left( \frac{k_{T,RANS}}{C_{INT} k_{TOT,RANS}}, 1 \right). \quad (5.51)$$

$$f_{SS} = \exp \left[ - \left( \frac{C_{SS}\nu\Omega}{k_{T,RANS}} \right)^2 \right], \quad (5.52)$$

$$\phi_{BP} = \max \left[ \left( \frac{k_{T,RANS}}{\nu\Omega} - C_{BP,crit} \right), 0 \right] \quad (5.53)$$

and

$$f_{NAT,crit} = 1 - \exp \left( - C_{NC} \frac{\sqrt{k_{L,RANS}d}}{\nu} \right). \quad (5.54)$$

The reduced sub-grid scale kinetic energy resulted in incorrect values for  $f_\nu$ ,  $f_{INT}$  and  $f_W$ . Using the following modified forms

$$f_\nu = \max \left[ 1 - \exp \left( - \frac{\sqrt{Re_T}}{A_\nu} \right), (1 - f_d) \right], \quad (5.55)$$

$$f_{INT} = \max \left[ \min \left( \frac{k_T}{C_{INT}k_{TOT}}, 1 \right), (1 - f_d) \right] \quad (5.56)$$

and

$$f_W = \max \left[ \left( \frac{\lambda_{eff}}{\lambda_T} \right)^{2/3}, (1 - f_d) \right], \quad (5.57)$$

it is made sure that their values return to 1, inside the LES region of the turbulent boundary layer.

- Modifications to PISO

Development of turbulent structures at the transition location is promoted by the addition of velocity fluctuations

$$u' = \alpha \cdot V_{SS,d} \cdot RANDOM \cdot \sqrt{\frac{2}{3}k_L}, \quad (5.58)$$

where  $V_{SS,d}$  is one where the laminar boundary layer becomes turbulent and zero everywhere else,  $\alpha$  is a scaling function,  $RANDOM$  assigns a random intensity between -0.5 and 0.5 and  $k_L$  is the pretransitional kinetic energy.



---

## Chapter 6

---

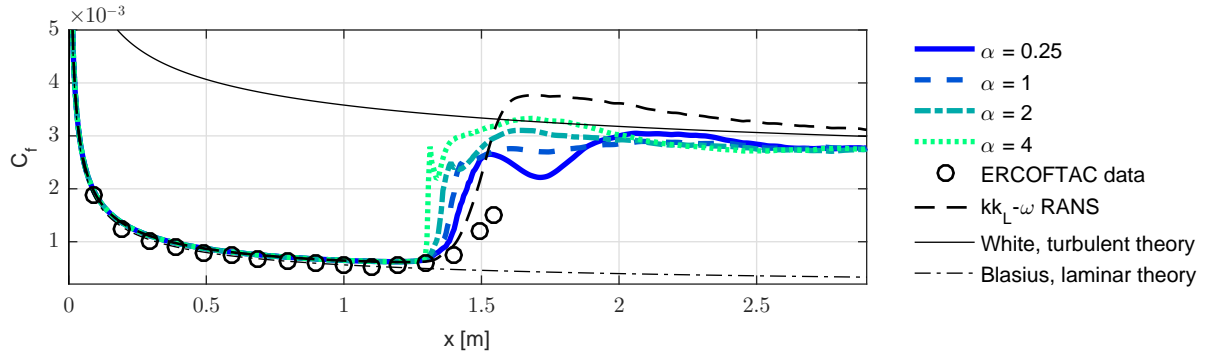
# Verification and Validation

Two new model properties have been introduced, the RANS elevating function and the tripping mechanism. Their response to different mesh resolutions and inflow conditions are examined in this Section 6.1 and Section 6.2, respectively. A mesh convergence study is presented in Section 6.3 and a validation study is performed Section 6.4.

### 6.1 Effect of tripping intensity

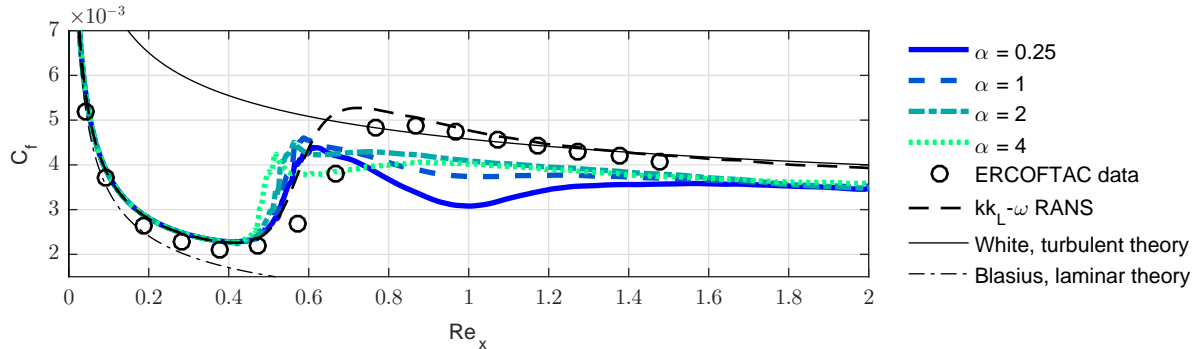
The stochastic velocity fluctuations do not realistically represent the fluid motion typically present in the transitional boundary layer. It might therefore be the case that the development of coherent turbulent structures is delayed or exaggerated. In order to establish the response of the turbulent boundary layer, flat-plate simulations are performed with different intensities of the velocity fluctuations. The simulations are performed with a turbulent inflow condition with  $Tu = 0.8\%$  and  $Tu = 3\%$ .

When the velocity fluctuations are scaled down by a factor of four, the typical *grey area* can be observed in Figure 6.1. The initial increase in wall-friction is due to the modelled turbulence. Only at  $x = 1.7\text{m}$  the wall friction increases again due to the development of turbulence. The opposite behaviour is observed when the fluctuations are increased. The maximum wall-friction increases and also the transition location moves forward. For all four cases, the wall friction converges downstream of the transition location.



**Figure 6.1:** Wall friction coefficient with  $Tu = 0.8\%$  for different values of the tripping intensity  $\alpha$ .

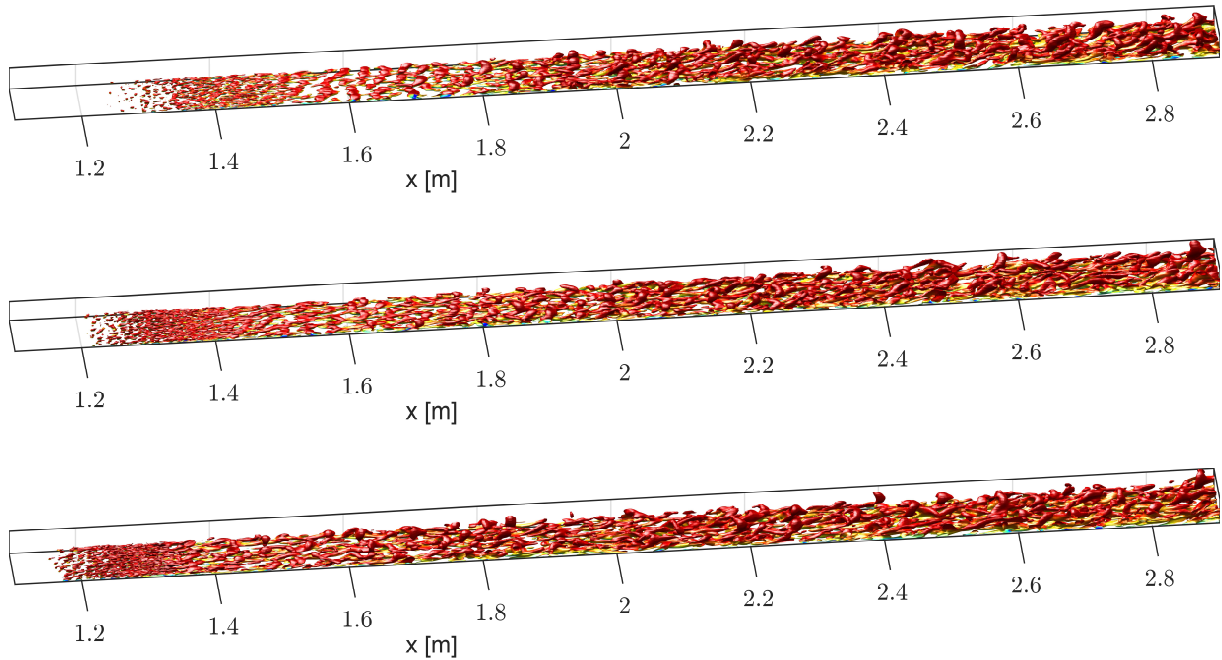
The simulation with an increases free-stream turbulence level,  $Tu = 3\%$ , returns a slightly different behaviour of the wall friction. Not only the down scaled fluctuations but also the non-scaled fluctuations show a slight decrease in the wall friction after the modelled transition location. Increasing the fluctuations to a factor of 4 completely removes the dip in the skin-friction but does create an unrealistic sharp peak in the wall friction at  $x = 0.51\text{m}$ .



**Figure 6.2:** Wall friction coefficient with  $Tu = 3\%$  for different scaling factors (alpha) of the tripping intensity.

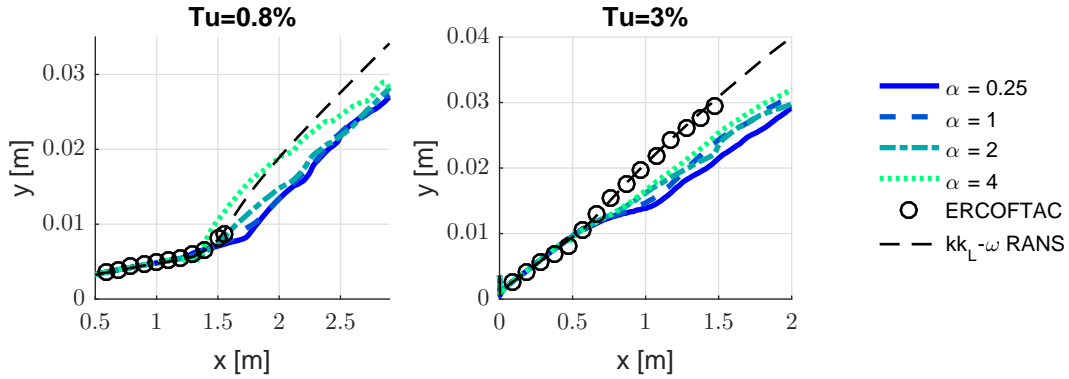
A visualization of the flow structure will aid in the interpretation of the transition process and is therefore depicted in Figure 6.3. The small turbulent structures in the region between  $x = 1.2\text{m}$  and  $x = 1.4\text{m}$  are mainly generated by the tripping fluctuations. The case of the down scaled fluctuations, two-dimensional vortical structures can be recognized at  $x = 1.6\text{m}$ . Only at  $x = 1.7\text{m}$  the forming of hairpin vortices take place. In the middle graph, the two-dimensional structures are completely replaced by three-dimensional structures. Increasing the intensity of the velocity fluctuation even further, continuous to increases of the size of the turbulent structures. This increase in turbulent structures is supported by the chordwise boundary layer thickness, Figure 6.4. Both cases indicate that with increasing fluctuation velocities, the boundary layer starts to grow sooner. When the fluctuating velocities are increased by a factor of 4, we observe a very rapid increase in boundary layer thickness for

the  $Tu = 0.8\%$  case. Because the rate at which the boundary layer grows, deviates from the results at lower values of  $\alpha$  and because it does not follow the trend of the RANS result, it is concluded that for this specific case the value of  $\alpha = 4$  is too large.



**Figure 6.3:** Vorticity contours ( $Q\text{-iso}=1e4$ ) coloured with flow velocity.  $Tu = 0.8\%$ . *Top:*  $\alpha = 0.25$ . *Middle:*  $\alpha = 1$ . *Bottom:*  $\alpha = 4$ .

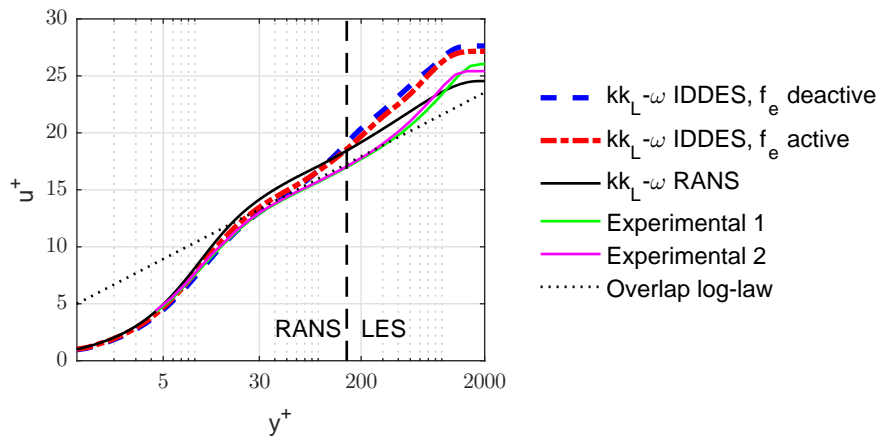
While the  $Tu = 0.8\%$  case shows that the boundary layer thickness converges to the same value downstream, it seems that the boundary layer thickness in the  $Tu = 3\%$  case is shifted upstream with increasing fluctuation intensity. It is noted that the introduced fluctuation only represent the pretransitional energy of the laminar boundary. Any free-stream fluctuations which are present, just above the turbulent boundary layer are not accounted for. This could have an effect on the development of the turbulent boundary layer when the free-stream turbulent energy is high. Furthermore, the tripping mechanism introduces random fluctuations and hence, does not distinguish between bypass transition and natural transition.



**Figure 6.4:** Boundary layer thickness ( $\delta_{99\%}$ ) along flat-plate for different intensities of the tripping scale  $\alpha$ .

## 6.2 RANS elevating-function

The purpose of the elevating function  $f_e$  (Equation 5.47) is to remove the mismatch of the boundary layer profile at the RANS/LES interface by increasing the eddy viscosity in the RANS region (Equation 2.74). The mean velocity profile at  $x = 2.6\text{m}$  for simulation with the elevating function deactivated and activated are visualized in Figure 6.5. The effect of the elevating function on the new model does reduce the miss-match in the log-layer partially and slightly approaches the experimental data by Osterlund (1999). However, based on the good results obtained by Shur et al. (2008) with the introduction of  $f_e$  for IDDES, a much better improvement was expected for the  $kk_L - \omega$  IDDES model. It is therefore questionable whether  $f_e$  works properly in combination with the  $kk_L - \omega$  turbulence model.



**Figure 6.5:** Effect of IDDES elevating function  $f_e$  on the non-dimensional velocity profile at  $x = 2.5\text{m}$ .

An additional improvement of the velocity profile could be obtained by calibration of model parameters. Due to differences in model equations between SA and  $kk_L - \omega$ , the effect of the



elevating function  $f_e$ , on the eddy viscosity could explain the discrepancies in the velocity profile. Shur et al. (2008) mentioned that calibration of the elevating coefficients  $C_t$  and  $C_l$  (Equations 2.69 and 2.70, respectively) is required when imposing the IDDES structure to a different RANS model. As was discussed in Section 2.5.1,  $C_t$  and  $C_l$  control the intensity of the additional eddy viscosity in the laminar viscous layer and the turbulent boundary layer, respectively. Several simulation have been performed in which only  $C_t$  and  $C_l$  have been scaled. However, the changes in the velocity profile showed little consistency. It remains therefore unclear whether  $f_e$  could remove the log-layer mismatch at all. Three additional simulations, with equal scaling of  $C_t$  and  $C_l$ , are performed to show the response of the elevating coefficients on the boundary layer profile. The results are depicted in Figure 6.6 and only show a minor response to the relative large changes of the elevating coefficients.

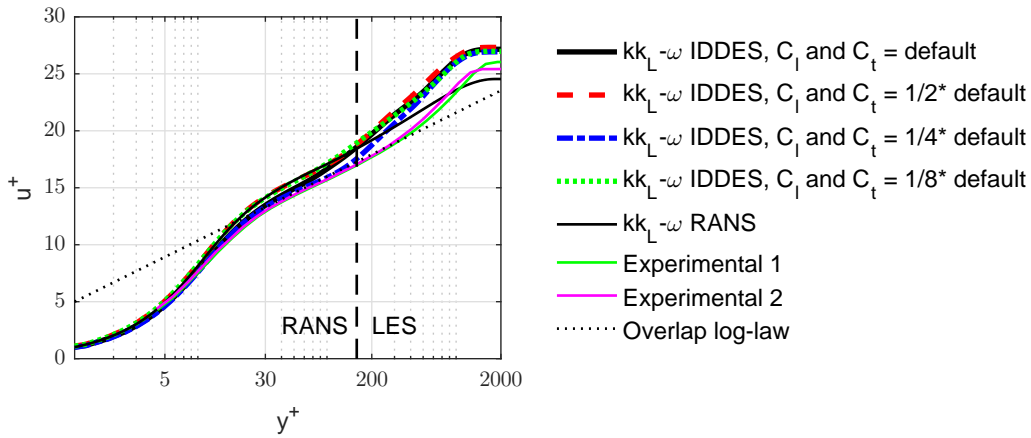


Figure 6.6: Effect of model coefficients on the non-dimensional velocity profile at  $x = 2.5m$ .

The velocity profile  $u^+$  improves slightly when  $C_t$  and  $C_l$  are decreased by a factor 0.5 and 0.25. Decreasing the elevating coefficients further results in an increased deviation of the velocity  $u^+$ . The coefficients are therefore chosen to be half of the value specified by Shur et al. (2008).

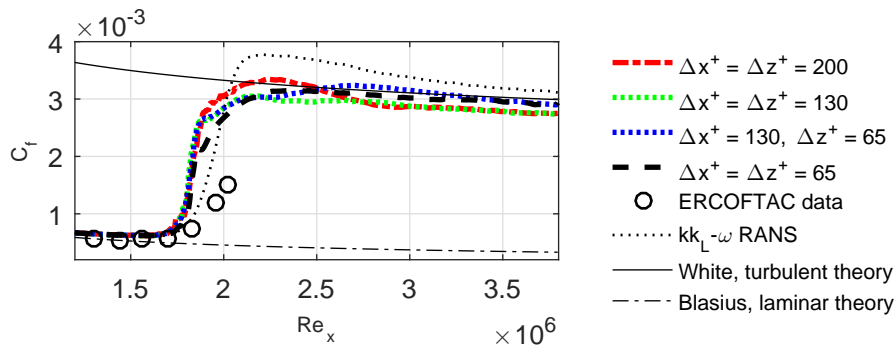
Table 6.1: New RANS elevating constants

$f_e$	$C_t$	$C_l$
active	0.41	0.89

### 6.3 Mesh sensitivity analysis

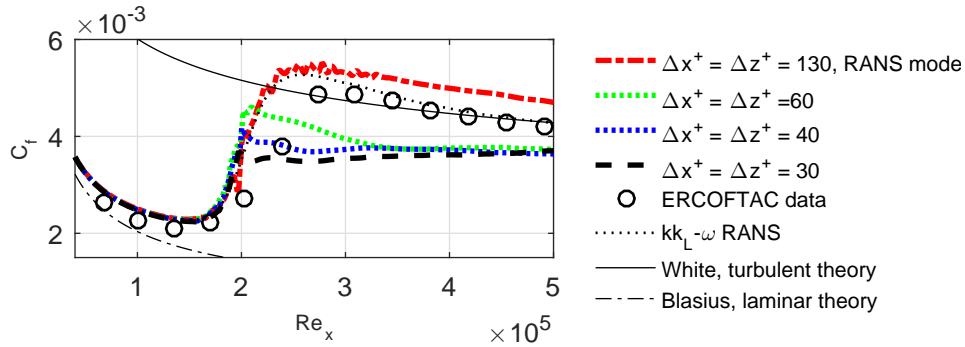
A mesh sensitivity analysis is performed on a flat-plate simulation with a turbulent free-stream intensity of  $Tu = 0.8\%$  and  $Tu = 3\%$ . The wall-normal grid resolution of these simulations is kept constant,  $y^+ < 1$ . Grid refinement is therefore only applied in the chordwise and spanwise direction. When the grid resolution is increased, not only the filter width of the

LES mode changes, but also the location of the RANS/LES interface. The wall-friction for four different grid resolution is depicted in the right graph of Figure 6.7. All four cases predict the same transition location and have an equal increase in the skin-friction. The coarsest grid ( $\Delta x^+ = \Delta z^+ = 200$ ) predicts the strongest peak in the skin-friction and is followed by a drop. Increasing the resolution ( $\Delta x^+ = \Delta z^+ = 130$ ) removes the peak, this is most likely the results of the decreased RANS region in the transitional boundary layer. Increasing the grid resolution in the spanwise direction ( $\Delta x^+ = 130, \Delta z^+ = 65$ ) increases the friction in the turbulent boundary layer, probably because more turbulence is resolved. Increasing also the chordwise resolution ( $\Delta x^+ = 65, \Delta z^+ = 65$ ) flattens the skin-friction slight near the transition location ( $Re_x \approx 1.9e6$ ).



**Figure 6.7:** Wall friction of transitional and turbulent boundary layer.  $Tu = 0.8\%$ .

When the free-stream level is increased, the difference in skin-friction deviates much more with increasing grid resolution, as can be seen in Figure 6.8. For the grid resolution  $\Delta x^+ = 60, \Delta z^+ = 60$  a peak in the skin-friction is observed at the transition location, which is reduced when the grid resolution increases.



**Figure 6.8:** Wall friction of transitional and turbulent boundary layer.  $Tu = 3\%$ .

It is concluded that for both free-stream levels, the transition location is insensitive to the grid resolution that is used in these simulations. Downstream of the transition point, where the turbulent boundary layer is developed, the skin-friction converges for both free-stream turbulence levels. Also, the skin-friction distribution right after transition converges, but to a lower value than what is predicted by the  $kk_L - \omega$  RANS, this is especially the case for the

highest free-stream level ( $Tu = 3\%$ ).

## 6.4 Validation

Flat-plate simulations with three different intensities of the turbulent inflow are performed to validate the transitional behaviour of the  $kk_L - \omega$  IDDES model.

### 6.4.1 Simulation details

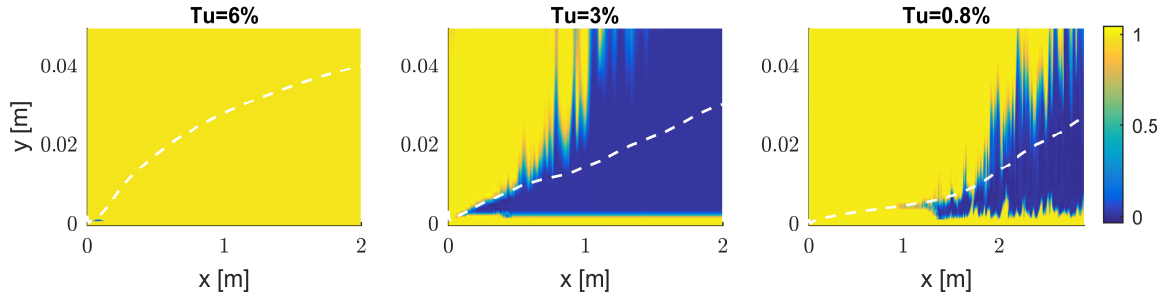
The three cases correspond with experimental simulations from ERCOFTAC, performed by [Coupland \(1990\)](#). The used numerical grid and solver setting were already presented in Section 3.2. The applied inlet boundary conditions are summarized in Table 6.2. These conditions have previously been used by [Fürst \(2013\)](#), whom verified that the decay of free-stream turbulence corresponds with the experimental results.

**Table 6.2:** Overview of the turbulent inlet boundary conditions for the three simulations.

inlet parameter	T3B	T3A	T3A-
$Tu$ [%]	9.43	3.3	0.9
$k_T$ [ $m^2s^{-2}$ ]	1.12827	0.04763	0.04857
$\omega$ [ $s^{-2}$ ]	23.8	56.8	23.8

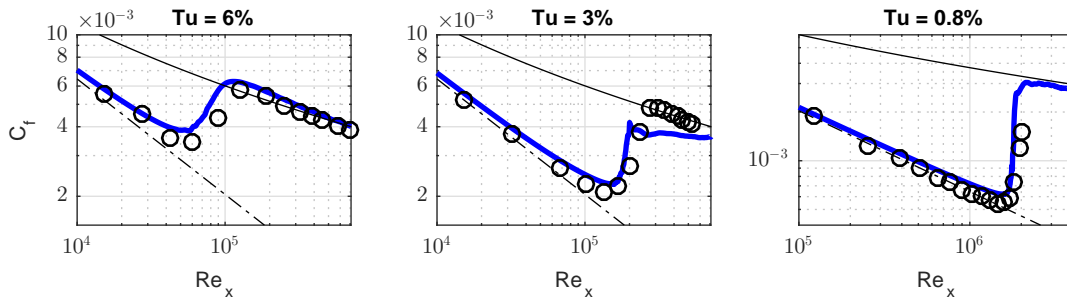
### 6.4.2 Results

Before interpretation of the flow results is possible, it should be known which flow regions are solved using RANS and LES. The RANS/LES blending function  $f_d$  is shown in Figure 6.9. Different results are obtained for the different cases. The left graph implies that nearly the complete flow field is solved in RANS, only right above the leading edge ( $x \approx 0.1$ ) a small region is solved using LES. This is because the WMLES branch ( $f_B$ ) is almost as thick as the laminar boundary layer. Recall that the RANS region from the WMLES branch is only dependent on the grid resolution. It is noted that the grid is already of an LES resolution. However, this is not enough to lower the near-wall delaying function  $f_B$  (Equation 2.64) such that  $u'$  can also become active deeper inside the laminar boundary layer. Decreasing the inflow turbulent content results in a thicker laminar boundary layer, which raises well above the WMLES blending function  $f_B$ , and the turbulent flow is solved using LES (middle graph). The same conclusion applies for the T3A- case with an inlet turbulent intensity of 0.8%, the turbulent boundary layer is solved in the LES mode.



