

Master's Thesis

# Leading Edge Erosion: A Review and Application of Modeling Strategies for the Investigation of Erosion-induced Effects on Wind Turbine Rotor Blades

under the supervision of

**Assistant Prof. Dipl.-Ing. Bernhard Semlitsch**

E302 - Institute of Energy Systems and Thermodynamics

submitted to the Faculty of Mechanical and Industrial Engineering

of the Technische Universität Wien

for the degree of Diplom-Ingenieur (Dipl.-Ing.)

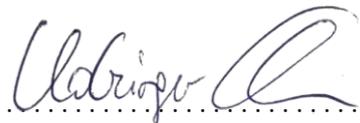
by

**Christoph Holzinger, BSc**

Matr.Nr. 01526455

Ortliebгasse 48/14, 1170 Wien

Wien, July 18, 2022



(Christoph Holzinger)

# Acknowledgements

I would like to thank Assistant Prof. Dipl.-Ing. Bernhard Semlitsch for his excellent supervision, without his advice and guidance, writing this thesis would not have been possible.

I would also like to thank Verbund for providing an operators' point of view.

Thanks to everyone who supported me during this time, especially to my girlfriend, Paige, for always believing in me. I would also like to mention my friends, Matthias and Jonathan, who at times provided much necessary relief.

The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC).

# Abstract

Wind energy as a renewable energy source has become increasingly important over the last two decades and plays a vital role in the transition to more sustainable energy. As wind turbine rotor blades are constantly exposed to weather phenomena, the surface near the leading edge erodes over time. Leading-edge erosion leads to a significant reduction in aerodynamic performance and, thus, losses in annual energy production. Various approaches to model erosion characteristics in computational fluid dynamics have been developed in the past few years.

This work encompasses two-dimensional numerical flow simulations around NACA 64-618 and NACA 63<sub>3</sub>-418 airfoils with different degrees of erosion severity using OpenFOAM and the validation of these results against experimental data. Heavily eroded airfoils are represented by a geometry-based model. A mathematical surface roughness-based model, implemented in OpenFOAM as part of this thesis, is used to represent slightly eroded airfoils. The steady-state flow simulation results agree well with experimental data and demonstrate the capability of both methodologies for predicting potential performance losses due to leading-edge erosion. The results of the transient simulations indicate that heavily eroded airfoils experience greater pressure fluctuation amplitudes on the surface, leading to increased loads and accelerated fatigue.

In addition, a single rotor blade of the WindPACT 1.5MW reference turbine is selected for both steady-state and transient three-dimensional flow simulations. The rough surface is placed within 10% of the leading edge in the outermost 40% of the blade, resulting in an 11% torque reduction compared to the undamaged blade. Sufficient results are also obtained from the transient simulations, where 1.2 ms of flow could be simulated with the available computational resources. This setup forms the basis for future investigations regarding the

## ABSTRACT

---

aeroelastic properties of the rotor blade. The work concludes with suggestions on how to combine the transient flow simulation and structural analysis using open-source software.

# Kurzfassung

Als erneuerbare Energiequelle hat Energie aus Windkraft in den letzten zwei Jahrzehnten zunehmend an Bedeutung gewonnen. Im Rahmen der Energiewende kommt der Windenergie zudem eine Schlüsselrolle zu. Die Rotorblätter einer Windkraftanlage sind den Umweltbedingungen permanent ausgesetzt, dadurch kommt es auf Dauer zu Erosionserscheinungen an der Vorderkante. Diese Schäden an der Vorderkante beeinträchtigen das aerodynamische Verhalten und können in der Folge zu erheblichen Leistungsverlusten führen. In der Vergangenheit sind verschiedene Ansätze entwickelt worden, um Erosionserscheinungen bei numerischen Strömungssimulationen zu berücksichtigen.

Diese Arbeit beinhaltet zweidimensionale Strömungssimulationen an NACA 64-618 und NACA 63<sub>3</sub>-418 Profilen mit unterschiedlichen Schadensausprägungen und die Validierung der Ergebnisse mit Messdaten. Die Berechnungen erfolgen mit der freien Software OpenFOAM. Rauigkeits- und Geometriebasierte Methoden werden zur Erosionsmodellierung eingesetzt. Das Rauigkeitsmodell, welches leichte Vorderkantenerosion als Rauigkeit mathematisch abbildet, wird im Zuge dieser Arbeit in OpenFOAM implementiert. Stationäre Simulationen liefern stimmige Ergebnisse für beide Modelle und zeigen ihre Eignung zur Vorhersage der Performanceverluste durch Erosion. Transiente Berechnungen legen erhöhte Druckschwankungen an der Oberfläche, und damit gesteigerte Materialermüdung stark erodierter Profile nahe.

Zusätzlich wird ein Rotorblatt der Referenzturbinen WindPACT 1.5MW für dreidimensionale Simulationen - stationäre und transiente - ausgewählt. Die äußersten 40% des Rotorblatts werden mit einer Rauigkeit von 101  $\mu\text{m}$  als leicht erodiert angenommen, mit dieser Konfiguration ergibt sich eine Reduktion des Drehmoments um 11%. Zufriedenstellende Ergebnisse liefern auch die instationären Simulationen; mit den verfügbaren Rech-

nerressourcen konnten 1.2 ms der Strömung simuliert werden. Damit wird eine Basis für zukünftige Untersuchungen hinsichtlich der aeroelastischen Eigenschaften des Rotorblatts geschaffen. Mit einer Anleitung zur Verbindung der transienten Strömungssimulation und der Strukturanalyse mit Open-Source-Software wird die Arbeit abgeschlossen.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Theory</b>	<b>4</b>
2.1	Blade Aerodynamics . . . . .	5
2.2	Flow Transition . . . . .	8
2.3	Leading-Edge Erosion . . . . .	10
2.4	Roughness Characterization . . . . .	13
<b>3</b>	<b>Governing Equations</b>	<b>15</b>
3.1	Shear Stress Transport Turbulence Model . . . . .	17
3.1.1	Note on BC for External Aerodynamic Flows . . . . .	18
3.2	Langtry-Menter $\gamma - \hat{Re}_{\theta t}$ Transition Model . . . . .	20
3.3	Amplification Roughness Model . . . . .	23
3.3.1	Modification to the SST-BC . . . . .	25
<b>4</b>	<b>The Amplification Roughness Model in OpenFOAM</b>	<b>26</b>
<b>5</b>	<b>Validation and 2D Results</b>	<b>34</b>
5.1	General Remarks . . . . .	34
5.2	Validation of a Smooth Airfoil . . . . .	39
5.3	Roughness Model . . . . .	47
5.3.1	Flat Plate Cases . . . . .	47
5.3.2	Airfoil Cases . . . . .	51
5.4	Delamination Model . . . . .	56
5.5	Model Comparison and Transient Results . . . . .	59

<b>6</b>	<b>3D Simulation</b>	<b>66</b>
6.1	Setting up the Simulation . . . . .	67
6.1.1	Grid Independence Study . . . . .	71
6.1.2	Setup of the Eroded Blade . . . . .	73
6.2	Results . . . . .	74
6.2.1	Steady State Simulations . . . . .	74
6.2.2	Transient Simulations . . . . .	79
6.3	Coupling the Simulation with FEA - Outline . . . . .	82
6.3.1	Translation of the FEA Model . . . . .	83
6.3.2	Setting up preCICE . . . . .	84
<b>7</b>	<b>Conclusion</b>	<b>87</b>
	<b>Bibliography</b>	<b>90</b>
<b>A</b>	<b>Estimation of the Discretization Error</b>	<b>99</b>
<b>B</b>	<b>Leading Edge Utility</b>	<b>102</b>
<b>C</b>	<b>Boundary Conditions OpenFOAM</b>	<b>106</b>
<b>D</b>	<b>Properties of the Rotor Blade</b>	<b>113</b>

# Nomenclature

## Latin symbols

$A$	surface area	$\text{m}^2$
$A_r$	amplification roughness	—
$C$	Courant number	—
$c$	chord length	$\text{m}$
$c_D$	drag coefficient	—
$c_F$	skin friction coefficient	—
$c_L$	lift coefficient	—
$c_M$	torque coefficient	—
$c_P$	power coefficient	—
$c_p$	pressure coefficient	—
$d_1$	first cell height	$\text{m}$
$e$	relative error	—
$E_x, E_y$	Young's modulus	$\text{GPa}$
$f$	frequency	$\text{Hz}$
$F_D$	drag force	$\text{N}$
$F_L$	lift force	$\text{N}$
$G_{xy}$	shear modulus	$\text{GPa}$
$h$	representative grid size	$\text{m}$
$I$	turbulence intensity	—
$k$	turbulent kinetic energy	$\text{m}^2/\text{s}^2$

## NOMENCLATURE

---

$k_r$	roughness height	m
$k_s$	equivalent sandgrain roughness height	m
$k_s^+$	dimensionless roughness parameter	—
$L$	reference length	m
$M$	torque	Nm
$Ma$	Mach number	—
$N$	total number	—
$P$	power	$\text{kgm}^2/\text{s}^3$
$p$	pressure	$\text{N}/\text{m}^2$
$P_1$	signal amplitude	—
$R$	convergence ratio	—
$Re$	Reynolds number	—
$Re_\nu$	vorticity/strain-rate Reynolds number	—
$Re_{\theta t}$	transition onset momentum thickness Reynolds number	—
$Re_\theta$	momentum thickness Reynolds number	—
$Re_{k_s}$	equivalent sand grain roughness Reynolds number	—
$Re_k$	roughness Reynolds number	—
$\hat{R}e_{\theta t}$	local transition onset momentum thickness Reynolds number	—
$r$	rotor radius	m
$r_{ij}$	refinement factor	—
$S$	strain-rate magnitude	1/s
$s_{ij}$	instantaneous strain-rate tensor	1/s
$t$	time	s
$u_\tau$	shear velocity	m/s
$\vec{u}$	velocity in vector notation	m/s
$u^+$	dimensionless velocity	—
$u_i$	velocity in tensor notation	m/s

## NOMENCLATURE

---

$u_r$	relative velocity	m/s
$u_\infty$	freestream velocity	m/s
$V$	volume	m <sup>3</sup>
$x, y, z$	spatial coordinates	—
$y^+$	dimensionless wall distance	—

### Greek symbols

$\alpha$	angle of attack	°
$\delta_{ij}$	Kronecker delta	—
$\epsilon_{ij}$	absolute error	—
$\gamma$	intermittency	—
$\lambda$	pressure gradient parameter	—
$\Lambda_k$	general density function	—
$\mu$	dynamic viscosity	kg/ms
$\mu_t$	dynamic eddy viscosity	kg/ms
$\nu$	kinematic viscosity	m <sup>2</sup> /s
$\nu_t$	kinematic eddy viscosity	m <sup>2</sup> /s
$\nu_{xy}$	Poisson's ratio	—
$\omega$	specific turbulent dissipation rate	1/s
$\Phi$	some quantity of interest	—
$\rho$	density	kg/m <sup>3</sup>
$\tau_w$	wall shear stress	N/m <sup>2</sup>
$\theta$	boundary layer momentum thickness	m
$\vartheta$	pitch angle	°
$\vec{\Omega}$	rotational vector	—

### Subscripts

1, 2, 3	grid indices
$\infty$	quantity in the free stream

*a* approximate

*ext* extrapolated

*n* normalized

### Superscripts

' fluctuating quantity

– temporal average

### Abbreviations

AEP Annual Energy Production

AMI Arbitrary Mesh Interface

BC Boundary Condition

CAD Computer Aided Design

CFD Computational Fluid Dynamics

ERCOFTAC European Research Community on Flow, Turbulence and Combustion

FEA Finite Element Analysis

FFT Fast Fourier Transform

FSI Fluid Structure Interaction

GCI Grid Convergence Index

HAWT Horizontal Axis Wind Turbine

LEE Leading Edge Erosion

NACA National Advisory Committee for Aeronautics

NREL National Renewable Energy Laboratory

NuMAD Numerical Manufacturing and Design

OpenFOAM Open Source Field Operation and Manipulation

SRF Single Rotating Frame

SST Shear Stress Transport

(U)RANS (Unsteady) Reynolds Averaged Navier Stokes

WindPACT Wind Partnership for Advanced Component Technology

# Chapter 1

## Introduction

Wind power plays an important role in increasing the sustainable energy supply. As of 2020, 743 GW of wind power capacity are installed, and a 4% annual growth of the industry is expected [1]. To further increase competitiveness, larger and more efficient turbines are being developed, and a reduction of operational (e.g. maintenance) costs is desired [2]. Surface degradation of the rotor blades is a challenge that affects both the development of new turbines and the maintenance of existing ones.



Figure 1.1: A severely eroded leading edge [3].

Rotor blades of wind turbines are constantly exposed to weather phenomena that depend on the turbine's location. The impact of rain, hail, insects, or other particles leads to erosion, which mostly appears on the leading edge in the outermost regions of the blade [4]. An example of severe, leading-edge erosion (LEE) is shown in Fig. 1.1. It is not hard to imagine that this kind of damage is unfavorable. It has been shown that LEE leads to a reduced aerodynamic performance, which consequently affects the annual energy production (AEP). Numerical studies point towards AEP losses of up to 10% for severe cases [5, 6, 7, 8]. Erosion-induced vibrations can “create undesirable loads that neither the turbine nor the blades were designed to carry” [3]. Understanding the loading cycles is crucial for damage prediction and remaining turbine lifetime assessment [9].

Mitigation strategies include the application of leading-edge protection, reducing the tip speed during extreme precipitation events [10], and repair. As the radius and, therefore, the tip speed of new turbines increases, leading-edge protection will become even more important in the future. Operators of existing wind parks have to assess the economic efficiency of repairing eroded rotor blades. However, understanding and predicting leading edge erosion and its effects are necessary to develop solutions further.

The objective of this thesis is to answer the research question if the damages on wind turbine blades can be modeled by numerical simulation. Furthermore, the assumption that LEE leads to increased rotor blade vibrations shall be investigated and answered. Most existing approaches to model LEE are based on either geometrical representation, i.e., modeling the erosion feature into the geometry, or mathematical representation. Geometrical representation is used to model the complete delamination of the outer layer over the leading edge. A suitable roughness model is introduced to represent the early stages of LEE mathematically. Such models are typically not included in commercial computational fluid dynamics (CFD) programs. Hence, the free, open-source CFD software OpenFOAM is chosen for the fluid-dynamic computations.

Chapter 2 contains a short introduction to wind turbines and blade aerodynamics. LEE is explained in more detail, and numerical modeling strategies are reviewed. Furthermore, flow transition and the importance of the roughness definition are discussed briefly. The turbulence closure, and the transition model based on it, are described in Chap. 3. Understanding them is necessary to properly follow the Amplification Roughness model, which is

outlined at the end of this chapter. The novel implementation of the Amplification Roughness model in OpenFOAM is outlined in Chap. 4.

Chapter 5 covers the validation of the numerical methods in general and the validation of the erosion models. Two well-investigated airfoil profiles are chosen for the validation process. The implementation of the Amplification Roughness model is tested thoroughly with flat plate test cases as well as airfoil profiles. Both erosion models are compared to clean simulations for various Reynolds numbers. Finally, the results of transient simulations are presented to identify differences in vibration behavior.

Chapter 6 deals with the setup of the simulation on a full-size rotor blade and the subsequent presentation of both steady-state and transient results. A possible way of simulating a fluid-structure interaction (FSI) problem using open-source software is also outlined here before some concluding remarks and suggestions for future work are given in Chap. 7.

# Chapter 2

## Theory

The basic working principles of wind turbines and blade aerodynamics are described at the beginning of this chapter. A definition of important parameters follows, which are used to characterize wind turbines and airfoils. After discussing the issue of LEE and its origins in more detail, literature concerning erosion modeling strategies in CFD is reviewed. Furthermore, this chapter deals with the laminar to turbulent boundary layer transition and the characterization of surface roughness.

## 2.1 Blade Aerodynamics

The most common type of wind turbine design in use today is the horizontal axis wind turbine (HAWT). Configurations with two or three blades exist. The object of this study is a more common three-bladed HAWT. A typical rotor blade hull is represented by several airfoil profiles that are swept along the span. Airfoils are described by their chord length  $c$  that connects the leading- and trailing edge, the thickness, and camber (Fig. 2.1).

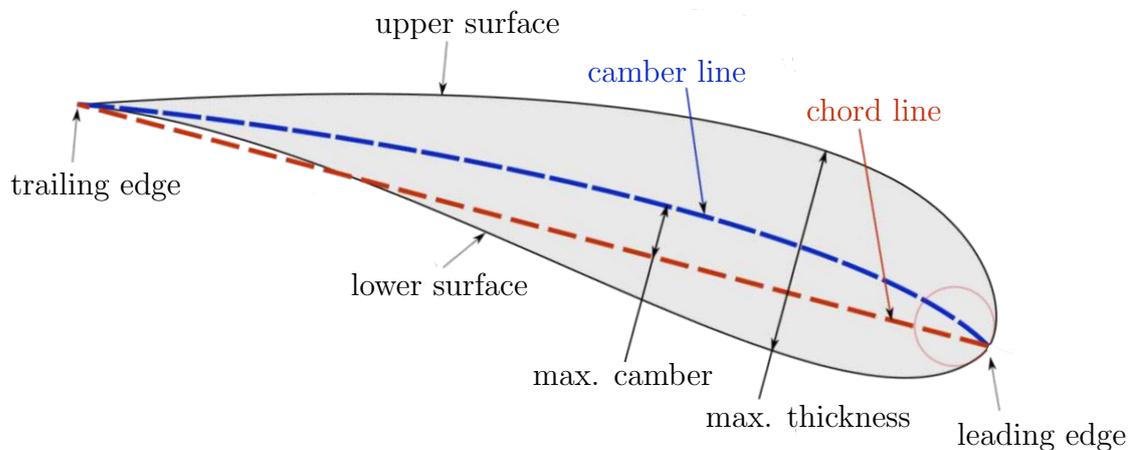


Figure 2.1: Airfoil nomenclature, adapted from [11].

The angle between the chord and the rotational plane is called pitch angle  $\vartheta$ . In pitch-controlled systems, it can be varied for power control by adjusting the angle of attack  $\alpha$ . While  $\vartheta$  is a quantity determined by design, the angle of attack  $\alpha$  is an aerodynamic quantity determined by the rotational speed and the free-stream velocity.

An airfoil that is exposed to a flow generates an aerodynamic force as depicted in Fig. 2.2. The force components, parallel and perpendicular to the relative velocity  $u_r$ , are termed drag  $F_D$  and lift  $F_L$  force, respectively.

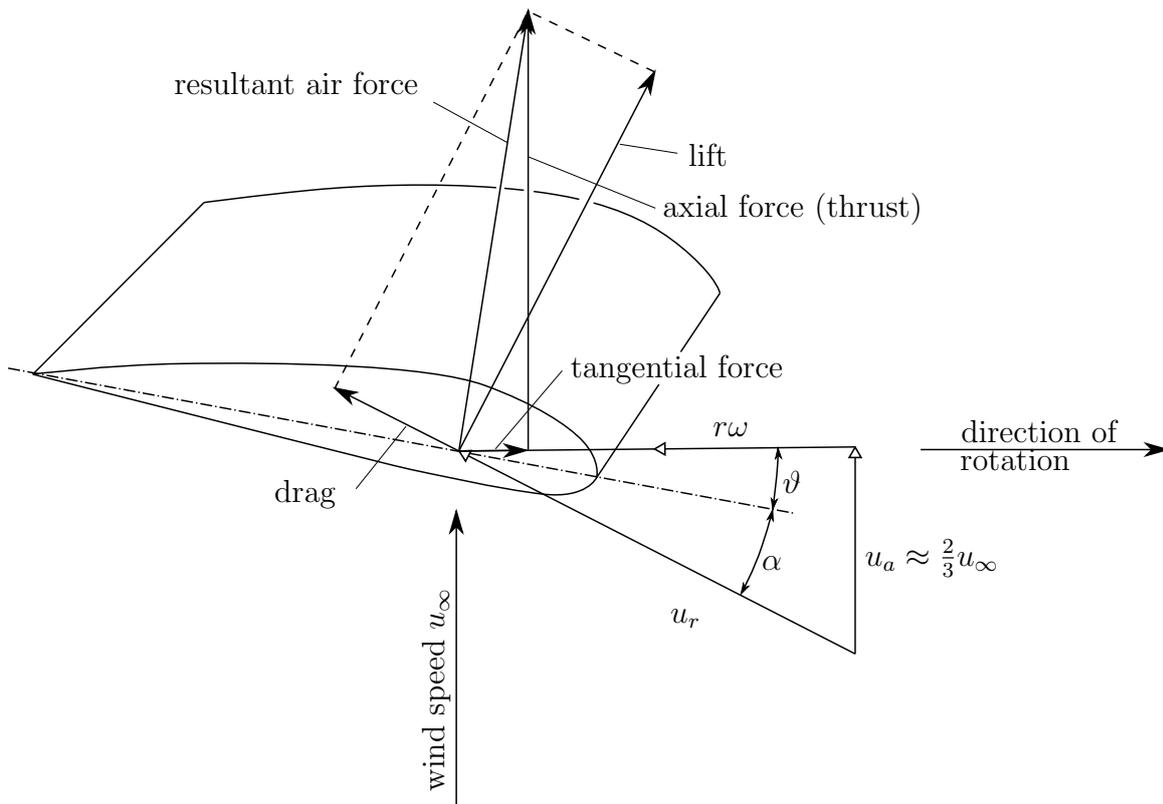


Figure 2.2: Flow conditions and aerodynamic forces on a blade element, adapted from [9].

Using the aerodynamic forces, one can define dimensionless coefficients:

$$c_D = \frac{F_D}{\frac{1}{2}\rho u_r^2 A}, \quad (2.1)$$

$$c_L = \frac{F_L}{\frac{1}{2}\rho u_r^2 A}, \quad (2.2)$$

where  $u_r$  is the relative velocity.  $A$  is a reference surface that is the product of  $c$  and the thickness in the spanwise direction<sup>1</sup>. The aerodynamic forces depend on the angle of attack  $\alpha$ . The Reynolds number  $Re$  is used to characterize the flow and is computed as:

$$Re = \frac{u_r c}{\nu}. \quad (2.3)$$

<sup>1</sup>Generally, the spanwise direction is perpendicular to the chordwise direction, along the length of the blade from root to tip.

Depending on the rotor size, Reynolds numbers at the blade tip are typically in a range from 1 to  $8 \cdot 10^6$  [9].