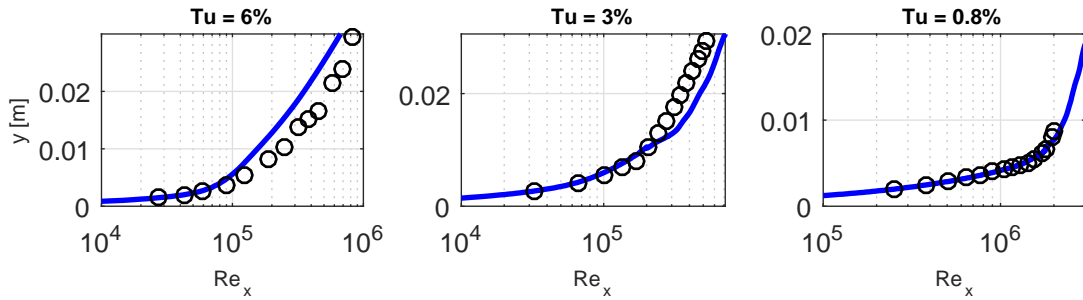
**Figure 6.9:** RANS/LES blending function  $f_d$ . RANS=1, LES=0. White dashed line: 99% boundary layer thickness.

Skin-friction values are depicted in Figure 6.10. The results from the  $Tu = 6\%$  will not be discussed any further as these all fully RANS mode results. For the other two cases, the transition location is predicted accurately, as was expected since the transition location is derived by the RANS model. The maximum value of the skin-friction peak of the  $Tu = 3\%$  case does show notable discrepancies with experimental data. This behaviour was already observed in the verification (Section 6.3), the skin-friction flattens right after the transition point. The results obtained for the lowest turbulent inflow conditions (right graph) returns a turbulent skin-friction which agrees with the turbulent theory from White (2006).



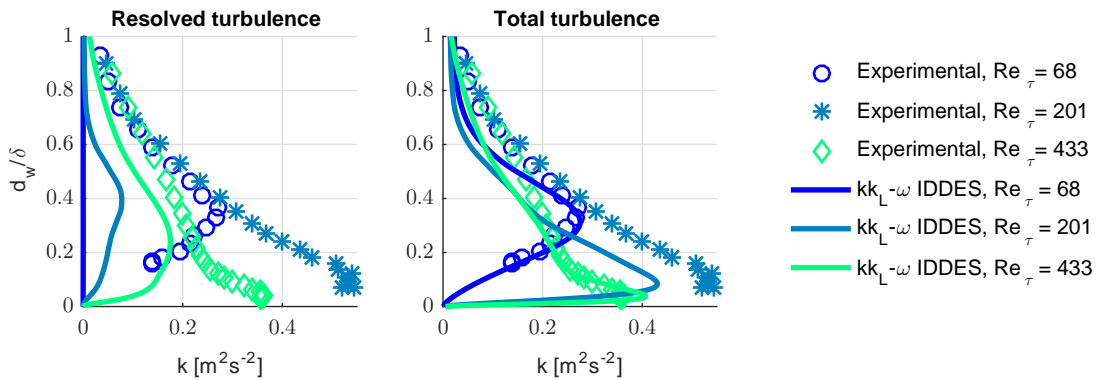
**Figure 6.10:** Wall friction coefficient along flat-plate. Blue line:  $kk_L - \omega$ -IDDES. Circles: experimental. Black line: turbulent theory White (2006). Black dashed line: laminar theory, Blasius.

An underprediction of the boundary layer thickness is observed from Figure 6.11, for  $Tu = 3\%$  and  $Tu = 0.8\%$ . However, the rate at which the boundary layer grows, agrees with experimental data. It is therefore concluded that right after transition the development of turbulence is delayed.



**Figure 6.11:** Boundary layer thickness. Blue line:  $kk_L - \omega$ -IDDES. Circles: experimental.

Another interesting property for a hybrid RANS/LES simulation is the prediction of modelled and resolved laminar and turbulent kinetic energy. Figure 8.4 shows the turbulent kinetic energy at three different stages of the experimental and simulated results ( $Tu = 3\%$ ). The data at  $Re_\tau = 68$  corresponds to the laminar boundary layer,  $Re_\tau = 201$  is located in the transition point and  $Re_\tau = 433$  corresponds to the turbulent boundary layer. The left graphs shows the resolved energy. It can be seen that the laminar resolved energy is zero, this corresponds with the RANS behaviour of the laminar boundary layer. The resolved energy at transition and in the turbulent region are both under predicted. The results improve when the modelled turbulent energy is added, this is shown in the right graph. However, there is still a significant underprediction at the transition point. Also, the profiles reduce to free-stream values much closer to the wall compared to the experimental data. It is as if the boundary layer is too thin, this corresponds with the boundary layer thickness from Figure 6.11.



**Figure 6.12:** Kinetic energy profiles. *Left:* resolved turbulent energy. *Right:* modelled plus resolved turbulent kinetic energy.



---

## Chapter 7

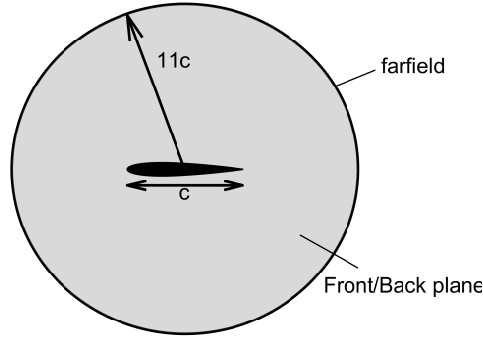
---

### Test Case: NACA 0012 Airfoil

The flow around a NACA 0012 airfoil with a chordwise based Reynolds number of 1 million is simulated under an angle of attack of zero degrees with OpenFOAM 4.1. Simulations are performed with the modified SA IDDES model and the new  $kk_L - \omega$  IDDES model. The results are compared with reference data from Verhoeven (2006), who used the original SA IDDES model on a NACA 0012 with a like-wise numerical mesh and inflow conditions as will be used in this test case. Numerical mesh characteristics and boundary conditions for this case are presented in Section 7.1 and Section 7.2, respectively. The flow results are presented and discussed in Section 7.3.

#### 7.1 Numerical grid

A three dimensional o-mesh is generated with Pointwise Inc. (2014). The trailing edge thickness is 1.4mm, the chord length 1m and the spanwise length is 10% of the chord length. In order to be able to obtain an accurate prediction of the drag, the size of the first of the wall cell height is kept below one ( $y^+ < 1$ ). The wall-normal cell size grows with a factor 1.15 until the maximum expected boundary layer thickness is reached. After that the grow factor is increased to 1.2 such that the number of cells in the farfield is minimized. The farfield boundary is located at minimum of 11 times the chord length. Clustering of the cells is applied towards the leading and trailing edge. A schematic representation of the mesh layout is depicted in Figure 7.1. In the chordwise direction 600 cells have been placed. The spanwise direction contains 49 cells. The trailing edge contains 40 cells. The mesh contains a total of 3.7 million hexahedra cells with a maximum non-orthogonality of 40.



**Figure 7.1:** Size view of the NACA 0012 simulation domain.

The results obtained with the simpleFOAM solver are used as initial conditions for the transient pisoFOAM simulation. The pressure field is computed with a PCG solver and a DIC preconditioner with GaussSeidel smoother. All other field are solved using a smooth solver with a GaussSeidel smoother. One non-orthogonal correction and 5 outer corrections are applied for the PISO algorithm.

## 7.2 Boundary conditions and solver settings

The flowing boundary conditions are applied to the steady state RANS simulation, which is used as an initial solution for the transient DES simulations. A chordwise Reynolds number of 1 million, a chord length of 1 meter, a farfield flow velocity of 70 m/s and a kinematic viscosity  $\nu$  of  $7e-05$  m<sup>2</sup>/s are used. The boundary conditions of the farfield inlet/outlet and the airfoil surface are summarized in Table 7.1. A cyclic boundary condition is applied to the front and back plane for all parameters.

**Table 7.1:** Boundary conditions for the  $kk_L - \omega$  SA simulations.  $Tu = 0.161\%$ .

	Farfield	Airfoil surface
$p$ [ $Nm^{-2}$ ]	0	$\frac{\partial p}{\partial n} = 0$
$U$ [ $ms^{-1}$ ]	(70 0 0)	(0 0 0)
$k_L$ [ $m^2s^{-2}$ ]	0	0
$k_T$ [ $m^2s^{-2}$ ]	0.02045	0
$\omega$ [ $s^{-2}$ ]	0.5386	$\frac{\partial \omega}{\partial n} = 0$
$\tilde{\nu}$	0.00028	0
$\nu_t$	$\frac{\partial \nu_t}{\partial n} = 0$	calculated



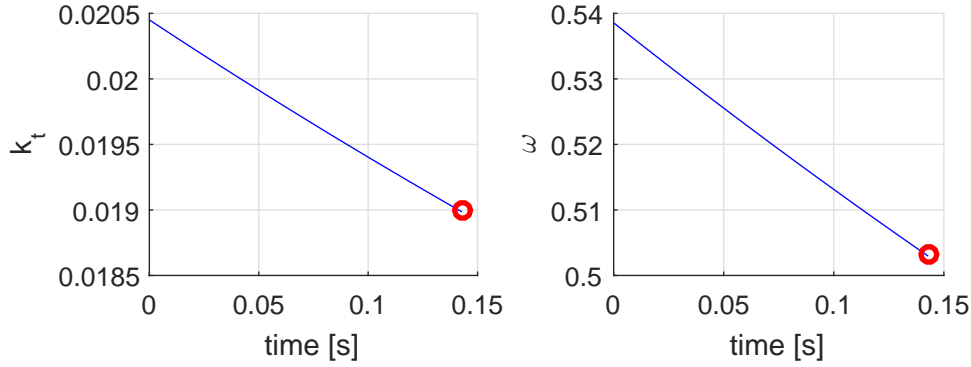
The inlet values for the  $kk_L - \omega$  turbulence model have been corrected for the decay of turbulent kinetic energy. The distance between the inlet and airfoil is 11 times the chord length. Therefore, a correction is necessary to obtain the desired ambient turbulence field near the airfoil. Without production of turbulence, the kinetic turbulent energy and length scale parameter are strongly simplified,

$$\frac{Dk_T}{Dt} = -\omega k_T \quad (7.1)$$

and

$$\frac{D\omega}{Dt} = -C_{\omega 2}\omega^2. \quad (7.2)$$

The decay of  $k_T$  and  $\omega$  are numerically predicted. For the desired ambient conditions  $k_{T,ambient} = 0.0019 \text{ m}^2\text{s}^{-2}$  and  $\omega_{ambient} = 0.5033 \text{ 1/s}$  an upstream decay between inlet and airfoil is predicted and visualized in Figure 7.2.

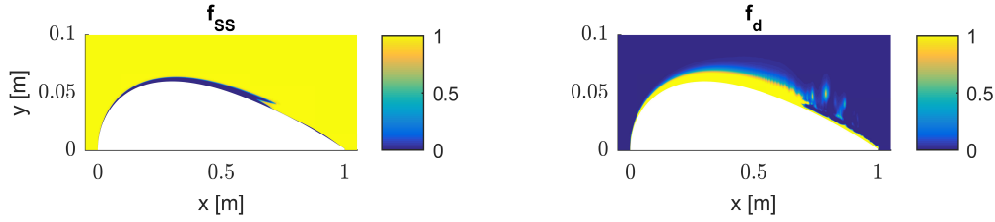


**Figure 7.2:** Predicted decay of turbulent parameters in the farfield. *Blue line:* decay of parameter. *Red circle:* desired ambient condition.

## 7.3 Results

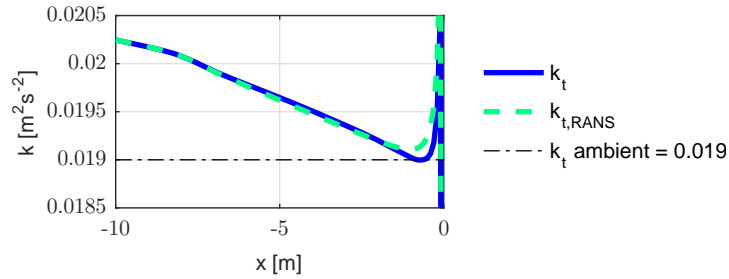
### 7.3.1 Blending functions

Before proceeding to the flow results, the location of the hybrid branches should be verified. Without the knowledge of the RANS/LES regions a proper interpretation of the results is difficult. The main indicator for the laminar and turbulent regions is the shear sheltering function ( $f_{SS}$ ), shown in the left graph of Figure 7.3. As expected, a laminar region is located around the leading edge of the airfoil. The DDES branch of IDDES should return an active RANS region in all areas without resolved turbulence (Equation 2.55). From the left graph of Figure 8.4 it can be seen that the farfield is operating in LES mode. This is a result of the very low free-stream modelled turbulence. Only the WMLES branch and the shear sheltering branch operate in RANS mode. The active LES mode in the free-stream region does not have an effect on  $f_{SS}$  as it is computed by the background RANS model.



**Figure 7.3:** Left: shear sheltering function. Turbulent=1, laminar=0. Right: hybrid blending function. RANS=1, LES=0.

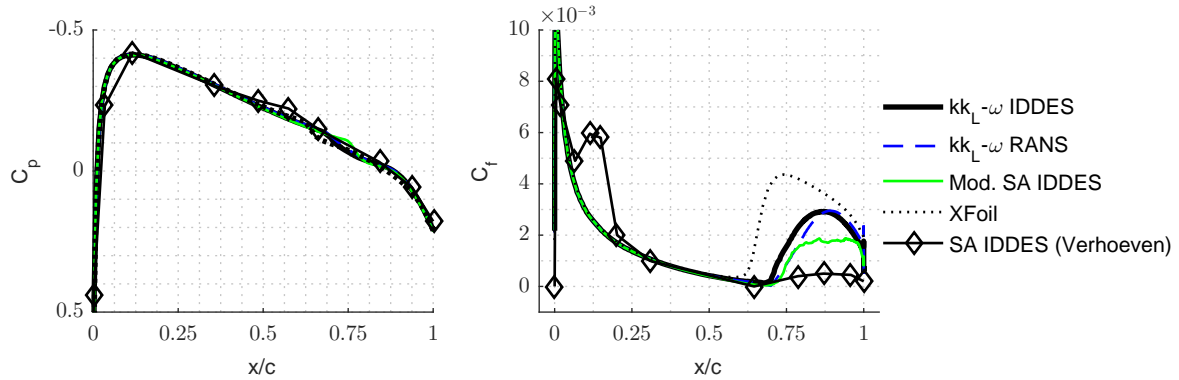
There is however an effect on  $k_t$  as the destruction term is now based on the sub-grid length scale. The decay of the turbulent quantities  $k_t$  and  $k_{t,RANS}$  are depicted in the Figure 7.4. The ambient turbulence level of 0.0019 is obtained just in front of the airfoil. Although the far-field LES region is undesirable, it does not seem to have a large effect on the turbulence decay.



**Figure 7.4:** Free-stream turbulence decay from inlet to airfoil.

### 7.3.2 Pressure and skin-friction

Small discrepancy in the skin-friction is found between the simulation results and XFOIL, left graph in Figure 7.5. XFOIL indicates slight decrease and increase peak in the pressure (laminar separation bubble) distribution around  $x/c = 0.62$ . This behaviour is observed at  $x/c = 0.68$  for the DES and RANS simulation result. The downstream prediction of the laminar separation bubble corresponds with the transition phenomena observed in the skin-friction distribution, right graph of Figure 7.5. It is unclear why the transition is postponed. Possible reason could be that the far-field boundary is located too close the airfoil. But the relative large ambient turbulent length scale could also be of influence. More interesting are the similarities in the pressure and skin-friction distribution between the RANS and DES results. The pressure distribution is almost a perfect match on both upper- and lower-side (not shown) of the airfoil. The skin-friction from the DES simulation indicates that the transition point is more upstream and increase faster compared to the RANS results, this behaviour was also observed in the flat-plate validation cases. Compared to the RANS result the DES simulation predicted the correct transition location within less than 2 percent of the chord length.

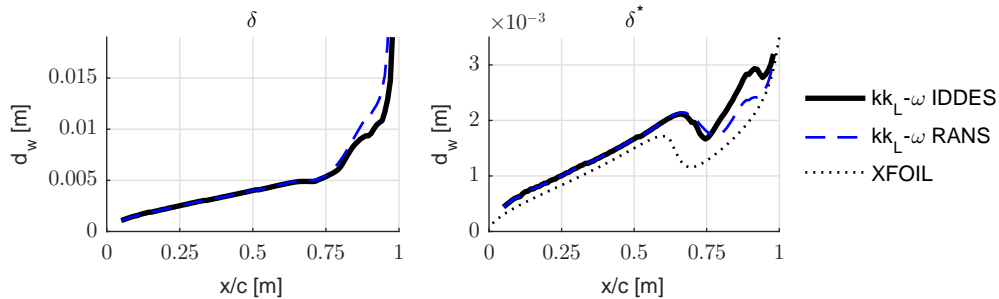


**Figure 7.5:** Mean chordwise wall friction of the NACA 0012. Tripping intensity  $\alpha = 1$ . Pressure and suction side overlap.

The advantage of the  $kk_L - \omega$  model becomes clear when the skin-friction is compared with the results of the SA IDDES model by Verhoeven (2006). Modelled transition takes place near the leading edge ( $x/c = 0.12$ ), followed by a large *grey area* region, transition due to development of turbulence takes place around  $x/c = 0.68$ . The predicted skin-friction in the turbulent boundary is far below the RANS results from  $kk_L - \omega$ . The discrepancies in the skin-friction imply that a different flow state is present above the airfoil. It could therefore be possible that the discrepancies will be reflected in the other boundary layer characteristics.

### 7.3.3 Boundary layer characteristics

The chordwise boundary layer thickness will indicate the development of turbulent structures along the airfoil. Both the RANS and DES results of the boundary layer thickness are depicted in Figure 7.6. An exact match in the thickness is obtained for the laminar boundary layer. The thickness of initial part of the turbulent boundary layer is captured accurately,  $0.7 < x/c < 0.8$ . However, beyond  $x/c > 0.8$  a too low value of the boundary layer thickness is observed.



**Figure 7.6:** Boundary layer profiles of upper side of NACA 0012. *Left:* boundary layer thickness. *Right:* displacement thickness.

A comparison between the results from  $kk_L\omega$  and reference data is possible through the value

of the displacement thickness,

$$\delta^* = \int_0^\infty \left(1 - \frac{u(y)}{U_\infty}\right) dy. \quad (7.3)$$

The displacement thickness near the trailing edge ( $\frac{x}{c} = 0.99$ ) is listed in Table 7.2, together with the results from Verhoeven (2006) and XFOIL. The displacement thickness from Verhoeven (2006) is roughly 3.5 time larger than those obtained by the  $kk_L - \omega$  turbulence model. Implying a significant distortion of the boundary layer characteristics. No chordwise data is provided by Verhoeven (2006) but it is expected that the displacement thickness is larger due to the premature transition at  $x/c = 0.12$  as was observed in Figure 7.5.

**Table 7.2:** Displacement thickness at the trailing edge of NACA 0012.

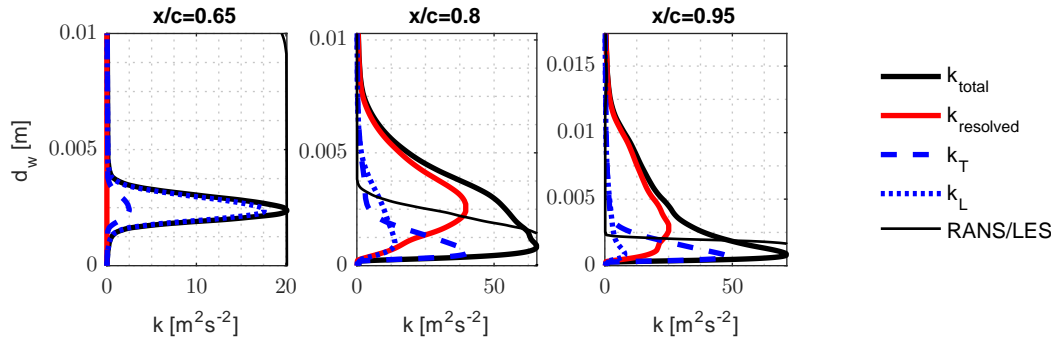
Model	$\delta^*[m]$
XFOIL	0.0030
$kk_L - \omega$ RANS	0.0030
$kk_L - \omega$ IDDES	0.0034
SA IDDES (Verhoeven)	0.011

### 7.3.4 Turbulent kinetic energy profiles

The purpose of a hybrid RANS/LES model is to resolve relevant large-scale turbulent structures while modelling the computationally expensive small-scale turbulence. Three energy profiles at different chordwise location are shown in Figure 7.7. The left graph corresponds to the laminar boundary layer, the middle graph shows the energy profiles right after the transition location and the right graph presents the energy profiles of the turbulent boundary layer near the trailing edge. For each chordwise station four energy profiles are shown. The total kinetic energy consists of the resolved and the modelled energy. The modelled energy itself consists of the laminar and turbulent kinetic energy.

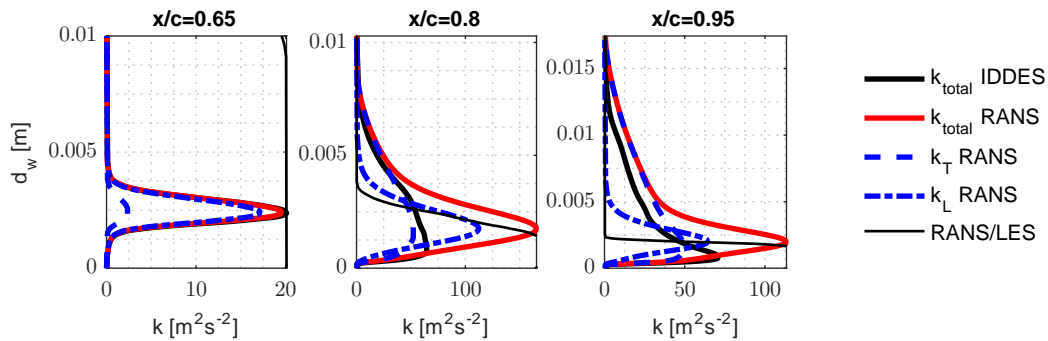
In the laminar boundary layer ( $x/c = 0.65$ ) the flow is solved using RANS and, as expected for a RANS simulation, all turbulence is modelled. Moving downstream, into the turbulent boundary layer ( $x/c = 0.8$ ), the RANS/LES blending takes place inside the boundary layer. The near wall flow is solved using RANS, while the outer region is solved using LES. Due to the activation of the LES mode, turbulence can develop and only the sub-grid scales are modelled. Inside the near wall RANS region the modelled energy is dominant.

The energy profiles near the trailing edge ( $x/c = 0.95$ ) indicate that the turbulent region has grown in the wall-normal direction which corresponds to the development of the boundary layer from Figure 7.6. This figure also shows that near the transition point ( $x/c = 0.8$ ) roughly half of the turbulent containing region is solved using RANS. This implies that only half of the turbulence is present as true unsteady motion. It can be expected that this deficiency of unsteady motion at the transition location will have an influence on the further development of the turbulent boundary layer.



**Figure 7.7:** Decomposition of the modelled and resolved kinetic energy profiles of the NACA 0012 boundary layer.

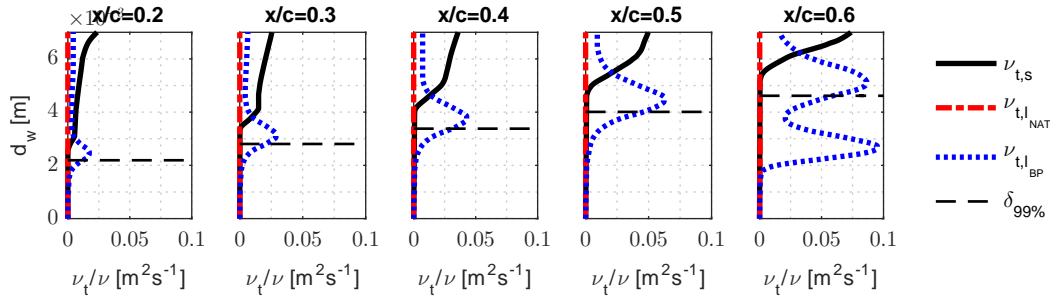
A comparison of the kinetic energy profiles from the  $kk_L - \omega$  IDDES simulation is made with that of the steady RANS simulation. Figure 7.8 presents the total kinetic energy from the IDDES simulation and from the RANS simulation the total-, laminar- and turbulent kinetic energy profiles. The left graph shows the results at the chordwise location  $x/c = 0.65$ , here the flow of the IDDES simulation is solved with RANS and does therefore correspond accurately with the results from the RANS simulation. More interesting are the results right after the transition process, these are shown in the middle graph. The IDDES simulation underpredict the kinetic energy peak at ( $d_w = 0.0025\text{m}$ ), by more than a factor of 2. A decomposition of the RANS results into the turbulent and laminar parts show that the IDDES results are in good resemblance with only the turbulent energy from the RANS simulation. The same observation is made for the chordwise location at  $x/c = 0.95$ . This may imply that the introduced velocity fluctuations do not realistically represent the motion present in the pretransitional boundary layer.



**Figure 7.8:** Kinetic energy profiles of the NACA 0012 boundary layer obtained from the IDDES and RANS simulation. RANS results have been decomposed in laminar and turbulent kinetic energy profiles.