In aerodynamics, a no-slip condition is assumed on the airfoil surface that requires the flow velocity on the wall to be zero. A thin boundary layer with a high-velocity gradient is formed. Boundary layers can be laminar, turbulent or in transition (see Sec. 2.2). The fluid viscosity causes skin friction drag, which is highest in turbulent boundary layers. Again, a dimensionless coefficient, the skin friction coefficient  $c_F$  is defined:

$$c_F = \frac{\tau_w}{\frac{1}{2}\rho u_\infty^2}, \quad (2.4)$$

where  $\tau_w$  is the skin shear stress on the surface. Using the skin friction coefficient, one can determine the location where the boundary layer transitions.

The aerodynamic performance can be evaluated by looking at the pressure distribution over an airfoil. It is typically expressed in terms of the pressure coefficient  $c_p$ :

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho u_\infty^2 A}. \quad (2.5)$$

Moving on to coefficients concerning the entire rotor, useful factors are the torque coefficient  $c_M$  and the power coefficient  $c_P$ , where  $R$  and  $A$  are the rotor radius and swept area by the rotor, respectively:

$$c_M = \frac{M}{\frac{1}{2}\rho u_\infty^2 AR}, \quad (2.6)$$

$$c_P = \frac{P}{\frac{1}{2}\rho u_\infty^3 A}. \quad (2.7)$$

## 2.2 Flow Transition

Flow transition in fluid dynamics is the process of a laminar flow becoming turbulent and was first demonstrated by Reynolds in 1883. Mostly, the laminar-turbulent transition is related to boundary layers in wall bounded uniform flow, though it applies to any fluid flow. Three primary modes of transition have been identified in boundary layers and are often referred to as natural, bypass, and separated flow transition [12].

It is generally accepted that natural transition occurs in flows with low free-stream turbulence intensities ( $I < 1\%$ ) beyond a critical Reynolds number [12]. As the boundary layer thickness increases, small disturbances can grow in magnitude through the destabilizing effect of viscosity [13]. The initial disturbances are known as Tollmien-Schlichting waves. They grow very slowly, but when the waves have grown until the mean flow profile is altered, secondary instabilities cause the appearance of turbulent spots [14]. These spots grow further until the laminar boundary layer breaks down into a turbulent one.

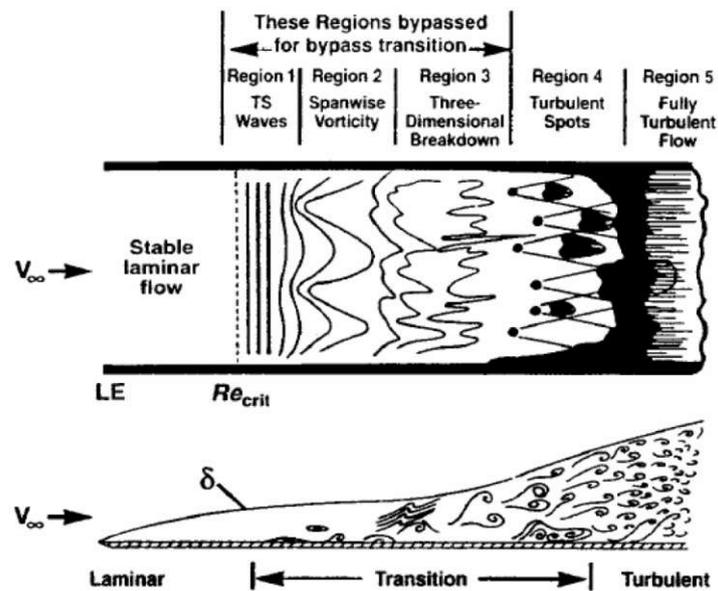


Figure 2.3: Stages of the natural transition process, adapted from [15].

Bypass transition occurs for flows with higher free-stream turbulence levels ( $>1\%$ ). Often  $I = 1\%$  is taken as the boundary between natural and bypass transition [12]. Here, the first three stages of the natural transition process are bypassed, and turbulent spots are

produced directly within the boundary layer [14]. Figure 2.3 shows the process of the flow transition process.

Instead of the turbulence intensity, surface roughness can lead to disturbances in the boundary layer as well and therefore induce bypass transition. Typically, bypass transition is considered to start when the skin friction starts to deviate from the laminar value. In airfoil theory, this is synonymous with the skin friction coefficient  $c_F$  reaching a minimum.

Various transition prediction models have been described in the literature. Besides models based on linear stability, e.g., the  $e^N$  method, several correlation-based models were developed. Most of them are based on the momentum thickness Reynolds number:

$$Re_\theta = \frac{u\theta}{\nu}, \quad (2.8)$$

with the characteristic length being the momentum thickness of the boundary layer  $\theta$ . Often, these models define a critical value,  $Re_{\theta_t}$ , at which the boundary layer is determined to transition.

However, the computation of  $\theta$  is a non-local operation, which makes it difficult to implement in unstructured, parallel CFD codes for three-dimensional RANS simulations. A model that is compatible with such codes is the Langtry-Menter  $\gamma - \hat{R}e_{\theta_t}$  model, which uses local properties and empirical correlations to estimate  $Re_\theta$ . A more detailed description of this model is given in Sec. 3.2.

## 2.3 Leading-Edge Erosion

Erosion on wind turbine blades is a gradual process that is caused by the repeated impact of airborne particles, rain, hail, or insects. Of course, the extent of erosion is dependent on the location due to regional differences in weather, insect occurrence, or air quality. Furthermore, on offshore turbines, the acid and salt content in the water can influence the erosion process by chemical interaction with the coating materials [16].

The erosion process of the rotor blade starts with a surface roughness increase at the leading edge. Small pits form, and, over time, they increase in density until gouges are formed [17]. Pits and gouges on the leading edge of a blade are shown in Fig. 2.4a. With increasing roughness, the aerodynamic performance suffers, i.e., the drag increases while the lift decreases. Blades may show signs of erosion after only three years of operation [3]. When not maintained properly, the gouges grow in size and density and can cause delamination of the underlying composite laminates [18]. At this stage (see Fig. 2.4b), the geometry alterations are so severe that mathematical roughness models cannot capture the fluid-dynamic effects anymore.

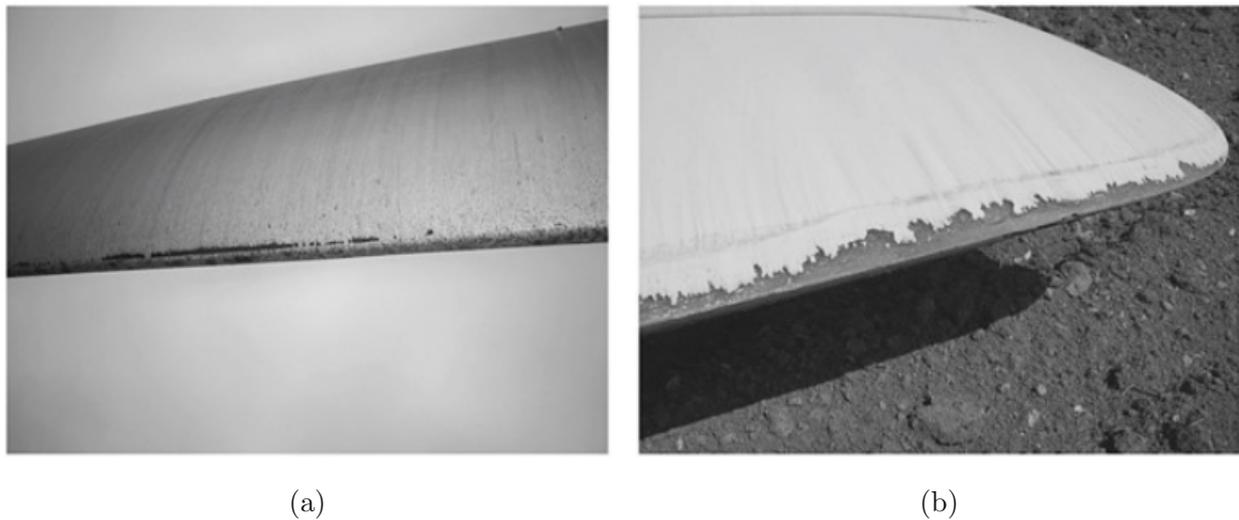


Figure 2.4: Stages of leading-edge erosion on real rotor blades with (a) pits and gouges, and (b) leading edge delamination [17].

The high-speed impact causes the surface damage over time. Blade tip velocities of up to

75 m/s are reported for medium sized turbines, larger turbines can reach speeds of up to 120 m/s [2]. Areas with higher impact velocities are more susceptible to erosion. Therefore, erosion typically starts to appear on the outermost parts of the blade.

The area most affected is near the leading edge, with a decrease in damage density in the chordwise direction. It has been reported that the damage is not symmetric, i.e., the lower surface of the blade is affected to a greater extent, “because the local average angle of attack is positive” [17]. Based on observations from photographic data, Sareen et al. [17] state a ratio of 1:1.3 to account for the greater extent on the lower surface.

Extensive wind tunnel measurements of a DU 96-W-180 airfoil with various stages of LEE were conducted by Sareen et al. [17]. They show a lift decrease of 17% compared to the clean case for the most severe LEE configurations and predict AEP losses of up to 25%. It should be noted that erosion was assumed on the full blade, which does not concur with practical experience.

More recently, Kruse et al. investigated a NACA 63<sub>3</sub>-418 airfoil affected by delamination and varying leading-edge roughness. They found that the drag continuously increased with the roughness height [19].

Moving on to numerical investigations, Schramm et al. [7] performed two-dimensional (2D) RANS simulations of high Reynolds number flows around a damaged NACA 64-618 airfoil. The delamination of the leading edge was directly modeled into the airfoil geometry. They predict a 7% AEP loss for a 5MW turbine. Wang et al. [20] used an airfoil geometry with pitting erosion for 2D RANS simulations. They modeled each pit as a semi-circular cavity and investigated the performance loss depending on the depth, density, extent and location of the erosion pits. A parametric CAD approach was used by Castorrini et al. [6] to generate the erosion pits. They account for the laminar-to-turbulent boundary layer with the Langtry-Menter  $\gamma - \hat{R}e_{\theta t}$  model. A maximum AEP loss of 2.5% is predicted for the NREL 5MW turbine. For the same turbine, Han et al. [5] predict AEP losses ranging from 2% to 3.7% depending on the extent of LEE.

Various approaches to represent surface roughness with numerical models have been proposed in the literature. Schramm et al. [7] assume a fully-turbulent flow and use the  $k - \omega$  SST model with rough wall functions. The  $k - \omega$  SST model is used in CFD simulations of Kruse et al. [21] as well. They account for wall roughness with an extension to the turbu-

lence model by Knopp et al. [22], which uses the equivalent sand grain roughness  $k_s$  as an input parameter. The authors conclude that more work has to be done on the calibration of the model and including the boundary layer transition could improve the results.

Langel et al. [14] propose a roughness extension to the Langtry-Menter  $\gamma - \hat{R}e_{\theta t}$  model to also include the effects of leading-edge roughness on the boundary layer transition. They investigated a NACA 63<sub>3</sub>-418 airfoil with leading-edge roughness from  $x/c = -0.13 : 0.02$ . The computed drag polars and transition locations are in good agreement with experimental data. However, it has to be stressed that the determination of the input parameter  $k_s$  is not universal and requires thorough consideration.

While most of the research seems to be mainly concerned with the effects of LEE on the annual power output, some questions arise regarding unsteady phenomena. One could assume that the dynamic behavior of a blade is negatively affected by LEE, meaning that the temporal course of the surface pressure distribution differs between the clean and the eroded profile in terms of frequency and amplitude. Consequences of such erosion-induced vibrations could include off-design loads and more frequent loading cycles affecting fatigue damage [3]. Furthermore, increased turbulence in the flow and higher pressure amplitudes can cause eroded blades to be noisier during operation. A study by Latoufis et al. [23], who investigated a small-scale, wooden wind turbine in a wind tunnel, reports that the acoustic noise emissions of the eroded rotor were increased by 10% on average. To investigate such aspects, URANS simulations are also performed in the course of this work. From a structural analysis point of view, the loss of material due to LEE leads to a blade-mass reduction and thus to changes in the blades' frequency response [24].

## 2.4 Roughness Characterization

Surface roughness can cause considerable alterations to the flow field. Included are disturbances of the velocity profile and the loss of flow momentum in the boundary layer. Rough surfaces can also encourage the early transition of a laminar boundary layer to a turbulent profile [25] and increase the turbulent fluctuations in the fully developed turbulent boundary layer [14].

Usually, surface roughness is described using the roughness Reynolds number:

$$Re_k = \frac{u_k k_r}{\nu}, \quad (2.9)$$

where  $k_r$  is the roughness height and  $u_k$  the flow velocity at  $k_r$ .

On an airfoil, the above mentioned early transition and alterations to the fully turbulent boundary layer can subsequently change the airfoil's polar and stall characteristics [14].

Surface roughness can be categorized into isolated 2D roughness, isolated 3D roughness, and distributed roughness. The isolated roughness elements have been studied extensively in the past, and experiments produced consistent results; however, generalizing the effects of distributed roughness has proven difficult [25].

For the purpose of this study, it is sufficient only to consider distributed roughness since the leading-edge erosion can be characterized as such.

Early investigations related to distributed surface roughness were done in the 1930s by Schlichting [26], and Nikuradse [27]. These experiments were focused on the friction factor in pipe flows, whereas the influence of roughness on transition was not studied until the 1950s, e.g., by Feindt [28]. Feindt varied the roughness height on flat plates and identified the transition location as a function of  $Re_k$ . A critical value for  $Re_k$ , below which surfaces could be considered “hydraulically smooth”, was reported as  $Re_k < 120$ . Experiments on an airfoil with leading-edge erosion were performed by Kerho [29].

Roughness features are often characterized by the equivalent sand grain roughness  $k_s$ , a term coined by Schlichting [26]. The intent of this concept is that a “roughness feature has the ‘equivalent’ effect on skin friction losses as a uniform layer of actual sand grains of

diameter  $k_s$ ” [30]. That implies that  $k_s$  is not a physically defined quantity, and its determination is not exactly straightforward. A wide variety of correlations, e.g., summarized by Bons [30], have been proposed to relate measurable surface parameters to  $k_s$ .

In this thesis, the actual roughness height  $k_r$  is related to  $k_s$  through a density function  $f(\Lambda_k)$ , which links to distribution density or other shape-defining parameters of the roughness elements:

$$\frac{k_s}{k_r} = f(\Lambda_k). \quad (2.10)$$

The relationship Eq. 2.10 is adopted from [14] to comply with the roughness model described in Sec. 3.3. Further information regarding the determination of  $f(\Lambda_k)$  is given by [14] as well.

The equivalent sand grain roughness can be used to define a dimensionless roughness parameter  $k_s^+$ :

$$k_s^+ = u_\tau \frac{k_s}{\nu}, \quad (2.11)$$

where  $u_\tau$  is the shear velocity:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}. \quad (2.12)$$

A second dimensionless roughness parameter is the equivalent sand grain roughness Reynolds number, defined as:

$$Re_{k_s} = \frac{u_\infty k_s}{\nu}. \quad (2.13)$$

# Chapter 3

## Governing Equations

The governing equations of the fluid flow are the continuity equation and the momentum equations (Navier-Stokes equations). The continuity equation can be written as:

$$\frac{\partial \rho}{\partial t} + \rho \frac{\partial u_i}{\partial x_i} = 0. \quad (3.1)$$

For incompressible fluids ( $\rho = \text{const.}$ ), Eq. 3.1 simplifies to:

$$\frac{\partial u_i}{\partial x_i} = 0. \quad (3.2)$$

Assuming a fluid with constant density and viscosity, the Navier-Stokes equations in conservation form read as follows:

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j} (u_j u_i) - \frac{\partial}{\partial x_j} (2\nu s_{ij}) = -\frac{1}{\rho} \frac{\partial p}{\partial x_i}, \quad (3.3)$$

where  $s_{ij}$  is the strain-rate tensor:

$$s_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (3.4)$$

Starting from Eq. 3.3, the Reynolds Averaged Navier-Stokes (RANS) equations can be derived using the Reynolds decomposition. Thereby, a flow variable is separated into a mean and a fluctuating component:

$$u_i(x, y, z, t) = \bar{u}_i(x, y, z, t) + u'_i(x, y, z, t). \quad (3.5)$$

After inserting the decomposition into Eq. 3.3 and some algebra, the RANS equation can be written as:<sup>1</sup>

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial (\overline{u'_i u'_j})}{\partial x_j}. \quad (3.6)$$

The most common approach to close the RANS equations is Boussinesq's approximation of the Reynolds stress tensor  $\overline{u'_i u'_j}$ :

$$\overline{u'_i u'_j} = -\nu_t \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} - \frac{2}{3} \frac{\partial \bar{u}_k}{\partial x_k} \delta_{ij} \right) + \frac{2}{3} \rho k \delta_{ij}, \quad (3.7)$$

where  $\delta_{ij}$  is the Kronecker delta and  $\nu_t$  is the kinematic eddy viscosity. A suitable turbulence closure model would then be used to represent  $\nu_t$ . Among the many proposed RANS turbulence models that rely on the Boussinesq approximation, the Shear Stress Transport (SST) turbulence model is used in the present work and described in more detail in the following section. It also forms the basis of the Langtry-Menter  $\gamma - \hat{R}e_{\theta t}$  transition model, which is discussed as well. Furthermore, this chapter deals with the theory of the Amplification Roughness model. Its implementation is presented thereafter in Chap. 4.

---

<sup>1</sup>For better readability, the overline denoting the mean velocity is omitted in subsequent sections.

### 3.1 Shear Stress Transport Turbulence Model

The SST turbulence model is a two-equation linear eddy viscosity model commonly used in CFD. Developed by Menter [31] and Menter et. al [32], it forms a hybrid model that combines the  $k - \epsilon$  and  $k - \omega$  models. The SST model is given by the following transport equations:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = \tilde{P}_k - \beta^* \rho k \omega + \frac{\partial}{\partial x_j} \left[ (\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right], \quad (3.8)$$

$$\begin{aligned} \frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} &= \frac{\alpha}{\nu_t} \tilde{P}_k - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[ (\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \right] \\ &+ 2(1 - F_1) \frac{\rho \sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}. \end{aligned} \quad (3.9)$$

A blending function  $F_1$  is used in Eq. 3.9 to switch between the two underlying models<sup>2</sup>. The  $k - \omega$  model is used in the inner region of the boundary layer, while the  $k - \epsilon$  model is active in the free stream. This formulation avoids the high sensitivity of the  $k - \omega$  model to turbulent properties in the free stream.  $F_1$  is also used to blend the individual model's constants.

A second blending function  $F_2$  - depending primarily on the wall distance - limits the eddy viscosity:

$$\mu_t = \frac{\rho a_1 k}{\max(a_1 \omega, S F_2)}, \quad (3.10)$$

and enhances the model's ability to predict flow detachment [33]. The factor  $a_1$  is one of many model constants and is typically given by  $a_1 = 0.31$  [34].

<sup>2</sup> $F_1 = 0$  in the free flow,  $F_1 = 1$  in the boundary layer.

### 3.1.1 Note on BC for External Aerodynamic Flows

The turbulent kinetic energy  $k$  at the inlet, or in the free stream of external aerodynamic problems, is typically set to a value that represents a turbulence intensity according to Eq. 3.11:

$$k_{\text{farfield}} = \frac{3}{2} (u_{\infty} I)^2. \quad (3.11)$$

However, assigning a boundary value for  $\omega$  is not always as straightforward and is somewhat open to interpretation<sup>3</sup>.

The following boundary conditions (BC) are recommended in the original reference [31]:

$$\frac{u_{\infty}}{L} < \omega_{\text{farfield}} < 10 \frac{u_{\infty}}{L}, \quad (3.12)$$

$$10^{-5} \frac{u_{\infty}^2}{Re_L} < k_{\text{farfield}} < 0.1 \frac{u_{\infty}^2}{Re_L}, \quad (3.13)$$

$$\omega_{\text{wall}} = 10 \frac{6\nu}{\beta_1 (d_1)^2}, \quad (3.14)$$

$$k_{\text{wall}} = 0. \quad (3.15)$$

with  $d_1$  being “the distance to the next point away from the wall” and  $L$  being “the approximate length of the computational domain” [31]. In this context, it is useful to define the dimensionless wall distance  $y^+$ :

$$y^+ = \frac{u_{\tau} d_1}{\nu}. \quad (3.16)$$

<sup>3</sup>Often, a value for  $\omega$  is chosen that gives a certain viscosity ratio or a certain decay rate of  $k$  [35].

In external aerodynamic flows, “the turbulence variables decay (sometimes dramatically) from their set values in the farfield” [34]. Spalart and Rumsey [36] suggested a modified formulation of the SST model to compensate for this non-physical decay. This form of the SST model merely adds sustaining terms to the original equations (Eq. 3.9 and 3.8) and introduces more precise farfield boundary conditions:

$$\omega_{\text{farfield}} = 5 \frac{u_{\infty}}{L}, \quad (3.17)$$

$$k_{\text{farfield}} = 10^{-6} u_{\infty}^2. \quad (3.18)$$

Now,  $L$  refers to a “defining length scale” of a certain problem [34], e.g., the chord length of an airfoil.

In OpenFOAM, this formulation of the SST model can be selected by the keyword *decayControl* in the *turbulenceProperties* dictionary. Ambient values for  $k$  and  $\omega$  have to be provided as well. Listing 3.1 shows an example of how the decay control is activated in OpenFOAM.

```

1   kOmegaSSTCoeffs
2   {
3     // Optional decay control
4     decayControl    yes;
5     kInf            \<far-field k value\>;
6     omegaInf        \<far-field omega value\>;
7   }

```

Listing 3.1: *turbulenceProperties* entry to activate the decay controlled variant of the SST model.

### 3.2 Langtry-Menter $\gamma - \hat{Re}_{\theta t}$ Transition Model

The widely used Langtry-Menter  $\gamma - \hat{Re}_{\theta t}$  model is a transition model developed by Langtry and Menter [13]. It is also known as  $\gamma - \hat{Re}_{\theta t}$  SST model since it is based on the SST turbulence model and uses two additional transport equations for the intermittency  $\gamma$  and the local transition onset momentum thickness Reynolds number  $\hat{Re}_{\theta t}$ . Only minor modifications of the blending function  $F_1$ , the source- and dissipation terms of the original SST model are necessary.