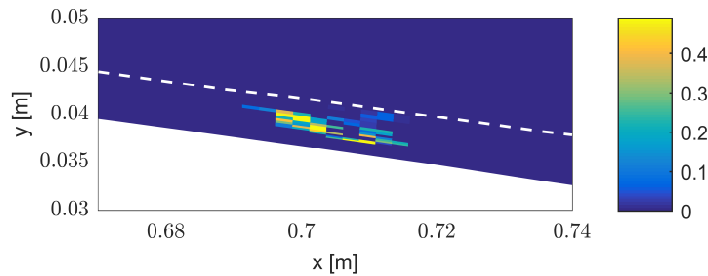
Near the trailing edge ( $x/c = 0.95$ ) it is also observed that the kinetic energy profile of IDDES is too thin, this again corresponds with the comparison of the boundary layer thickness which was made in Figure 7.6. Based on these observations it is concluded that the development of turbulence is delayed because the resolved kinetic energy near the transition location is too

low. A possible reason could be that the introduced fluctuations are only effective in tripping the flow, but do not effectively introduce kinetic energy. In the case of the NACA 0012 airfoil, the main contributor to the large-scale eddy viscosity ( $\nu_{t,l}$ ) of the pretransitional boundary layer, is the bypass mechanism. This is concluded from Figure 7.9, which depicts the different eddy viscosity terms. If the introduced velocity fluctuations are indeed responsible for the underprediction of the resolved turbulence, it is then because they do not effectively represent the bypass mechanism.



**Figure 7.9:** Instantaneous profiles of the small-scale eddy viscosity  $\nu_{t,s}$ , natural  $\nu_{t,l,NAT}$  and bypass  $\nu_{t,l,BP}$  eddy viscosity.

In Figure 7.10 the introduced velocity fluctuations ( $u'$ ) are visualized. From the shape of the individual cells, it can be seen that the velocity fluctuations are relative flat and elongated, a shape that may not be very effective to represent bypass energy. After the introduction of  $u'$  the boundary layer thickness does increase, as can be seen from the white dashed line. Roughly 80% of the boundary layer thickness is provided with  $u'$ , this is possible because the small local cell size allows for only in a very thin near-wall RANS region ( $f_B$ ).

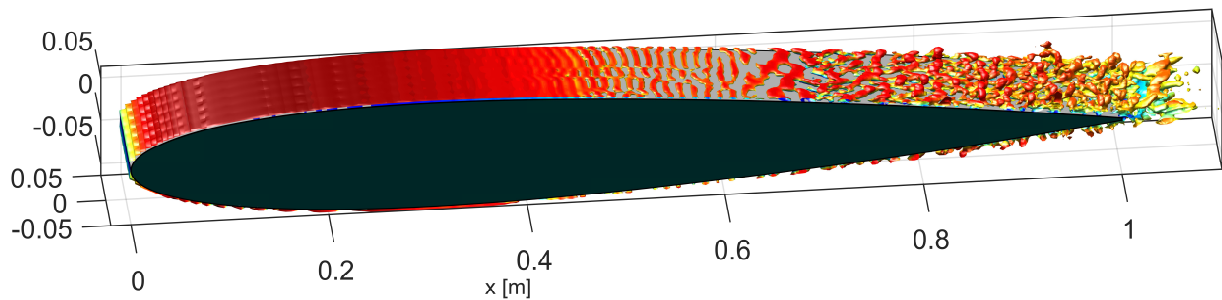


**Figure 7.10:** Magnitude of the introduced velocity fluctuations by the tripping mechanism. Upper side of NACA 0012. White dashed line: 99% boundary layer thickness.

A possible reason for an ineffective introduction of velocity fluctuations could be found in the small time step and small cell size. Therefore, introduced velocity fluctuations represent turbulence with a small turbulent length scale. The energy dissipation will be high and the introduced fluctuations might decay too fast.

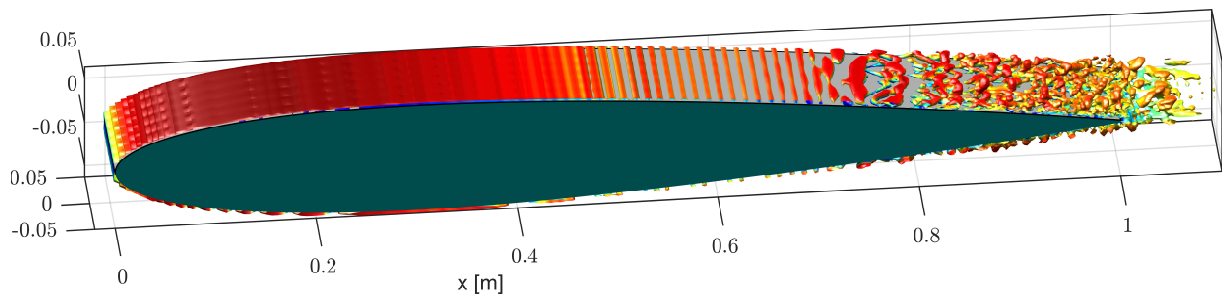
### 7.3.5 Vorticity contours

The Q-iso contours from Figure 7.11 indicate that two dimensional vortical structures start to form in the laminar RANS region,  $x < 0.7\text{m}$ . Although the laminar region is solved using RANS, it is not a true 'averaged' solution since the simulation is of a transient type. This type of simulation corresponds to an Unsteady RANS (URANS) simulations. Due to the high viscosity in the RANS region, the two-dimensional TS waves will develop very slowly, possibly resulting in a different flow field after a long period of simulation time.



**Figure 7.11:** Vorticity contours ( $Q\text{-iso}=2\text{e-}5$ ) around the NACA 0012 by  $kk_L - \omega$  IDDES. Coloured with the velocity magnitude.

When the vortical structures, obtained by the modified SA IDDES model (Figure 7.12), are compared with the results from  $kk_L - \omega$  IDDES a difference in the pretransitional boundary layer ( $x < 0.7\text{m}$ ) is observed. In the modified SA IDDES model only spanwise roll ups are present, representing TS waves. Opposite to the result from the modified SA IDDES model, the results from  $kk_L - \omega$  IDDES also show secondary instabilities in the two dimensional vortical structures in the pretransitional boundary layer. This difference in structures can be assigned to the addition of the bypass eddy viscosity of the laminar kinetic energy. This additional eddy viscosity reduces the critical Reynolds number. Possible instabilities in the flow will therefore be amplified.



**Figure 7.12:** Vorticity contours ( $Q\text{-iso}=2\text{e-}5$ ) around the NACA 0012 with the modified SA IDDES. Coloured with the velocity magnitude.

### 7.3.6 Surface noise sources

Since the  $kk_L-\omega$  IDDES model does predict with reasonable accuracy the transition location and boundary layer development, it could be used for noise predictions. One type of noise sources are related to the surface pressure fluctuations (not including the typical trailing edge noise).

The surface pressure fluctuations are generated by the vortical structures present in the flow. The left graph of Figure 7.13 shows the noise sources predicted by the modified SA IDDES model. Near the transition point ( $x/c = 0.75$ m) a strong peak is observed. This peak might therefore be generated by the presents of laminar separation bubbles. In the flow downstream of the transition point less pressure fluctuations are present. Opposite to the laminar boundary layer, where no pressure fluctuations are present, the pressure fluctuations in the turbulent boundary layer do not return to zero.

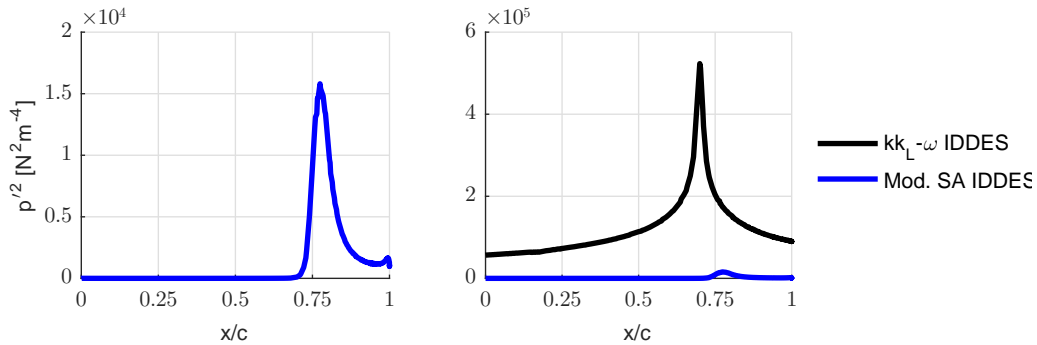


Figure 7.13:

Results from the  $kk_L-\omega$  IDDES model predict pressure fluctuation which are roughly 40 times stronger than those generated by the modified SA IDDES model. However, these pressure fluctuations are not only generated by the vortical structures present in the flow, but are a result of the added non-divergence free velocity fluctuations at the transition point. This behaviour is a result of the OpenFOAM PISO solver. The solver removes the non-divergent free velocity fluctuations through the pressure-velocity corrector. It is this corrector which generates non-physical pressure fluctuations throughout the whole numerical domain. It is therefore concluded that the current model is not suitable for aeroacoustic simulations until a method is used which can introduce divergence free fluctuations.



---

## Chapter 8

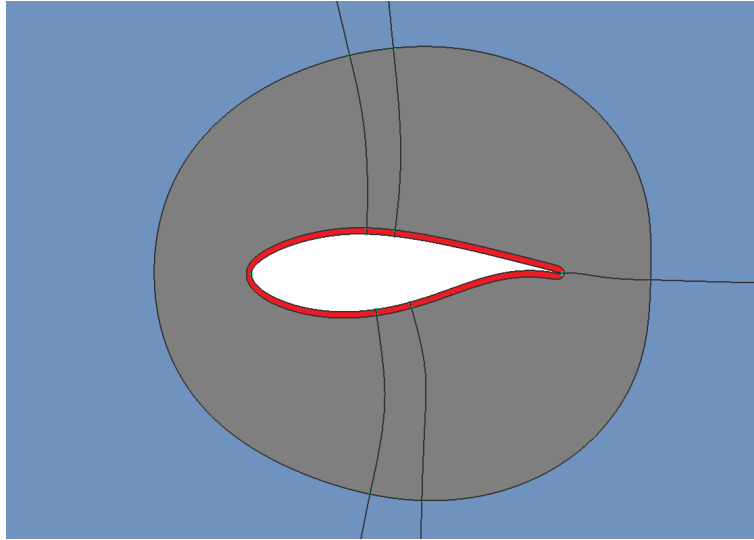
---

# Test Case: DU91W250 Airfoil

A more challenging simulation case is a high Reynolds number (3.2 million) simulation over a cambered wind turbine airfoil (DU91W250). The airfoil is was designed as an inboard wind turbine section and has a large region of laminar flow over the leading edge. Two simulations are performed, one at operating-lift ( $\alpha = 6^\circ$ ) conditions and the other at zero-lift ( $\alpha = -2.85^\circ$ ) conditions. A procedure for the construction of an efficient numerical grid is presented in Section 8.1. Boundary conditions, simulation models and the computational resources can be found in Section 8.2. The simulation results are presented in Section 8.3.

### 8.1 Numerical grid

Only one mesh is generated for both the operating- and zero-lift case. The computational time for the simulation, will be high due to the high Reynolds number and because of the required transient simulation. It is therefore important to develop a mesh with sufficient resolution and a minimum number of cells. A method is chosen that decreases the mesh density in areas away from the boundary layer, where the velocity gradients are low. This depreciation in radial direction is visualized in Figure 8.1. The radial section which is adjacent to the airfoil, has the highest spanwise number of cells, 128. This radial section covers the boundary layer. The outer segment only contains 1 cell in the spanwise direction. Besides a very low mesh density and a low cell aspect ratio is used in the farfield. An intermediate radial section connects the dense and coarse radial segments with each other, without this segment the non-orthogonality would become too high.



**Figure 8.1:** Spanwise cell distribution for 10% span. Blue = 1 cell. Gray = 32 cells. Red = 128 cells in spanwise direction.

The mesh is made in Pointwise [Inc. \(2014\)](#) version V17.2R1 and finalized using multiple OpenFOAM utilities. Final mesh characteristics are summarized in Table 8.1. The listed values of the number of cells per radial segment indicate that 90% of the cells are located in the inner radial segment where the turbulent boundary layer should be resolved.

**Table 8.1:** Mesh summary for DU91W250 mesh after refinement,  $Re$  of 3.2 Million, span = 10% of chord,  $\alpha = -2.85^\circ$  and  $6^\circ$ .

Parameter	value
Total No. of cells	20.5 million
No. of cells in b.l. segment	18.6 million
No. of cells in intermediate segment	1.1 million
No. of cells in farfield segment	0.8 million
$\Delta y^+$	1
$\Delta x^+$	165
$\Delta z^+$	97
Far-field boundary	$85 \times chord$
No. of cells x-dir.	1200
No. of cells z-dir.	128
No. of cells normal-dir.	119
No. of cells on trailing edge	12
First cell length	0.4 mm
Last cell length	0.2 mm

An extensive description of the mesh generation procedure can be found in Appendix A.

## 8.2 Simulation details

### 8.2.1 Boundary conditions

Applied boundary conditions can be found in Table 8.2. These boundary conditions have been applied to both steady state and transient simulations. The values presented are used to initialize the steady state simulation. With a chordwise Reynolds number of 3.2 million, a chord length of 0.9 meter and a farfield flow velocity of 58 m/s, the kinematic viscosity  $\nu$  is set to  $1.63125e-05 \text{ m}^2/\text{s}$ . The required ambient turbulent intensity ( $Tu$ ) and turbulent energy ( $k_T$ ) are, 0.07 and  $0.00247 \text{ m}^2/\text{s}^2$ , respectively. This corresponds with an ambient  $n_{crit}$  value of 9.

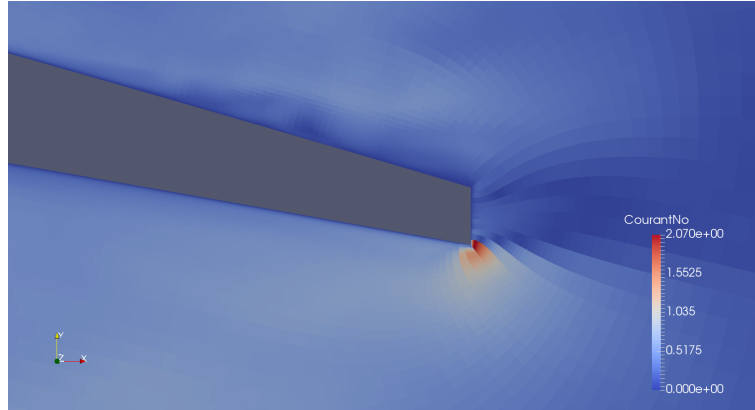
**Table 8.2:** Boundary conditions applied to the RANS and IDDES case.

	Farfield	Airfoil surface
$p \text{ [Nm}^{-2}\text{]}$	0	$\frac{\partial p}{\partial n}=0$
$U \text{ [ms}^{-1}\text{]}$	$(57.68 \ 6.06 \ 0)_{\alpha=6^\circ} \parallel (57.93 \ -2.88 \ 0)_{\alpha=-2.85^\circ}$	$(0 \ 0 \ 0)$
$k_L \text{ [m}^2\text{s}^{-2}\text{]}$	0	0
$k_T \text{ [m}^2\text{s}^{-2}\text{]}$	0.00324	0
$\omega \text{ [s}^{-2}\text{]}$	0.64	$\frac{\partial \omega}{\partial n}=0$

The value of the turbulent parameters at the inlet are corrected for the decay of turbulence. An explanation of the methodology was already given in Section 7.2.

### 8.2.2 Solver settings

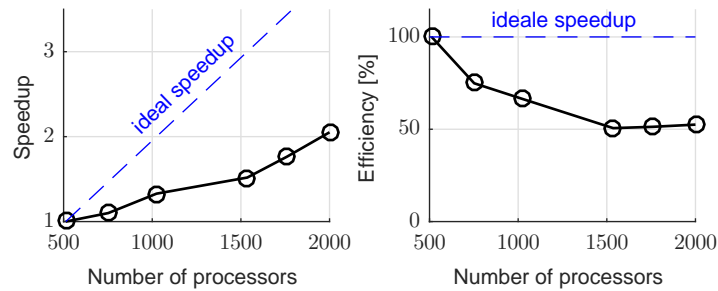
The steady state solver (SIMPLE) is used to generate an initial converged flow field for the transient simulation. The transient simulation is performed using the PISO solver. This is currently the only solver which works with the IDDES model. The PISO solver uses an implicit scheme which allows time steps that exceed a Courant number of 1. This advantage is used in both the operating and zero lift conditions. At the same time, a LES simulation prefers a Courant number well below 1 in order to maintain its accuracy. Figure 8.2 shows the Courant number near the trailing edge. It can be seen that the maximum Courant number is reached in a local area around the lower trailing edge corner. This is a result of the boundary layer cells of the trailing edge which are extruded into the airfoils pressure side boundary layer. The average Courant number in the turbulent boundary layer is estimated to be a factor of 2.5 lower. The time step is chosen such that the maximum Courant number remains below 2, while the mean Courant number in the turbulent boundary layer stays below 1.



**Figure 8.2:** Courant number near the trailing edge of the DU91W250 airfoil for a time step of  $1.8e-6$  seconds.  $AoA = 6^\circ$

### 8.2.3 Computational resources

Due to the large size of the simulation cases, the majority of simulations are performed on SuperMUC. Thin Nodes (Phase 1), class name general (2.7 GHz). Before the large case simulation are started a small investigation of the scaling of processors is performed. For this a converged solution of the steady state DU91W250 simulation case is used. The left graph in Figure 8.3 indicates a far from ideal speedup. Increasing the number of processors also results in a decreased efficiency, Figure 8.3. However, when using more than 1500 cores, the efficiency remains almost constant.



**Figure 8.3:** Scaling on SuperMUC with OpenFOAM v4.1

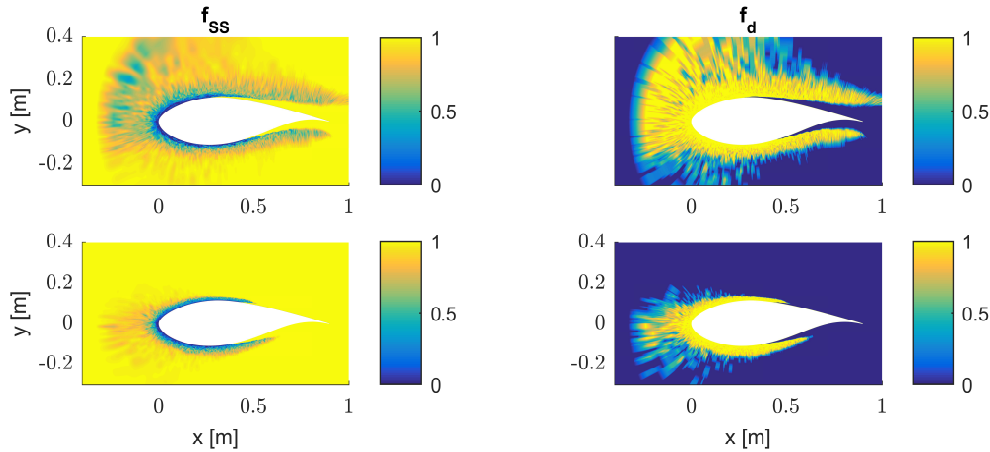
The DU91W250 airfoil with the modified SA IDDES is simulated with OpenFOAM v3.0.1. With a maximum of 112 cores (more cores resulted in errors) the computational time for the transient simulation takes roughly 7 second per time step and in total 18 days. All other simulation are performed with OpenFOAM v4.1, using a minimum of 528 and a maximum of 2000 cores. The DU91W250 simulation with  $kk_L - \omega$  IDDES is performed with 2000 cores. However, due to the addition of the velocity fluctuations one time step still requires 5 seconds. Without the addition the time step reduced to roughly 0.8 seconds. This poses a significant shortcoming of the tripping mechanism.

## 8.3 Results

The results obtained with the new  $kk_L - \omega$  IDDES model are presented and compared with the  $kk_L - \omega$  RANS and the modified SA IDDES (Section 3.1.2) simulation. Predicted noise level by the modified SA IDDES model can be found in D.

### 8.3.1 Blending functions

The shear sheltering function ( $f_{SS}$ ), shown in the left graph of Figure 8.4, predicts a laminar region around the leading edge. The shear sheltering function is also responsible for the leading edge RANS region, this is indeed the case, as can be seen from the RANS/LES blending function  $f_d$  (right graph of Figure 8.4). Just as the NACA 0012 case, the far-field region is solved using LES instead of RANS. The decay of the turbulent quantities  $k_t$  and  $k_{t,RANS}$  are depicted in the Figure 8.5. The ambient turbulence level of 0.00247 is obtained just in front of the airfoil. Although the far-field LES region is undesirable, it does not seem to have a large effect on the turbulence decay.



**Figure 8.4:** Blending functions *Left column:* shear sheltering function. Turbulent=1, laminar=0. *Right column:* RANS/LES blending function. RANS=1, LES=0. *Top row:*  $AoA = 6^\circ$  and tripping intensity  $\alpha = 1$ . *Bottom row:*  $AoA = -2.85^\circ$  and tripping intensity  $\alpha = 1$ .

Furthermore, it is observed that irregularities in the shear sheltering and RANS/LES blending are present in front of the leading edge. These irregularities start at the edge of the radial mesh segments (Figure 8.1). It is expected that the increase in the number of spanwise cells cause small fluctuations in the velocity field. The turbulence fields respond to these fluctuations and irregularities form in the shear sheltering function. The process is strengthened by the addition of small velocity fluctuations by the tripping mechanism. The reduced tripping intensity  $\alpha = 1$  for  $AoA = -2.85^\circ$  does suffer less from this problem.

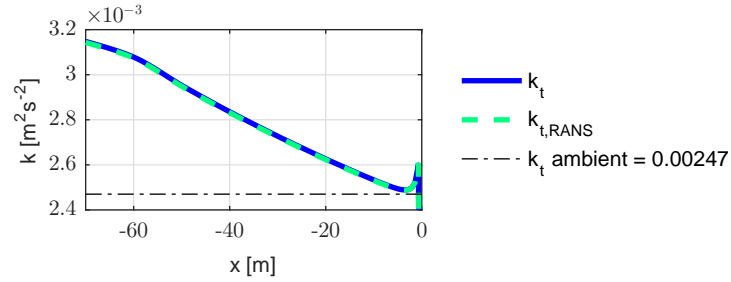


Figure 8.5: Free-stream turbulence decay from inlet to airfoil,  $AoA = 6^\circ$

### 8.3.2 Pressure and skin-friction

The pressure distribution of all three simulations agree with the predicted values from XFOIL (Figure 8.6). The largest discrepancies are found near the transition location. A close up of the pressure distribution can be found in Figure 8.7.

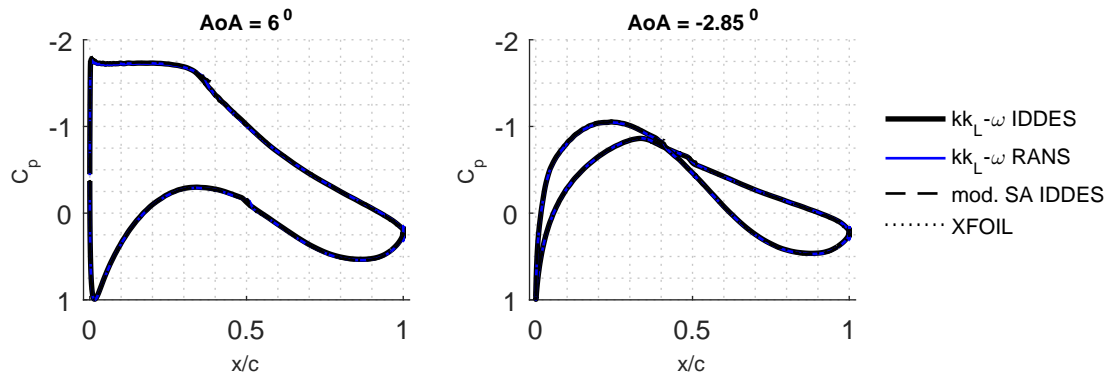
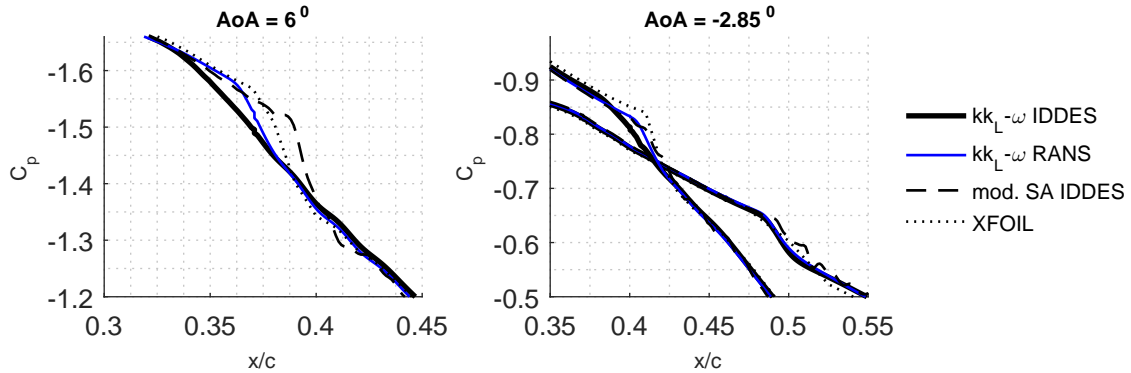


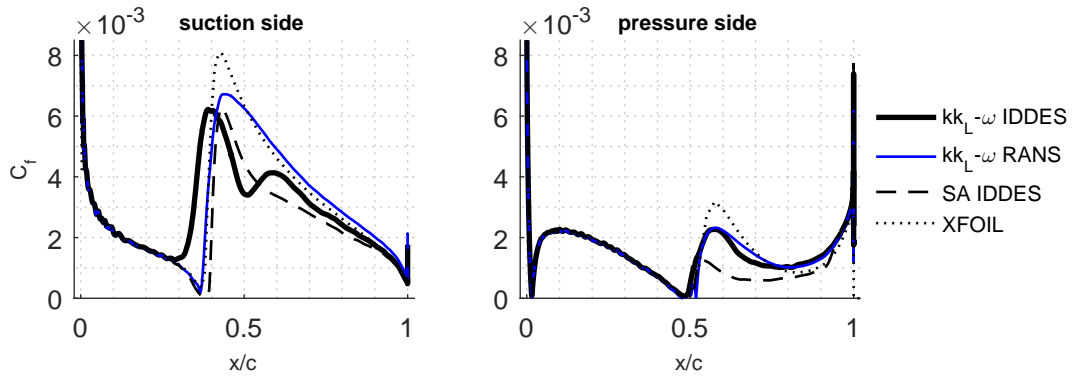
Figure 8.6: Mean pressure distribution along surface chord. *Left:* Tripping intensity  $\alpha = 2$ . *Right:* Tripping intensity  $\alpha = 1$ .

A small bump is observed in the pressure distribution, near the expected transition location. This bump is expected to represent a laminar separation bubble. For all three simulations, the modified SA IDDES model predicts the bump too far downstream, compared with XFOIL, while the  $kk_L - \omega$  IDDES model predicts a too smooth pressure distribution.



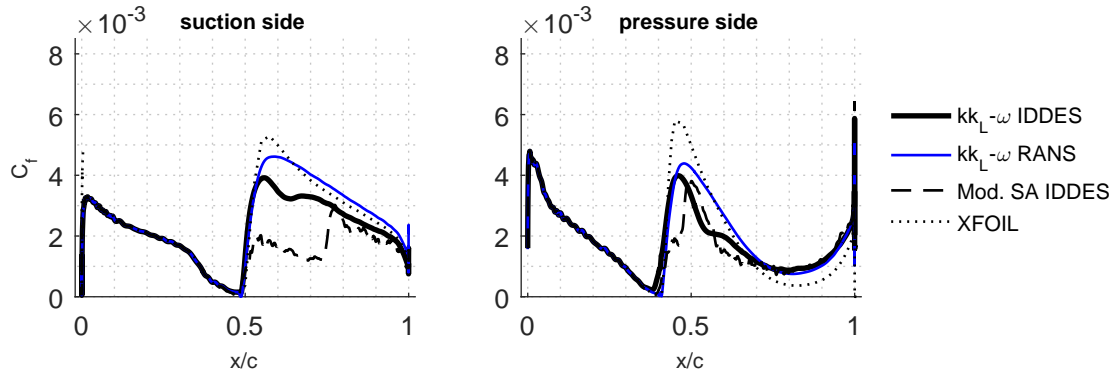
**Figure 8.7:** Mean pressure distribution near the transition location. *Left:*  $\alpha = 2$ . *Right:*  $\alpha = 1$ .

From the wall-friction coefficient on the suction side (left graph of Figure 8.8), of the operating-lift case ( $\alpha = 6^\circ$ ), it is observed that the transition location is slightly shifted upstream and that addition of the artificially generated turbulence results in a dip in the skin-friction between  $0.4 < x/c < 0.5$ . Note that the fluctuation intensity is already scaled by a factor of 2. Due to the premature development of turbulence, a laminar separation bubble can not form, explaining the absence of a laminar separation bubble of Figure 8.7. The skin-friction on the pressure side corresponds quite accurate with the RANS result. Besides, a much more accurate skin-friction is obtained compared with the SA IDDES model.



**Figure 8.8:** Span-averaged skin-friction coefficient on suction and pressure sides of DU91W250,  $AoA = 6^\circ$ . Tripping intensity  $\alpha = 2$ .

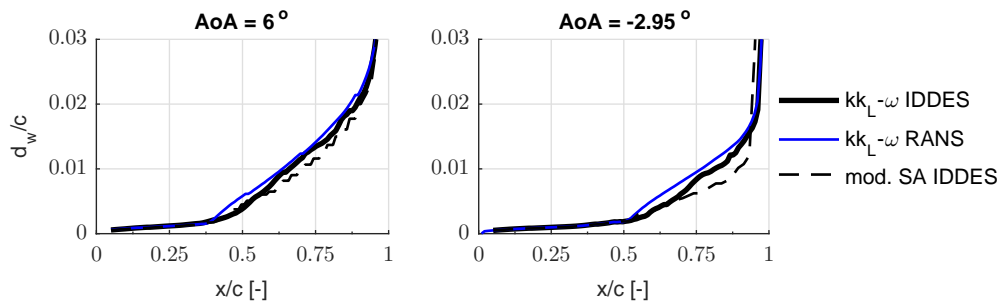
For the zero-lift case ( $AoA = -2.85^\circ$ ) the modified SA IDDES predicts a very weak increase in the skin-friction at the transition location. Further downstream, a second increase in skin-friction is observed. This *grey area* is not present in the skin-friction predicted by  $kk_L-\omega$  IDDES since development of turbulence is excited by the introduced velocity fluctuations.



**Figure 8.9:** Span-averaged skin-friction coefficient on suction and pressure sides of DU91W250,  $AoA = -2.85^\circ$ . Tripping intensity  $\alpha = 1$ .

### 8.3.3 Boundary layer thickness

If the development of turbulent structures would indeed be delayed, it should be reflected in the thickness of the boundary layer. In Figure 8.10, the boundary layer thickness of the airfoil's upper side is shown for the operating-lift case ( $AoA = 6^\circ$ ) and zero-lift case ( $AoA = -2.95^\circ$ ). The lower (concave) side is not depicted since no clear boundary layer edge could be determined (using Equation 3.1) from the velocity field. For both angles of attack, the modified SA IDDES and  $kk_L - \omega$  IDDES model predict a delayed growth of the turbulent boundary layer. This is remarkable since the skin-friction does predict a correct transition location. This inconsistent result (which was also observed from the validation case, Chapter 6) could mean that the introduced (stochastic) velocity fluctuations are unable to quickly develop into true turbulent structures. Although, the  $kk_L - \omega$  IDDES model underpredicts the boundary layer thickness, it does predict a more accurate boundary layer thickness than the modified SA IDDES model.

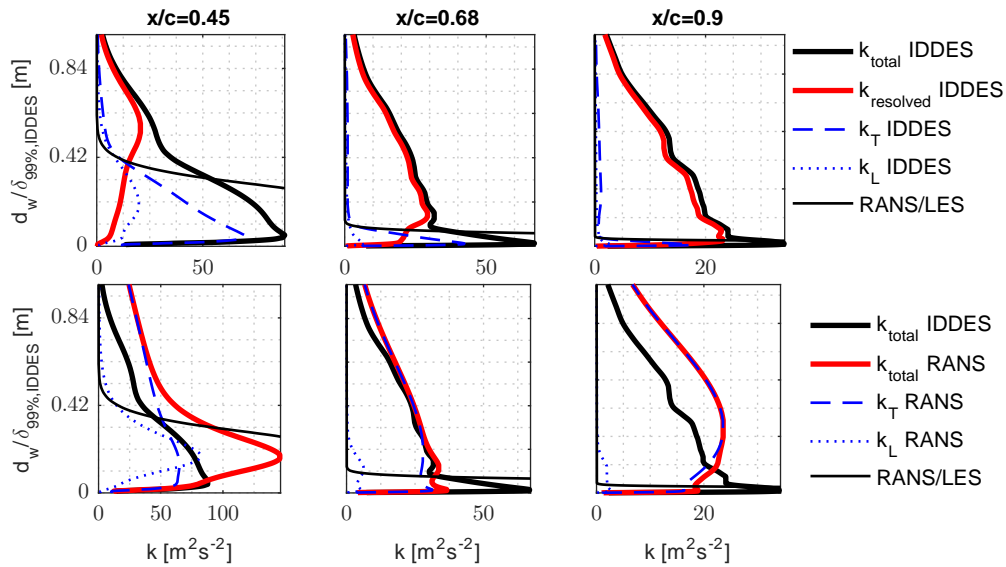


**Figure 8.10:** Thickness of the boundary layer on the upper side. *Left:*  $AoA = 6^\circ$  and tripping intensity  $\alpha = 2$ . *Right:*  $AoA = -2.85^\circ$  and tripping intensity  $\alpha = 1$ .

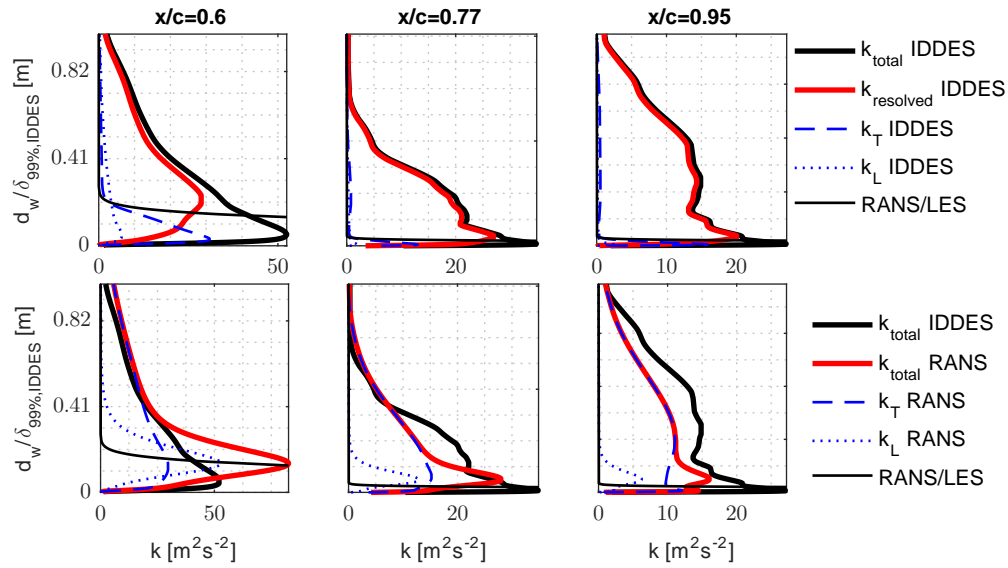


### 8.3.4 Turbulent kinetic energy profiles

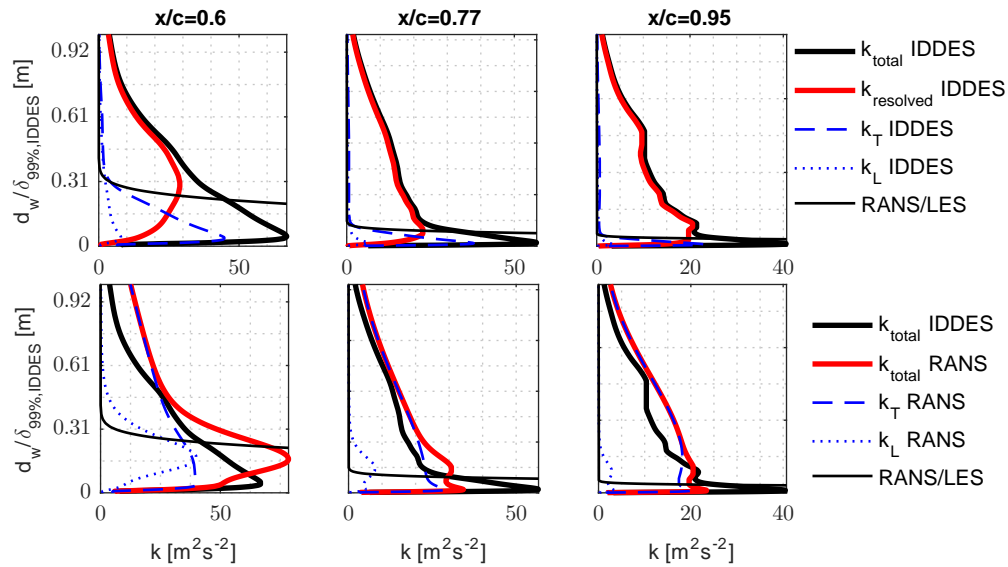
Kinetic energy profiles of the modelled and resolved vortical structures are depicted in the top row of Figures 8.11, 8.12, 8.13 and 8.14, for the upper and lower side of case  $AoA = 6^\circ$  and  $AoA = -2.85^\circ$ , respectively. The total kinetic energy consists of the modelled (sub-grid scale) and the resolved part. The modelled energy consists of the laminar kinetic energy and the turbulent kinetic energy. For all cases it is observed that the modelled energy is dominant in the near-wall RANS region, while the resolved turbulence is most present in the LES region. From these graphs it is concluded that downstream of the transition point, relative more turbulent structures develop. The simulation that solves most of the boundary layer using RANS, also predicts the strongest drop in the skin-friction of the turbulent boundary layer. Also, the maximum kinetic energy is located near the transition point, but at this location most of the energy is modelled, this could therefore result in an deficient of the unsteady motion in the downstream boundary layer. This is especially the case at the wall-normal height  $d_w/\delta_{99\%} \approx 0.2$ , because the RANS model predicts the strongest energy here. A detailed examination of the RANS components  $k_T$  and  $k_L$  shows that  $k_L$  is the main contributor to the near-wall energy peak ( $d_w/\delta_{99\%} \approx 0.2$ ). It could therefore be assumed that the IDDES simulation does not accurately predict the (large-scale) laminar kinetic energy. This could be a valid assumption since the introduced velocity fluctuations are introduced with a relative heigh frequency and a small size (size of the local cell). Therefore more resemblance with small-scale turbulence, than with large-scale turbulence, can be expected. Half way the turbulent boundary layer (middle graph) the energy profile from IDDES does agree with the RANS results, except for the lower side for  $AoA = -2.85^\circ$ . Results near the trailing edge show large discrepancies. Another possible reason for the discrepancies near the transition location and the trailing edge, is that the assumption of an equilibrium flow state (Section 5.1) is invalid.



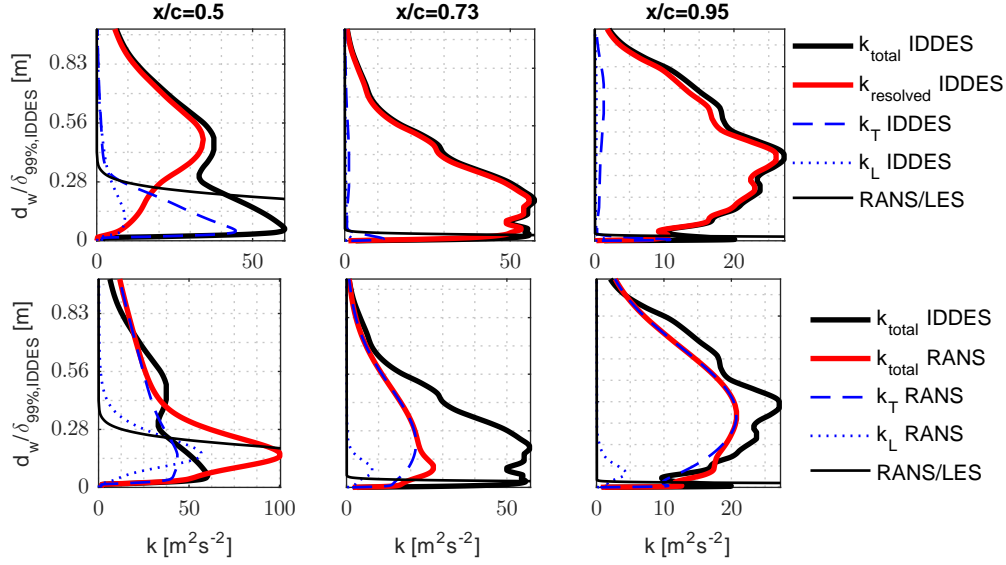
**Figure 8.11:** Kinetic energy profiles on suction side of DU91,  $AoA = 6^\circ$ . *Top:* decomposition of the modelled and resolved energy. *Bottom:* IDDES and RANS comparison. RANS results have been decomposed in their laminar and turbulent parts.



**Figure 8.12:** Kinetic energy profiles on pressure side of DU91,  $AoA = 6^\circ$ . *Top*: decomposition of the modelled and resolved energy. *Bottom*: IDDES and RANS comparison. RANS results have been decomposed in their laminar and turbulent parts.

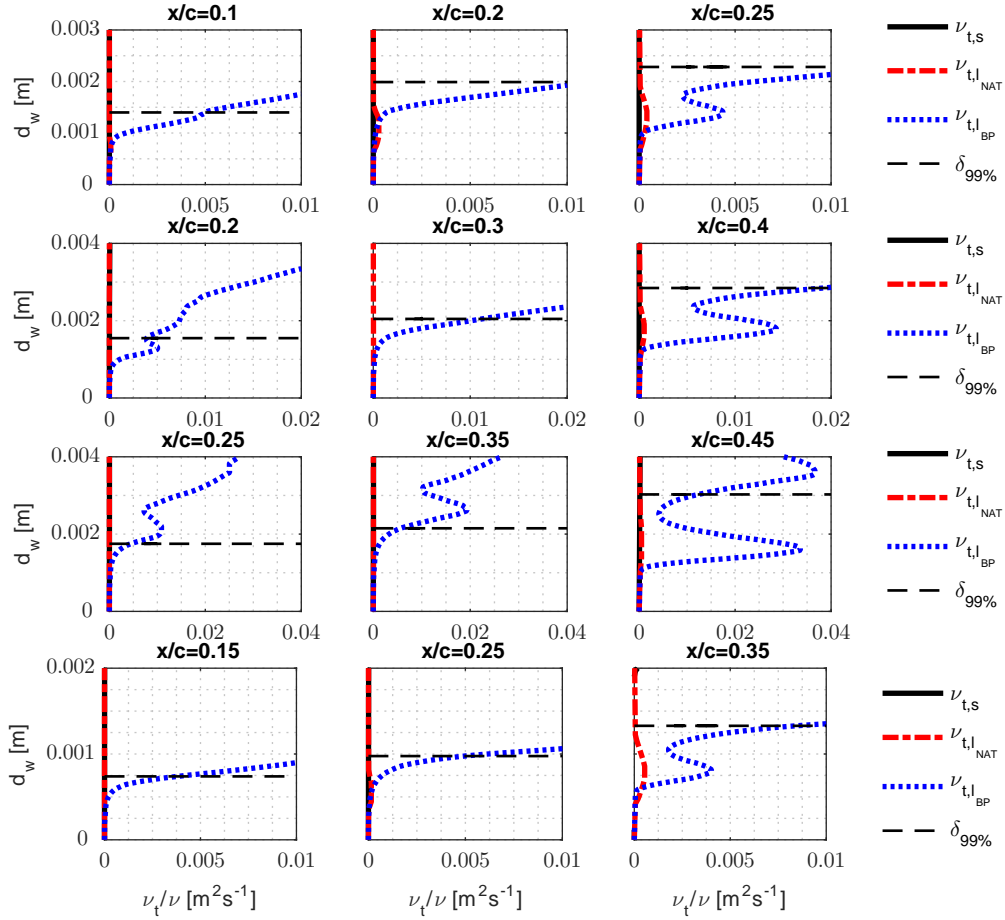


**Figure 8.13:** Kinetic energy profiles on upper side of DU91,  $AoA = -2.85^\circ$ . *Top*: decomposition of the modelled and resolved energy. *Bottom*: IDDES and RANS comparison. RANS results have been decomposed in their laminar and turbulent parts.



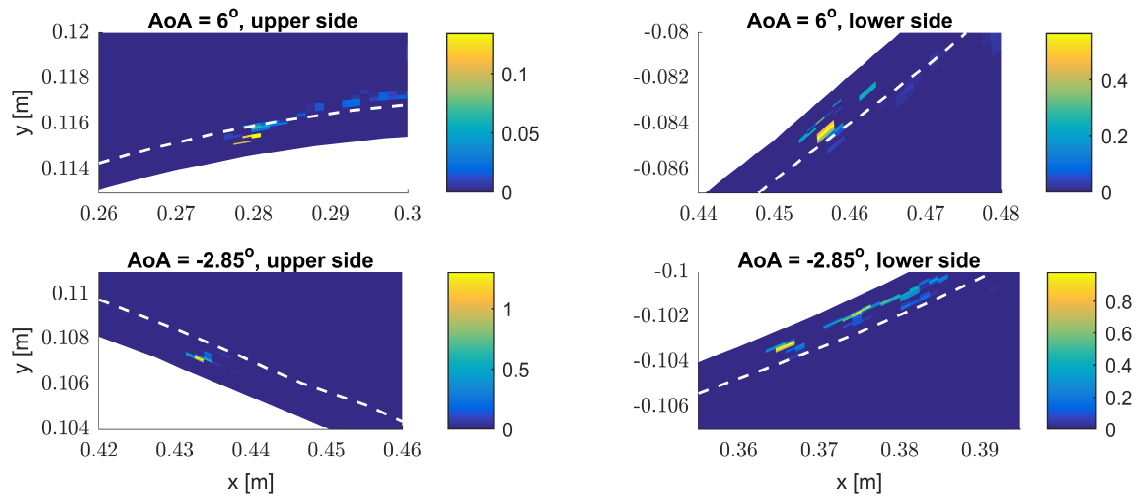
**Figure 8.14:** Kinetic energy profiles on lower side of DU91W250,  $AoA = -2.85^\circ$ . *Top:* decomposition of the modelled and resolved energy. *Bottom:* IDDES and RANS comparison. RANS results have been decomposed in their laminar and turbulent parts.

Just as for the NACA 0012 case, the most dominant eddy viscosity near the laminar boundary layer is the large-scale  $\nu_{t,lBP}$ , see Figure 8.15. If the delayed development of turbulence is a result of an unrealistic introduction of the velocity fluctuations, it would then be because the tripping mechanism does not represent the turbulent structures of  $\nu_{t,lBP}$ . This figure also displays the eddy viscosity at different chordwise locations of the laminar boundary layer. The small-scale eddy viscosity is almost complete absent, opposite to what was seen for the NACA 0012 case. The natural eddy viscosity  $\nu_{t,lNAT}$  increase inside the laminar boundary layer at downstream stations. The large-scale bypass eddy viscosity is also present outside the laminar boundary layer and only increases strongly close to the transition location.



**Figure 8.15:** Instantaneous profiles of the small-scale  $\nu_{t,s}$ , natural  $\nu_{t,l,NAT}$  and bypass  $\nu_{t,l,BP}$  eddy viscosity near the laminar boundary.

The introduced velocity fluctuations from the tripping mechanism are shown in Figure 8.16. It is observed that the fluctuations are only present in the outer 60% of the boundary layer. Remember that for the NACA 0012 case the fluctuations were present in 80% of the boundary layer. This difference is a result of the relative large cells size of the DU91W250 airfoil, which has a large near-wall RANS region. Differences are also observed in the scatter of the velocity fluctuations between the different angles of attack and between the lower and upper size of the airfoil. This scatter of velocity fluctuations was not intended. Besides, little consistency can be observed, therefore the behaviour of the model to different flow simulations remains uncertain.



**Figure 8.16:** Magnitude of the introduced velocity fluctuations by the tripping mechanism. *Top:* tripping intensity  $\alpha = 2$ . *Bottom:* tripping intensity  $\alpha = 2$ . White dashed line: 99% boundary layer thickness.



# Conclusion and Recommendations

Current hybrid RANS/LES models fail to accurately predict transition from a laminar to turbulent boundary layer. The goal of this thesis was to explore the possibilities of developing a transitional hybrid RANS/LES model.

## 9.1 Conclusion

A study into current hybrid RANS/LES models shows that the Spalart-Allmaras IDDES model predicts premature transition and development of turbulent structures is delayed. De-activation of eddy viscosity in the laminar boundary layer results in a delayed transition location but turbulent structures develop much faster. After including a quasi two-dimensional tripping edge, the transition location is predicted correctly, but turbulent structures still do not develop fast enough to reduce a dip in the skin-friction. The  $kk_L - \omega$  SST-DES model does predict a correct transition location but is unable to develop turbulent.

Based on these results, the existing  $kk_L - \omega$  turbulence model of Walters and Cokljat (2008) and the IDDES framework by Shur et al. (2008) are used to develop a new hybrid RANS/LES model.

### Proposed transitional hybrid RANS/LES model

Several modifications were required before the transition location was predicted with satisfactory accuracy. A tripping mechanism was proposed, which forces the steady flow from the RANS region to become unsteady in the LES region, reducing the typical *grey area* observed in hybrid RANS/LES simulations. The following modification were studied:

1. The classical DES modification is made to the turbulent kinetic energy ( $k_T$ ) equation.

The destruction  $k_T\omega$  remains in its original form when the flow is solved using RANS. When the flow is solved in the LES mode, the destruction term becomes a function of the filter width ( $\Delta$ ) and reads  $k_T \frac{\sqrt{k_T}}{C_{DES}\Delta}$ . This modification of the sub-grid scale destruction term, introduces a length scale into the equations. Therefore, the LES region behaves as a Smagorinsky like sub-grid scale model. The original IDDES model contains an elevating function for the eddy viscosity at the interface of the near-wall RANS/LES transition region. However, this function is based on the Spalart-Allmaras turbulence variable  $\tilde{\nu}$  which is not present in the new model. Therefore, an approximation of the relation of  $\frac{\tilde{\nu}}{\nu}$  and the elevation strength is made. At last, the low-Reynolds number correction terms in the laminar and turbulent kinetic energy equations are deactivated when the flow is solved in LES.

Unfortunately, the pretransitional flow is solved in LES. Therefore, the destruction of  $k_T$  is increased which results in a lower value of  $k_T$ . So low in fact that the terms responsible for transition never become active and the flow remains fully laminar.

2. The turbulent transition parameters of the  $kk_L - \omega$  model are separated from the hybrid RANS/LES turbulence parameters. This requires the introduction of the following *background RANS* fields,  $k_{L,RANS}$ ,  $k_{T,RANS}$  and  $\omega_{RANS}$ . These fields are solved using the original equations of  $kk_L - \omega$ .