This model correlates  $Re_{\theta}$  to the vorticity or strain-rate Reynolds number  $Re_{\nu}$  by:

$$Re_{\theta} = \frac{\max(Re_{\nu})}{2.193}. \quad (3.19)$$

$Re_{\nu}$  is locally formulated using the strain-rate magnitude  $S = \sqrt{2s_{ij}s_{ji}}$  and is defined as:

$$Re_{\nu} = \frac{\rho y^2}{\mu} \left| \frac{\partial u}{\partial y} \right| = \frac{\rho y^2}{\mu} S, \quad (3.20)$$

where  $y$  is taken to be the wall-normal. A trigger function  $F_{onset} = f(Re_{\nu}, Re_{\theta c})$  is used to produce the intermittency  $\gamma$  variable once the ratio  $\frac{Re_{\nu}}{2.193}$  reaches a threshold value, defined as  $Re_{\theta c}$ . The production of intermittency initiates the transition process by switching on the underlying turbulence model. The intermittency  $\gamma$  indicates if the local boundary layer flow<sup>4</sup> is laminar ( $\gamma = 0$ ) or turbulent ( $\gamma = 1$ ) and is governed by the following transport equation:

$$\frac{\partial(\rho\gamma)}{\partial t} + \frac{\partial(\rho u_j \gamma)}{\partial x_j} = P_{\gamma} - E_{\gamma} + \frac{\partial}{\partial x_j} \left[ \left( \mu + \frac{\mu_t}{\sigma_f} \right) \frac{\partial \gamma}{\partial x_j} \right]. \quad (3.21)$$

The threshold value,  $Re_{\theta c} = f(\hat{Re}_{\theta t})$ , describes “the position where turbulent fluctuations begin in the boundary layer” [14]. Equation 3.22 governs the distribution of the empirical correlation  $Re_{\theta t}$  in the flow field through the use of an additional variable  $\hat{Re}_{\theta t}$ .

<sup>4</sup> $\gamma = 1$  in the free stream.

$$\frac{\partial (\rho \hat{R}e_{\theta t})}{\partial t} + \frac{\partial (\rho u_j \hat{R}e_{\theta t})}{\partial x_j} = P_{\theta t} + \frac{\partial}{\partial x_j} \left[ \sigma_{\theta t} (\mu + \mu_t) \frac{\partial \hat{R}e_{\theta t}}{\partial x_j} \right]. \quad (3.22)$$

$Re_{\theta t}$  is included in the source term

$$P_{\theta t} = c_{\theta t} \frac{\rho}{t} (Re_{\theta t} - \hat{R}e_{\theta t}) (1 - F_{\theta t}), \quad (3.23)$$

and defined as:

$$Re_{\theta t} = \begin{cases} (1173.51 - 589.428I + \frac{0.2196}{I^2}) F(\lambda_{\theta}), & I \leq 1.3 \\ 331.50 (I - 0.5658)^{-0.671} F(\lambda_{\theta}), & I > 1.3 \end{cases} \quad (3.24)$$

The function  $Re_{\theta t}$  depends on the free-stream turbulence intensity  $I$  and a pressure gradient parameter  $\lambda_{\theta}$ :

$$F(\lambda_{\theta}) = \begin{cases} 1 + (12.986\lambda_{\theta} + 123.66\lambda_{\theta}^2 + 405.689\lambda_{\theta}^3) e^{(-\frac{I}{1.5})^{1.5}}, & \lambda_{\theta} \leq 0 \\ 1 + 0.275 (1 - e^{-35\lambda_{\theta}}) e^{-\frac{I}{0.5}}, & \lambda_{\theta} > 0 \end{cases} \quad (3.25)$$

with:

$$\lambda_{\theta} = \frac{du}{ds} \frac{\theta^2}{\nu}, \quad (3.26)$$

where  $du/ds$  is the acceleration in the streamwise direction. Due to the appearance of  $\theta$  in Eq. 3.26,  $Re_{\theta t}$  has to be computed iteratively. Further information concerning this model can be found in [14], [37], or in the original paper [13].

The equation of the turbulent kinetic energy in the original SST model has to be modified so that the source term  $\tilde{P}_k$  scales with  $\gamma_{eff}$  and the dissipation term  $\tilde{D}_k$  is limited:

$$\tilde{P}_k = \gamma_{eff} \tilde{P}_{k|SST}, \quad (3.27)$$

$$\tilde{D}_k = \min(\max(\gamma_{eff}, 0.1), 1.0) \tilde{D}_{k|SST}. \quad (3.28)$$

Also, the blending function  $F_1$  has to be changed so that  $F_1 = 0$  does not appear in the laminar boundary layer.

The boundary conditions for  $\gamma$  and  $\hat{R}e_{\theta t}$  suggested in [37] are:

$$\frac{\partial \gamma}{\partial n}|_{wall} = 0, \quad (3.29)$$

$$\gamma_{farfield} = 1, \quad (3.30)$$

$$\frac{\partial \hat{R}e_{\theta t}}{\partial n}|_{wall} = 0, \quad (3.31)$$

$$\hat{R}e_{\theta t, farfield} = \begin{cases} (1173.51 - 589.428I + 0.2196I^{-2}), & I \leq 1.3 \\ 331.50 (I - 0.5658)^{-0.671}, & I > 1.3 \end{cases} \quad (3.32)$$

### 3.3 Amplification Roughness Model

The version of the roughness model described in this section is an extension to the Langtry-Menter  $\gamma - \hat{R}e_{\theta t}$  model and was formulated by Langel et al. in 2017 [14]. This model introduces a scalar field quantity, the amplification roughness  $A_r$ , that is defined on rough walls. An additional transport equation distributes this quantity throughout the flowfield. The idea behind this is that information about the roughness is not just available at the rough boundary patch itself but can be passed downstream as well. Finally,  $A_r$  modifies the onset criteria of the Langtry-Menter  $\gamma - \hat{R}e_{\theta t}$  model through a change in the source term.

The concept of using an additional transport equation to account for surface roughness was originally proposed by Dassler et al. [38].

The transport equation for  $A_r$  is given by:

$$\frac{\partial(\rho A_r)}{\partial t} + \frac{\partial(\rho u_j A_r)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ \sigma_{Ar} (\mu + \mu_t) \frac{\partial A_r}{\partial x_j} \right], \quad (3.33)$$

with the constant  $\sigma_{Ar} = 30.0$ .

$A_r$  is defined through a boundary condition at the rough wall by the sigmoid function

$$A_{r|wall} = f(k_s^+) = \frac{C_{Ar1}}{1 + e^{-(C_{Ar2}k_s^+ + C_{Ar3})}}, \quad (3.34)$$

with the constants  $C_{Ar1} = 2000$ ,  $C_{Ar2} = 1$  and  $C_{Ar3} = -13.5$ . The function is plotted in Fig. 3.1 and was calibrated to match the flow behavior over flat plates; an in-depth description of that process is given in the original reference [14].<sup>5</sup>

Information about the roughness feature is passed to the model only through the boundary condition for  $A_r$ , i.e., only one input parameter  $k_s$ . It is correlated to the physical roughness height  $k_r$  through a density function  $f(\Lambda_k)$  (see Eq. 2.10). Ideally, the input is assumed to

<sup>5</sup>The constant  $C_{Ar3}$  is given with 13.5 in the original reference [14]. That would shift the function into the negative, non-physical range of  $k_s^+$ . Therefore, in this work,  $C_{Ar3}$  is set to  $-13.5$ , which is also consistent with a graph provided in the original reference.

be  $k_s = k_r$  or  $f(\Lambda_k) = 1$ . However, since Eq. 3.34 was calibrated to a flat plate, airfoil cases require a change of  $f(\Lambda_k)$ . Furthermore,  $f(\Lambda_k)$  changes with density distribution. For the airfoil validation cases in the original paper, the input  $k_s$  was varied until experimental results matched.

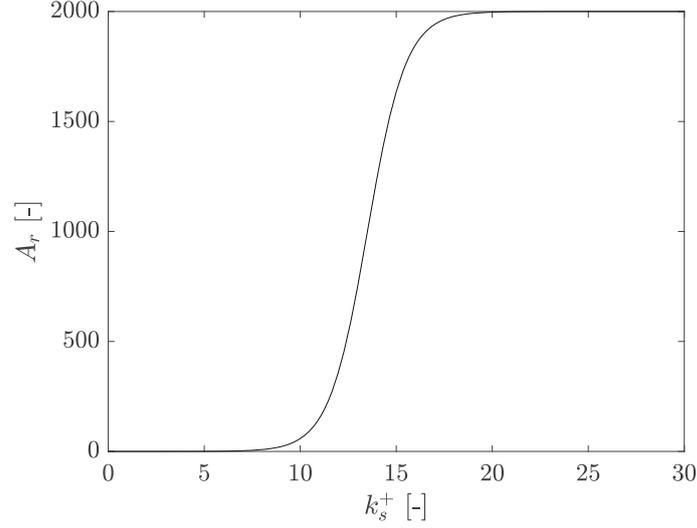


Figure 3.1: Rough wall boundary condition for  $A_r$ .

The  $\hat{R}e_{\theta t}$  source term is modified as follows:

$$P_{\theta t, mod} = c_{\theta t} \frac{\rho}{t} \left[ \left( Re_{\theta t} - \hat{R}e_{\theta t} \right) (1 - F_{\theta t}) - b F_{Ar} \right], \quad (3.35)$$

with a blending function  $b$ :

$$b = 0.5 \sin \left( \frac{\pi}{155} \hat{R}e_{\theta t} - \frac{97\pi}{155} \right) + 0.5. \quad (3.36)$$

The function  $F_{Ar}$  is determined according to Eq. 3.37, in which  $F(\lambda_{\theta})$  is taken directly from the Langtry-Menter  $\gamma - \hat{R}e_{\theta t}$  model (see Eq. 3.25).

$$F_{Ar} = \begin{cases} 4.0 A_r^{0.42} F_{\theta t} + 4.0 A_r \left( \frac{1}{F(\lambda_{\theta})} \right) (1 - F_{\theta t}), & F(\lambda_{\theta}) \leq 1.0 \\ 4.0 A_r^{0.42} F_{\theta t} + 4.0 A_r \left( \frac{1}{F(\lambda_{\theta})} \right)^{0.6} (1 - F_{\theta t}), & F(\lambda_{\theta}) > 1.0 \end{cases} \quad (3.37)$$

### 3.3.1 Modification to the SST-BC

With the newly introduced variable  $A_r$ , the influence of surface roughness on the transition point can be included, but the effects on the turbulent boundary layer and the reduction of the turbulent dissipation rate cannot be captured with this variable alone [14].

Surface roughness causes a downward shift of the logarithmic law of the wall [39], which can be expressed by a function  $\Delta B$ :

$$u^+ = \frac{1}{\kappa} \ln(y^+) + B - \Delta B. \quad (3.38)$$

Here, the constants have been determined to be  $\kappa = 0.41$  and  $B = 5.0$  [40],  $\Delta B$  is typically a function of a roughness parameter.

These effects are taken into account by modifying the boundary condition for the specific dissipation rate  $\omega$ . Instead of the boundary condition for hydraulically smooth walls (Eq. 3.14), Langel et al. [14] suggest the following:

$$\omega_{wall|rough} = \frac{u_\tau^2 S_r}{\nu}, \quad (3.39)$$

with  $S_r$  being a function that depends on the dimensionless parameter  $k_s^+$ :

$$S_r = \begin{cases} \left(\frac{50}{k_s^+}\right)^2, & k_s^+ \leq 25 \\ \frac{100}{k_s^+}, & k_s^+ > 25 \end{cases} \quad (3.40)$$

In short, as roughness  $k_s^+$  increases, the modified BC essentially lowers  $\omega$  and therefore increases local skin friction.

## Chapter 4

# The Amplification Roughness Model in OpenFOAM

In this thesis, the Amplification Roughness model [14] - along with the modified boundary condition - was implemented into OpenFOAM-v1912. No literature regarding previous attempts to implement this model in OpenFOAM was found. Key aspects of this procedure are presented in this section.

Without getting into too much detail, some basics about the file structure of C++, the base programming language of OpenFOAM, have to be explained. In object-oriented programming languages, such as C++, classes are used to create objects and provide member variables and functions. These classes are typically built in pairs, the class definition file takes a *.C* extension, and the class declaration file (or header file) a *.H* extension.

OpenFOAM uses *wmake* to compile applications and libraries, which requires a certain directory structure (see Fig. 4.1). The *Make* subdirectory contains two files, *options* and *files*. The file *files* contains name and location of the respective library or executable. The paths to source files, which the compiler needs, are specified in *options*. This file further contains a list of libraries that are linked to the current application.

The turbulence model framework is organized somewhat differently than shown in Fig. 4.1. It makes use of templates that allows all solvers - compressible, incompressible, single- and multiphase - to use the framework for turbulence models. However, a custom turbulence

model is created with *.C* and *.H* files, just as any other application. Further details on this topic can be found in [41] or [42].

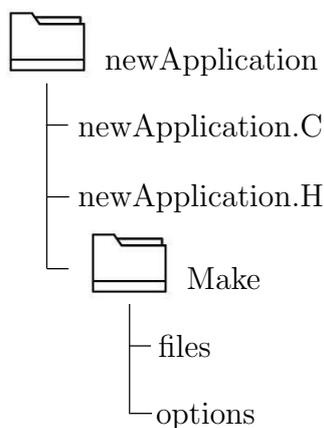


Figure 4.1: Typical file structure for applications compiled with *wmake*, adapted from [43].

A new turbulence model termed *kOmegaSSTLM\_AR*, based on the already established Langtry-Menter  $\gamma - \hat{R}e_{\theta t}$  model<sup>1</sup>, was created. In the header file *kOmegaSSTLM\_AR.H*, some additional quantities had to be declared, i.e., the constant  $\sigma_{Ar}$ , the new field  $A_r$  and its access function. Furthermore, a protected member function  $F_{Ar}$  was declared as shown in List. 4.1.

```

1 // - Return FAR for modified source-term
2 tmp<volScalarField::Internal> FAR
3 (
4     const volScalarField::Internal& Fthetat ,
5     const volScalarField::Internal& ReThetat0 ,
6     const volScalarField::Internal& Us
7 ) const ;
  
```

Listing 4.1: Declaration of  $F_{Ar}$  in the file *kOmegaSSTLM\_AR.H*

<sup>1</sup>called *kOmegaSSTLM* in OpenFOAM syntax.

The definition of  $F_{Ar}$  in the main file *kOmegaSSTLM\_AR.C* is done according to Eq. 3.37. Note that  $F(\lambda_\theta)$  is constructed through  $Re_{\theta t}$  with the use of an auxiliary variable in line 17 of List. 4.2. This approach avoids the need to iterate again and keeps the programming effort low. See Eq. 3.24 for reference. A part of the code is given below:

```

1   forAll(FAr, celli)
2   {
3       const scalar Tu
4       (
5           max(100*sqrt((2.0/3.0)*k[celli])/Us[celli], scalar(0.027))
6       )
7       scalar aux;
8       scalar Flambda;
9       if (Tu <= 1.3)
10      {
11          aux = (1173.51 - 589.428*Tu + 0.2196/sqr(Tu));
12      }
13      else
14      {
15          aux = 331.50*pow((Tu - 0.5658), -0.671);
16      }
17      Flambda = ReThetat0[celli] / aux;
18      if (Flambda <= 1)
19      {
20          FAr[celli] = 4*pow(Ar[celli],0.42)*Fthetat[celli]
21                      + 4*Ar[celli]*pow(Flambda,-1)*(1-Fthetat[celli]);
22      }
23      else
24      {
25          FAr[celli] = 4*pow(Ar[celli],0.42)*Fthetat[celli]
26                      + 4*Ar[celli]*pow(Flambda,-0.6)*(1-Fthetat[celli]);
27      }
28  }
29  return tFAr;

```

Listing 4.2: Definition of  $F_{Ar}$  in the file *kOmegaSSTLM\_AR.C*

As for the amplification roughness  $A_r$ , an *IObject* is constructed and its transport equation added in the member function *correctReThetatGammaInt*:

```

1 // - Amplification Roughness equation
2 tmp<fvScalarMatrix> ArEqn
3 (
4     fvm::ddt(alpha, rho, Ar_)
5     + fvm::div(alphaRhoPhi, Ar_)
6     - fvm::laplacian(alpha*rho*DArEff(), Ar_)
7     ==
8     fvOptions(alpha, rho, Ar_)
9 );

```

Listing 4.3: The transport equation for  $A_r$  in the file *kOmegaSSTLM\_AR.C*

Lastly,  $F_{Ar}$  is used to modify the source term in the  $\hat{R}e_{\theta t}$  transport equation.

```

1 // - Transition onset momentum-thickness Reynolds number equation
2 tmp<fvScalarMatrix> ReThetatEqn
3 (
4     fvm::ddt(alpha, rho, ReThetat_)
5     + fvm::div(alphaRhoPhi, ReThetat_)
6     - fvm::laplacian(alpha*rho*DReThetatEff(), ReThetat_)
7     ==
8     Pthetat*ReThetat0 - fvm::Sp(Pthetat, ReThetat_)
9     - Paux*b*FAr //adding the modified source term FAr*b explicitly
10    + fvOptions(alpha, rho, ReThetat_)
11 );

```

Listing 4.4: The transport equation for  $\hat{R}e_{\theta t}$  in the file *kOmegaSSTLM\_AR.C*

The term  $-P_{aux}bF_{Ar}$  is added to the transport equation (see Line 9 in List. 4.4) in an explicit manner.<sup>2</sup> Due to its negative sign, it is considered a sink term, and it is generally recommended to handle sink terms implicitly [44]. OpenFOAM provides mechanisms for the explicit or implicit discretization of source (or sink) terms, such as  $fvm::Sp(\rho\phi)$  and

<sup>2</sup> $P_{aux}$  is an auxiliary term only used for programming elegance.

fvm::SuSp( $\rho\phi$ ). That requires a term to be expressed as  $\rho\phi$ , where  $\rho$  is a scalar, and  $\phi$  is a volume field of some type [43]. In this particular case,  $\phi$  would correspond to  $\hat{R}e_{\theta t}$ . Expressing  $-P_{aux}bF_{Ar}$  in such a way proved difficult due to the nature of the blending function  $b$  (see Eq. 3.36). However, the implicit discretization of the sink term is considered an improvement to the convergence and stability behavior of the algorithm and could be implemented in future works.

The complete implementation of the roughness model also necessitates a new boundary condition. This new boundary condition is called *omegaRoughWallFunction* and supplies boundary conditions for  $\omega$  and  $A_r$ . Three input arguments, as shown in List. 4.5, have to be specified: the equivalent sand grain roughness  $k_s$ , the kinematic viscosity  $\nu$ , and a boundary condition type.

```

1   roughWall
2   {
3       type            omegaRoughWallFunction;
4       ks              \<wall ks value\>;
5       nu              \<nu value\>;
6       BCtype         omega;
7       value          \<dummy value for initialization only\>;
8   }
```

Listing 4.5: Example of the boundary condition specification.

Options for the *BCtype* entry are *omega* or *amplificationRoughness* to select the appropriate method, which computes values for  $\omega_{wall}$  and  $A_{r|wall}$ , respectively. Note that if  $k_s = 0$  is specified,  $\omega_{wall}$  will take the value specified by the entry *value*. Therefore, it is recommended to supply the value for smooth walls at this point.

The BC is created using the utility *foamNewBC*, which constructs the *.C* and *.H* source files as well as the compilation files. With the command *foamNewBC -f -s omegaRoughWallFunction*, a template code for a fixed value (*-f*) BC of a scalar field (*-s*) is generated. Further options are available and can be viewed with *foamNewBC -help*.

Necessary member data are declared in the header file *omegaRoughWallFunction.H* and classes used by the main file are included:

```
1 #include "fixedValueFvPatchFields.H"
```

Listing 4.6: Including the *fixedValueFvPatchFields* class in the file *omegaRoughWallFunction.H*

In the main file, the calculations are done in a loop over each face of the patch. Depending on the *BCType* entry, the respective value is computed according to Eq. 3.39 or Eq. 3.34. After these rather straightforward computations, the patch value is set to the newly computed quantity.

```
1 // - Set the value_ of this patch to the newly computed quantity
2 this->operator==(outVal);
3 }
```

Listing 4.7: Statement in the file *omegaRoughWallFunction.C* to set the patch value to *outVal*.

However, the calculation of the wall shear stress<sup>3</sup> is more challenging.  $\tau_w$  is computed with the patch normal vector  $n$  and the symmetric Reynolds stress tensor  $R$ :

$$\tau_w = R \cdot n. \quad (4.1)$$

While the computation of  $n$  is quite simple,  $R$  had to be retrieved from the turbulence model first. Listing 4.8 shows how the object registry is checked for a present turbulence model and how it is retrieved if one is present. A turbulence model is present when a solver runs the case. It is typically not loaded when preprocessing is performed. This check ensures that the BC works for preprocessing operations, e.g., *decomposePar* as well.

<sup>3</sup>The wall shear stress  $\tau_w$  is required for both quantities  $\omega$  and  $A_r$  through the dimensionless roughness parameter  $k_s^+$  (see Eq. 2.11 and Eq. 2.12).

```

1   if
2   (
3       db().found
4       (
5           IOobject::groupName
6           (
7               turbulenceModel::propertiesName ,
8               internalField().group()
9           )
10      )
11  )
12  {
13      const turbulenceModel& turbModel = db().lookupObject<turbulenceModel>
14      (
15          IOobject::groupName
16          (
17              turbulenceModel::propertiesName ,
18              internalField().group()
19          )
20      );
21      ...
22      // - Remaining code in this section
23  }

```

Listing 4.8: Checking if a turbulence model is present in the object registry in file *omegaRoughWallFunction.C*, following [45].

Then, a check is performed if the turbulence model is of incompressible type. If it is, a *refCast* gives access to the deviatoric part of the effective Reynolds stress tensor  $R_{eff}$ . In OpenFOAM, this part is accessed by *devReff()* as shown in List. 4.9.

```

1   if (isA<incompressibleTurbulenceModel>(turbModel))
2   {
3       const incompressibleTurbulenceModel& incompTurbModel =
4           refCast<const incompressibleTurbulenceModel>(turbModel);
5
6       // - Deviatoric Reynolds Stress Tensor at boundary patch
7       const tmp<volSymmTensorField> Re1 = incompTurbModel.devReff();
8       const tmp<symmTensorField> Re = Re1().boundaryField()[patch().index()];
9       ...
10      // - Remaining code in this section
11  }

```

Listing 4.9: Retrieving *devReff* from the turbulence model in file *omegaRoughWallFunction.C*, following [45].

Due to the way this BC is programmed, it only works for incompressible cases. That could be extended further to compressible cases in the future.

After compiling both models using *wmake*, libraries are created with the name and location specified in the files *Make/files*. Using them necessitates an entry in each case's *controlDict*:

```

1   libs
2   (
3       "libomegaRoughWallFunction.so"
4       "libmyIncompressibleTurbulenceModels.so"
5   );

```

Listing 4.10: *controlDict* entry to add user generated libraries.