Transition from laminar to turbulent flow is predicted but takes place too much upstream compared to experimental data. The reason for this behaviour is as follows. At the outer edge of the laminar boundary layer the flow is solved using LES. Therefore, the strain-rate near the laminar boundary layer is reduced. Subsequently, the *background RANS* model compensates for the reduced strain-rate and increases the value of  $k_{T,RANS}$ . As a result the transitional threshold value is reached too soon and transition takes place.

3. A shear-sheltering limiter is introduced to deactivate the LES mode near the laminar boundary layer. It was also observed that the viscous damping function, the intermittency and the near wall damping do not return the correct value of one in the turbulent boundary layer. This is also a result of the reduced size of the turbulent kinetic energy field  $k_T$ . The problem was solved by setting the value of the aforementioned fields to unity when the flow is solved in the LES mode.

These modifications result in a predicted transition location which corresponds to the RANS result of  $kk_L - \omega$ . However, the flow from the laminar boundary layer is solved in the RANS mode and does therefore contain little to no unsteady motion. When this flow enters the LES region resolved turbulence is developed too slow, in DES simulation this behaviour is referred to as the *grey area* and is associated with a reduced skin-friction.

4. A tripping mechanism is added to the model. This is deemed necessary to remove the observed delayed development of turbulent structures. Velocity fluctuations with the intensity of  $\sqrt{\frac{2}{3}k_L}$  are added to the velocity field at the location where the flow leaves the laminar region and enters the turbulent region.

Due to the tripping mechanism, development of turbulent structures now take place directly after transition. The *grey area* problem is therefore reduced. In this study it was unfortunately not possible to add divergence free velocity fluctuation to the



flow. This poses a significant problem as many more iterations of the pressure-velocity corrector are required before the non-divergence free fluctuations are filtered out of the velocity field. The required computational time increased by roughly a factor 5. A second problem is that large pressure fluctuations are introduced by the pressure-velocity corrector when the non-divergence free fluctuations are filtered. This made it impossible to subtract realistic pressure field data for aero-acoustic computations.

## Verification and validation

When the intensity of the introduced velocity fluctuations are decreased by a factor of four, the development of turbulence is again delayed and a decrease in skin-friction is observed. Increasing the fluctuation intensity moves the transition location slightly forward and increases the skin-friction slope at transition significantly, transition becomes more or less instantaneous. It is concluded that although the skin-friction distribution changes with varying scaled velocity fluctuations, the overall behaviour of the transitional boundary layer and the location of transition are only influenced slightly.

Verification of the  $kk_L - \omega$  model shows that the skin-friction along a flat-plate converges when the resolution of the numerical mesh is refined. It is also observed that the predicted skin-friction of the low turbulence inflow condition ( $Tu = 0.8\%$ ) matches much better with reference data than the simulation with a much higher turbulence intensity ( $Tu = 3\%$ ). Possible reason can be that the stochastic velocity fluctuations of the tripping mechanism do not effectively introduce the turbulent kinetic energy. Validation also shows that for increasing turbulence levels, the mesh resolution near the transition location needs to be refined in order to be able to solve the transition process with LES.

## Test cases

Most important observations and conclusions from the NACA 0012 ( $Re_x = 1e6$ ) and DU91W250 ( $Re_x = 3.2e6$ ) simulations are:

- Predicted skin-friction distribution over the NACA 0012 agree well with RANS results. A comparison with the SA IDDES reference data indicates a superior performance of  $kk_L - \omega$  IDDES. A slightly underprediction in the skin-friction is obtained for the DU91W250 airfoil, the transition location does correspond with reference data.
- For both the NACA 0012 and DU91W250, the boundary layer thickness is only slightly underpredicted. For the NACA 0012 the displacement thickness is overpredicted by 14%. However, the SA IDDES reference data, overpredicted the displacement thickness by 266%.
- For all simulations, a comparison with the RANS results indicate that right after transition the large-scale kinetic energy is underdeveloped.

- Discrepancies in the turbulent boundary layer are believed to originate from the added velocity fluctuations of the tripping mechanism. These velocity fluctuations probably do not represent the pretransitional structures accurately enough.
- When the RANS/LES blending takes place deeper inside the transitional region, improved skin-friction distributions are obtained. This is believed to be caused by the increased boundary layer penetration of the added velocity fluctuations, the increased unsteady motion in the near-wall region improves the development of the downstream turbulent boundary layer.
- Surface pressure fluctuations, along the full chord length, are corrupted by the non-divergence free velocity fluctuations.

## 9.2 Recommendations

From the results and conclusions it became clear that several subjects are required to be studied further, before an accurate and cost-efficient hybrid RANS/LES model is reached. The following list describes some ideas for future work:

- Addition of divergence free velocity fluctuations could decrease computational time and provide non-corrupted surface pressure fluctuations for aeroacoustic simulations.
- Other means to promote the development of turbulent structures, could be through pressure fluctuations or by introducing a fluctuating eddy viscosity field in the pretransitional boundary layer. For each of these methods, a length scale can be computed from  $\omega$ .
- Improved skin-friction results were obtained when the near-wall RANS limiter of the tripping mechanism became thinner. Completely removing the near-wall RANS limiter from the tripping mechanism might improve the development of turbulence and make the model less dependent on the local mesh resolution.
- The *background* RANS model is able to predict the correct transition location but requires to solve three additional equations. Other RANS models that can predict laminar to turbulent flow, such as  $\gamma - Re_{\Theta}$ , might be able to predict transition without the need of a *background* model, reducing computational time.
- Improved turbulent boundary layer characteristics might be obtained when the introduced velocity fluctuations more accurately represent the physical motion present in pretransitional boundary layer.
- A more simple model that can define the RANS/LES blending could reduce computational time and result in a more predictive behaviour. The  $kk_L - \omega$  model itself might be able to provide such a RANS/LES blending function.

---

# Bibliography

- P. Andersson, M. Berggren, and D.S. Henningson. Optimal disturbances and bypass transition in boundary layers. *Physics of Fluids*, 11(1), 1991.
- A. Beechook. Development and implementation of a new hybrid RANS/LES model for transitional boundary layers in OpenFOAM. Master's thesis, Coventry University, 2016.
- M.R. Castelli, G. Grandi, and E. Benini. Numerical analysis of laminar to turbulent transition on the DU91-W2-250 airfoil. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 6(3), 2012.
- E.R. Corino and R.S. Brodkey. A visual investigation of the wall region in turbulent flow. *Journal of Fluid Mechanics*, 37(1):1–30, 1968.
- J. Coupland. ERCOFTAC special interest group on laminar to turbulent transition and retransition, technical report a309514, 1990. URL [http://www.ercoftac.org/products\\_and\\_services/classic\\_collection\\_database/](http://www.ercoftac.org/products_and_services/classic_collection_database/).
- J. W. Deardorff. A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics*, 41(2):453–480, 1970.
- D.J.C Dennis. Coherent structures in wall-bounded turbulence. *Anais da Academia Brasileira de Ciencias*, 87(2), 2015.
- OpenFOAM Foundation. OpenFOAM 4.1., 2016. URL <https://openfoam.org/release/4-1/>.
- J. Fürst. Numerical simulation of transitional flows with laminar kinetic energy. *Engineering Mechanics*, 20(5):379–388, 2013.
- M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids*, 3(7), 1991.
- B. Greschner, J. Grilliat, MC. Jacob, and F. Thiele. Measurements and wall modeled LES simulation of trailing edge noise caused by a turbulent boundary layer. *International journal of aeroacoustics*, 9(3):32 – 355, 2010.
- Pointwise Inc. V17.2r1, 2014. URL <http://www.pointwise.com/theconnector/April-2014-Special/Pointwise-V172.shtml>.

- A. Keating and U. Piomelli. A dynamic stochastic forcing method as a wall layer model for large-eddy simulation. *Journal of Turbulence*, 7(12), 2006.
- P.S. Klebanoff. characteristics of turbulence in a boundary layer with zero pressure gradient. Technical report, National Advisory Committee for Aeronautics technical note ; 3178, July 1954. NACA TN No. 3178.
- O. Marsden, C. Bogey, and C. Bailly. Noise radiated by a high-Reynolds-number 3-D airfoil. *AIAA/CEAS Aeroacoustics Conference*, 11, 2005.
- H.J. Medina and J. Early. Modelling transition due to backward-facing steps using the laminar kinetic energy concept. *European Journal of Mechanics -B/Fluids*, 44:60–68, 2016.
- F.R. Menter. Zonal two equation k-omega turbulence models for aerodynamic flows. *AIAA, Fluid Dynamics Conference*, 24, 1993.
- U. Michel, D. Eschricht, B. Greschner, T. Knacke, C. Mockett, and F. Thiele. *Advanced DES Methods and Their Application to Aeroacoustics*, pages 59–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- B. Nebenfuhr. Aerodynamic and aeroacoustic analysis of a multi-element airfoil using hybrid RANS/LES modeling approaches. Master’s thesis, Chalmers university of technology, 2012.
- N.V. Nikitin. An approach to wall modeling in large-eddy simulations. *Physics of Fluids*, 12, 2000.
- J.M. Osterlund. *Experimental Studies of Zero Pressure-Gradient Turbulent Boundary-Layer Flow*. PhD thesis, Department of Mechanics, Royal Institute of Technology, Stockholm, 1999.
- U. Piomelli. Wall-layer models for large-eddy simulations. *Progress in Aerospace Sciences*, 44:437–446, 2008.
- C.L. Rumsey. Apparent transition behavior of widely-used turbulence models. *International Journal of Heat and Fluid Flow*, 28(6):1460 – 1471, 2007. ISSN 0142-727X. URL <http://www.sciencedirect.com/science/article/pii/S0142727X07000392>. Revised and extended papers from the 5th conference in Turbulence, Heat and Mass Transfer.
- H. Schlichting. *Boundary layer theory*. McGraw-Hil, 1979.
- M.L. Shur, P.R. Spalart, M. Strelets, and A.K. Travin. A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities. *International Journal of Heat and Fluid Flow*, 29(6):1638 – 1649, 2008. URL <http://www.sciencedirect.com/science/article/pii/S0142727X08001203>.
- J. Smagorinsky. General circulation experiments with the primitive equations. *General Circulation Research Laboratory, U.S. Weather Bureau, Washington, D.C.*, 91(3), 1963.
- P. Spalart. A one-equation turbulence model for aerodynamic flows. *Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings*, 30, 1992.

- P. R. Spalart, S. Deck, M. L. Shur, K. D. Squires, M. Kh. Strelets, and A. Travin. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theoretical and Computational Fluid Dynamics*, 20(3):181, 2006. ISSN 1432-2250. doi: 10.1007/s00162-006-0015-0. URL <http://dx.doi.org/10.1007/s00162-006-0015-0>.
- P.R. Spalart, W-H. Jou, M. Strelets, and S.R. Allmaras. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. *Proceedings of first AFOSR international conference on DNS/LES*, 1997.
- S. Stryzhak. *Creating your own libraries for acoustic analogy in OpenFOAM*, 2015. URL <https://github.com/unicfdlab/libAcoustics/tree/master/OpenFOAM-3.0.0>.
- T. Theodorsen. Mechanism of turbulence. *Midwestern Conference on Fluid Mechanics*, 2:1–18, 1952.
- E.R. van Driest. On turbulent flow near a wall. *Journal of the Aeronautical Sciences*, 23(11):1007–1011, 1956.
- O. Verhoeven. Trailing edge noise simulations. Master’s thesis, Delft University of Technology, 2006.
- D.K. Walters and D. Cokljat. A three-equation eddy-viscosity model for Reynolds-Averaged NavierStokes simulations of transitional flow. *Journal of Fluids Engineering*, 130(12), 2008.
- H. White. *Viscous Fluid Flow*. McGraw-Hill, 2006.
- J. Winkler and S. Moreau. LES of the trailing-edge flow and noise of a NACA6512-63 airfoil at zero angle of attack. *Center for Turbulence Research Proceedings of the Summer Program*, 2008.
- W.R. Wolf and S.K. Lele. Trailing edge noise predictions using compressible LES and acoustic analogy. *AIAA/CEAS Aeroacoustics Conferences*, 17, 2011a.
- W.R. Wolf and S.K. Lele. Investigation of noise generated by a DU96 airfoil. *AIAA/CEAS Aeroacoustics Conferences*, 18, 2011b.
- Y. Zhang, T. Gillebaart, G.J.W van Bussel, and H.Bijl. Validation of a transition model for the DU91-W2-250 airfoil. *Proceedings of 9th PhD Seminar on Wind Energy in Europe*, 2013. Uppsala University Campus Gotland, Sweden.



---

# Appendix A

---

## Mesh generation procedure

The upcoming sections will explain the processes that lead to a proper OpenFOAM mesh for the DU91W250 airfoil.

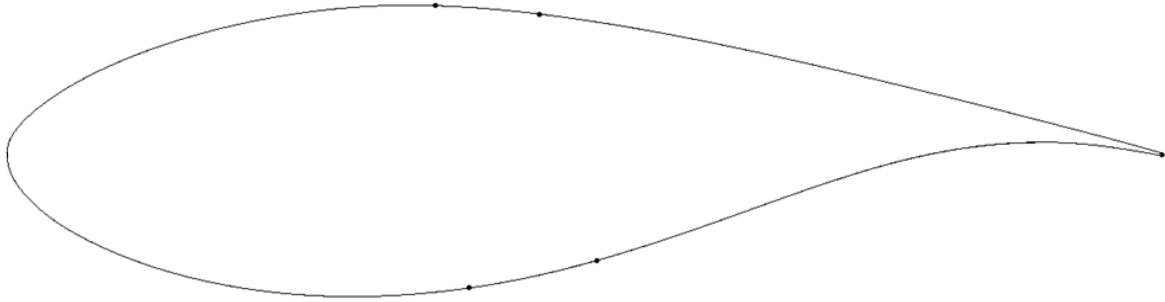
### A.1 Zones

This section specifies the zones over which control of the turbulence modelling by the modified SA IDDES model is required, see Section 3.1.2. Split the airfoil at the locations of expected transition. The chordwise division of the laminar and turbulent zones are based on the transition locations obtained from XFOIL with the  $e^n$  method, these locations are shown in Table A.1.

**Table A.1:** Predicted transition locations from XFOIL for DU91W250,  $n_{crit} = 9$ .

AoA [ $^\circ$ ]	Suction Side [x/c]	Pressure side [x/c]
6	0.376	0.512
-2.85	0.496	0.412

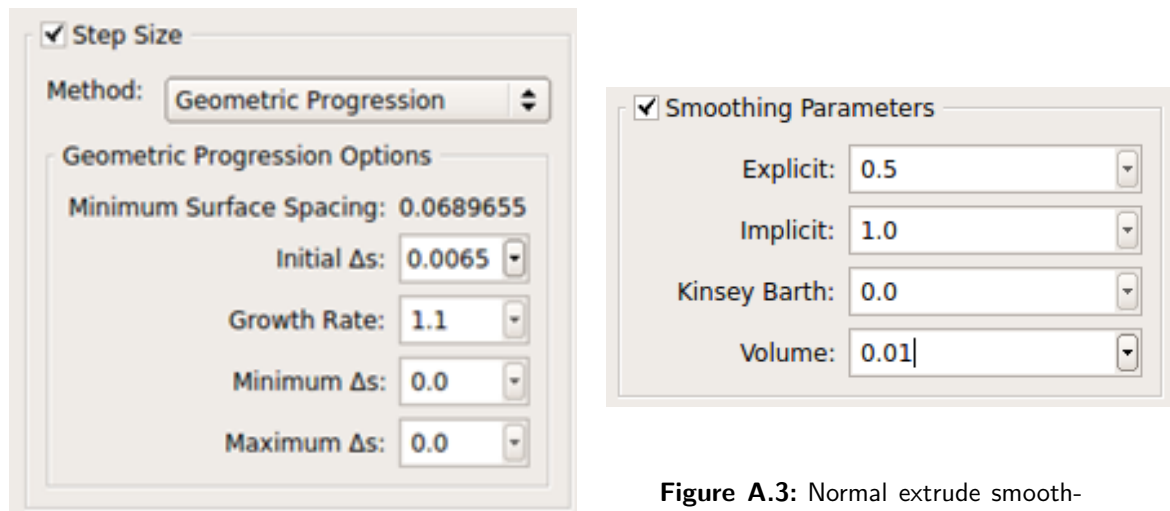
There are now 4 different zones. The first zone at the leading edge will be laminar for both angles of attack. The upper zone will only be laminar if  $AoA = -2.85^\circ$ , while the lower zone is only laminar in the case  $AoA = 6^\circ$ . The trailing edge zone is always turbulent.



**Figure A.1:** DU91W250 outline. Dots indicate the zones, based on the expected transition location(A.1).

## A.2 Extrude

Under *create*, select *extrude normal*. This will layer cells around the airfoil. The initial step should equal  $y^+$ , the growth rate should equal 1.1. Setting the Volume parameter to a low value for the first few cell layers will assure orthogonal cells. Extrude the mesh until the full boundary layer would be covered, this will be around 60 cell layers.



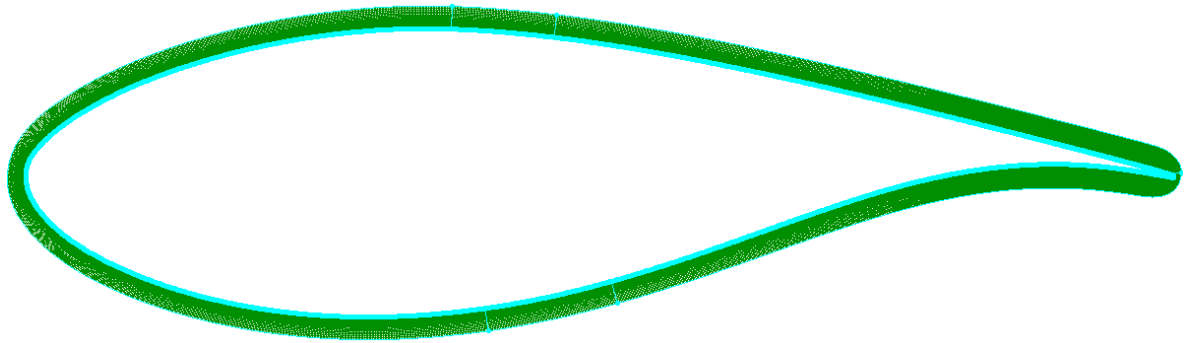
**Figure A.2:** stepsize

**Figure A.3:** Normal extrude smoothing parameters.

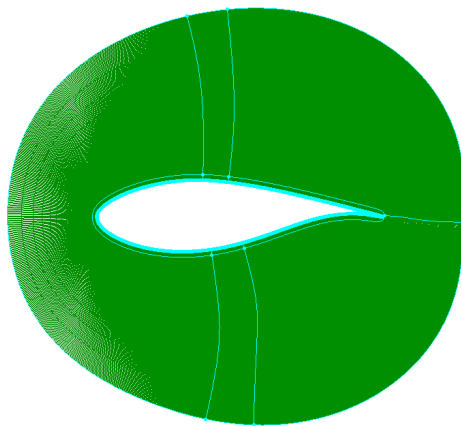
Subsequently the domain is split at the prescribed locations (Figure A.1), the first radial segment is finished. The next radial segment will act as a transition mesh between the farfield - course mesh- and the fine - boundary layer - mesh. The transition mesh should have its outer boundary at least  $\frac{1}{2}$  chord away from the airfoil. This will assure a low non-orthogonality at the location of mesh refinement. The subsequent normal extrusions are shown in Figure A.4,



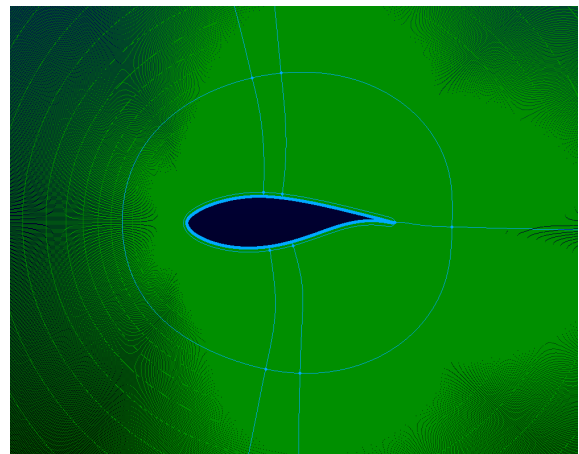
A.5 and A.6. For the transition and farfield mesh a growth rate of 1.2 can be applied and the volume parameter can be set to 0.5.



**Figure A.4:** Third radial segment, covers the boundary layer and will have a high mesh density.



**Figure A.5:** Second radial segment, acts as the transition between the third and first radial segments.



**Figure A.6:** First radial segment will have a low mesh density.

## A.3 Cyclic boundaries

Using the *periodic* function, a 3D periodic mesh is generated. Use connectors to connect the intersections of the left and right plane with each other.

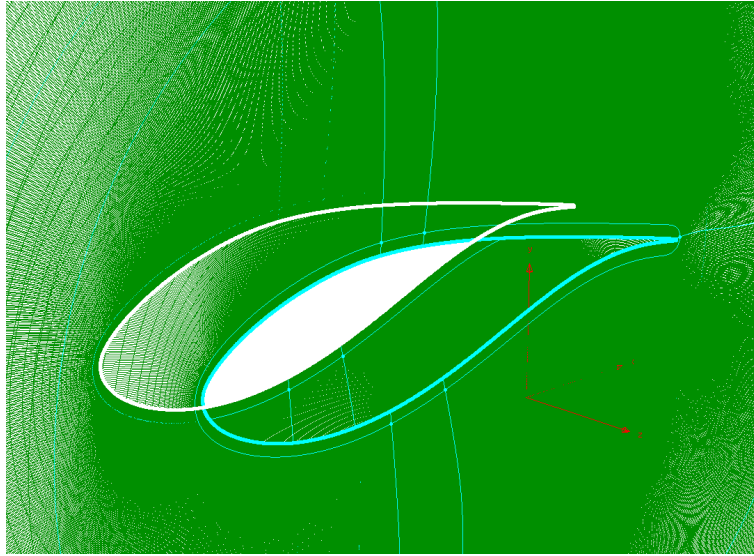


Figure A.7: MakeCyclicPlane

## A.4 Domains and blocks

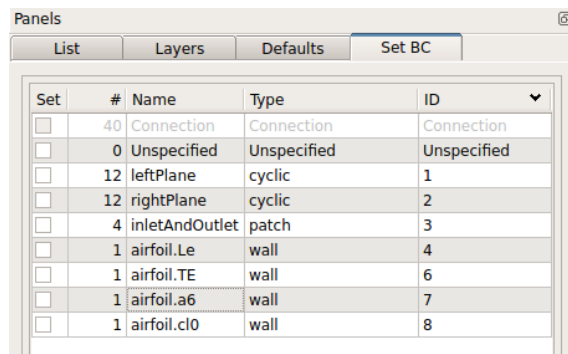
Create domains on the airfoil surface and between all the connectors on the left and right plane. Using these domains, 3D blocks can be generated. Each radial segment will consist of 4 blocks.

## A.5 Boundary and volume conditions

A name will be assigned to the domain and blocks. These are required for OpenFOAM to distinct the laminar and turbulent zones and to specify the areas which will be treated with spanwise cell refinements. Name and distribute the domains and blocks as specified by Figures A.8 and A.9, respectively.

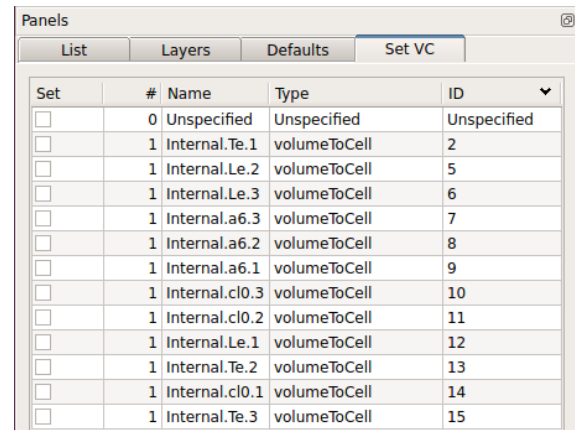
## A.6 Solver attributes

Set *CellExport* in the solver attributes to *SetAndZones* such that Pointwise will export the boundary and volume conditions when written to file.



Set	#	Name	Type	ID
<input type="checkbox"/>	40	Connection	Connection	Connection
<input type="checkbox"/>	0	Unspecified	Unspecified	Unspecified
<input type="checkbox"/>	12	leftPlane	cyclic	1
<input type="checkbox"/>	12	rightPlane	cyclic	2
<input type="checkbox"/>	4	inletAndOutlet	patch	3
<input type="checkbox"/>	1	airfoil.Le	wall	4
<input type="checkbox"/>	1	airfoil.TE	wall	6
<input type="checkbox"/>	1	airfoil.a6	wall	7
<input type="checkbox"/>	1	airfoil.cI0	wall	8

Figure A.8: Boundary conditions.



Set	#	Name	Type	ID
<input type="checkbox"/>	0	Unspecified	Unspecified	Unspecified
<input type="checkbox"/>	1	Internal.Te.1	volumeToCell	2
<input type="checkbox"/>	1	Internal.Le.2	volumeToCell	5
<input type="checkbox"/>	1	Internal.Le.3	volumeToCell	6
<input type="checkbox"/>	1	Internal.a6.3	volumeToCell	7
<input type="checkbox"/>	1	Internal.a6.2	volumeToCell	8
<input type="checkbox"/>	1	Internal.a6.1	volumeToCell	9
<input type="checkbox"/>	1	Internal.cI0.3	volumeToCell	10
<input type="checkbox"/>	1	Internal.cI0.2	volumeToCell	11
<input type="checkbox"/>	1	Internal.Le.1	volumeToCell	12
<input type="checkbox"/>	1	Internal.Te.2	volumeToCell	13
<input type="checkbox"/>	1	Internal.cI0.1	volumeToCell	14
<input type="checkbox"/>	1	Internal.Te.3	volumeToCell	15

Figure A.9: Volume conditions.