# Chapter 5

## Validation and 2D Results

The numerical methods for the 2D airfoil simulations are described in this chapter. The general information about the simulation setup and the airfoils used is presented first.

Two models are used to characterize LEE; the Amplification Roughness model for slightly eroded airfoils and the so-called delamination model representing severe coating damage via geometrical discretization. The Amplification Roughness model's implementation is validated using flat-plate test cases.

Both RANS and URANS airfoil simulations are conducted with each model using OpenFOAM-v1912. The results are validated using experimental and numerical reference data. A comparison of both models and a discussion about their suitability for application in 3D simulations follows.

### 5.1 General Remarks

The validation process of the numerical methods, which is presented in the following sections, was performed on two different airfoils. Both airfoils belong to the NACA six-digit series with sharp trailing edges. Their contours are shown in Fig. 5.1. Even though they were designed in the 1940s, they are still being used on today's wind turbine blades [46].

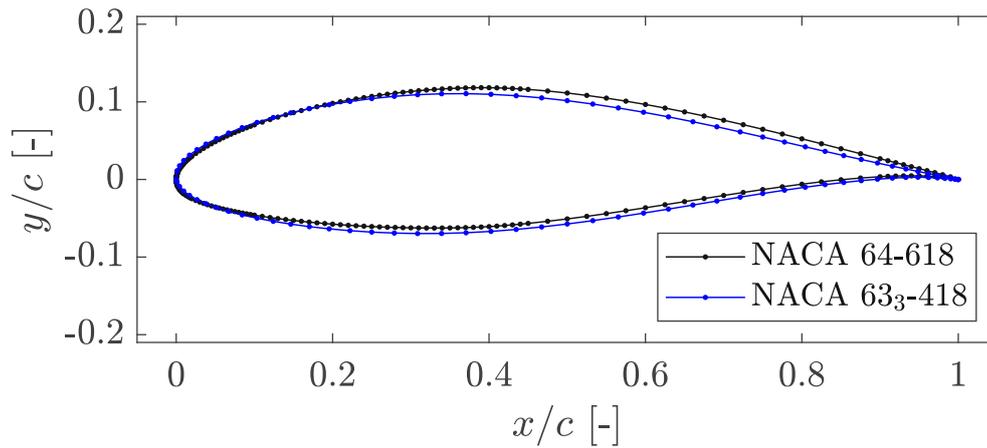


Figure 5.1: NACA 64-618 and NACA 63<sub>3</sub>-418 airfoil coordinates.

The selection of the first airfoil, the cambered NACA 64-618 airfoil with a maximum thickness of 18% of the chord, was motivated by the good accessibility of geometric and aerodynamic data. It is also utilized at the blade tip of the NREL reference wind turbine [47] and is widely used in wind energy research.

However, experimental data for that airfoil with LEE could not be found. Therefore, it was required to switch to the NACA 63<sub>3</sub>-418 airfoil, which has been subject to various investigations in terms of LEE [48, 19, 49].

While the coordinates of the NACA 64-618, obtained from the NREL project [50], show a sufficient amount of data points, the freely available coordinates for the NACA 63<sub>3</sub>-418 [51] lack data resolution. The contour point distribution had to be improved first to eliminate discontinuities on the airfoil surface and avoid problems later. That was done using the open-source airfoil contour analysis and CFD meshing tool PyAero [52], which is also able to spline and refine airfoil contours. Additional points are added by “producing a spline representation through the initial airfoil contour” [53], and then distributing a specified number of points along the spline. The leading and trailing edges are further refined. After this, the number of contour points had been increased from 97 to 549.

A large number of points arise due to excessive refinement in the leading edge area. That was done with consideration to the modeling of delamination features in Sec. 5.4.

Airfoils are commonly meshed with hexahedral cells using structured O-type or C-type grid topologies. C-type structured meshes generally come with a higher computational cost due to the clustered accumulation of grid points in the downstream wake. The beneficial wake resolution is an important factor for predicting the drag forces accurately [14]. So, C-type grids were chosen for all further analysis. Furthermore, O-type structured grids come with highly skewed cells for airfoils with sharp trailing edges. Figure 5.2 shows the C-type mesh around the NACA 64-618 airfoil.

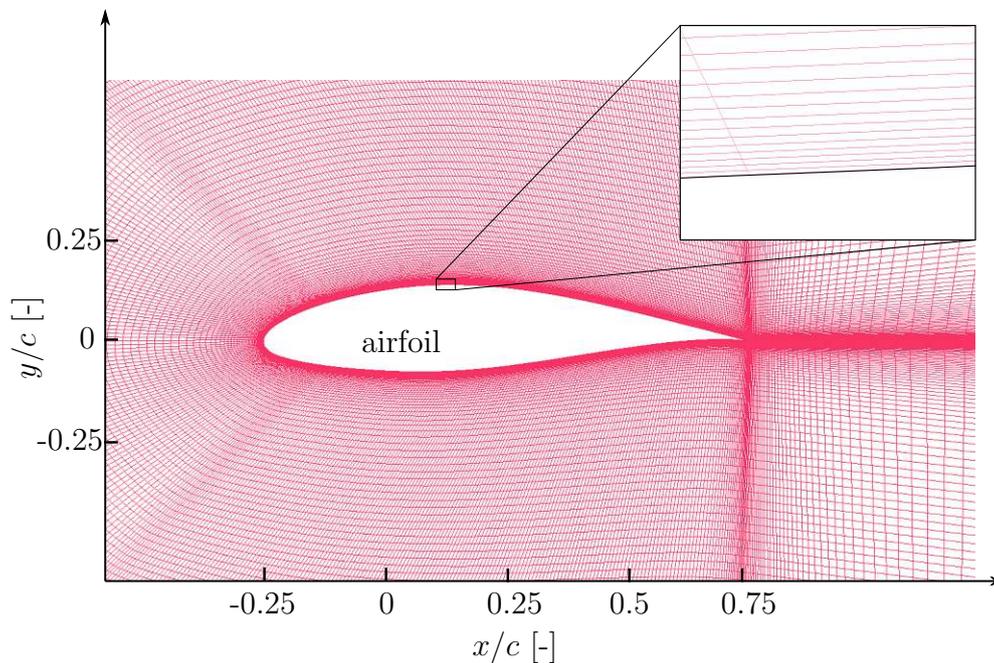


Figure 5.2: C-type mesh around NACA 64-618 airfoil.

Standard chord lengths of both airfoils are given as  $c = 1$  m, and it was assumed that the aerodynamic centers are located at the 25% chord position.

The airfoil surface is modeled as a no-slip wall without wall functions. The first cell height was set to  $d_1 = 3 \cdot 10^{-6}$  m with a growth rate of 1.1 to ensure  $y^+ < 1$  for all angles of attack and Reynolds numbers.

The computational domain's size is chosen such that every boundary is  $20c$  away from the airfoil to allow for proper flow development. The dimensions, along with the boundary patches, are visualized in Fig. 5.3. Note that OpenFOAM always requires a 3-dimensional mesh. Therefore, all grids for 2D computations have one cell in the z-direction. By speci-

fying a special *empty* condition on the frontAndBack patch, OpenFOAM understands that no solution is required along the z-axis.

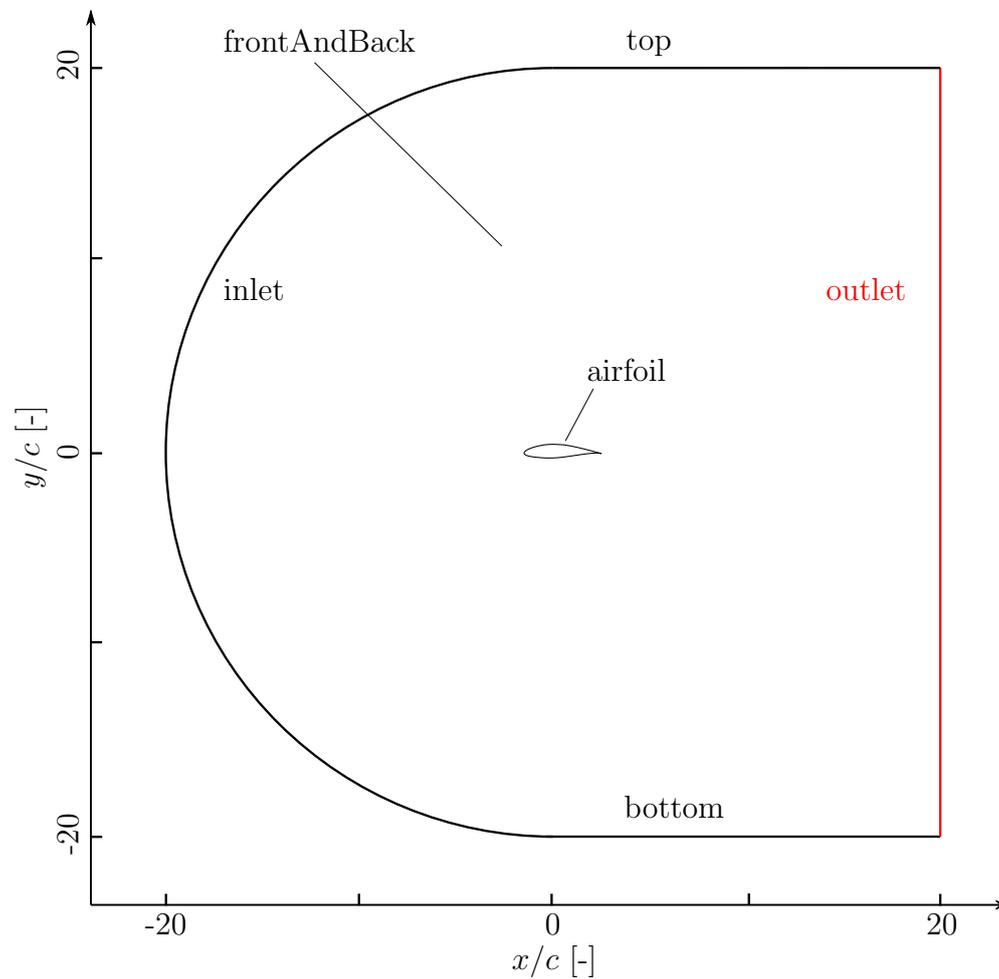


Figure 5.3: Computational domain for the 2D-simulations.

At the inlet, outlet, top, and bottom patches, free-stream BC are specified for the velocity and pressure fields<sup>1</sup>. A detailed list of BC for the basic fields is given in Appendix C. In order to change the angle of attack, the internal vector field of the velocity was changed accordingly.

Steady-state simulations (RANS) ran until lift and drag coefficients converged ( $\Delta < 10^{-5}$ ). However, at higher angles of attack, when flow separation occurs, convergence in terms of the force coefficients could not be achieved. Therefore, the periodically oscillating results were averaged.

<sup>1</sup> $u$ : freestreamVelocity,  $p$ : freestreamPressure

The numerical discretization schemes for terms are specified in the *system/fvSchemes* dictionary. Time derivatives are set to zero by specifying *steadyState* for RANS simulations. In transient simulations (URANS), the time derivative is discretized by a second-order implicit *CrankNicolson* scheme. The *Gauss linear* scheme is used for gradient terms. A gradient limiting version *cellLimited Gauss linear 1* is chosen for the velocity and turbulence fields to improve the numerical stability. The keywords for the advective terms are specified in the form *div(phi,...)*, where *phi* refers to the volumetric flux for incompressible flows [54]. The *linearUpwind* and *limitedLinear* schemes, which are second-order accurate and based on Gauss integration, are used for these terms. Note that in the steady-state simulations, *bounded* variants of the schemes are applied. Listing 5.1 shows the exemplary syntax for some fields.

```

1   gradSchemes
2   {
3       grad(U)           cellLimited Gauss linear 1;
4       // ...
5   }
6
7   divSchemes
8   {
9       div(phi,U)        bounded Gauss linearUpwind grad(U);
10      div(phi,k)         bounded Gauss limitedLinear 1;
11      // ...
12  }
```

Listing 5.1: Extract from the *fvSchemes* dictionary.

The consistent formulation of the SIMPLE algorithm with under-relaxation factors of 0.95 is used for steady-state computations. The PIMPLE algorithm is employed for transient simulations. Linear solvers used for the discretized terms are *GAMG* for the pressure field and *PBiCGStab* for the remaining fields. The equation solvers, tolerances, and algorithms are specified in the *system/fvSolution* dictionary. More information concerning discretization schemes and solution control can be found in the OpenFOAM User Guide [54].

A measure of flow compressibility is the Mach number  $Ma$ . Air flows obeying the ideal

gas law can be considered incompressible if  $Ma$  is less than 0.3. This is true for most of the investigated cases. However, it was observed that in some cases with high angles of attack, the local Mach numbers exceeded that value but remain below 0.6. While being aware of potential inaccuracies due to the increasing influence of compressible effects, they were deemed negligible for the purpose of this study, and incompressibility was assumed in all cases.

## 5.2 Validation of a Smooth Airfoil

A smooth NACA 64-618 airfoil was used to validate the CFD method. Experimental data from wind tunnel measurements by Timmer [46] were used to validate the simulation methodology. To match the test conditions, the simulations are carried out at  $Re = 6 \cdot 10^{-6}$ , which corresponds to an inflow wind speed of  $u_\infty = 90.36$  m/s. The inflow turbulence intensity is set to 0.1%. The objective of this comparison is to define the simulation settings, motivate the selection of the turbulence model, and predict drag and lift forces for angles of attack ranging from  $-10^\circ$  to  $18^\circ$ .

Figure 5.4 shows that the Langtry-Menter  $\gamma - \hat{Re}_{\theta t}$  transition model yields better results than the SST model. Both models are able to capture the lift characteristics for low angles of attack. However, the transition model can predict  $c_L$  accurately even for higher angles of attack when flow separation occurs. The transition model also exhibits superior drag prediction for the majority of angles of attack. Only at very low angles of attack the drag is somewhat underestimated. Although the transition model comes with a higher computational cost, it yields superior results and is therefore used throughout this study.

It should be noted that to achieve good agreement with the experimental data, the empirical constant  $a_1$ , which appears in the definition of the eddy viscosity, had to be reduced from 0.31 to 0.28. This was done according to the guideline in [55]. The results shown in Fig. 5.4 are obtained on the *medium* mesh. A mesh independence study is presented later on.

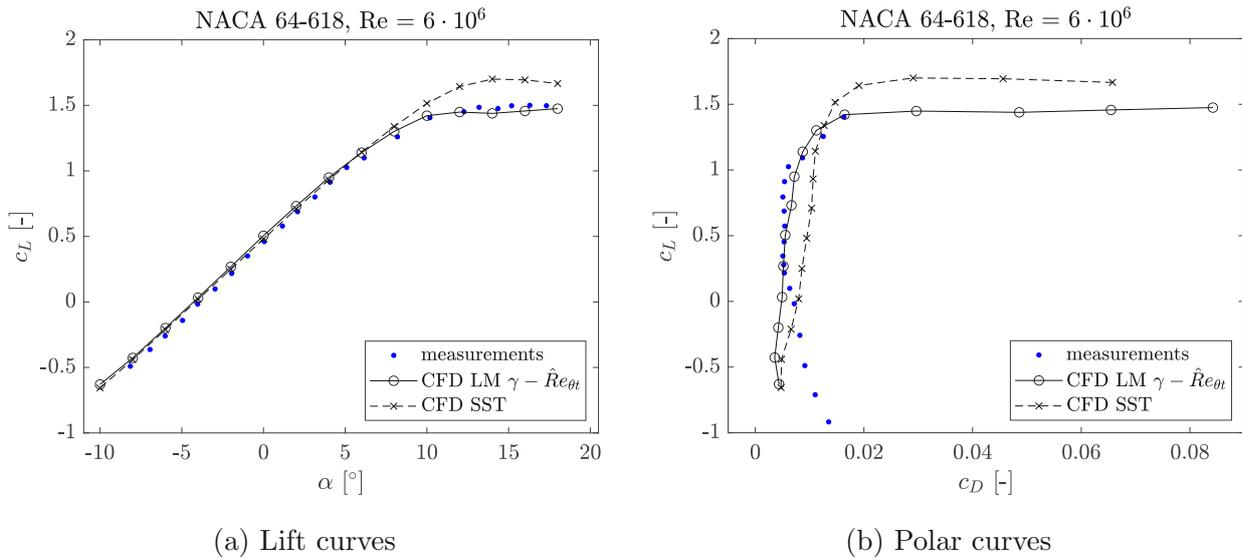


Figure 5.4: Comparison of numerical results against wind tunnel measurements [46] of a NACA 64-618 airfoil.

The two main principles for assessing the accuracy and credibility of CFD simulations are verification and validation [56]. The process of validation typically involves comparisons of the computational simulations with experimental results. The identification and quantification of errors and uncertainties in the numerical methodology, on the other hand, is part of the verification assessment. A method frequently used for uncertainty estimation is the Grid Convergence Index (GCI) method, which is briefly described in Appendix A. The goal is to show a reduction of the discretization error for quantities of interest by systematically refining the grid. In this section, results obtained on three gradually refined meshes are presented and analyzed. Furthermore, this process allows for optimizing the grid size in terms of reliability and computation time.

Meshes are created in three different resolutions to get a first estimation regarding the grid design. The properties of the grids, labeled as *coarse*, *medium*, and *fine*, are listed in Tab. 5.1. Only the number of nodes on the airfoil surface and the number of nodes in the radial direction are changed. The overall grid structure and the first cell height are retained.

Label	Airfoil nodes	Radial nodes	Elements
coarse	384	200	≈ 155k
medium	448	250	≈ 233k
fine	640	300	≈ 367k

Table 5.1: Properties of the grids used in first grid independence study.

Figure 5.5 shows the lift and polar curves obtained on those three grids. While the results on all three grids are acceptable for moderate angles of attack, it is clear that the coarse grid is insufficient for angles of attack above 10°. However, at  $\alpha = -10^\circ$ , the drag prediction is more accurate on the coarse mesh than on the medium grid.

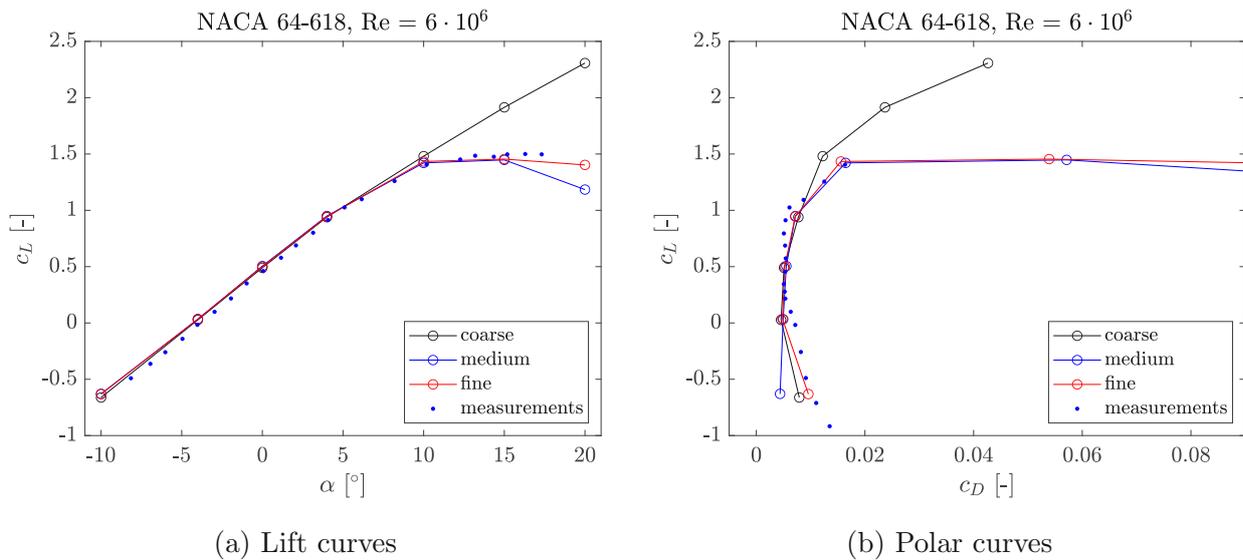


Figure 5.5: Comparison of airfoil characteristics resulting from the first grid independence study.

The results for the grid convergence according to the GCI method are shown in Tab. 5.2 for  $c_L$  and  $c_D$  at  $\alpha = 4^\circ$  and  $\alpha = 10^\circ$ . In Fig. 5.6a, convergence is shown for both coefficients normalized to the extrapolated values. It can be observed that most quantities exhibit an oscillating convergence behavior. That is also indicated by a negative value of  $R$ . It is unclear where this behavior originates from.

Angle of attack $\alpha$	4°		10°	
Parameter	$c_L$	$c_D$	$c_L$	$c_D$
Convergence ratio $R$ [-]	-0.491	0.054	-0.209	-0.207
Relative error $e_a^{21}$ [%]	0.531	0.412	0.874	5.610
Extrapolated error $e_{ext}^{21}$ [%]	0.475	0.017	0.200	1.288
Grid convergence index $GCI^{21}$ [%]	0.591	0.022	0.251	1.589

Table 5.2: Calculation of discretization uncertainties.

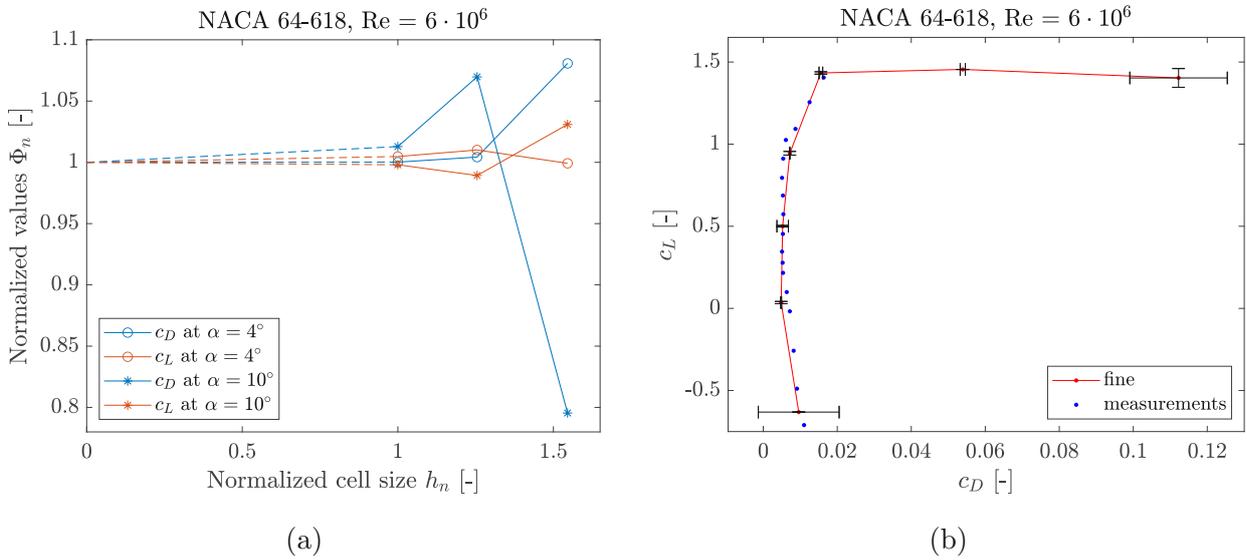


Figure 5.6: a) Convergence behavior of  $c_L$  and  $c_D$  at two angles of attack and b) polar curve of the *fine*-grid solution, with discretization error bars computed using the GCI.

The GCI represents an error estimate and is plotted as error bars in Fig. 5.6b to show the uncertainty of the *fine*-grid solution. The solution on the *fine* grid agrees very well with the experimental data, and, for the most part, GCI values are within an acceptable range. Nevertheless, it was decided to conduct a second mesh independence study aiming at optimizing the grid and understanding the effects of certain mesh parameters better. A new set of grids with the specifics listed in Tab. 5.3 was used. In addition to the previously defined *medium* and *fine* grids, two modified versions with a reduced number of radial nodes were created. Note that the results shown in Fig. 5.7 were obtained on the slightly different NACA 633-418 airfoil at a Reynolds number  $Re = 5 \cdot 10^6$ . For this part of the grid study,

simulations were limited to angles of attack between  $\alpha = 6^\circ$  and  $\alpha = 15^\circ$ .

Label	Airfoil nodes	Radial nodes	Elements
medium2	448	200	$\approx 166\text{k}$
medium	448	250	$\approx 233\text{k}$
fine2	640	200	$\approx 204\text{k}$
fine	640	300	$\approx 367\text{k}$

Table 5.3: Properties of the grids used in second mesh independence study.

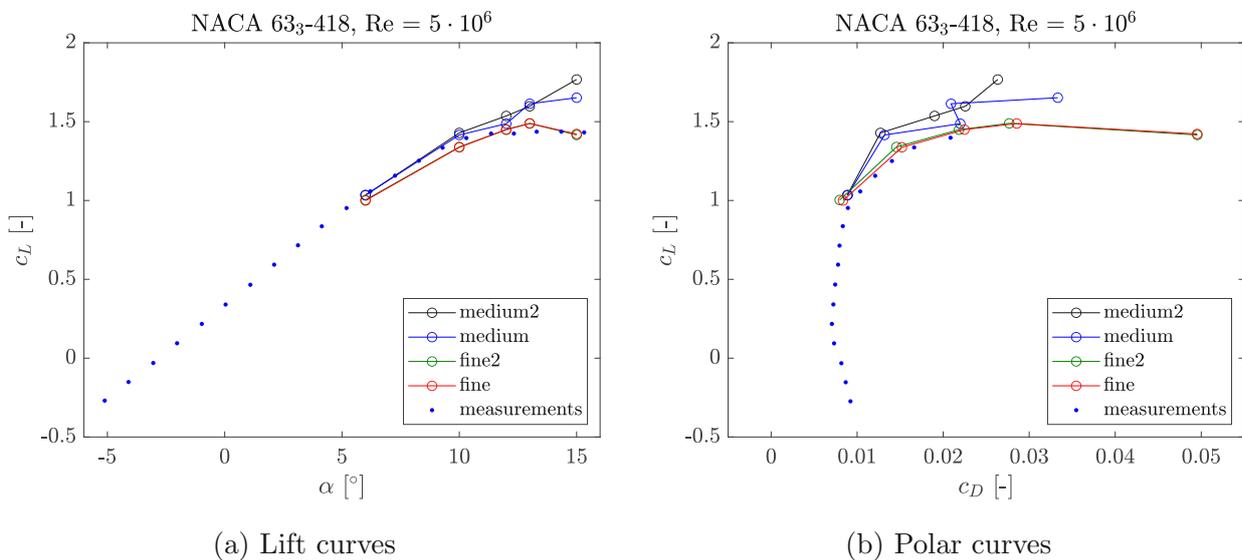


Figure 5.7: Comparison of airfoil characteristics resulting from the second grid independence study.

The above figure shows that a sufficiently large number of nodes on the airfoil surface is crucial for accurately predicting the aerodynamic coefficients. The number of radial nodes, however, seems to be of lower importance. Both results obtained on the *fine* and *fine2* grids match the measured characteristic precisely, but since the *fine2* grid only uses  $\approx 55\%$  of the *fine* grid’s cells, it is used for further analysis.<sup>2</sup> Again, convergence behavior is investigated through the GCI method. The *medium* grid was disregarded for this analysis. Table 5.4 shows the results computed with the GCI method; Fig. 5.8a shows the normalized values

<sup>2</sup>The computational time on the *fine2* grid is about 50% of the *fine* grid.

of the aerodynamic coefficients.

Angle of attack $\alpha$	$10^\circ$		$15^\circ$	
Parameter	$c_L$	$c_D$	$c_L$	$c_D$
Convergence ratio $R$ [-]	0.022	0.349	-0.018	-0.001
Relative error $e_a^{21}$ [%]	0.144	4.249	0.440	0.054
Extrapolated error $e_{ext}^{21}$ [%]	2.39e-6	9.66e-2	4.58e-6	2.39e-10
Grid convergence index $GCI^{21}$ [%]	2.99e-6	0.120	5.73e-6	2.98e-10

Table 5.4: Calculation of discretization uncertainties.

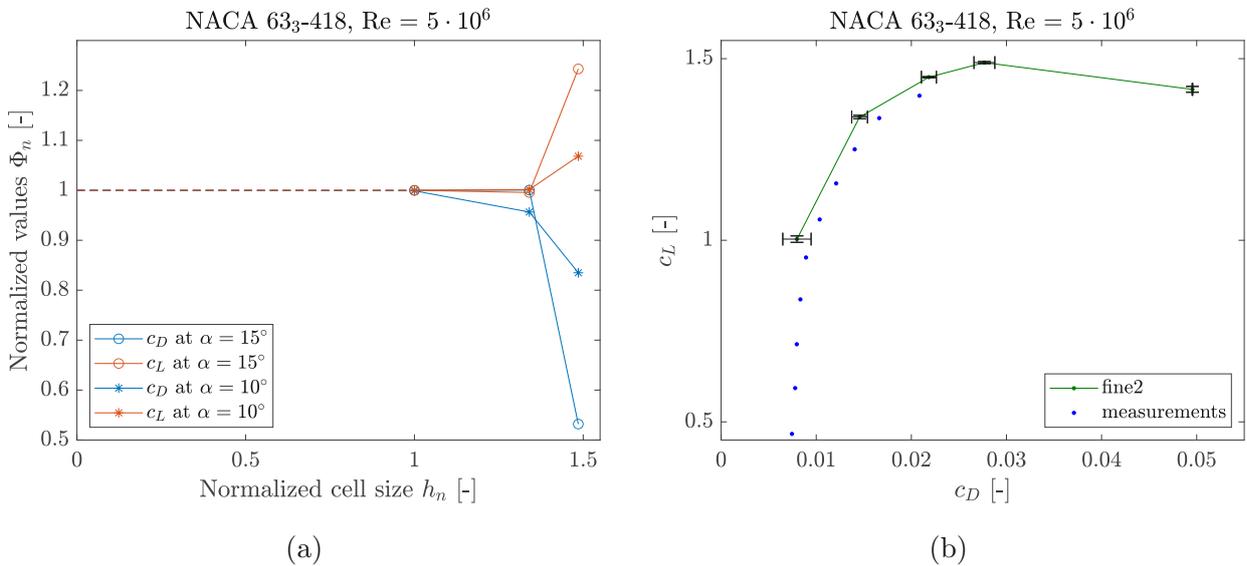


Figure 5.8: a) Convergence behavior of  $c_L$  and  $c_D$  at two angles of attack and b) polar curve of the *fine2*-grid solution, with discretization error bars computed using the GCI.

While low values are desirable for the GCI, the extremely low values in Tab. 5.4 have to be questioned. They possibly indicate that the simulations are not yet in an asymptotic range of convergence or that some other problem occurred. Some reasons seem plausible:

- Strongly differing results on the *medium2* grid: If one of the absolute errors  $\epsilon_{32} = \Phi_3 - \Phi_2$  or  $\epsilon_{21} = \Phi_2 - \Phi_1$  is “very close” to zero, the procedure fails, according to Celik et al. [57]. In this particular case, while  $\epsilon_{21}$  is very small for both coefficients and angles of attack,  $\epsilon_{32}$  is greater by a factor of at least 100.

- Unsystematic (unstructured) mesh refinement: The refinement procedure should be done systematically. Here, it has been carried out in a non-ideal way, i.e., the refinement factor is not constant ( $r_{21} = 1.34$ ,  $r_{32} = 1.11$ ), and only parts of the grid were refined while others remained unaltered.
- The apparent order  $p$  that is calculated during the GCI procedure does not agree well with the formal order  $\hat{p}$  of the schemes. While the solution is obtained using second-order schemes ( $\hat{p} = 2$ ),  $p$  takes values between 12 and 65. Agreement between the formal and apparent order is a good indication “of the grids being in the asymptotic range” [57]. However, Celik et al. [57] also state that disagreement “should not necessarily be taken as a sign of unsatisfactory calculations.”

Leaving that aside, the aerodynamic coefficients obtained on the *fine2* mesh are in excellent agreement with the experimental data. Therefore, the *fine2* grid is used to test the numerical erosion models in the remainder of this chapter. Additional validation is provided by the pressure coefficient  $c_p$  shown in Fig. 5.9.

For a smooth NACA 63<sub>3</sub>-418 airfoil at  $Re = 5 \cdot 10^6$ ,  $c_p$  is computed for angles of attack  $\alpha = -10^\circ, 0^\circ, 10^\circ$ , and  $15^\circ$ . Again, the results obtained on the *fine2* grid agree generally well with the experimental data by Kruse [8]. It is noticeable that the pressure at the stagnation point was underpredicted in all cases. While the pressure coefficient on the lower surface of the airfoil fits the experimental data quite well in all cases, some deviations can be seen on the upper surface. A trend towards overestimating  $c_p$  from the leading edge until 60% $c$  is present for  $\alpha = -10^\circ, 0^\circ$  and  $10^\circ$ . At  $\alpha = 15^\circ$ , deviations in the front part can be seen as well, but most significantly, they are present near the trailing edge. The experimental data shows an adverse pressure gradient from 40% $c$  until the trailing edge, indicating flow separation. The simulation shows a similar trend, although it delays the separation point and overestimates the pressure.

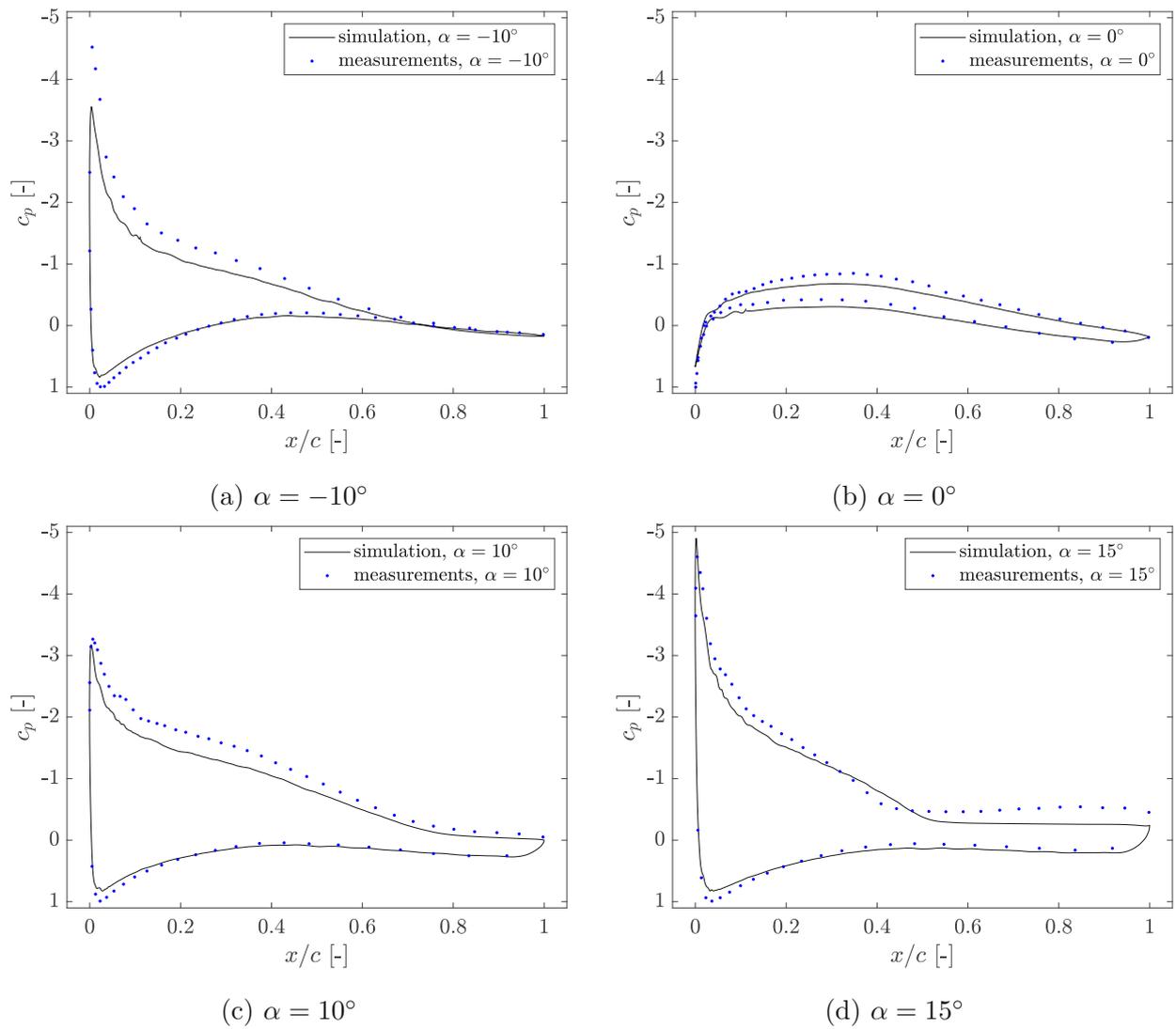


Figure 5.9: Comparison of the pressure coefficient  $c_p$  between simulations and experimental data for  $\alpha = -10^\circ, 0^\circ, 10^\circ$  and  $15^\circ$ .

## 5.3 Roughness Model

In this section, the Amplification Roughness model is used to represent the early erosion phase mathematically. The theoretical concept and the model’s implementation in OpenFOAM are described in detail in Sec. 3.3 and Chap. 4, respectively. The implemented model will be validated on flat plates and airfoils against experimental or numerical reference data in this section. Quantities used to validate the implemented model are the lift and drag coefficients, as well as the location of the boundary layer transition. Airfoils were tested for flows with Reynolds numbers ranging from  $1.6 \cdot 10^6$  to  $5 \cdot 10^6$ .

### 5.3.1 Flat Plate Cases

The validation includes the comparison of simulation results for flows over a flat plate with experimental data. The ERCOFTAC flat plate test cases, which were defined in the 1990s, are used as reference cases. This database has become a standard test case for validating numerical transition models. A flat plate with a rounded leading edge was investigated in a wind tunnel. Experimental data are available for a variety of turbulence intensities and pressure gradients.

Here, the testing was limited to two cases with zero pressure gradient, i.e., the T3A and T3A- cases. The boundary conditions of these configurations are given in Tab. 5.5.

Case	Velocity $u_\infty$ [m/s]	Turbulence intensity $I$ [%]	Viscosity ratio $\frac{\nu_t}{\nu}$ [-]
T3A	5.4	3.3	12.0
T3A-	19.8	0.874	8.72

Table 5.5: Boundary conditions of the test cases T3A und T3A-;  $\rho = 1.225 \text{ kg/m}^3$  and  $\nu = 15 \cdot 10^{-6} \text{ m}^2/\text{s}^2$

Both test configurations were simulated using the provided mesh in the T3A tutorial case, which was described by [58] and is part of the OpenFOAM distribution [59]. These cases

were used since they are well-defined, all parameters are known, and a very good convergence behavior can be achieved. It has proved difficult to reproduce other CFD flat plate cases, often due to a lack of known input parameters.

A two-dimensional plate of length 3 m and height  $7.5 \cdot 10^{-4}$  m is discretized in a domain  $3.04 \text{ m} \times 1 \text{ m}$  with 26820 cells. The wall resolution is fine enough to ensure  $y^+ < 1$  for both cases. Figure 5.10 shows the meshed plate.

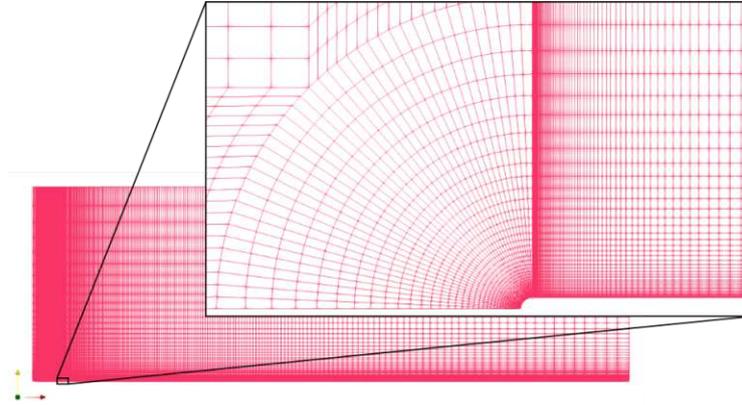


Figure 5.10: Mesh used in the flat plate simulations and detail near rounded leading edge.

The boundary conditions of the T3A case are identical to those from the OpenFOAM tutorial. Inlet values for  $k$  and  $Re_{\theta t}$  were computed with Eq. 3.11 and 3.32, respectively. Note that the value for  $\omega_{inlet}$  was calculated based on the viscosity ratio  $\nu_t/\nu$  according to:

$$\omega_{inlet} = \frac{k}{\frac{\nu_t}{\nu}}. \quad (5.1)$$

The flow over a smooth flat plate, i.e., with  $k_s = 0$ , was simulated first. Figure 5.11 shows the results of a laminar and a fully turbulent flow, as well as the results obtained with the Langtry Menter  $\gamma - \hat{R}e_{\theta t}$  transitional model with (LM\_AR) and without (LM) the roughness extension. The skin friction factor  $c_F$  is plotted against the Reynolds number  $Re_x$ :

$$Re_x = \frac{ux}{\nu}, \quad (5.2)$$

where  $x$  is the distance from the leading edge of the flat plate. The transition location is specified by  $c_F$  reaching a minimum before increasing rapidly. Both transitional models give quite similar results and are consistent with the experimental data, although the point of transition is predicted slightly upstream.

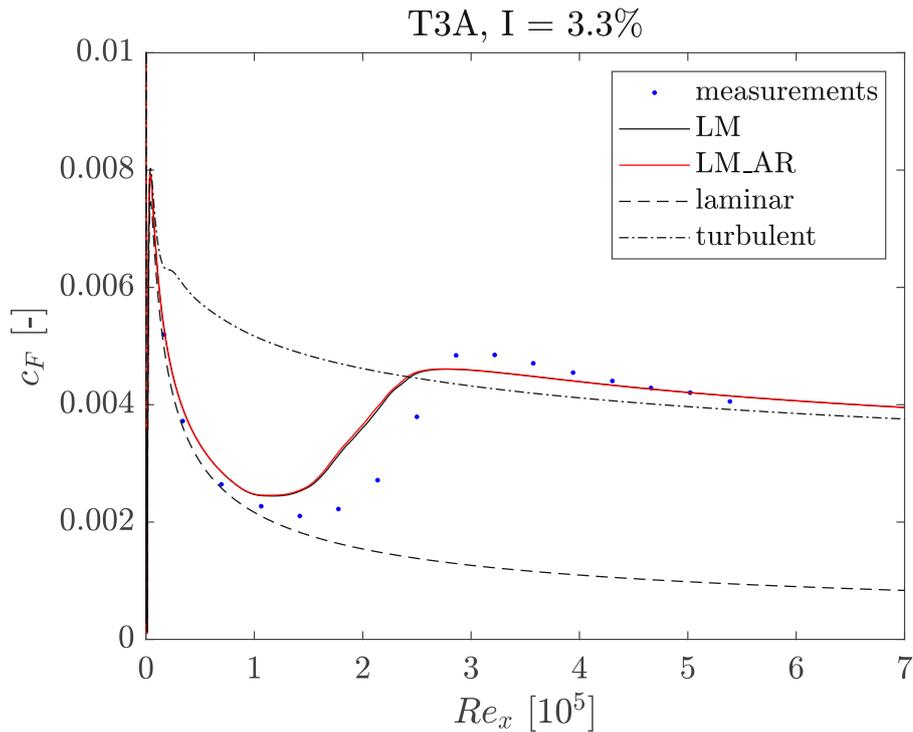


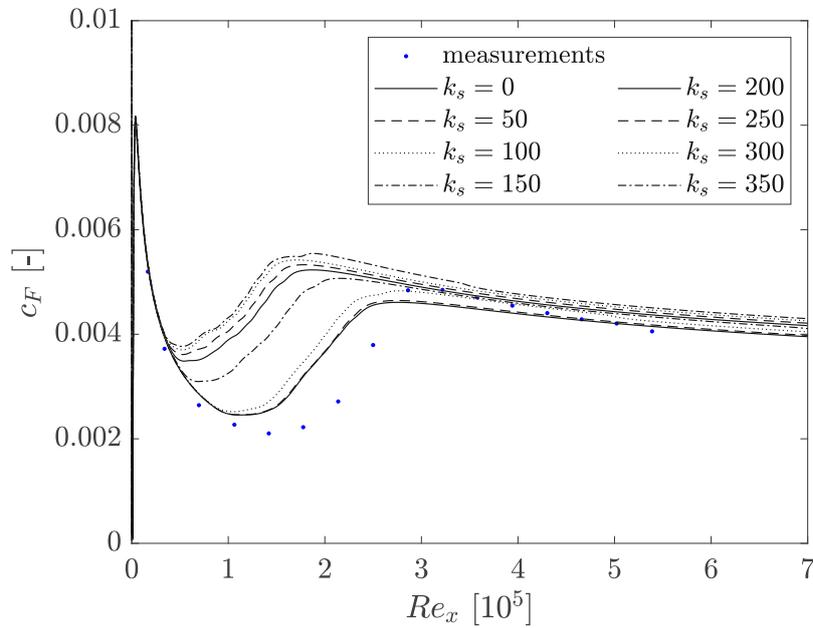
Figure 5.11: Visualization of the transition behavior on a smooth, flat plate. Measurements for the T3A test case are taken from [60].