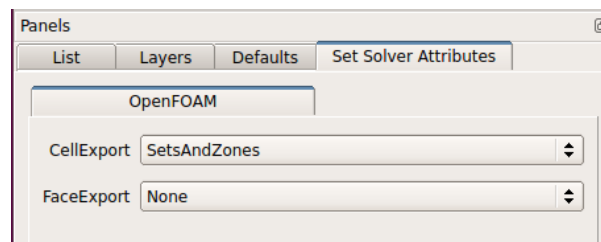


Figure A.10: Solver attributes.

## A.7 Export

Select all domains, boundaries and connectors and use *export*, *CAE* to create the OpenFOAM mesh.

## A.8 OpenFOAM mesh generation

## A.9 Compatibility

The output format of the mesh by Pointwise is not complete. The cyclic boundaries in the boundary file need to be redefined. The first step is to add two lines to the boundary file: the line `neighbourPatch leftPlane` and `neighbourPatch rightPlane` need to be added to the patches `rightPlane` and `leftPlane`, respectively. An example is given in Figure A.11.

```

leftPlane
{
    type cyclic;
    nFaces 217120;
    startFace 436080;
    neighbourPatch rightPlane;
}

```

**Figure A.11:** Correct representation of cyclic patch

The second step is to set-up the `createPatchDict`. Here the name of the cyclic patches should be used, e.g. `leftPlane` or `rightPlane`. Also the `neighbourPatch` should be defined, e.g. `rightPlane` or `leftPlane`.

## A.10 Scale mesh

The chord length of the Pointwise mesh is 1000 meters. This needs to be transformed into meters for OpenFOAM. The following command is used:

`transformPoints -scale "(0.0009 0.0009 0.0009)"`. Before performing the scaling, the `writePrecision` should be increased to 24 digits in order to have a successful scaling. The write precision can be changed in the `controlDict`.

## A.11 Mesh refinement

At this point a useful OpenFOAM mesh is generated. However, this mesh still has only 1 cell in the spanwise direction. The `refineMesh` utility is used for mesh refinements in specified spanwise direction. Each time a refinement is applied, the number of cells in the specified direction doubles. Apply the refinement for each radial segment until the desired number of cells has been reached.

## A.12 Mesh quality

The final mesh is assessed using the OpenFOAM utility `checkMesh`. The results are summarized in Table A.2. Since the full mesh is structured, the non-orthogonality is low. Therefore, only one non-orthogonal corrector is required. The cells with the highest non-orthogonality are the polyhedral cells which are located at the transition between the radial mesh zones.

Non-orthogonality of these cell can be reduced further when the aspect ratio, in radial direction, are increased.

**Table A.2:** Mesh quality.

Property	Maximum	Average
aspect ratio	190	-
Non-orthogonality	57	2
Skewness	2.9	-

### A.13 Specify the laminar and turbulent regions

Using the `setFieldDict`, the laminar regions can be specified. These zones where already created in Pointwise (Section A.1) and will now be assigned the value of zero if there boundary layer is of the laminar type. After executing the `setFieldDict`, the file `LamTurRegion` will adjusted accordingly. Section 3.1.2 provides an explanation of the structure and effect of the modified `nuTilda`.



---

## Appendix B

---

### Substitution of divergence free fluctuations

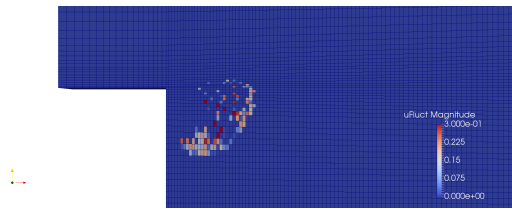
The generated fluctuations  $u'$ , from the trpping mechanisme are random and will not be divergence free. Adding this fluctuation field to the velocity field will therefore results in a non-divergence free velocity field. In this Section a method is suggested that converts the velocity fluctuation  $u'$  into a divergence free field. This divergence free field is then added to the velocity field. Unfortunately, the method is unstable and could therefore not be used.

The correction for the divergence free fluctuations is performed inside the PISO solver. The modification is as follows,

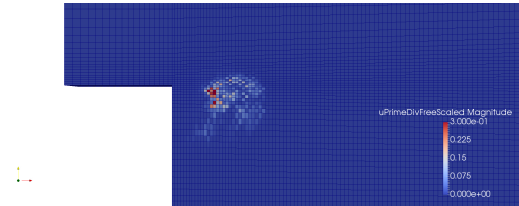
1. Solve  $U$
2. Perform  $p - U$  correction  $\rightarrow p_{new}$  and  $U_{new}$
3. Temporally save  $p_{new}$  and  $U_{new}$  as  $p_{tmp}$  and  $U_{tmp}$
4. Generate random fluctuations  $u'$
5. Perform  $p - (U + u')$  correction  $\rightarrow p_{new,div.-free}$  and  $U_{new,div.-free}$
6. Compute divergence free fluctuation  $u'_{div.-free} = U_{new,div.-free} - U$
7. Reset  $U_{new} = U_{tmp} + u'_{new,div.-free}$  and  $p_{new} = p_{tmp}$
8. Subtract  $u'_{new,div.-free}$  from  $k_L$

In set 5, the pressure-velocity corrector removes the non-divergence free velocities from the the velocity field  $U$ . Ideally, the absolute difference between  $u'_{new,div.-free}$  and  $u'$  is zero and the location of both fluctuations remain in the same location as well as the random distribution of the fluctuations in  $u'$ . A low cost computational simulation is performed to analyse the way this method behaves. For this the 2-dimensional PitzDaily tutorial case of OpenFOAM [Foundation \(2016\)](#) is used. The initial velocity fluctuations  $u'$  show little to non correlation

in intensity as is shown in Figure B.1. After the fluctuations have been made divergence free the randomness of the fluctuations and intensity have also been filtered out, this can be seen in Figure B.2. The random fluctuations  $u'$  have been replaced by a fluctuation field, which contains unwanted coherence  $u'_{new,div.-free}$ . As a result of the coherent  $u'_{new,div.-free}$ , the strain rate is increased, which then increases the kinetic energy. A feedback loop is created which eventually leads to a blow up of the turbulence fields, making this method unsuitable for tripping the laminar flow.



**Figure B.1:** Magnitude of initial random velocity fluctuation  $u'$ .



**Figure B.2:** Magnitude of the divergence free velocity fluctuation  $u'_{div.-free}$ , lacking randomness in the fluctuation magnitude and location.



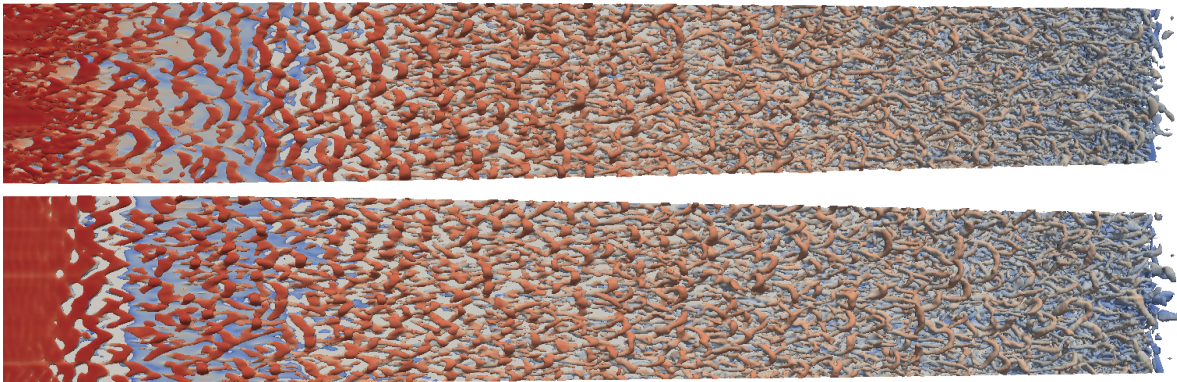
---

## Appendix C

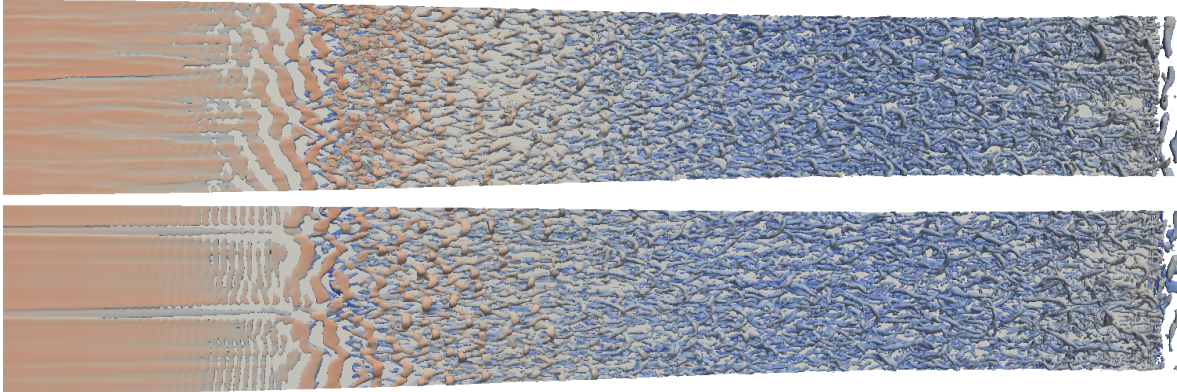
---

### Vorticity contours DU91W250

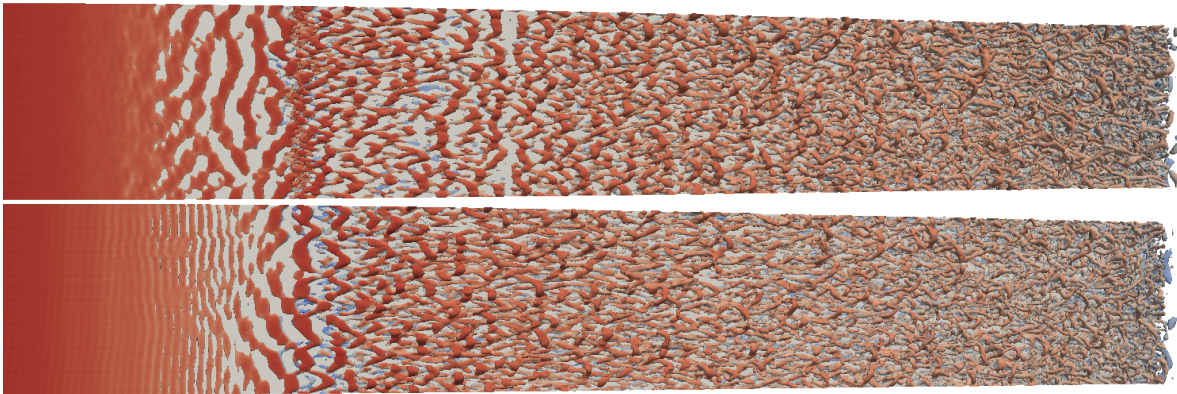
This appendix provides a comparison of the vorticity contours obtained with the  $kk_L - \omega$  IDDES model and the modified SA IDDES model (Section 2.3.1).



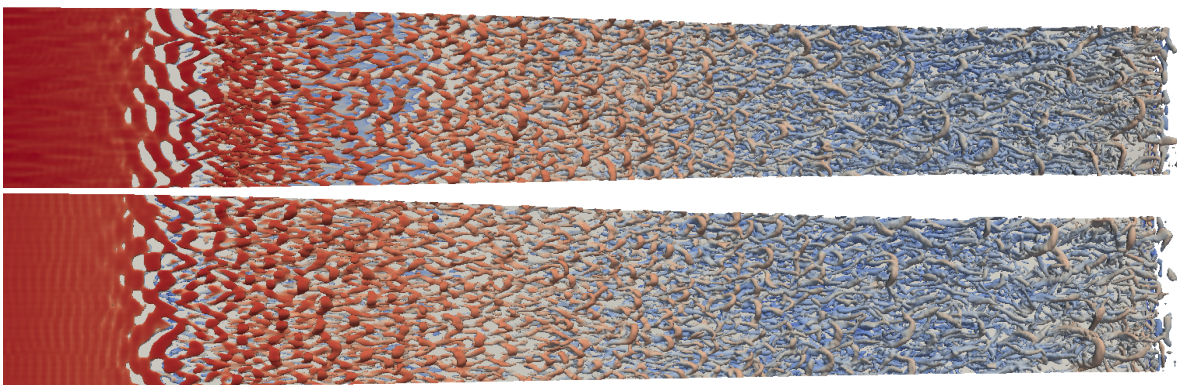
**Figure C.1:** Vorticity contours ( $Q - iso = 3e6$ ) on upper side of DU91W250.  $AoA = 6^\circ$ . *Top:*  $kk_L - \omega$  IDDES. *Bottom:* modified SA IDDES.



**Figure C.2:** Vorticity contours ( $Q - iso = 3e6$ ) on lower side of DU91W250.  $AoA = 6^\circ$ . *Top:*  $kk_L - \omega$  IDDES. *Bottom:* modified SA IDDES.



**Figure C.3:** Vorticity contours ( $Q - iso = 3e6$ ) on upper side of DU91W250.  $AoA = -2.85^\circ$ . *Top:*  $kk_L - \omega$  IDDES. *Bottom:* modified SA IDDES.



**Figure C.4:** Vorticity contours ( $Q - iso = 3e6$ ) on lower side of DU91W250.  $AoA = -2.85^\circ$ . *Top:*  $kk_L - \omega$  IDDES. *Bottom:* modified SA IDDES.

---

# Appendix D

---

## Acoustics of DU91W250

In this Appendix the methodology used for the aeroacoustic simulations of the DU91W250 airfoil is presented, Section D.1. Predicted noise level by the modified SA IDDES model (Section 3.1.2) for  $AoA = 6^\circ$  and  $AoA = -2.85^\circ$  can be found in Sections D.2.1 and D.2.2, respectively. Because of the tripping mechanism of  $kk_L - \omega$  IDDES, which corrupts the pressure field, no aeroacoustic simulation was performed.

### D.1 Acoustic analogy

An existing OpenFOAM library is used. The library was developed in 2013 by Sorguven [Stryzhak \(2015\)](#) and came online available for OpenFOAM 3.0.0 in 2015. It is based on Curle's analogy and derives the pressure fluctuation for the far-field of a compact, rigid body.

An integral form of Curle's equation is

$$p'(x, t) = \frac{x_i x_j}{4\pi ||x||^2 c_0^2} \frac{\partial^2}{\partial t^2} \int_V \left[ \frac{T_{ij}}{r} \right]_{\tau=t_e} dV_y + \frac{1}{4\pi} \frac{\partial}{\partial t} \int_S \left[ \rho v_i r \right]_{\tau=t_e} n_i dS + \frac{x_i}{4\pi ||x|| c_0} \frac{\partial}{\partial t} \int_S \left[ \frac{p_{ij} + \rho v_i v_j}{r} \right]_{\tau=t_e} n_i dS, \quad (\text{D.1})$$

and consists of three terms. Here it is assumed that  $p' = c_0^2 \rho'$  is valid at the observer location. The three terms are the volume sources, mass flux sources and surface loading sources, respectively.

The following simplifications are made: for low Mach numbers the volume sources can be neglected. The object surface will be impermeable and therefore the mass flux sources will also be neglected. Assuming that the object is stationary will lead the simplification  $\rho v_i v_j = 0$ .



Furthermore, it is assumed sound sources originate from a compact body, meaning that the retarded time can be neglected and that  $|x| = r$ . Rewriting Equation D.1 using these assumptions and placing the temporal derivative inside the integral will lead to Equation D.2 as used in the acoustic Library from Sorguven [Stryzhak \(2015\)](#).

$$p'(x, t) = \frac{1}{4\pi c_0} \frac{x_i}{r^2} \left[ \frac{\partial F_i}{\partial t} + \frac{c_0 F_i}{r} \right] \quad (\text{D.2})$$

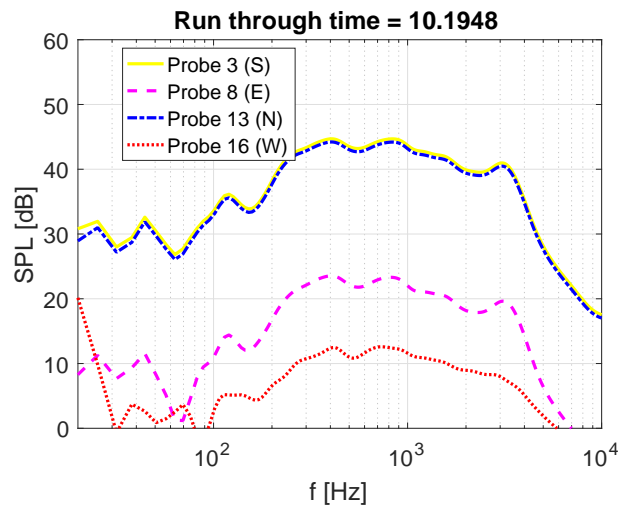
Curle's analogy assumes that the source region is compact,  $\frac{\lambda}{c} \ll 1$ , which dictates that the source length should be much smaller than the sound wave length. This implies that all noise sources can be regarded to originate from one single point. When the source length is longer than the wave length, the time which the signal requires to travel from source to observer (retarded time or emission time) should be taken into account. Since Curle's analogy is not capable of this, deviations will emerge in the results. A solution to this problem would be to use the Ffowcs William-Hawkings analogy. Physical boundaries allow for reflection, absorption and damping of pressure fluctuations. However, the CAA methods used in this report do not have the capabilities to process the effect of surface properties. The acoustic pressure fluctuations could therefore be under or over estimated. A physical acoustic pressure fluctuations travels with the speed of sound and can also be convected with the flow. The implemented CAA method does not make use of the flow field velocity and will therefore not be able to transport the pressure fluctuations with the flow. For simulations with low Mach numbers this does not impose a problem. The demand to analyse a flow over a range of frequencies will create requirements for the CFD simulation, leading to increased computational time. The highest resolvable frequency will constrain the maximum allowable time step. On the other hand, the minimum required frequency will dictate a minimum sampling time. These requirements for an upper and lower frequency will be further explained. According to the Nyquist criterion: the sampling frequency should be at least twice the highest frequency. The highest audible frequency is 20 kHz. Hence, the maximum timestep should stay below 2.5e-05 seconds. However, the maximum timestep used for the operating and zero lift case is limited by the Courant number to even lower values (1.8e-6 s).

## D.2 Results

Acoustic results are presented for the operating and zero lift case. SPL reported are unscaled, they reflect the noise from the airfoil with a span of 10 percent of the chord. Frequency filtering of 1/3 Octave band is applied to the signal. The pressure samples for both operating and zero lift have a length of roughly 10 run through times. Therefore, most of the frequencies in the range of 20 Hz-20 kHz will be captured. The lowest frequencies, 20 Hz-100 Hz will require a pressure signal which is 2-4 time longer, in order to be presented accurately. Probe locations are centred around the trailing edge, at a distance of 1700mm.

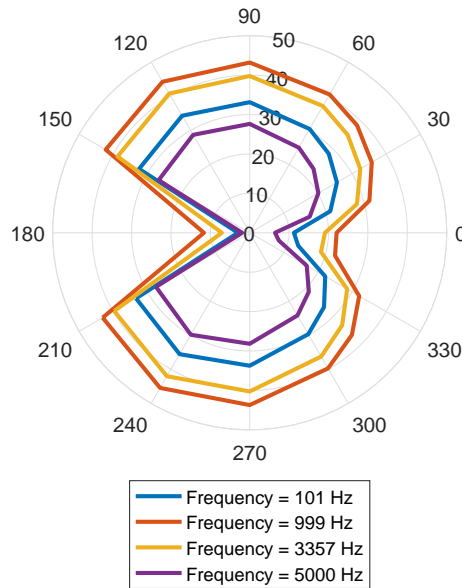
### D.2.1 Operating lift

Figure D.1 presents the SPL range for 4 different probe locations, south, east, north and west. The frequency range from 200 Hz to 2000 Hz is the typical broadband noise generated by a turbulent boundary. The most dominant noise frequency varies between 400 Hz and 1000 Hz and has a maximum SPL of 44 dB. The small peak near 3000 Hz is presumed to be generated by the vortex shedding of the blunt trailing edge and has a maximum SPL of 40 dB. The dominant noise mechanism is therefore the broadband noise.



**Figure D.1:** SPL with 1/3 Octave band filter.  $AoA = 6^\circ$ .

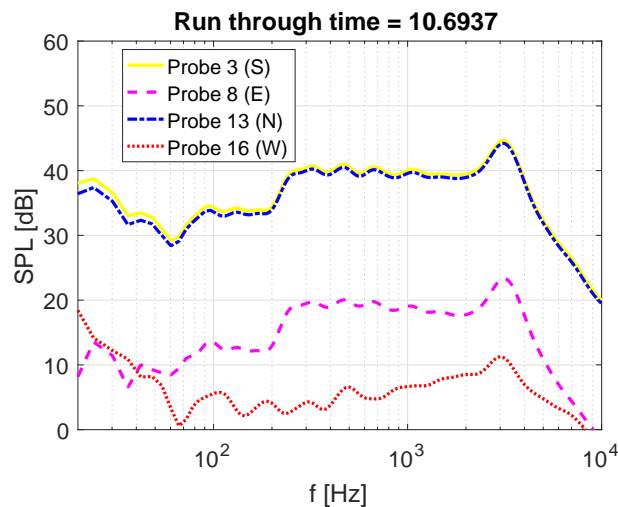
The directivity plot can be found in Figure D.2. This figure presents the SPL of the 16 probes. For each probe four different frequencies are shown. The directivity pattern shows two lobes, pointing in the north and south direction. This is a typical dipole pattern observed for airfoils.



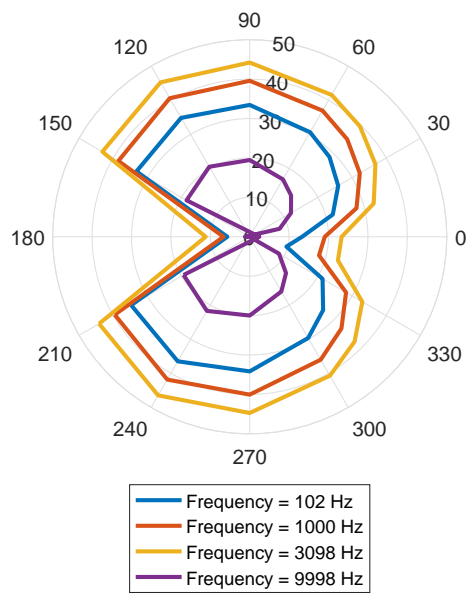
**Figure D.2:** Directivity polar of different frequencies.  $AoA = 6^\circ$ .

### D.2.2 Zero lift

The SPL levels for the south, east, north and west locations at zero-lift conditions are presented in Figure D.3. From this figure it is observed that the broadband noise varies from 300 Hz to 2000 Hz and has a maximum SPL of 40 dB. The SPL peak at 3000 Hz is generated by the vortex shedding from the blunt trailing edge. Opposite to the operating lift case, the tonal frequency from the vortex shedding is the dominating noise source with a SPL of 44 dB. Figure D.4 depicts the directivity of the noise source. Just as the operating lift case this figure shows the typical dipole behaviour of the noise source.