The results for both test cases with roughness ranging from  $k_s = 0$  to  $350 \mu\text{m}$  are plotted in Fig. 5.12. The roughness was introduced on the whole plate by a boundary condition for the Amplification Roughness as described in Sec. 3.3. As expected, the model predicts the movement of the transition location further upstream with increasing roughness. However, without surface roughness ( $k_s = 0$ ), an upstream shift of the transition location compared to the experimental data is noticeable for both cases.

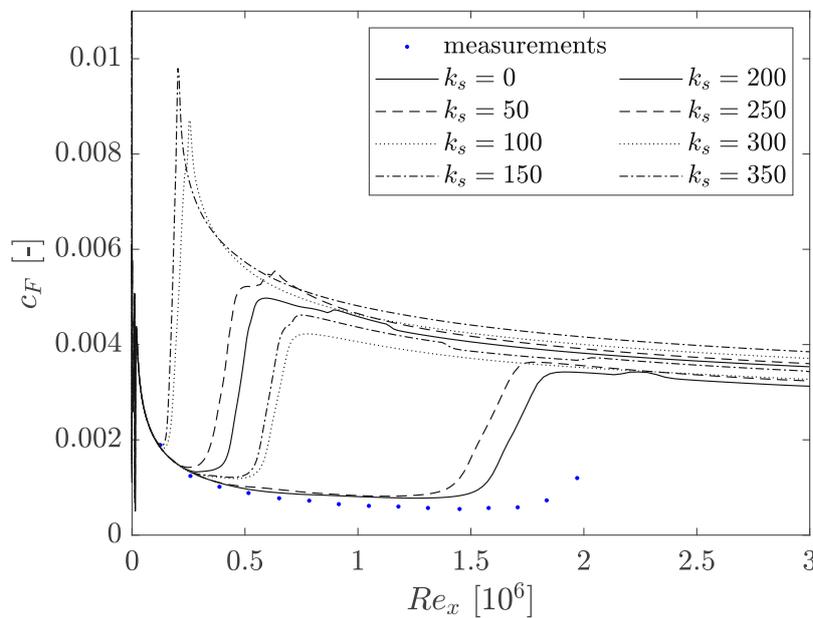
Another observable effect in both cases is the amplification of  $c_F$  with increasing  $k_s$  in the turbulent section following the transition. This increase originates from the modification of the boundary condition for  $\omega$  at rough walls [14].

The influence of small roughness heights on the transition location is different in the two

cases; Fig. 5.12a shows only minor differences for  $k_s < 100 \mu\text{m}$ . For the T3A- case, the same could be said for  $k_s < 50 \mu\text{m}$  only.



(a) T3A,  $I = 3.3 \%$



(b) T3A-,  $I = 0.874 \%$

Figure 5.12: Simulated influence of varying roughness  $k_s [\mu\text{m}]$  on transition location. Measurements of the (a) T3A and (b) T3A- cases are both taken from [60].

### 5.3.2 Airfoil Cases

The flat plate tests described above give some insight into the general behavior of the Amplification Roughness model. However, the input values of  $k_s$  were chosen arbitrarily, and a closer examination of the equivalent sand grain roughness as an input parameter is necessary. Additional tests on a NACA 63<sub>3</sub>-418 airfoil have been conducted to validate the model further. The reference cases were taken from Langel et al. [14], who provided both numerical and experimental results for several roughness configurations and Reynolds numbers in the validation process of their Amplification Roughness model.

As outlined in Sec. 3.3, the determination of the roughness parameter  $k_s$  is not straightforward and, in this case, depends on the density distribution and the calibration process. The input for the reference cases was “selected by determining what input produces the best output” [14]. That is to say that the input parameter is not universal and should be related to experimental data for each case. The input values  $k_s$  listed in Tab. 5.6 were used to test the current implementation of the Amplification Roughness model against the original numerical and experimental results. A NACA 63<sub>3</sub>-418 airfoil is investigated at  $Re = 1.6 \cdot 10^6$  and  $3.2 \cdot 10^6$ . The Reynolds number was changed by varying the free-stream velocity. Surface roughness is applied from the leading edge until 2%*c* on the upper surface and 13%*c* on the lower surface.

$k_r$	Density	$f(\Lambda_k)$	$k_s$
[ $\mu\text{m}$ ]	[%]	[-]	[ $\mu\text{m}$ ]
100	15	0.47	57
200	3	0.41	101

Table 5.6: Summary of roughness configurations, taken from [14].

Figure 5.13 compares the simulated lift and polar curves for  $Re = 1.6 \cdot 10^6$  and  $3.2 \cdot 10^6$ . In general, the lift coefficient decreases - primarily in the stall area - from its smooth reference, while the drag coefficient increases. That is well observable in the  $Re = 3.2 \cdot 10^6$  case, where increased roughness leads to larger deviations. A leading-edge roughness  $k_s = 101 \mu\text{m}$  leads to a significant drag increase in the linear lift region.

The  $Re = 1.6 \cdot 10^6$  results show that a Reynolds number dependency is evident, e.g.,  $k_s = 57 \mu m$  has no relevant effect on the airfoil characteristics, but it has at the higher  $Re$  case. To summarize, the effects of roughness (lift decrease, drag increase) are enhanced with increased roughness height and Reynolds number.

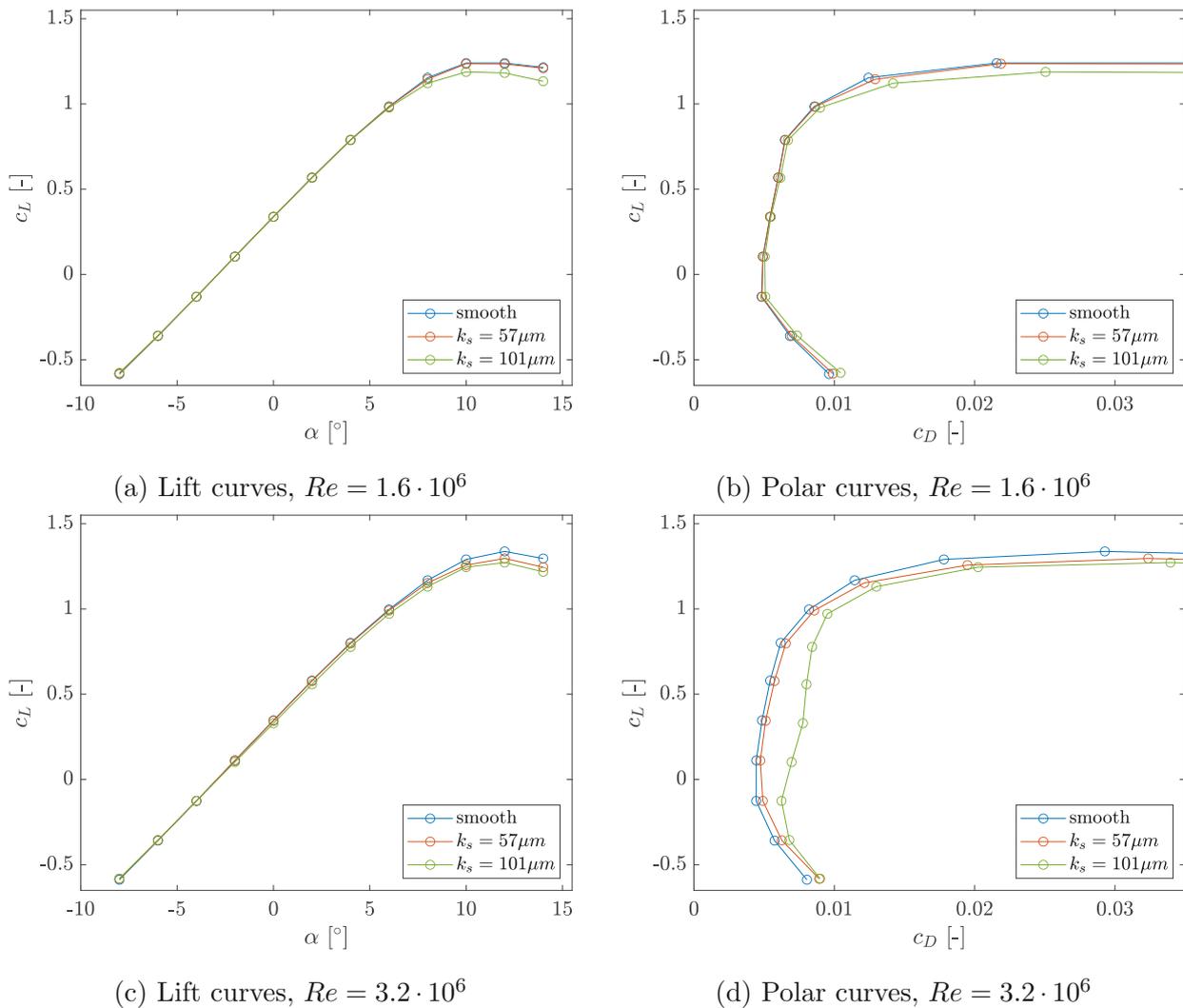
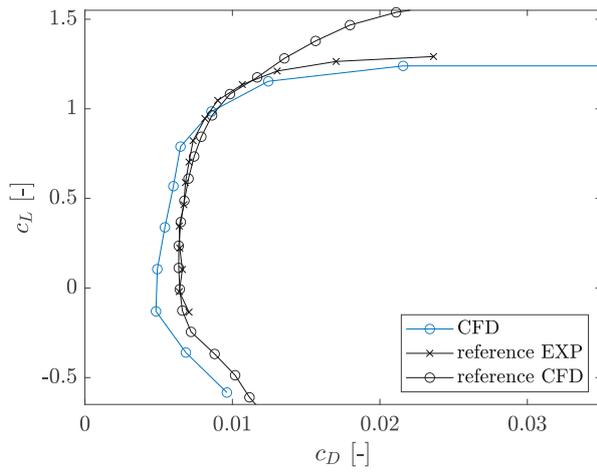
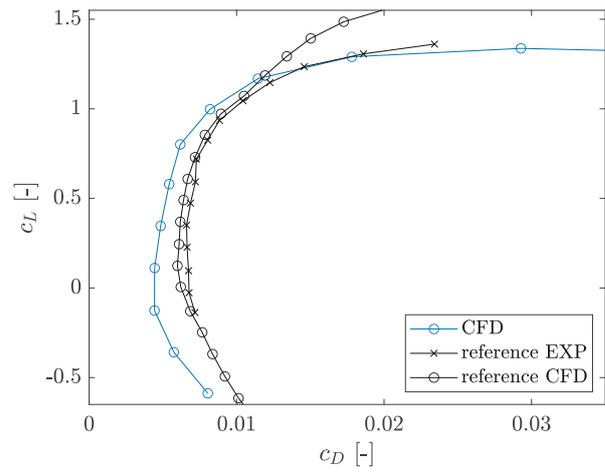


Figure 5.13: Comparison of NACA 63<sub>3</sub>-418 airfoil characteristics with different roughness configurations and Reynolds numbers, roughness applied from  $-13\%c$  to  $2\%c$ .

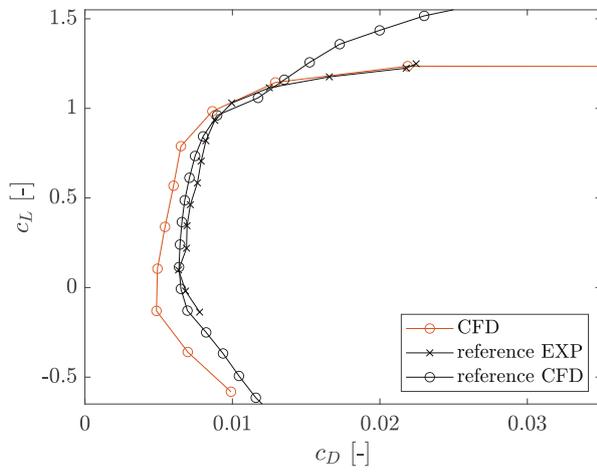
Furthermore, the results are compared to numerical and experimental data from the reference case [14]. (Note that the reference data had to be interpolated.) In the high-lift region, the simulated coefficients agree well with the experimental data, but a general underprediction of the drag is noticeable for all configurations (Fig. 5.14).



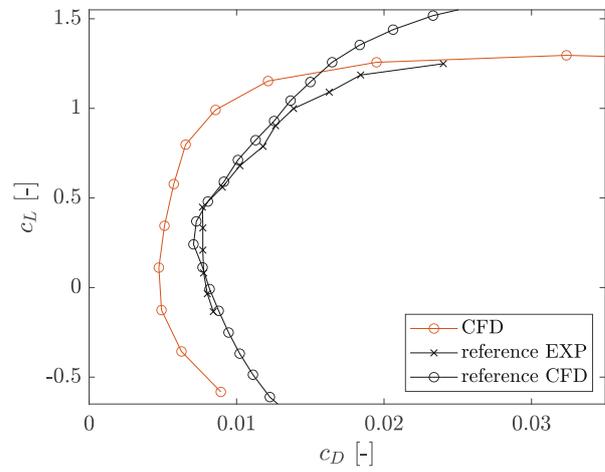
(a) smooth,  $Re = 1.6 \cdot 10^6$



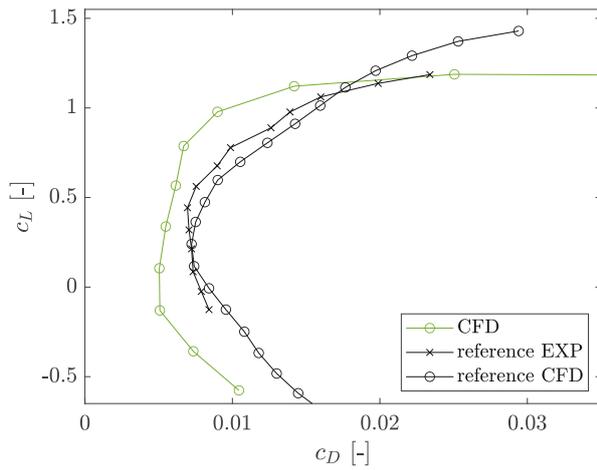
(b) smooth,  $Re = 3.2 \cdot 10^6$



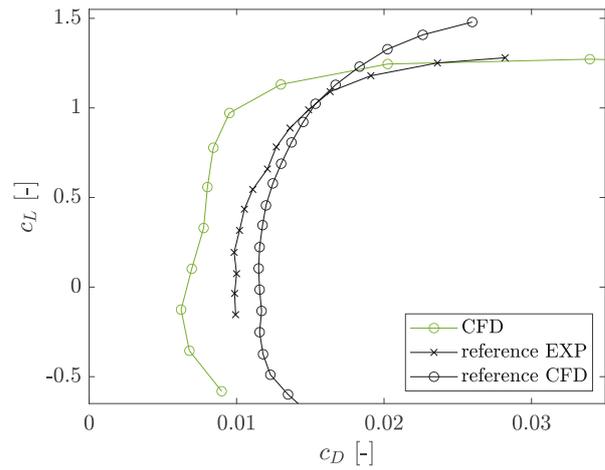
(c)  $k_s = 57 \mu m$ ,  $Re = 1.6 \cdot 10^6$



(d)  $k_s = 57 \mu m$ ,  $Re = 3.2 \cdot 10^6$



(e)  $k_s = 101 \mu m$ ,  $Re = 1.6 \cdot 10^6$



(f)  $k_s = 101 \mu m$ ,  $Re = 3.2 \cdot 10^6$

Figure 5.14: Comparison of simulated drag polars with reference numerical and experimental data from [14] and roughness applied from  $-13\%c$  to  $2\%c$ .

While the drag increases with increasing roughness, it seems that its extent is underestimated. That can be seen in Fig. 5.15a, where the drag of the rough cases at  $Re = 3.2 \cdot 10^6$  is related to the smooth surface case according to Eq. 5.3. The simulated drag increase in the low-lift region ( $c_L < 0.5$ ) shows similar behavior as shown in the experimental data. In the high-lift region, however, the increase of  $c_D$  is underestimated for both roughness configurations.

$$\Delta c_D = \frac{c_{D|rough} - c_{D|smooth}}{c_{D|smooth}} \cdot 100. \quad (5.3)$$

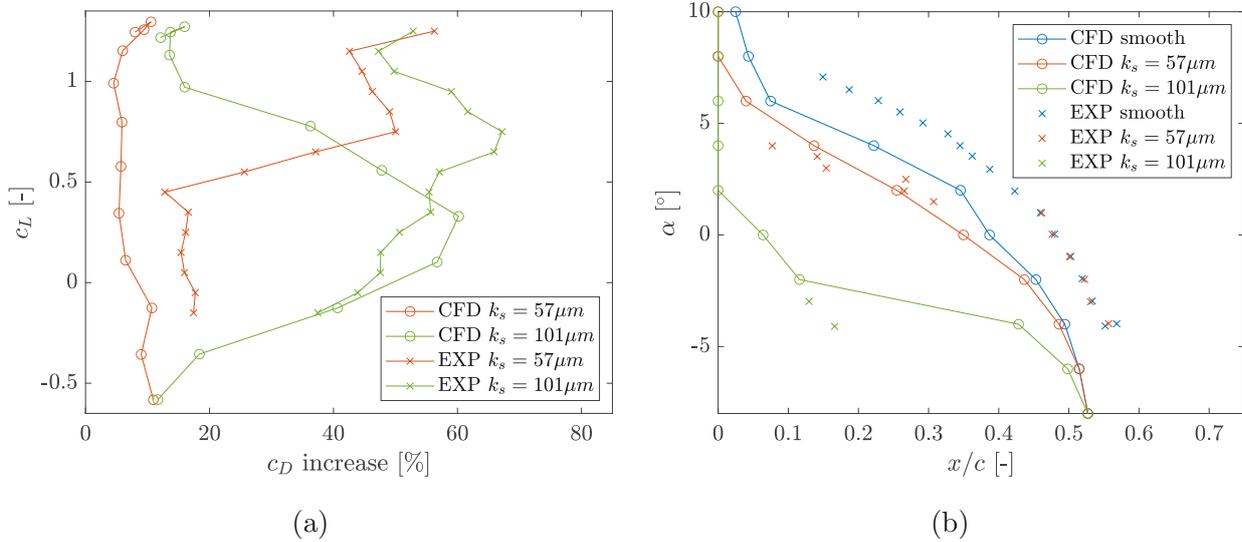


Figure 5.15: NACA 63<sub>3</sub>-418 at  $Re = 3.2 \cdot 10^6$ ; a) rough drag increase relative to the smooth case; b) location of boundary layer transition.

The location of boundary layer transition - where the laminar boundary layer transitions to a turbulent state - was defined as the point where the skin friction factor  $c_F$  reaches a minimum before increasing rapidly. Numerical results along with experimental data are depicted in Fig. 5.15b.

The expected upstream shift of the transition location with increasing roughness is captured well. An angle of attack, where the transition onset starts to move from the smooth reference towards the leading edge, can be identified and roughly matches the experimentally determined location. Note that with increasing roughness, this “critical” angle decreases. Below  $\alpha_{crit}$ , roughness does not notably influence the location of transition. At  $\alpha = -8^\circ$ , the transition location is predicted at  $x/c \approx 0.52$  for both roughness configurations and the smooth surface case. However, it should be noted that the transition location is predicted

too early for the smooth surface case in general and for the rough cases at  $\alpha < \alpha_{crit}$ . The results show the overall flow transition behavior due to roughness and are considered sufficient for this thesis, but it is believed that they can be improved. As outlined in detail by Langel et al. [14], adding the Kato-Launder production limiter to the SST  $k-\omega$  turbulence model leads to superior transition predictions. The limiter was “designed to address the overproduction of  $k$  near a stagnation point and in regions of recirculation” [61]. Without the limiter, overproduction of  $k$  leads to lower levels of  $\hat{Re}_{\theta t}$  in the boundary layer, which moves the transition onset forward [14].

At the tip of the rotor blade investigated in Chap. 6, Reynolds numbers beyond  $3.2 \cdot 10^6$  occur. Therefore, simulations at  $Re = 5 \cdot 10^6$  were carried out with the results shown in Fig. 5.16. Again, as seen in previous simulations, the drag coefficient of the smooth airfoil is predicted to be too low, while the lift coefficient agrees well with the experimental data. Both values of  $k_s$  yield the expected output, with a more pronounced drag increase for  $k_s = 57 \mu m$  compared to cases with lower Reynolds numbers. Unfortunately, no suitable rough airfoil wind tunnel measurements could be found for this setup. Apart from the drag underestimation, which is insignificant due to the comparative nature of this study, the model shows the expected behavior and is deemed suitable for further application.

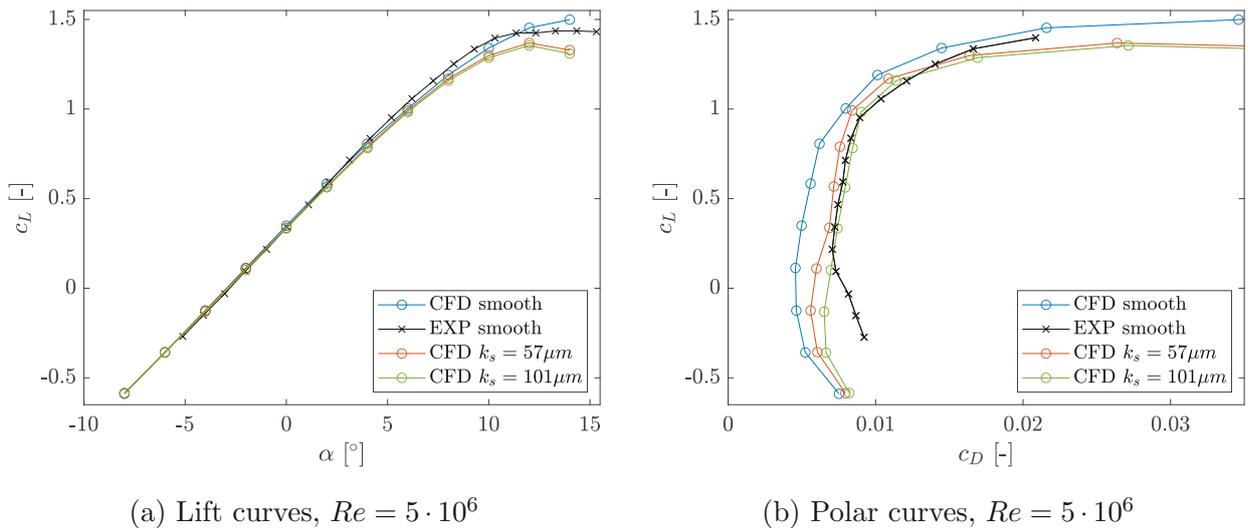


Figure 5.16: Simulated characteristics of a NACA 633-418 airfoil with roughness applied from  $-13\%c$  to  $2\%c$ , experimental data from [19].