**Figure D.3:** SPL for different frequencies.



**Figure D.4:** Directivity polar of different SPL frequencies.





---

# Appendix E

---

## Modified PISO solver

```
// * * * * * // 1
int main(int argc, char *argv[])
{
    #include "postProcess.H" 6
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "createControl.H" 11
    #include "createFields.H"
    #include "createFvOptions.H"
    #include "initContinuityErrs.H"

    turbulence->validate(); 16

    // * * * * * //
    Info<< "\nStarting time loop\n" << endl;

    while (runTime.loop()) 21
    {
        Info<< "Time = " << runTime.timeName() << nl << endl;

        #include "CourantNo.H" 26

        // Pressure-velocity PISO corrector
        // {

            #include "UEqn.H" 31

            // Modified section
            // Modified section

            const volVectorField uFluct_ = mesh.lookupObject<volVectorField>("uFluct"); 36
            const volScalarField alphaFrac_ = mesh.lookupObject<volScalarField>("alphaFractionField");

            const vectorField uFluctTMP = uFluct_.internalField();
            scalar meanMaguFluct = gSum( ag(uFluctTMP) * mesh.V() ); 41

            Info << "Power uPrime = " << meanMaguFluct/mesh.time().deltaT().value() << " [Nm/s]" <<
                endl;

            // --- PISO loop 46
            while (piso.correct())
            {
                #include "pEqn.H"
            }

            laminarTransport.correct(); 51

            turbulence->correct();

            U = U + uFluct_; 56

            // Modified section
            // Modified section

```

```
////////////////////////////////////
runTime.write();
Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
    << " ClockTime = " << runTime.elapsedClockTime() << " s"
    << nl << endl;
}
Info<< "End\n" << endl;
return 0;
}
// ***** //
```

61

66

71

---

# Appendix F

---

## $kk_L - \omega$ IDDES OpenFOAM implementation

### F.1 kkLOmegaIDDES.H

```
/*-----*/
|
|  F i e l d           |   O p e n F O A M :   T h e   O p e n   S o u r c e   C F D   T o o l b o x
|  O p e r a t i o n   |   |
|  A n d               |   C o p y r i g h t   ( C )   2 0 1 1 - 2 0 1 5   O p e n F O A M   F o u n d a t i o n
|  M a n i p u l a t i o n |
|
|-----|
License
  This file is part of OpenFOAM.

  OpenFOAM is free software: you can redistribute it and/or modify it
  under the terms of the GNU General Public License as published by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version.

  OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
  ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
  FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
  for more details.

  You should have received a copy of the GNU General Public License
  along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Class
  Foam::incompressible::RASModels::kkLOmegaIDDES

Group
  grpIcoRASTurbulence

Description
  Low Reynolds-number k-kl-omega turbulence model for
  incompressible flows.

  This turbulence model is described in:
  \verbatim
    Walters, D. K., & Cokljat, D. (2008).
    A three-equation eddy-viscosity model for Reynolds-averaged
    Navier Stokes simulations of transitional flow.
    Journal of Fluids Engineering, 130(12), 121401.
  \endverbatim
  however the paper contains several errors which must be corrected for the
  model to operation correctly as explained in
  \verbatim
    Furst, J. (2013).
    Numerical simulation of transitional flows with laminar kinetic energy.
    Engineering MECHANICS, 20(5), 379-388.
  \endverbatim
  All these corrections and updates are included in this implementation.

  The default model coefficients are
```

```

\verbatim
  kkLOmegaIDDESCoeffs
  {
    A0          4.04
    As          2.12
    Av          6.75
    Abp         0.6
    Anat       200
    Ats       200
    CbpCrit   1.2
    Cnc       0.1
    CnatCrit  1250
    Cint     0.75
    CtsCrit  1000
    CrNat    0.02
    C11     3.4e-6
    C12     1.0e-10
    CR      0.12
    CalphaTheta 0.035
    Css     1.5
    CtauL   4360
    Cw1     0.44
    Cw2     0.92
    Cw3     0.3
    CwR     1.5
    Clambda 2.495
    CmuStd  0.09
    Prtheta 0.85
    Sigmak  1
    Sigmax  1.17
  }
\endverbatim
SourceFiles
  kkLOmegaIDDES.C
/*-----*/
#ifdef kkLOmegaIDDES_H
#define kkLOmegaIDDES_H
  // #include "RASModel.H"
  // #include "turbulentTransportModel.H"
  // #include "eddyViscosity.H"
  #include "LESeddyViscosity.H"
  #include "IDDESdelta.H"
  // * * * * * //
namespace Foam
{
  // namespace incompressible
  // {
  namespace LESModels
  {
    /*-----*/
    Class kkLOmegaIDDES Declaration
    /*-----*/
    template<class BasicTurbulenceModel>
    class kkLOmegaIDDES
    :
      //public eddyViscosity<RASModel<BasicTurbulenceModel>>
      public LESeddyViscosity<BasicTurbulenceModel>
    {
      // Private Member Functions
      tmp<volScalarField> alpha() const;
      tmp<volScalarField> ft(const volScalarField& magGradU, const volScalarField& nuts) const;
      tmp<volScalarField> fl(const volScalarField& magGradU) const;
      tmp<volScalarField> rd
      (
        const volScalarField& nur,
        const volScalarField& magGradU
      ) const;
      // Delay function
      tmp<volScalarField> fd(const volScalarField& magGradU, const volScalarField& nuts) const;
      // Disallow default bitwise copy construct and assignment
      kkLOmegaIDDES(const kkLOmegaIDDES&);
      void operator=(const kkLOmegaIDDES&);
    }
  }
}

```

```

protected: 145

// Protected Member Functions

virtual tmp<volScalarField> Psi(const volTensorField& gradU, const volScalarField& Omega, 150
    const volScalarField& ResInf, const volScalarField& nuts) const;

//-- Length scale
virtual tmp<volScalarField> dTilda
(
    const volTensorField& gradU, 155
    const volScalarField& S2,
    const volScalarField& kForm,
    const volScalarField& omegaForm,
    const volScalarField& Omega, 160
    const volScalarField& ResInf,
    const volScalarField& nuts
) const;

tmp<volScalarField> fv(const volScalarField& Ret) const; 165
tmp<volScalarField> fINTMIX() const;
tmp<volScalarField> fINTRANS() const;
tmp<volScalarField> fSS(const volScalarField& omega) const;

tmp<volScalarField> Cmu(const volScalarField& S, const volScalarField& omega) const; 170

tmp<volScalarField> BetaTS(const volScalarField& Rew) const; 175

tmp<volScalarField> fTaul
(
    const volScalarField& lambdaEff,
    const volScalarField& ktL,
    const volScalarField& omega
) const; 180

tmp<volScalarField> fOmega
(
    const volScalarField& lambdaEff,
    const volScalarField& lambdaT
) const; 185

tmp<volScalarField> phiBP(const volScalarField& Omega) const;

tmp<volScalarField> phiNAT
(
    const volScalarField& ReOmega,
    const volScalarField& fNatCrit
) const; 190

tmp<volScalarField> D(const volScalarField& k) const; 195
// Model coefficients
dimensionedScalar kappa_;
dimensionedScalar A0_;
dimensionedScalar As_;
dimensionedScalar Av_; 200
dimensionedScalar Abp_;
dimensionedScalar Anat_;
dimensionedScalar Ats_;
dimensionedScalar CbpCrit_;
dimensionedScalar Cnc_; 205
dimensionedScalar CnatCrit_;
dimensionedScalar Cint_;
dimensionedScalar CtsCrit_;
dimensionedScalar CrNat_;
dimensionedScalar C11_; 210
dimensionedScalar C12_;
dimensionedScalar CR_;
dimensionedScalar CalphaTheta_;
dimensionedScalar Css_;
dimensionedScalar CssBP_; 215
dimensionedScalar CtauL_;
dimensionedScalar Cw1_;
dimensionedScalar Cw2_;
dimensionedScalar Cw3_;
dimensionedScalar CwR_; 220
dimensionedScalar Clambda_;
dimensionedScalar CmuStd_;
dimensionedScalar Prtheta_;
dimensionedScalar Sigmak_;
dimensionedScalar Sigmaw_; 225
dimensionedScalar CDES_;
dimensionedScalar CPsiRestore_;
dimensionedScalar CfRestore_;
dimensionedScalar alpha_;

// Model constants 230

```

```

        dimensionedScalar fwStar_;
        dimensionedScalar cl_;
        dimensionedScalar ct_;
235

    // Fields

    const IDDESDelta& IDDESDelta_;
240

    // Fields
    volScalarField omega_;
    volScalarField omegaRANS_;
    volScalarField kl_;
    volScalarField klRANS_;
    volScalarField kt_;
    volScalarField ktRANS_;
    volVectorField uFluct_;
    volScalarField alphaFractionField_;
    volScalarField epsilon_;
245

    Random ranGen_;

    tmp<volScalarField> alphaT
    (
255
        const volScalarField& lambdaEff,
        const volScalarField& fv,
        const volScalarField& kts
    ) const;

    // - Wall distance
    // Note: different to wall distance in parent RASModel
    // which is for near-wall cells only
    const volScalarField& y_;
260

    // Protected Member Functions

    virtual void correctNut();
265

public:

    typedef typename BasicTurbulenceModel::alphaField alphaField;
    typedef typename BasicTurbulenceModel::rhoField rhoField;
    typedef typename BasicTurbulenceModel::transportModel transportModel;
275

    // - Runtime type information
    TypeName("kkLOmegaIDDES");
280

    // Constructors

    // - Construct from components
    kkLOmegaIDDES
    (
285
        const alphaField& alpha,
        const rhoField& rho,
        const volVectorField& U,
        const surfaceScalarField& alphaRhoPhi,
        const surfaceScalarField& phi,
        const transportModel& transport,
        const word& propertiesName = turbulenceModel::propertiesName,
        const word& type = typeName
    );
290

    // - Destructor
    virtual ~kkLOmegaIDDES()
    {}
295

    // Member Functions

    // - Read model coefficients if they have changed
    virtual bool read();
305

    // - Return the effective diffusivity for k
    tmp<volScalarField> DkEff(const volScalarField& alphaT) const
    {
310
        return tmp<volScalarField>
        (
            new volScalarField("DkEff", alphaT/Sigmak_ + this->nu())
        );
315
    }

    // - Return the effective diffusivity for omega
    tmp<volScalarField> DomegaEff(const volScalarField& alphaT) const
    {
320
        return tmp<volScalarField>

```

```

        (
            new volScalarField("DomegaEff", alphaT/Sigmaw_ + this->nu())
        );
    }
}

// - Return the turbulence specific dissipation rate
virtual tmp<volScalarField> omega() const
{
    return omega_;
}
// - Return the turbulence specific dissipation rate
virtual tmp<volScalarField> omegaRANS() const
{
    return omegaRANS_;
}

// - Return the laminar kinetic energy
virtual tmp<volScalarField> kl() const
{
    return kl_;
}
// - Return the laminar kinetic energy
virtual tmp<volScalarField> klRANS() const
{
    return klRANS_;
}

// - Return the turbulence kinetic energy
virtual tmp<volScalarField> kt() const
{
    return kt_;
}

// - Return the turbulence kinetic energy
virtual tmp<volScalarField> ktRANS() const
{
    return ktRANS_;
}

// // - Return the total fluctuation kinetic energy
// - Return SGS kinetic energy
virtual tmp<volScalarField> k() const
{
    return tmp<volScalarField>
    (
        new volScalarField
        (
            IObject
            (
                "k",
                this->mesh_.time().timeName(),
                this->mesh_
            ),
            kl_+kt_
        )
    );
}

// - Return the total fluctuation kinetic energy dissipation rate
virtual tmp<volScalarField> epsilon() const
{
    return epsilon_;
}

// - Return the LES field indicator
virtual tmp<volScalarField> LESRegion() const;

// - Validate the turbulence fields after construction
// Update turbulence viscosity and other derived fields as requires
virtual void validate();

// - Solve the turbulence equations and correct the turbulence viscosity
virtual void correct();
};

// * * * * * //
} // End namespace LESModels
//} // End namespace incompressible
} // End namespace Foam

// * * * * * //
#ifdef NoRepository
#include "kkLOmegaIDDES.C"
#endif

```

```
// * * * * * // 415
#endif
// * * * * * //
```

## F.2 kkLOmegaIDDES.C

```
/*-----*/ 2
      F ield      | OpenFOAM: The Open Source CFD Toolbox
      O peration  |
      A nd        | Copyright (C) 2011–2016 OpenFOAM Foundation
      M anipulation |
/*-----*/ 7

License 9
  This file is part of OpenFOAM.

  OpenFOAM is free software: you can redistribute it and/or modify it
  under the terms of the GNU General Public License as published by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version. 12

  OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
  ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
  FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
  for more details. 17

  You should have received a copy of the GNU General Public License
  along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>. 22

/*-----*/

#include "kkLOmegaIDDES.H"
#include "fvOptions.H"
#include "bound.H"
#include "wallDist.H"
#include "volFields.H"

//#include "addToRunTimeSelectionTable.H"

// * * * * * //

namespace Foam
{
  // namespace incompressible
  // {
  namespace LESModels
  {
    // * * * * * Static Data Members * * * * * //
    // defineTypeNameAndDebug(kkLOmegaIDDES, 0);
    // addToRunTimeSelectionTable(RASModel, kkLOmegaIDDES, dictionary);
    // * * * * * Protected Member Functions * * * * * //

    /// IDDES blending function ///
    template<class BasicTurbulenceModel>
    tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::alpha() const 52
    {
      return max
      (
        0.25 - this->y_/static_cast<const volScalarField&>(IDDESDelta_.hmax()),
        scalar(-5)
      ); 57
    }

    template<class BasicTurbulenceModel>
    tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::ft 62
    (
      const volScalarField& magGradU,
      const volScalarField& nuts
    ) const 67
    {
      return tanh(pow3(sqr(this->ct_)*rd(this->nut_, magGradU)));
    }

    template<class BasicTurbulenceModel>
    tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::fl 72
    (
      const volScalarField& magGradU
    )
  }
}

```



```

) const
{
    return tanh(pow(sqr(this->cl_)*rd(this->nu(), magGradU), 10));
}
77

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::rd
(
    const volScalarField& nur,
    const volScalarField& magGradU
) const
{
    return min
    (
        nur
        /(
            max
            (
                magGradU,
                dimensionedScalar("SMALL", magGradU.dimensions(), SMALL)
            )
            *sqr(this->kappa_*this->y_)
        ),
        scalar(10)
    );
}
82
87
92
97
102

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::fd
(
    const volScalarField& magGradU,
    const volScalarField& nuts
) const
{
    return 1 - tanh(pow3(8*rd(this->nuEff(), magGradU)));
}
107
112

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::Psi(const volTensorField& gradU, const
volScalarField& Omega, const volScalarField& ResInf, const volScalarField& nuts) const
{
    const volScalarField alpha(this->alpha());
    const volScalarField expTerm(exp(sqr(alpha)));
    const volScalarField magGradU(mag(gradU));

    tmp<volScalarField> fStep = min(2*pow(expTerm, -9.0), scalar(1));
    tmp<volScalarField> fHyb = max(1 - fd(magGradU, nuts), fStep);
    //tmp<volScalarField> fLimiter = exp(-max((fINTMIX() - 0.90), scalar(0))*1000) ;
    tmp<volScalarField> fLimiterfSS = ( 1-exp(-10000*max(fSS(Omega) - 0.95, scalar(0))) ) ;
    //const volScalarField fLimiterf(1-exp(-max((fINTMIX() - 0.99), scalar(0))*1000) ) ;
    //tmp<volScalarField> fLimiter = min(fLimiterfSS, fLimiterf);
    //tmp<volScalarField> fSSRANS = exp(- max( (fSS(Omega) - 0.99) , scalar(0)) *1000) ;
    return
    (
        max
        (
            fHyb()
            ,
            1-fLimiterfSS()
        ) * (1-ResInf)
    );
}
117
122
127
132
137

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::dTilda
(
    const volTensorField& gradU,
    const volScalarField& S2,
    const volScalarField& kForm,
    const volScalarField& omegaForm,
    const volScalarField& Omega,
    const volScalarField& ResInf,
    const volScalarField& nuts
) const
{
    const volScalarField alpha(this->alpha());
    const volScalarField expTerm(exp(sqr(alpha)));
    const volScalarField magGradU(mag(gradU));

    tmp<volScalarField> fHill =
        2*(pos(alpha)*pow(expTerm, -11.09) + neg(alpha)*pow(expTerm, -9.0));

    tmp<volScalarField> fStep = min(2*pow(expTerm, -9.0), scalar(1));
    tmp<volScalarField> fLimiterfSS = 1-exp(-10000*max(fSS(Omega) - 0.95, scalar(0))) ;
    //const volScalarField fLimiterf(1-exp(-max((fINTMIX() - 0.99), scalar(0))*1000) ) ;
    //tmp<volScalarField> fLimiter = min(fLimiterfSS, fLimiterf);
    //tmp<volScalarField> fSSRANS = exp(- max( (fSS(Omega) - 0.99) , scalar(0)) *1000) ;
}
142
147
152
157
162

```

```

const volScalarField fHyb
(
    max
    (
        max
        (
            1 - fd(magGradU, nuts)
            ,
            fStep()
        )
        ,
        l-fLimiterfSS()
    ) * (1-ResInf)
);

tmp<volScalarField> fAmp = 1 - max(ft(magGradU, nuts), f1(magGradU));
tmp<volScalarField> f1_ = f1(magGradU);
tmp<volScalarField> ft_ = ft(magGradU, nuts);
tmp<volScalarField> expTermNu (exp(pow(this->nut_/this->nu(), 1.0/9.0)));
const volScalarField f_tmp4(expTermNu());
tmp<volScalarField> fINTMIXLim = 1-exp(-100*max(fINTMIX() - 0.9, scalar(0)));
const volScalarField f_tmp2(fINTMIXLim());

tmp<volScalarField> PsiRestore = this->CPsiRestore_*max(500*pow(expTermNu, -5.5)+1, scalar
(1)) * fINTMIXLim;
const volScalarField f_tmp3(PsiRestore());
tmp<volScalarField> fRestore = max(fHill - 1, scalar(0))*fAmp*this->CfRestore_;
tmp<volScalarField> lRANS = kForm / (omegaForm + this->omegaMin_);

if(this->runTime_.outputTime())
{
    volScalarField flimiter
    (
        IOobject
        (
            "flimiter",
            this->runTime_.timeName(),
            this->mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        fLimiterfSS()
    );
    Info << "Writing flimiter to file" << endl;
    flimiter.write();

    volScalarField f_Hyb
    (
        IOobject
        (
            "f_Hyb",
            this->runTime_.timeName(),
            this->mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        fHyb
    );
    Info << "Writing f_Hybrid to file" << endl;
    f_Hyb.write();

    const volScalarField f_tmp(fRestore());

    volScalarField f_Restore
    (
        IOobject
        (
            "f_Restore",
            this->runTime_.timeName(),
            this->mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        f_tmp
    );
    Info << "Writing f_Restore to file" << endl;
    f_Restore.write();
}

return max
(
    dimensionedScalar("SMALL", dimLength, SMALL),
    fHyb*(1 + fRestore()*PsiRestore())*lRANS
    + (1 - fHyb)*this->CDES_*this->delta()
);
}

```

```

/// END IDDES blending function ///
257

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::fv
(
    const volScalarField& Ret
262
) const
{
    return(1.0 - exp(-sqrt(Ret)/Av_));
}
267

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::fINTMIX() const
272
{
    return
    (
        min
        (
            ktRANS_/(Cint_*(kl_ + ktRANS_ + this->kMin_)),
            dimensionedScalar("1.0", dimless, 1.0)
277
        )
    );
}
282

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::fINTRANS() const
{
    return
    (
        min
        (
            ktRANS_/(Cint_*(klRANS_ + ktRANS_ + this->kMin_)),
            dimensionedScalar("1.0", dimless, 1.0)
287
        )
    );
}
292

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::fSS(const volScalarField& Omega) const
297
{
    return(exp(-sqr(this->Css_ * this->nu()*Omega/(ktRANS_ + this->kMin_)));
}
302

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::Cmu(const volScalarField& S, const
volScalarField& omega) const
307
{
    return(1.0/(A0_ + As_*(S/(omega + this->omegaMin_)));
}

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::BetaTS
312
(
    const volScalarField& ReOmega
) const
{
    return(scalar(1) - exp(-sqr(max(ReOmega - CtsCrit_, scalar(0))/Ats_));
}
317

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::fTauL
322
(
    const volScalarField& lambdaEff,
    const volScalarField& ktL,
    const volScalarField& Omega
) const
327
{
    return
    (
        scalar(1)
        - exp
        (
            -CtauL_*ktL
            /
            (
                sqr
                (
                    lambdaEff*Omega
                    + dimensionedScalar
                    (
                        "ROOTVSMALL",
                        dimLength*inv(dimTime),
332
                    )
                )
            )
        )
    );
}
337
342

```

```

        ROOTVSMALL
    )
)
);
}
template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::alphaT
(
    const volScalarField& lambdaEff,
    const volScalarField& fv,
    const volScalarField& ktS
) const
{
    return(fv*CmuStd_*sqrt(ktS)*lambdaEff);
}
template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::f0omega
(
    const volScalarField& lambdaEff,
    const volScalarField& lambdaT
) const
{
    return
    (
        scalar(1)
        - exp
        (
            -0.41
            *pow4
            (
                lambdaEff
                / (
                    lambdaT
                    + dimensionedScalar
                    (
                        "ROTVSMALL",
                        lambdaT.dimensions(),
                        ROOTVSMALL
                    )
                )
            )
        )
    );
}
template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::phiBP(const volScalarField& Omega) const
{
    return
    (
        min
        (
            max
            (
                ktRANS_/this->nu()
                / (
                    Omega
                    + dimensionedScalar
                    (
                        "ROTVSMALL",
                        Omega.dimensions(),
                        ROOTVSMALL
                    )
                )
            )
            - CbpCrit_,
            scalar(0)
        ),
        scalar(50.0)
    );
}
template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::phiNAT
(
    const volScalarField& ReOmega,
    const volScalarField& fNatCrit
) const
{
    return
    (
        max
        (
            ReOmega
            - CnatCrit_
            / (

```

```

        fNatCrit + dimensionedScalar("ROTVSMALL", dimless, ROOTVSMALL)
    ),
    scalar(0)
);
}
437

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::D
(
const volScalarField& k
) const
{
    return this->nu()*magSqr(fvc::grad(sqrt(k)));
}
442
447

template<class BasicTurbulenceModel>
void kkLOmegaIDDES<BasicTurbulenceModel>::correctNut()
{
    // Currently this function is not implemented due to the complexity of
    // evaluating nut. Better calculate nut at the end of correct()
    NotImplemented;
}
452
457

// * * * * * Constructors * * * * * //
462

template<class BasicTurbulenceModel>
kkLOmegaIDDES<BasicTurbulenceModel>::kkLOmegaIDDES
(
    const alphaField& alpha,
    const rhoField& rho,
    const volVectorField& U,
    const surfaceScalarField& alphaRhoPhi,
    const surfaceScalarField& phi,
    const transportModel& transport,
    const word& propertiesName,
    const word& type
)
:
LESeddyViscosity<BasicTurbulenceModel>
(
    type,
    alpha,
    rho,
    U,
    alphaRhoPhi,
    phi,
    transport,
    propertiesName
),
kappa_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "kappa",
        this->coeffDict_,
        0.41
    )
),
A0_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "A0",
        this->coeffDict_,
        4.04
    )
),
As_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "As",
        this->coeffDict_,
        2.12
    )
),
Av_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Av",
        this->coeffDict_,
        6.75
    )
),
)
467
472
477
482
487
492
497
502
507
512
517
522

```

```

Abp_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Abp",
        this->coeffDict_,
        0.6
    )
),
Anat_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Anat",
        this->coeffDict_,
        200
    )
),
Ats_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Ats",
        this->coeffDict_,
        200
    )
),
CbpCrit_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "CbpCrit",
        this->coeffDict_,
        1.2
    )
),
Cnc_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Cnc",
        this->coeffDict_,
        0.1
    )
),
CnatCrit_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "CnatCrit",
        this->coeffDict_,
        1250
    )
),
Cint_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Cint",
        this->coeffDict_,
        0.75
    )
),
CtsCrit_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "CtsCrit",
        this->coeffDict_,
        1000
    )
),
CrNat_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "CrNat",
        this->coeffDict_,
        0.02
    )
),
C11_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C11",
        this->coeffDict_,
        3.4e-6
    )
),

```

```

C12_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "C12",
    this->coeffDict_,
    1.0e-10
  )
),
CR_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "CR",
    this->coeffDict_,
    0.12
  )
),
CalphaTheta_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "CalphaTheta",
    this->coeffDict_,
    0.035
  )
),
Css_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "Css",
    this->coeffDict_,
    1.5
  )
),
CssBP_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "CssBP",
    this->coeffDict_,
    1.5
  )
),
CtauL_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "CtauL",
    this->coeffDict_,
    4360
  )
),
Cw1_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "Cw1",
    this->coeffDict_,
    0.44
  )
),
Cw2_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "Cw2",
    this->coeffDict_,
    0.92
  )
),
Cw3_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "Cw3",
    this->coeffDict_,
    0.3
  )
),
CwR_
(
  dimensioned<scalar >::lookupOrAddToDict
  (
    "CwR",
    this->coeffDict_,
    1.5
  )
),
),

```

```

Clambda_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Clambda",
        this->coeffDict_,
        2.495
    )
),
CmuStd_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "CmuStd",
        this->coeffDict_,
        0.09
    )
),
Prtheta_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Prtheta",
        this->coeffDict_,
        0.85
    )
),
Sigmak_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Sigmak",
        this->coeffDict_,
        1
    )
),
Sigmax_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Sigmax",
        this->coeffDict_,
        1.17
    )
),
CDES_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "CDES",
        this->coeffDict_,
        0.65
    )
),
CPsiRestore_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "CPsiRestore",
        this->coeffDict_,
        1.0
    )
),
CfRestore_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "CfRestore",
        this->coeffDict_,
        1.0
    )
),
alpha_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "alpha",
        this->coeffDict_,
        1
    )
),
fwStar_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "fwStar",
        this->coeffDict_,

```



```

        0.424
    )
),
cl_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "cl",
        this->coeffDict_,
        3.55
    )
),
ct_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "ct",
        this->coeffDict_,
        1.63
    )
),
IDDESDelta_(refCast<IDDESDelta>(this->delta_())),
omega_
(
    IOobject
    (
        IOobject::groupName("omega", U.group()),
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    this->mesh_
),
omegaRANS_
(
    IOobject
    (
        IOobject::groupName("omegaRANS", U.group()),
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    this->mesh_
),
kl_
(
    IOobject
    (
        IOobject::groupName("kl", U.group()),
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    this->mesh_
),
klRANS_
(
    IOobject
    (
        IOobject::groupName("klRANS", U.group()),
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    this->mesh_
),
kt_
(
    IOobject
    (
        IOobject::groupName("kt", U.group()),
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    this->mesh_
),
ktRANS_
(
    IOobject
    (
        IOobject::groupName("ktRANS", U.group()),
        this->runTime_.timeName(),

```

```

        this->mesh_ ,
        IOobject :: MUST_READ ,
        IOobject :: AUTO_WRITE
    ),
    this->mesh_
),
uFluct_
(
    IOobject
    (
        "uFluct",
        this->runTime_.timeName(),
        this->mesh_ ,
        IOobject :: NO_READ ,
        IOobject :: AUTO_WRITE
    ),
    this->mesh_ ,
    dimensionedVector
    (
        "uFluct",
        dimVelocity ,
        vector :: zero
    )
),
alphaFractionField_
(
    IOobject
    (
        "alphaFractionField",
        this->runTime_.timeName(),
        this->mesh_ ,
        IOobject :: NO_READ ,
        IOobject :: NO_WRITE
    ),
    this->mesh_ ,
    dimensionedScalar
    (
        "alphaFractionField",
        dimless ,
        1
    )
),
/* epsilon_
(
    IOobject
    (
        "epsilon",
        this->runTime_.timeName(),
        this->mesh_
    ),
    kt_*omega_ + D(kl_) + D(kt_)
),
*/
epsilon_
(
    IOobject
    (
        "epsilon",
        this->runTime_.timeName(),
        this->mesh_ ,
        IOobject :: NO_READ ,
        IOobject :: AUTO_WRITE
    ),
    this->mesh_ ,
    dimensionedScalar
    (
        "epsilon",
        dimVelocity ,
        0
    )
),
ranGen_(label(0)),
y_(wallDist :: New(this->mesh_).y())
{
    bound(kt_ , this->kMin_);
    bound(ktRANS_ , this->kMin_);
    bound(kl_ , this->kMin_);
    bound(klRANS_ , this->kMin_);
    bound(omega_ , this->omegaMin_);
    bound(omegaRANS_ , this->omegaMin_);
    // bound(epsilon_ , this->epsilonMin_);

    if (type == typeName)
    {
        // Evaluating nut_ is complex so start from the field read from file
        this->nut_.correctBoundaryConditions();
    }
}