## 5.4 Delamination Model

The delamination of the leading edge coating was modeled into the airfoil surface by changing the leading edge's shape to model a severely eroded blade. Two parameters that define the extent of the delamination - the depth  $d$  and the chord-wise extent  $s$  - are visualized in Fig. 5.17. For the purpose of modifying the geometry, a MATLAB script that allows the generation of parameterized geometries was developed. The script takes the coordinates of the original airfoil and the parameters  $d$  and  $s$  as input. All points located between the leading edge and  $s$  are then moved by  $d$  in the surface normal direction.

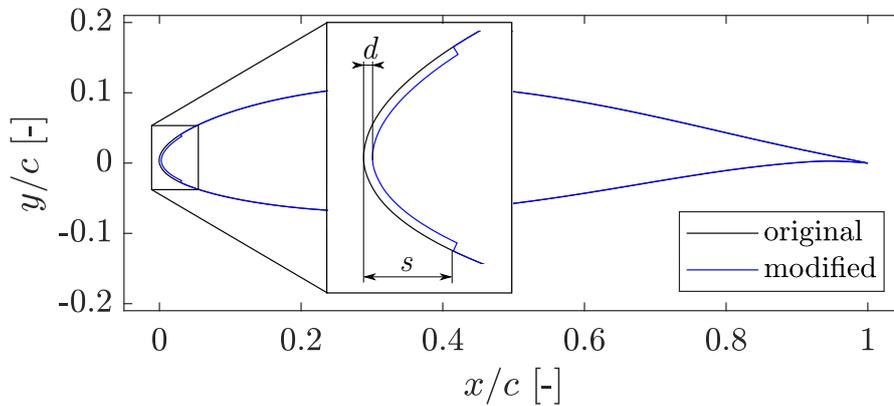


Figure 5.17: Original and modified contour of the NACA 63<sub>3</sub>-418 airfoil.

The results presented in this section were obtained for an eroded NACA 63<sub>3</sub>-418 airfoil with  $s = 3\%c$  and  $d = 0.3\%c$ . These values were chosen to match the experimental setup from [19]. It should be noted that the delamination feature at  $s = 3\%c$  is modeled as a forward-facing step with sharp 90° corners.

Generally, the simulations were carried out using the same boundary conditions and settings described in Sec. 5.2. Based on the results of the mesh independence study, the *fine2* grid was used. Due to the alteration of the airfoil surface, the mesh had to be changed by adding a block in the delaminated section, which is shown in Fig. 5.18. A cell-count increase of 6% or  $\approx 12k$  cells is the consequence. Note that with this modeling approach, for every geometry alteration, the mesh has to be manually adjusted as well. Developing a more automated process could be of interest in the future.

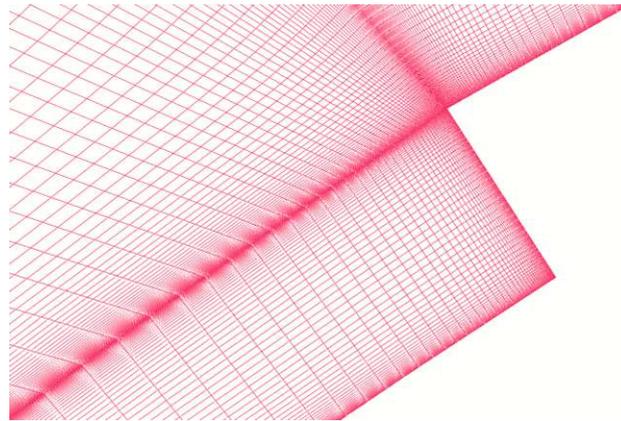
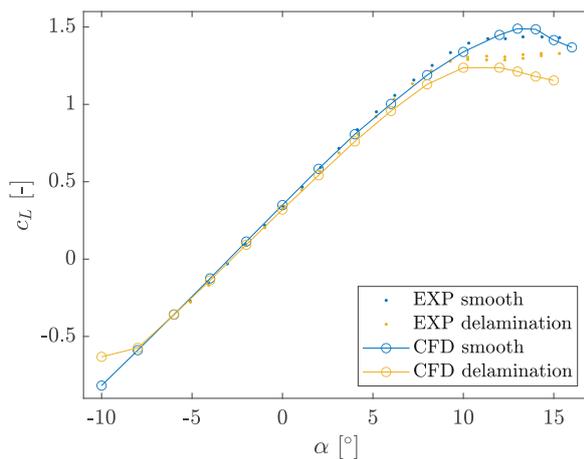


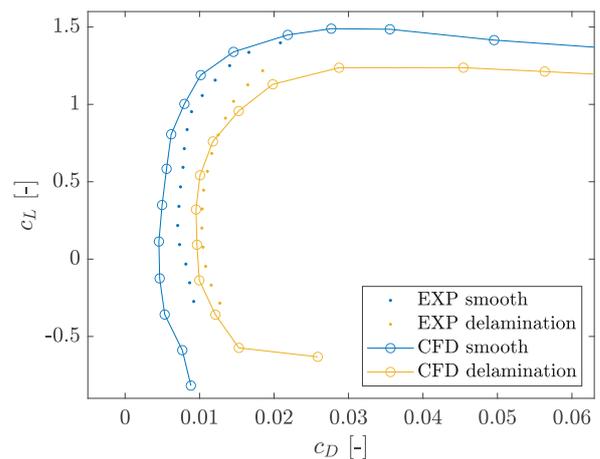
Figure 5.18: A detailed view of the *fine2* grid in the region of the step.

Simulations for  $\alpha$  ranging from  $-10^\circ$  to  $15^\circ$  have been carried out at  $Re = 5 \cdot 10^6$ , with the results depicted in Fig. 5.19. In general, the computational results of the delamination case show good agreement with experimental data. Especially in the linear lift region, the agreement is excellent. Unfortunately, no values of the drag coefficient for high angles of attack were recorded in the wind tunnel experiment [19].

The computed lift coefficient matches the experimental data well; only at very high (and low) angles of attack  $c_L$  starts to deviate from the experimental values. A reduction of the maximum lift by 14 % is observed in the numerical results.



(a) Lift curves,  $Re = 5 \cdot 10^6$



(b) Polar curves,  $Re = 5 \cdot 10^6$

Figure 5.19: Comparison of numerical results and experimental data from [19] for a NACA 63<sub>3</sub>-418 airfoil with a delaminated leading edge.

Figure 5.20 shows the pressure distribution and the velocity field in the delaminated area of the airfoil. It can be observed that the step disrupts the pressure recovery on the upper surface and causes the formation of a separation bubble downstream. The separation bubbles lengthen towards the trailing edge as the angle of attack increases.

Further investigation of the influence of the edge shape might be of interest, considering that in reality, the shape of the edge will most likely differ from a 90° step.

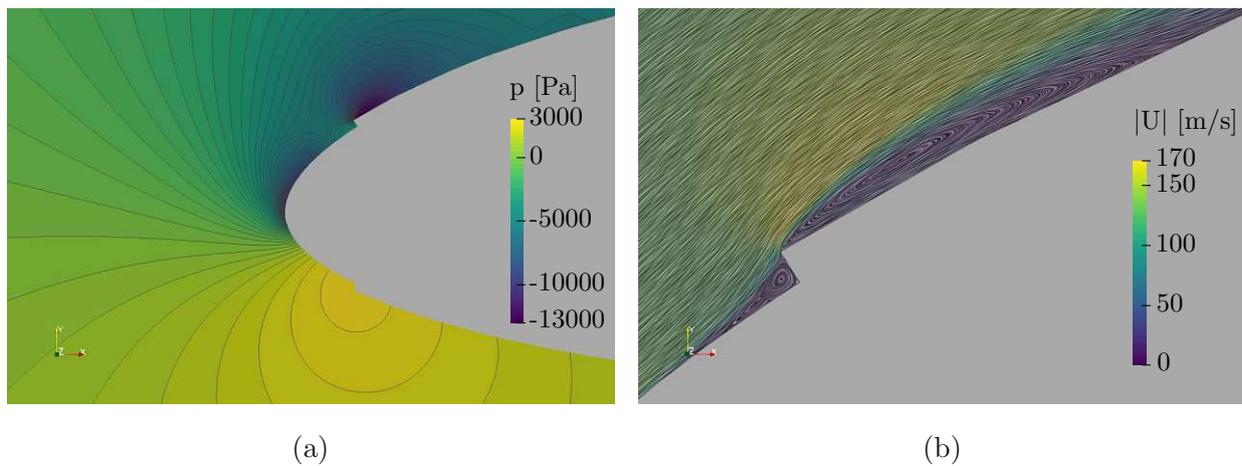


Figure 5.20: The a) pressure contour and b) velocity field for a delaminated NACA 63<sub>3</sub>-418 airfoil at  $Re = 5 \cdot 10^6$  and  $\alpha = 12^\circ$ .

## 5.5 Model Comparison and Transient Results

In this section, the simulated effects of different degrees of LEE are compared to each other and the smooth airfoil. The LEE configurations investigated include the following:

1. Delamination of the leading edge up to  $3\%c$  as described in Sec. 5.4.
2. Rough leading edge with  $k_s = 57 \mu\text{m}$  from  $-13\%c$  to  $2\%c$  as described in Sec. 5.3.
3. A combination of 1. and 2.

Figure 5.21 shows the results of the steady-state computations. Both the delamination and roughness models reveal the expected results on their own, in the sense that the aerodynamic performance is reduced most in the case of the delaminated leading edge. That is reflected in a greater drag increase and a further decrease in lift compared to the case with roughness ( $k_s = 57 \mu\text{m}$ ).

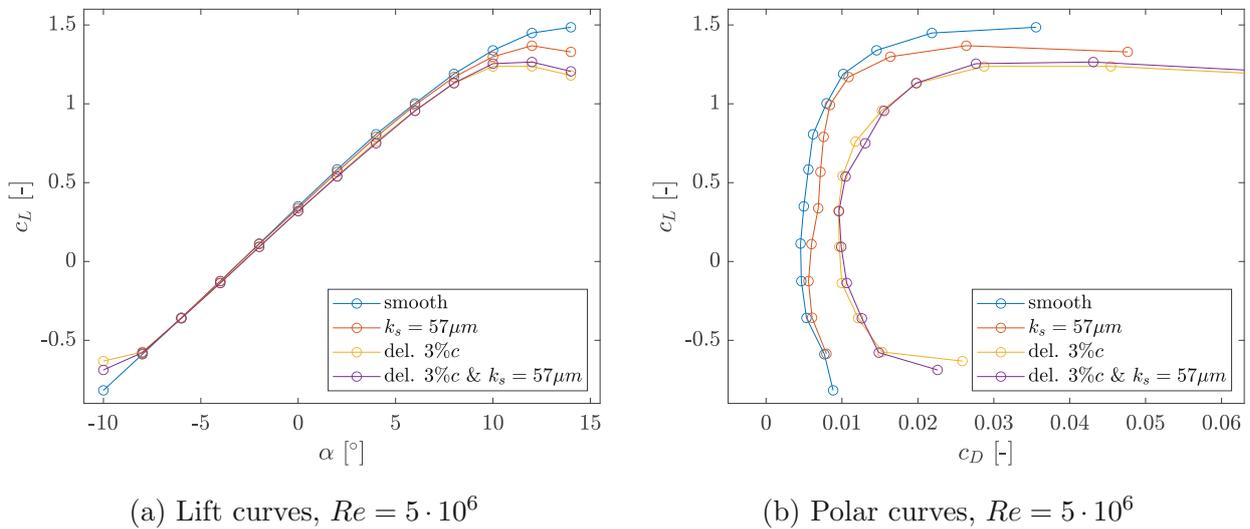


Figure 5.21: Comparison of the results of several LEE models on a NACA 633-418 airfoil.

The combination of the two models, however, produces unexpected results. Due to the additional roughness, one would presume a further reduction of the lift coefficient compared to the smooth delaminated case. Instead,  $c_L$  is almost the same for the majority of the

angles of attack. At  $\alpha > 10^\circ$ , the lift coefficient even exceeds the values of the smooth delaminated case. The drag coefficient also shows counter-intuitive behavior. In the “drag bucket”, the values for  $c_D$  are more or less the same, but at higher angles, it tends to be underestimated, which is visualized in Fig. 5.22.

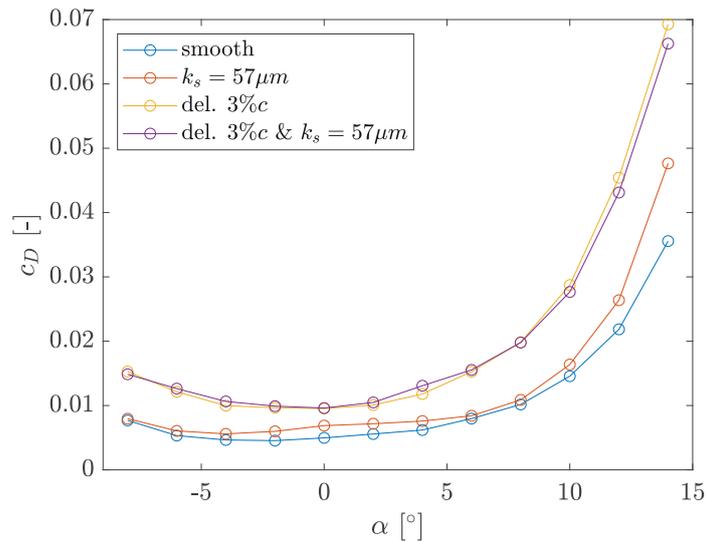


Figure 5.22: Comparison of LEE models by means of  $c_D$  at  $Re = 5 \cdot 10^6$ .

The pressure distribution on the surface of the profile is used to examine the results at a more in-depth level. Fig. 5.23 shows the simulated pressure distribution through  $c_p$  for  $\alpha = 12^\circ$ . That angle was chosen since the combination of roughness, and delamination models show “unexpected” results at large angles of attack especially.

On the lower surface of the airfoil, there are only minor differences between the four configurations, i.e.,  $c_p$  is highest for the smooth airfoil profile and generally decreases slightly with increasing damage. The situation is different on the upper surface, where the differences are clearly visible - in particular, the pressure spike caused by the step at 3% $c$ .

Looking at the section between the step and the point where an adverse gradient starts (at  $x/c \approx 0.6$ ), one can see that  $c_p$  of the rough case ( $k_s = 57 \mu m$ ) *increases* slightly from the smooth reference value. In other words, the higher pressure on the upper surface and the lower pressure on the lower surface reduce the performance of the airfoil with a rough leading edge.

However, the roughness model in combination with delamination leads to contrasting results, i.e., the pressure on the upper surface is slightly *lower* than in the case of the smooth, delaminated surface. That leads to the aforementioned increase in the lift coefficient.

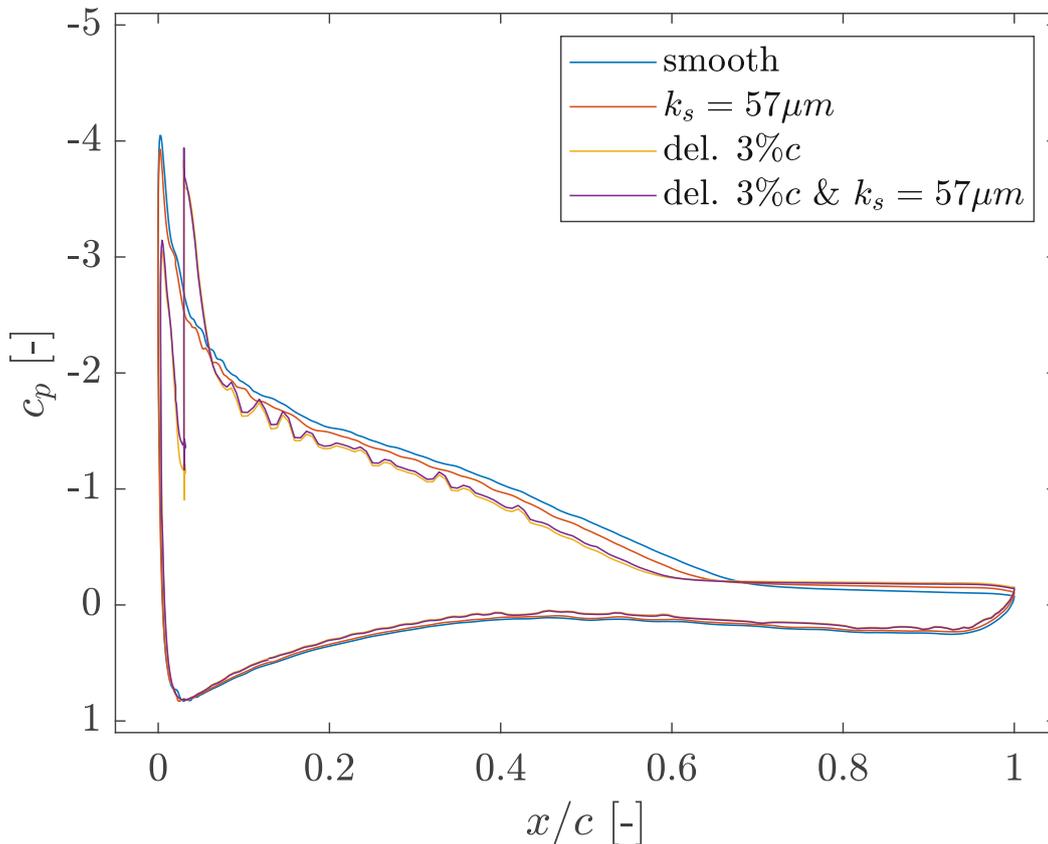


Figure 5.23: Comparison of the pressure coefficient  $c_p$  between different LEE models at  $\alpha = 12^\circ$  and  $Re = 5 \cdot 10^6$ .

The separation bubbles downstream of the step for the smooth and rough delaminated leading edges are compared to investigate the origin of the lift coefficient differences. Figure 5.24 shows the results at  $\alpha = 12^\circ$  and reveals that the rough-surface bubble with a length of 30.8 mm is shortened by  $\approx 11\%$  compared to the smooth case. The length of the bubble is defined as the distance from the step to the point where the flow reattaches.

This trend of the Amplification Roughness model producing a reduced separation bubble size continues for high angles of attack. Below  $\alpha = 8^\circ$ , this trend reverses and the separation bubble tends to be larger for the rough surface.

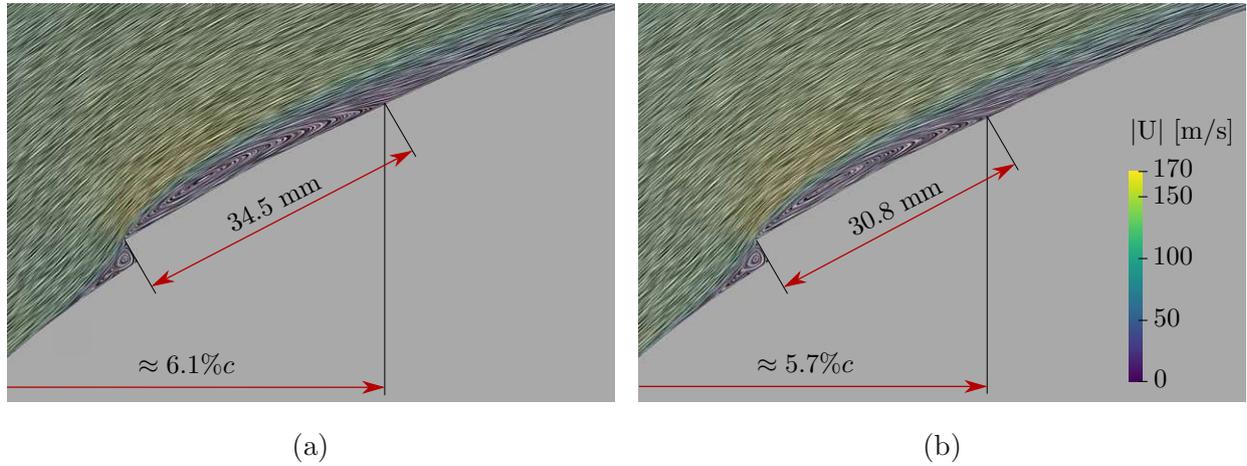


Figure 5.24: The separation bubble caused by the step for a delaminated NACA 63<sub>3</sub>-418 airfoil with a) smooth and b) rough surface at  $Re = 5 \cdot 10^6$  and  $\alpha = 12^\circ$ .

Due to the unsteady nature of the flow at higher angles of attack, unsteady RANS simulations were carried out at  $\alpha = 12^\circ$ . At this angle, pressure fluctuations on the airfoil surface and, therefore, unsteady behavior of the lift- and drag coefficients are expected. The LEE configurations examined are the same as those listed above, except that the combination of the roughness and delamination models was discarded as it failed to deliver consistent results in the steady-state simulations. In general, the simulations were set up as previously described. Only the time marching algorithm is now a transient one, namely *pimpleFoam*. Relevant settings for the time-discretization are summarized in Tab. 6.5:

Subject	Property
Algorithm/Solver	pimpleFoam
Time discretization scheme	Crank-Nicolson 0.5
Time step	$1 \cdot 10^{-6}$ s
max. Courant number $C$	0.95
Simulation time	5.5 ms

Table 5.7: Time-related simulation settings.

The Crank-Nicolson discretization scheme is a second-order, implicit scheme. In the present case, it uses a blending factor of 0.5, which means that the pure Crank-Nicholson scheme is blended with the Euler scheme to improve stability.

Figure 5.25 shows the computed lift coefficients for the smooth, rough, and delaminated airfoil over time. For presentation purposes, the data were smoothed using the Savitzky-Golay filter.