```

```

        this->printCoeffs (type);
    }
}
977

// * * * * * Member Functions * * * * * //
982
template<class BasicTurbulenceModel>
bool kkLOmegaIDDES<BasicTurbulenceModel>::read()
{
    if (LESddyViscosity<BasicTurbulenceModel>::read())
    {
        A0_ .readIfPresent (this->coeffDict ());
        As_ .readIfPresent (this->coeffDict ());
        Av_ .readIfPresent (this->coeffDict ());
        Abp_ .readIfPresent (this->coeffDict ());
        Anat_ .readIfPresent (this->coeffDict ());
        Abp_ .readIfPresent (this->coeffDict ());
        Ats_ .readIfPresent (this->coeffDict ());
        CbpCrit_ .readIfPresent (this->coeffDict ());
        Cnc_ .readIfPresent (this->coeffDict ());
        CnatCrit_ .readIfPresent (this->coeffDict ());
        Cint_ .readIfPresent (this->coeffDict ());
        CtsCrit_ .readIfPresent (this->coeffDict ());
        CrNat_ .readIfPresent (this->coeffDict ());
        C11_ .readIfPresent (this->coeffDict ());
        C12_ .readIfPresent (this->coeffDict ());
        CR_ .readIfPresent (this->coeffDict ());
        CalphaTheta_ .readIfPresent (this->coeffDict ());
        Css_ .readIfPresent (this->coeffDict ());
        CssBP_ .readIfPresent (this->coeffDict ());
        CtauL_ .readIfPresent (this->coeffDict ());
        Cw1_ .readIfPresent (this->coeffDict ());
        Cw2_ .readIfPresent (this->coeffDict ());
        Cw3_ .readIfPresent (this->coeffDict ());
        CwR_ .readIfPresent (this->coeffDict ());
        Clambda_ .readIfPresent (this->coeffDict ());
        CmuStd_ .readIfPresent (this->coeffDict ());
        Prtheta_ .readIfPresent (this->coeffDict ());
        Sigmak_ .readIfPresent (this->coeffDict ());
        Sigmax_ .readIfPresent (this->coeffDict ());
        cl_ .readIfPresent (this->coeffDict ());
        ct_ .readIfPresent (this->coeffDict ());
        CDES_ .readIfPresent (this->coeffDict ());
        CPsiRestore_ .readIfPresent (this->coeffDict ());
        alpha_ .readIfPresent (this->coeffDict ());
        return true;
    }
    else
    {
        return false;
    }
}
1022

template<class BasicTurbulenceModel>
tmp<volScalarField> kkLOmegaIDDES<BasicTurbulenceModel>::LESRegion() const
{
    // const volScalarField chi (this->chi ());
    // const volScalarField fv1 (this->fv1 (chi));
    1032

    tmp<volScalarField> tLESRegion
    (
        new volScalarField
        (
            IObject
            (
                "DES::LESRegion",
                this->mesh_.time().timeName(),
                this->mesh_,
                IObject::NO_READ,
                IObject::NO_WRITE
            ),
            y_
        )
    );
    1042
    1047
    return tLESRegion;
    1052
}

template<class BasicTurbulenceModel>
void kkLOmegaIDDES<BasicTurbulenceModel>::validate()
{}
1057

template<class BasicTurbulenceModel>
void kkLOmegaIDDES<BasicTurbulenceModel>::correct()
1062

```

```

{
    if (!this->turbulence_)
    {
        return;
    }

    //eddyViscosity<incompressible::RASModel>::correct();
    LESeddyViscosity<BasicTurbulenceModel>::correct();

    tmp<volTensorField> tgradU(fvc::grad(this->U_));
    const volTensorField& gradU = tgradU();
    const volScalarField S2(2.0*magSqr(dev(symm(gradU))));
    const volScalarField Omega(sqrt(2.0)*mag(skew(gradU)));

    //////////////////////////////////////
    // Insert turbulent fluctuations
    // set up unity vector and scalar field
    volVectorField unityVF
    (
        IObject
        (
            "unityVF",
            this->runTime_.timeName(),
            this->mesh_,
            IObject::NO_READ,
            IObject::NO_WRITE
        ),
        this->mesh_,
        dimensionedVector
        (
            "unityVF",
            dimless,
            vector(1,1,1)
        )
    );
    volScalarField unitySF
    (
        IObject
        (
            "unitySF",
            this->runTime_.timeName(),
            this->mesh_,
            IObject::NO_READ,
            IObject::NO_WRITE
        ),
        this->mesh_,
        dimensionedScalar
        (
            "unitySF",
            dimless,
            1
        )
    );
};

const volScalarField fLimiter( 1 - exp(-1000*max((fSS(Omega) - 0.95), scalar(0))) );

// make inflow edge from fSS Limited

const volVectorField GradfLimiter = fvc::grad(fLimiter);
const volScalarField magGradfLimiter = mag( GradfLimiter)*dimensionedScalar("1",dimLength,1) ;

const volScalarField alpha(this->alpha());
const volScalarField expTerm(exp(sqr(alpha)));
const volScalarField fStep = min(2*pow(expTerm, -9.0), scalar(1));

const volScalarField EdgefINT
(
    (1-exp
    (
        -max
        (
            magGradfLimiter
            ,
            scalar(0)
        )
        ) * scalar(1)
    ) * (1 - fStep)*fLimiter
);

//derive smooth gradient from fINTMIX .
const volVectorField gradfINT = fvc::grad(fINTMIX());
const volScalarField magGradfINT = mag(gradfINT)+dimensionedScalar("1e-05",inv(dimLength),1e-05);

const volScalarField GradXfINT = gradfINT.component(0)/magGradfINT ;
const volScalarField GradYfINT = gradfINT.component(1)/magGradfINT ;

```

```

const volScalarField GradZfINT = gradfINT.component(2)/magGradfINT ;

const volScalarField limGradXfINT( 1-exp(-max((mag(GradXfINT) - 0.001) , scalar(0))*1000) );
const volScalarField limGradYfINT( 1-exp(-max((mag(GradYfINT) - 0.001) , scalar(0))*1000) );
const volScalarField limGradZfINT( 1-exp(-max((mag(GradZfINT) - 0.001) , scalar(0))*1000) );
1157

// derive velocity component
const volScalarField UxComp = this->U_.component(0)/( mag(this->U_)+dimensionedScalar("1e-15",
dimVelocity,1e-15) );
const volScalarField UyComp = this->U_.component(1)/( mag(this->U_)+dimensionedScalar("1e-15",
dimVelocity,1e-15) );
const volScalarField UzComp = this->U_.component(2)/( mag(this->U_)+dimensionedScalar("1e-15",
dimVelocity,1e-15) );
1162

const volScalarField LimUxGrad = 1-exp(-max(mag(UxComp) - 0.1 , scalar(0) ) *100);
const volScalarField LimUyGrad = 1-exp(-max(mag(UyComp) - 0.1 , scalar(0) ) *100);
const volScalarField LimUzGrad = 1-exp(-max(mag(UzComp) - 0.1 , scalar(0) ) *100);
1167

const volScalarField LimXInf = LimUxGrad*limGradXfINT*UxComp*GradXfINT;
const volScalarField LimYInf = LimUyGrad*limGradYfINT*UyComp*GradYfINT;
const volScalarField LimZInf = LimUzGrad*limGradZfINT*UzComp*GradZfINT;
1172

const volScalarField EdgeInflow
(
    sqrt( sqrt(max( LimXInf , scalar(0))) + sqrt(max( LimYInf , scalar(0))) + sqrt(max( LimZInf ,
    scalar(0))) )
    * EdgefINT
);
1177

const volScalarField ResInf( (1-exp(-EdgeInflow*1e5) ) );

// calculate fraction Alpha fluctuations
const dimensionedScalar deltaT(this->mesh().time().deltaT());
const volScalarField alphaFraction(unitySF*this->alpha_);
alphaFractionField_ = unitySF * this->alpha_;
1182

const volScalarField uPrime
(
    sqrt(2*(k1_)/3) // for 3D (3*3) for 2D (3*2)
);
1192

// Correction-factor to compensate for the loss of RMS fluctuation
// due to the temporal correlation introduced by the alpha parameter.
//const volScalarField rmsCorr = sqrt(12*(2*alphaFraction - sqrt(alphaFraction)))
//((alphaFraction + dimensionedScalar("1e-15", dimless, 1e-15) );
// calculate turbulent fluctuation.
// generate random fluctuations varyig between 0 and 1
volScalarField randomField = unitySF * dimensionedScalar("0", dimless, 0);
forAll(randomField.internalField(), i)
{
    randomField[i] = ranGen_.scalar01();
}
//reduce (randomField, sumOp<scalarField>());
1197

//uFluctOld_ = uFluct;
//uFluct_ = (1-uFluctOld_) + alphaFraction*ResInf*(randomField-dimensionedScalar("0.5", dimless, 0.5))
// *rmsCorr*uPrime*unityVF;
//uFluct_ = alphaFraction*ResInf*(randomField-dimensionedScalar("0.5", dimless, 0.5))*rmsCorr*uPrime*
// unityVF;
uFluct_ = alphaFraction*ResInf*(randomField-dimensionedScalar("0.5", dimless, 0.5))*uPrime*unityVF;
1202

if(this->runTime_.outputTime())
{
1207

/*
volVectorField gradFINT
(
    IOobject
    (
        "gradFINT",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    gradfINT
);
Info << "gradFINT" << endl;
gradFINT.write();
1212

*/
volScalarField fINTMix
(
    IOobject
    (
        "fINTMix",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
1217
1222
1227
1232
1237

```

```

        IObject::AUTO_WRITE
    ),
    fINTMIX ()
);
Info << "fINTMix" << endl;
fINTMix.write();
volScalarField EdgeFINT
(
    IObject
    (
        "EdgeFINT",
        this->runTime_.timeName(),
        this->mesh_,
        IObject::NO_READ,
        IObject::AUTO_WRITE
    ),
    EdgefINT
);
Info << "EdgeFINT" << endl;
EdgeFINT.write();

volScalarField ResINF
(
    IObject
    (
        "ResINF",
        this->runTime_.timeName(),
        this->mesh_,
        IObject::NO_READ,
        IObject::AUTO_WRITE
    ),
    ResInf
);
Info << "ResINF" << endl;
ResINF.write();
}
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

const volScalarField lambdaTRANS(sqrt(ktRANS_)/(omegaRANS_ + this->omegaMin_));
const volScalarField lambdaEffRANS(min(Clambd*_y_, lambdaTRANS));

const volScalarField lambdaT(sqrt(kt_)/(omega_ + this->omegaMin_));
const volScalarField lambdaEff(min(Clambd*_y_, lambdaT));

// fw based on rans equations
const volScalarField fw
(
    pow
    (
        lambdaEff
        /(lambdaT + dimensionedScalar("SMALL", dimLength, ROOTVSMALL)),
        2.0/3.0
    )
);

const volScalarField fwRANS
(
    pow
    (
        lambdaEffRANS
        /(lambdaTRANS + dimensionedScalar("SMALL", dimLength, ROOTVSMALL)),
        2.0/3.0
    )
);

const volScalarField fINTfnuHybrid
(
    max
    (
        fINTRANS() * fv(sqrt(fw)*kt_/this->nu()/(omega_ + this->omegaMin_))
        ,
        1-this->Psi(gradU, Omega, ResInf, this->nut_)
    )
);

const volScalarField fwHybrid
(

```

```

        max
        (
            fw
            ,
            l~this->Psi(gradU, Omega, ResInf, this->nut_)
        )
    );
    const volScalarField ktS(fSS(Omega)*fwHybrid*kt_);
    // const volScalarField ktS(fSS(Omega)*fw*kt_);
    const volScalarField nuts
    (
        fINTfnuHybrid * Cmu(sqrt(S2), omega_)
        *sqrt(ktS)*lambdaEff
    );
    const volScalarField Pkt(nuts*S2);
    const volScalarField ktSRANS(fSS(Omega)*fwRANS*ktRANS_);
    const volScalarField nutsRANS
    (
        fv(sqrt(fwRANS)*ktRANS_/this->nu()/(omegaRANS_ + this->omegaMin_))
        *fINTRANS()
        *Cmu(sqrt(S2), omegaRANS_)*sqrt(ktSRANS)*lambdaEffRANS
    );
    const volScalarField PktRANS(nutsRANS*S2);
    const volScalarField ktLRANS(ktRANS_ - ktSRANS);
    const volScalarField ktL(kt_ - ktS);
    const volScalarField ReOmega(sqrt(y_)*Omega/this->nu());
    const volScalarField nutLRANS
    (
        min
        (
            C11_*fTaul(lambdaEffRANS, ktLRANS, Omega)*Omega*sqr(lambdaEffRANS)
            *sqrt(ktLRANS)*lambdaEffRANS/this->nu()
            + C12_*BetaTS(ReOmega)*ReOmega*sqr(y_)*Omega
            ,
            0.5*(klRANS_ + ktLRANS)/(sqrt(S2) + this->omegaMin_)
        )
    );
    const volScalarField nutl
    (
        min
        (
            C11_*fTaul(lambdaEff, ktL, Omega)*Omega*sqr(lambdaEff)
            *sqrt(ktL)*lambdaEff/this->nu()
            + C12_*BetaTS(ReOmega)*ReOmega*sqr(y_)*Omega
            ,
            0.5*(kl_ + ktL)/(sqrt(S2) + this->omegaMin_)
        )
    );
    const volScalarField Pkl(nutl*S2);
    const volScalarField PklRANS(nutLRANS*S2);
    const volScalarField fnuHybrid
    (
        max
        (
            fv(sqrt(fw)*kt_/this->nu()/(omega_ + this->omegaMin_))
            ,
            l~this->Psi(gradU, Omega, ResInf, nuts)
        )
    );
    const volScalarField alphaTEff
    (
        alphaT(lambdaEff, fnuHybrid, ktS)
    );
    const volScalarField alphaTEffRANS
    (
        alphaT(lambdaEffRANS, fv(sqrt(fwRANS)*ktRANS_/this->nu()/(omegaRANS_ + this->omegaMin_)),
            ktSRANS)
    );
    const dimensionedScalar fwMin("SMALL", dimless, ROOTVSMALL);
    const volScalarField fNatCrit(1.0 - exp(-Cnc_*sqrt(klRANS_)*y_/this->nu()));
    // By pass source term divided by kl_
    const volScalarField RbpRANS

```

```

(
    CR_*(1.0 - exp(-phiBP(Omega())/Abp_))*omegaRANS_
    /(fwRANS + fwMin)
);
// By pass source term divided by kl_
const volScalarField Rbp
(
    CR_*(1.0 - exp(-phiBP(Omega())/Abp_))*omega_
    /(fw + fwMin)
);
// Natural source term divided by kl_
const volScalarField Rnat
(
    CrNat_*(1.0 - exp(-phiNAT(ReOmega, fNatCrit)/Anat_))*Omega
);

const surfaceScalarField& Phi = this->phi_;
const dimensionedScalar& KMin = this->kMin_;

omega_.boundaryFieldRef().updateCoeffs();

// Turbulence specific dissipation rate equation
tmp<fvScalarMatrix> omegaEqn
(
    fvm::ddt(omega_)
    + fvm::div(Phi, omega_)
    - fvm::laplacian(DomegaEff(alphaTEff), omega_)
    ==
    Cw1_*Pkt*omega_/(kt_ + KMin)
    - fvm::SuSp
    (
        (1.0 - CwR_/(fw + fwMin))*kl_*(Rbp + Rnat)/(kt_ + KMin)
        , omega_
    )
    - fvm::Sp(Cw2_*sqr(fw)*omega_, omega_)
    + (
        Cw3_*fOmega(lambdaEff, lambdaT)*alphaTEff*sqr(fw)*sqrt(kt_)
    )()()/pow3(y_)
);
omegaEqn.ref().relax();
omegaEqn.ref().boundaryManipulate(omega_.boundaryFieldRef());

solve(omegaEqn);
bound(omega_, this->omegaMin_);

omegaRANS_.boundaryFieldRef().updateCoeffs();
// Turbulence RANS specific dissipation rate equation
tmp<fvScalarMatrix> omegaRANSEqn
(
    fvm::ddt(omegaRANS_)
    + fvm::div(Phi, omegaRANS_)
    - fvm::laplacian(DomegaEff(alphaTEffRANS), omegaRANS_)
    ==
    Cw1_*PktRANS*omegaRANS_/(ktRANS_ + KMin)
    - fvm::SuSp
    (
        (1.0 - CwR_/(fwRANS + fwMin))*klRANS_*(RbpRANS + Rnat)/(ktRANS_ + KMin)
        , omegaRANS_
    )
    - fvm::Sp(Cw2_*sqr(fwRANS)*omegaRANS_, omegaRANS_)
    + (
        Cw3_*fOmega(lambdaEffRANS, lambdaTRANS)*alphaTEffRANS*sqr(fwRANS)*sqrt(ktRANS_)
    )()()/pow3(y_)
);
omegaRANSEqn.ref().relax();
omegaRANSEqn.ref().boundaryManipulate(omegaRANS_.boundaryFieldRef());

solve(omegaRANSEqn);
bound(omegaRANS_, this->omegaMin_);

const volScalarField Psi(this->Psi(gradU, Omega, ResInf, nuts));
const volScalarField Dl(Psi*D(kl_));
const volScalarField DlRANS(D(klRANS_));
const volScalarField Dt(Psi*D(kt_));
const volScalarField DtRANS(D(ktRANS_));

// Laminar kinetic energy equation
tmp<fvScalarMatrix> klEqn
(
    fvm::ddt(kl_)
    + fvm::div(Phi, kl_)
    - fvm::laplacian(this->nu(), kl_)
    ==

```



```

    Pk1
    - fvm::Sp(Rbp + Rnat + Dl/(kl_ + KMin), kl_)
);
1507

klEqn.ref().relax();
klEqn.ref().boundaryManipulate(kl_.boundaryFieldRef());
1512

solve(klEqn);

bound(kl_, this->kMin_);
1517

tmp<fvScalarMatrix> klRANSEqn
(
    fvm::ddt(klRANS_)
    + fvm::div(Phi, klRANS_)
    - fvm::laplacian(this->nu(), klRANS_)
    ==
    Pk1RANS
    - fvm::Sp(RbpRANS + Rnat + DlRANS/(klRANS_ + KMin), klRANS_)
);
1522

klRANSEqn.ref().relax();
klRANSEqn.ref().boundaryManipulate(klRANS_.boundaryFieldRef());
1527

solve(klRANSEqn);
bound(klRANS_, this->kMin_);
1532

const volScalarField dTildaKt( this->dTilda(gradU, S2, sqrt(kt_), omega_, Omega, ResInf, nuts)
);
const volScalarField destruction( sqrt(kt_)/dTildaKt );
const volScalarField deltaDestruction( destruction - omega_ );
// Turbulent kinetic energy equation
tmp<fvScalarMatrix> ktEqn
(
    fvm::ddt(kt_)
    + fvm::div(Phi, kt_)
    - fvm::laplacian(DkEff(alphaTEff), kt_)
    ==
    Pkt
    + kl_ * (Rbp + Rnat)
    - fvm::Sp(deltaDestruction + Dt/(kt_ + KMin), kt_)
);
1542

ktEqn.ref().relax();
ktEqn.ref().boundaryManipulate(kt_.boundaryFieldRef());
1547

solve(ktEqn);
bound(kt_, KMin);
1552

// Turbulent RANS kinetic energy equation
tmp<fvScalarMatrix> ktRANSEqn
(
    fvm::ddt(ktRANS_)
    + fvm::div(Phi, ktRANS_)
    - fvm::laplacian(DkEff(alphaTEffRANS), ktRANS_)
    ==
    PkRANS
    + klRANS_ * (RbpRANS + Rnat)
    - fvm::Sp(omegaRANS_ + DtRANS/(ktRANS_ + KMin), ktRANS_)
);
1557

ktRANSEqn.ref().relax();
ktRANSEqn.ref().boundaryManipulate(ktRANS_.boundaryFieldRef());
1562

solve(ktRANSEqn);
bound(ktRANS_, KMin);
1567

// Update total fluctuation kinetic energy dissipation rate
// epsilon_ = kt_*omega_ + Dt + Dl;
// bound(epsilon_, this->epsilonMin_);
1572

// Re-calculate turbulent viscosity
this->nut_ = nuts + nutl;
this->nut_.correctBoundaryConditions();
1577

if(this->runTime_.outputTime())
{
1582

volScalarField DeltaDestruction
(
    IObject
    (
        "DeltaDestruction",
        this->runTime_.timeName(),
        this->mesh_,
        IObject::NO_READ,
        IObject::AUTO_WRITE
    ),
    deltaDestruction
);
1587

Info << "Writing DeltaDestruction to file" << endl;
1592

```

```

DeltaDestruction.write();
volScalarField RBP
(
    IOobject
    (
        "RBP",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    (1.0 - exp(-phiBP(Omega())/Abp_))
);
Info << "Writing large scale bypass production (RBP*kl) to file" << endl;
RBP.write();

volScalarField RNAT
(
    IOobject
    (
        "RNAT",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    (1.0 - exp(-phiNAT(ReOmega, fNatCrit)/Anat_))
);
Info << "Writing large scale natural production (RNAT) to file" << endl;
RNAT.write();

volScalarField PkL
(
    IOobject
    (
        "PkL",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    PkL
);
Info << "Writing laminar production (PkL) to file" << endl;
PkL.write();

volScalarField PkT
(
    IOobject
    (
        "PkT",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    PkT
);
Info << "Writing turbulent production (PkT) to file" << endl;
PkT.write();

volScalarField nutL
(
    IOobject
    (
        "nutL",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    nutL
);
Info << "Writing large scale eddy viscosity (nutL) to file" << endl;
nutL.write();

volScalarField nutS
(
    IOobject
    (
        "nutS",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    nuts
);

```

```

);
Info << "Writing small scale eddy viscosity (nutS) to file" << endl;
nutS.write();
1687

    volScalarField nutLnat
    (
        IOobject
        (
            "nutLnat",
            this->runTime_.timeName(),
            this->mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        min
        (
            C12_*BetaTS(ReOmega)*ReOmega*sqr(y_)*Omega
            ,
            0.5*(k1_ + ktLRANS)/(sqrt(S2) + this->omegaMin_)
        )
    );
Info << "Writing laminar natural eddy viscosity (nutLnat) to file" << endl;
nutLnat.write();
1692
1697
1702

volScalarField dTildaKt
(
    IOobject
    (
        "dTildaKt",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    dTildaKt
);
Info << "Writing turbulent SGS length scale (dTildaKt) to file" << endl;
dTildaKt.write();
1712
1717
1722

volScalarField DT
(
    IOobject
    (
        "DT",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    Dt
);
Info << "Writing near wall turbulent dissipation (DT) to file" << endl;
DT.write();
1727
1732

volScalarField DL
(
    IOobject
    (
        "DL",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    Dl
);
Info << "Writing near wall laminar dissipation (DL) to file" << endl;
DL.write();
1737
1742
1747
1752

volScalarField kts
(
    IOobject
    (
        "kts",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    ktS
);
Info << "Writing small-scale turbulent energy (kts) to file" << endl;
kts.write();
1757
1762
1767

volScalarField lambdat
(
    IOobject

```

```

        (
            "lambdat",
            this->runTime_.timeName(),
            this->mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        lambdat
    );
Info << "Writing turbulent length scale (lambdat) to file" << endl;
lambdat.write();
volScalarField lambdaeff
(
    IOobject
    (
        "lambdaeff",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    lambdaEff
);
Info << "Writing effective turbulent length scale (lambdaeff) to file" << endl;
lambdaeff.write();
/*
volScalarField Delta
(
    IOobject
    (
        "Delta",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    this->delta()
);
Info << "Writing filter width (delta) to file" << endl;
Delta.write();
*/
volScalarField fSs
(
    IOobject
    (
        "fSs",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    fSS(Omega)
);
Info << "Writing shear sheltering function (fSs) to file" << endl;
fSs.write();
volScalarField fvisc
(
    IOobject
    (
        "fvisc",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    fv(sqrt(fw)*kt_/this->nu()/(omega_ + this->omegaMin_))
);
Info << "Writing viscous damping term (fvisc) to file" << endl;
fvisc.write();
volScalarField fviscRANS
(
    IOobject
    (
        "fviscRANS",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    fv(sqrt(fwRANS)*ktRANS_/this->nu()/(omegaRANS_ + this->omegaMin_))
);
Info << "Writing viscous damping term (fviscRANS) to file" << endl;
fviscRANS.write();

```

```

volScalarField fIntrANS
(
    IOobject
    (
        "fIntrANS",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    fINTRANS()
);
Info << "Writing RANS intermittency (fIntrANS) to file" << endl;
fIntrANS.write();

volScalarField cMU
(
    IOobject
    (
        "cMU",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    Cmu(sqrt(S2), omega_)
);
Info << "Writing cmu (cMU) to file" << endl;
cMU.write();

}

// * * * * * //

} // End namespace LESModels
} // End namespace Foam

// * * * * * //

```