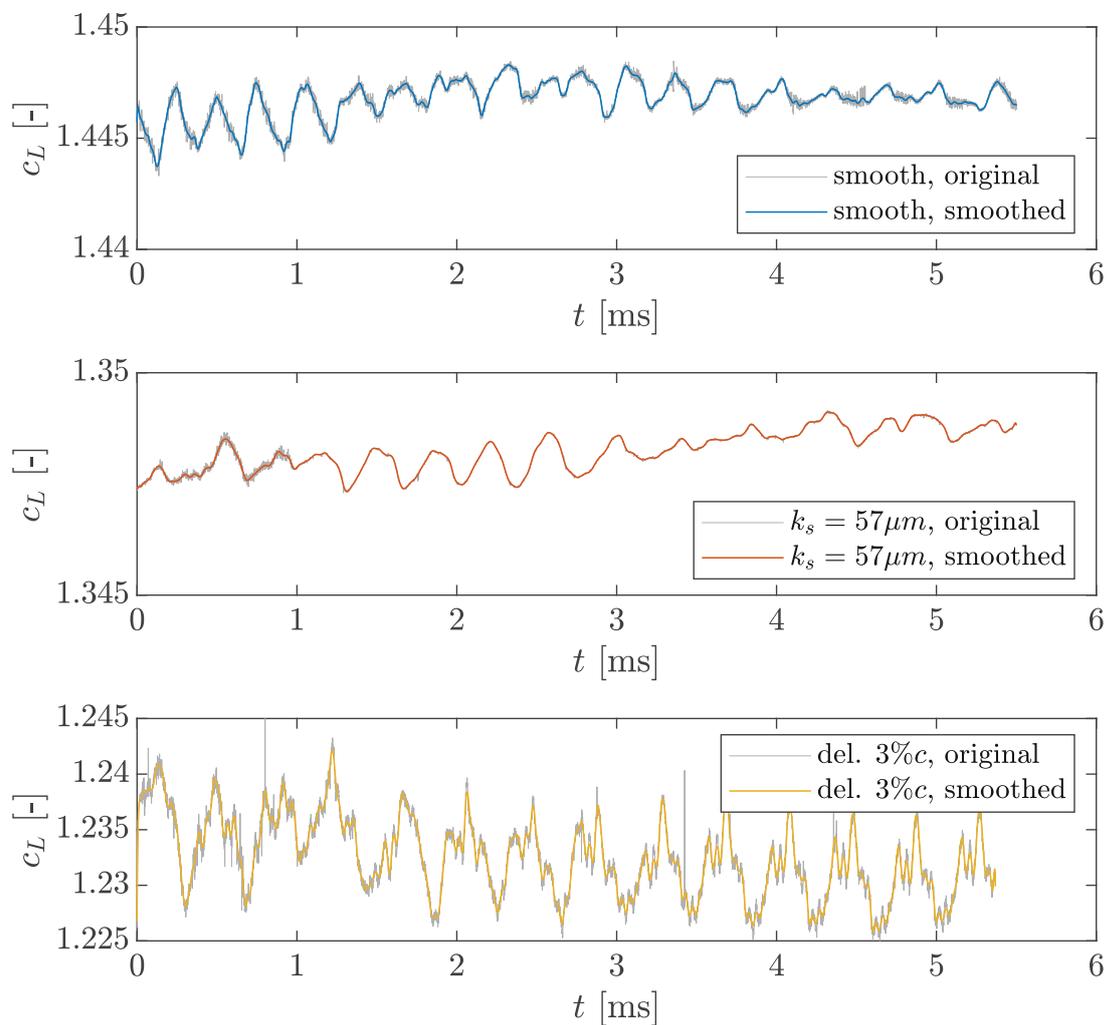


Figure 5.25: Simulated lift coefficient  $c_L(t)$  for a smooth, rough and delaminated NACA 63<sub>3</sub>-418 airfoil.

In all three cases, some fluctuations with periodic components occur but at different amplitudes. The objective is to determine the amplitudes and frequency components of the signal. This can be achieved with the use of the Fourier transformation. The discrete Fourier transform was computed from the original data with MATLAB using the built-in fast Fourier transform (FFT) algorithm.

The results are plotted in Fig. 5.26 as the single-sided amplitude spectrum of  $c_L(t)$ . Note that the amplitude  $|P_1|$  is related to the corresponding mean lift coefficient  $\bar{c}_L$  and plotted as relative amplitude  $A$ .

The lowest frequency peaks, as well as the frequencies connected to the largest amplitudes, are highlighted in the diagram and listed in Tab. 5.8. First of all, the amplitudes of the delamination case are significantly larger than for the other two cases. One can conclude that such a severely damaged profile is subject to increased material fatigue due to the higher amplitudes. The rough surface case, on the contrary, shows only slight deviations from the smooth surface case. Most notably, the amplitude of the high-frequency component is reduced; the frequencies have also decreased compared to the smooth surface case. Thus, the Amplification Roughness Model with the current setup does not confirm increased vibrational intensity for slightly eroded airfoils. The frequencies detected for both damaged airfoils are in a similar range but below those of the smooth ones. However, additional low-frequency components cannot be ruled out. Much longer simulation times are necessary for their determination.

Configuration	1 <sup>st</sup> Frequency		n <sup>th</sup> Frequency	
	$f$ [Hz]	$A$ [%]	$f$ [Hz]	$A$ [%]
smooth	168	0.046	3473	0.033
rough: $k_s = 57 \mu\text{m}$	127	0.042	2897	0.014
del.: $c = 3\%$	128	0.224	2524	0.269

Table 5.8: The identified amplitudes and frequencies for each case.

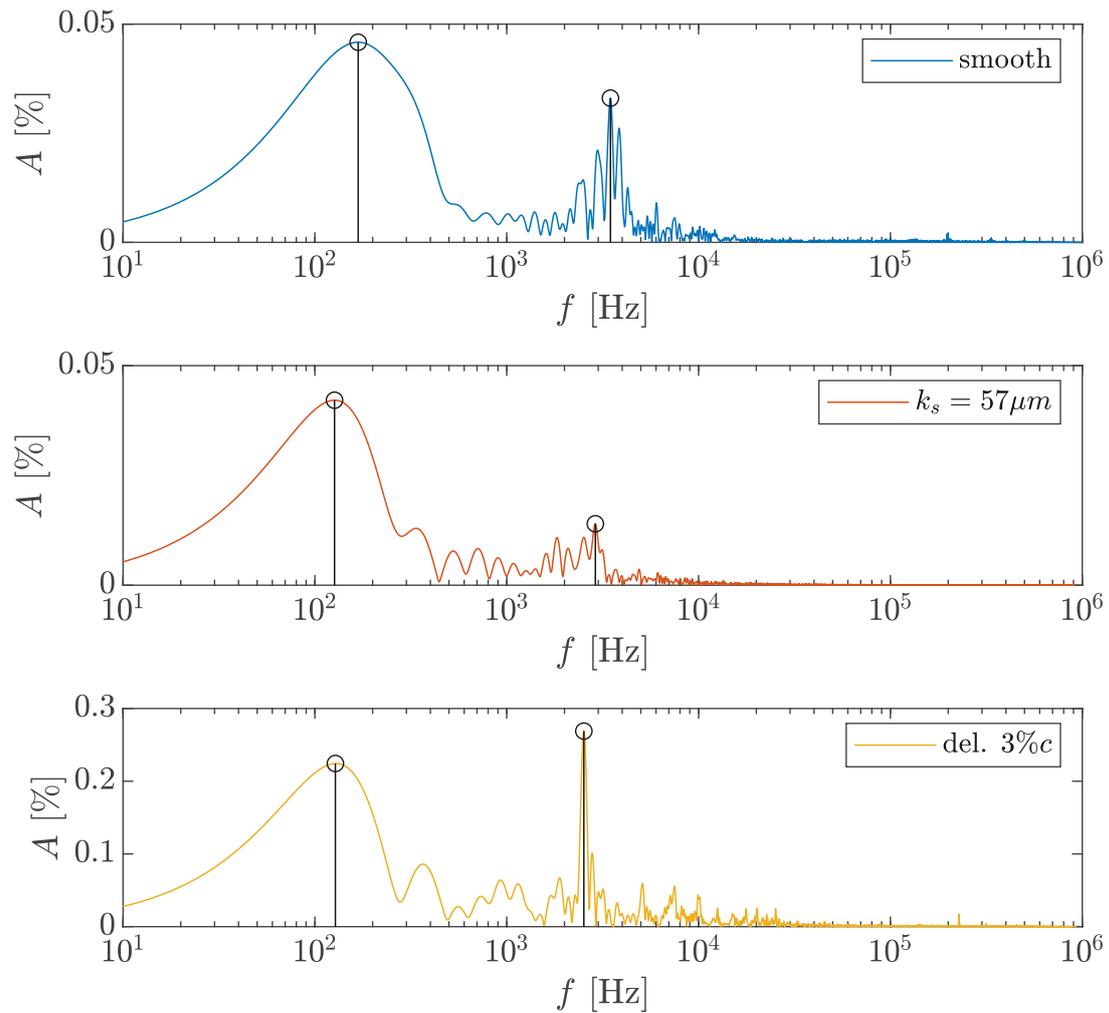


Figure 5.26: A relative amplitude  $A = |P_1|/\bar{c}_L$  and a corresponding frequency  $f$  can be identified for each roughness configuration.

# Chapter 6

## 3D Simulation

The WindPACT 1.5MW wind turbine model is used for the three-dimensional simulations. It is a conventional three-bladed HAWT and was designed as a reference wind turbine by National Renewable Energy Laboratory (NREL) during the WindPACT project [62, 63]. The turbines' main design parameters are summarized in Tab. 6.1. The blade geometry is included as a generic example in the rotor blade design tool NuMAD [64] and is obtained from there. The blade consists of a cylinder at the base and three airfoil profiles. It has a length of 33.25 m and is mounted on a hub with a radius of 1.75 m.

Parameter	Quantity
Rotor radius	35 m
Rated power	1.5 MW
Rated tip speed	75 m/s
Rated wind speed	11.5 m/s

Table 6.1: Main parameters of the WindPACT 1.5MW turbine.

The operating condition of the wind turbine for the simulations is taken from [65]. The wind speed, pitch angle, and rotor rotational speed are 8 m/s, 2.6°, and 15 rpm, respectively.

## 6.1 Setting up the Simulation

The three-bladed wind turbine has rotational symmetry. Therefore it can be modeled using a single blade in a  $120^\circ$ -cylindrical segment with periodic boundaries on the sides. A velocity inlet with a 120 m radius is placed 90 m upstream of the blade; the pressure outlet has a radius of 240 m. The dimensions and shape of the computational domain depicted in Fig. 6.1 are reproduced from Wang et al. [65] - the “conical shape allows for wake expansion” behind the blade. Note that the artificial hub extends over the whole domain and is modeled as a slip wall. A summary of the properties of the boundary patches is given in Tab. 6.2.

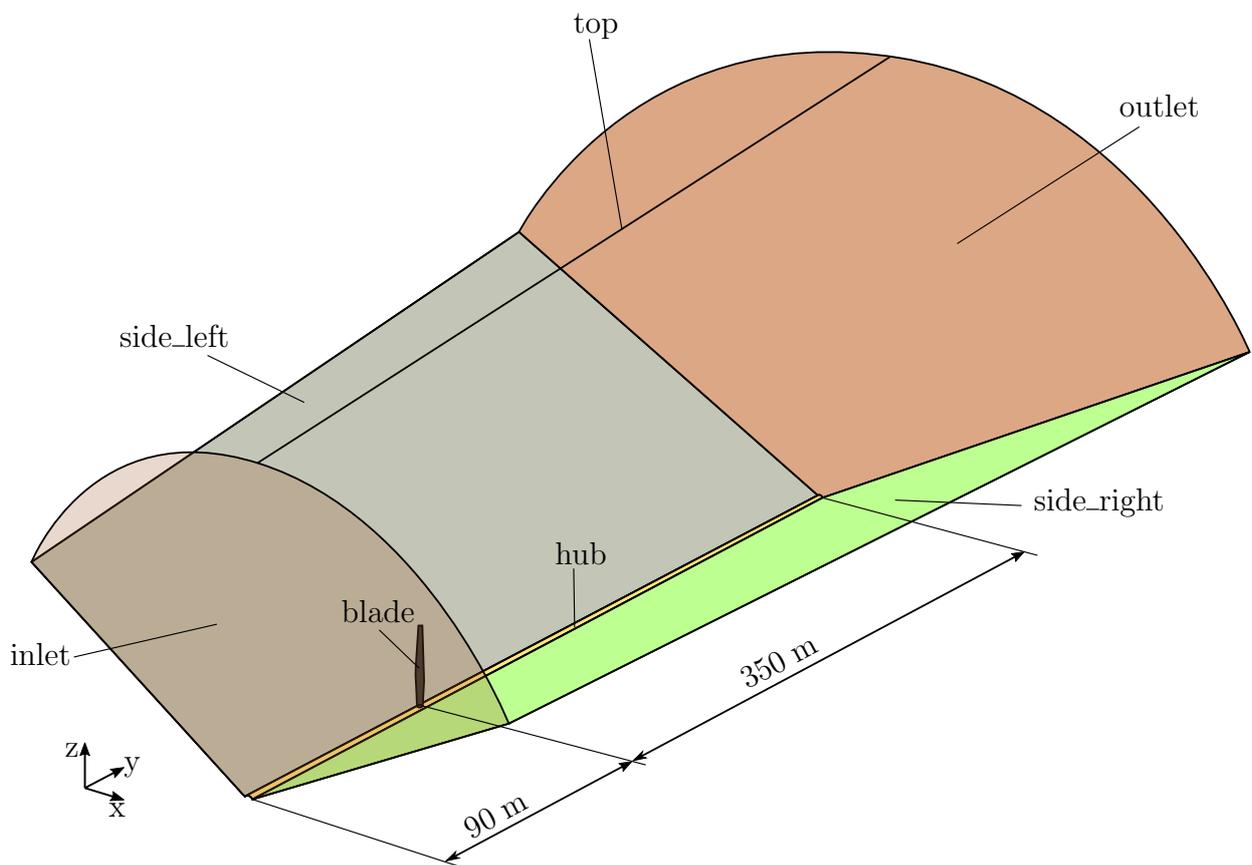


Figure 6.1: Computational domain and boundary patches of the 3D CFD simulation.

At the inlet, the turbulence variables  $k$  and  $\omega$  are set to  $6.4 \cdot 10^{-5} \text{ m}^2/\text{s}^2$  and  $16.36 \text{ 1/s}$ , respectively, resulting in a turbulence intensity of  $I = 0.08\%$ . The BC for  $\hat{R}e_{\theta t}$  and  $\gamma$  are set according to Eq. 3.32 and 3.31.

Patch label	Description
blade	Solid rotor blade, stationary wall (no-slip)
hub	Artificial hub, Euler wall (slip)
outlet	Atmospheric pressure outlet (fixedValue)
inlet	Velocity inlet (SRFVelocity)
top	Farfield (SRFFreestreamVelocity)
side_left	Rotational periodicity (cyclicAMI)
side_right	Rotational periodicity (cyclicAMI)

Table 6.2: Overview of boundary patch labels and their physical properties, including the corresponding BC in OpenFOAM.

A structured multi-block grid is created using Ansys ICEM CFD, following a similar blocking strategy described by Suarez et al. [66]. An H-type grid topology is used in the spanwise and free-stream directions. In the chordwise direction, the C-type grid is created analogous to the two-dimensional one described in Sec. 5.1. The grid topology is shown in Fig. 6.2. Note that the unstructured block above the blade tip is meshed with triangular prisms and hexahedrons as indicated in Fig. 6.3a. Structured hex-elements are used in the rest of the domain.

The mesh is refined in the leading and trailing edge regions and towards the blade tip (see Fig. 6.3a). The first cell height is set to  $d_1 = 4 \cdot 10^{-6}$  m. That gives  $y^+ < 1$  on both sides of the blade, which is visible in Fig. 6.3b. Note that on the blade tip surface,  $y^+ < 1$  is not satisfied. However, this aspect is neglected as this area is only of minor importance for the overall result.

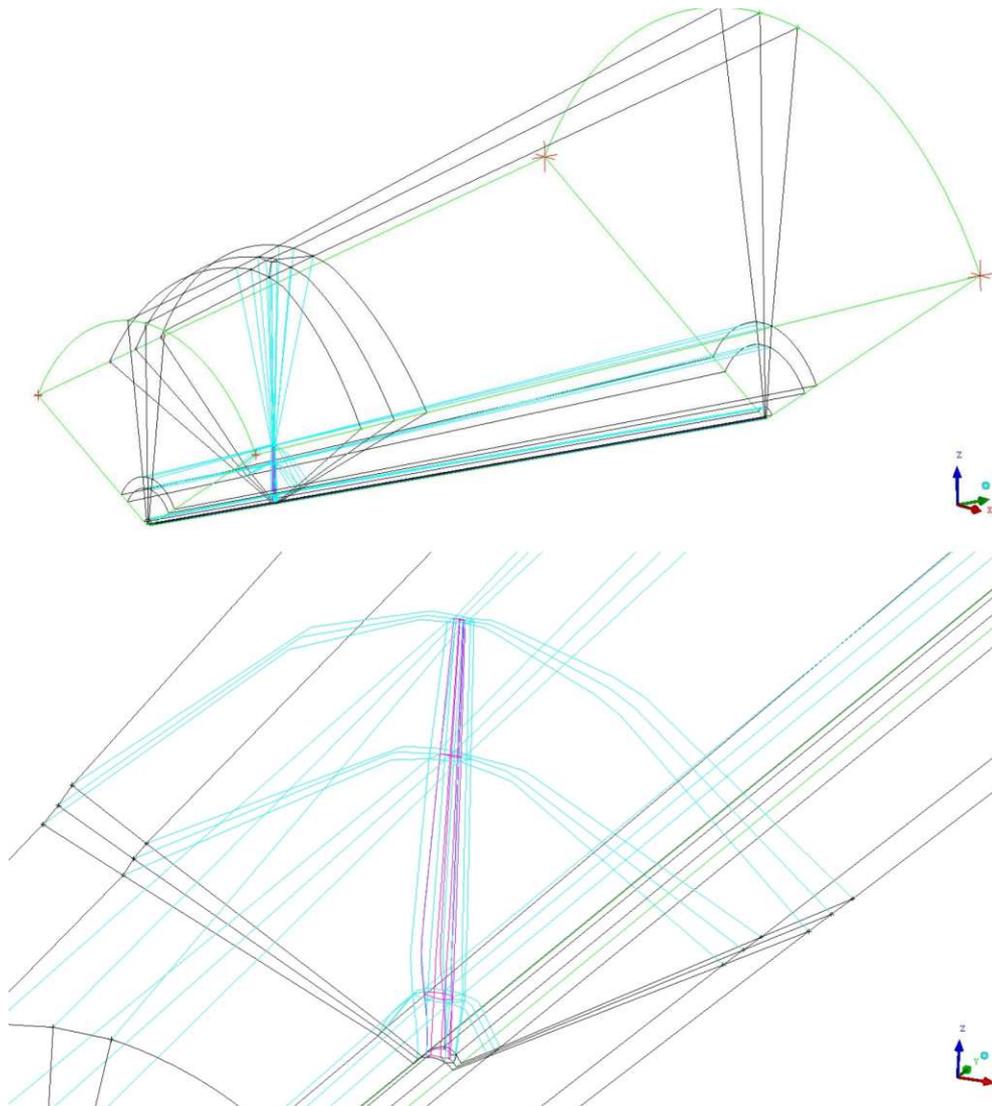


Figure 6.2: The grid topology around the WindPACT 1.5MW rotor blade consists of 75 blocks.



Figure 6.3: a) Surface mesh and b) contour of  $y^+$  in the blade tip region.

To account for rotation, the fluid flow is computed in a rotating frame of reference. The single rotating frame (SRF) approach eliminates the need to move the mesh and is, consequently, less computationally intensive. In OpenFOAM, the SRF model requires the *SRFProperties* file in the *constant* folder, which specifies the rotation axis, speed, and direction<sup>1</sup> (List. 6.1).

```

1   SRFModel      rpm;
2       origin      (0 0 0);
3       axis        (0 1 0);
4   rpmCoeffs
5   {
6       rpm         15;
7   }

```

Listing 6.1: *SRFProperties* entry, the rotational speed is specified in rounds per minute.

The velocity field is represented by the relative velocity  $\vec{u}_R$ , which can be expressed as:

$$\vec{u}_R = \vec{u}_I - \vec{\Omega} \times \vec{r}, \quad (6.1)$$

where  $\vec{\Omega}$  and  $\vec{r}$  are the rotational vector and position vector, respectively<sup>2</sup>. The *SRFVelocity* boundary condition associated with the rotating frame is prescribed at the inlet:

```

1   inlet
2   {
3       type        SRFVelocity;
4       relative    no;
5           // no: Urel = inletValue
6           // yes: Urel = inletValue - omega x r
7       inletValue  uniform (0 8 0);
8       value       uniform (0 8 0);
9   }

```

Listing 6.2: Specification of the inlet boundary condition in the *Urel* file.

<sup>1</sup>When the rotational axis vector points to you: rpm>0: clockwise rotation, rpm<0: counterclockwise rotation [67].

<sup>2</sup>In OpenFOAM, the relative velocity field is called *Urel*.

### 6.1.1 Grid Independence Study

A grid independence study was conducted on three meshes for a smooth, uneroded blade. The meshes were created so that the number of cells approximately doubled with each refinement step. Due to limitations of the available computational resources, a maximum cell count of 30 million was imposed.

The number of the span- and chordwise nodes were set according to the configuration shown in Tab. 6.3. Additionally, the number of cells normal to and downstream of the blade (in chordwise direction) was increased by 10% at each refinement step.

Label	Chordwise nodes	Spanwise nodes	Elements
coarse	200	175	$\approx 6.93\text{M}$
medium	420	200	$\approx 15.47\text{M}$
fine	640	250	$\approx 27.57\text{M}$

Table 6.3: Properties of the grids used in grid independence study.

The steady-state computations have been performed with OpenFOAM's *SRFSimpleFoam* algorithm. The convergence behavior of two parameters  $\Phi$  - the torque and the thrust - is assessed using the GCI-Method (see Appendix A). The main calculated uncertainty parameters are summarized in Tab. 6.4, and the convergence behavior of the two parameters normalized to the extrapolated values are shown in Fig. 6.4a. For both parameters, monotonic convergence can be assumed as indicated by  $0 < R < 1$ .

Parameter	Torque	Thrust
Convergence ratio $R$ [-]	0.269	0.233
Relative error $e_a^{21}$ [%]	13.919	5.427
Extrapolated error $e_{ext}^{21}$ [%]	10.175	3.489
Grid convergence index $GCI^{21}$ [%]	14.159	4.519

Table 6.4: Calculation of discretization uncertainties.

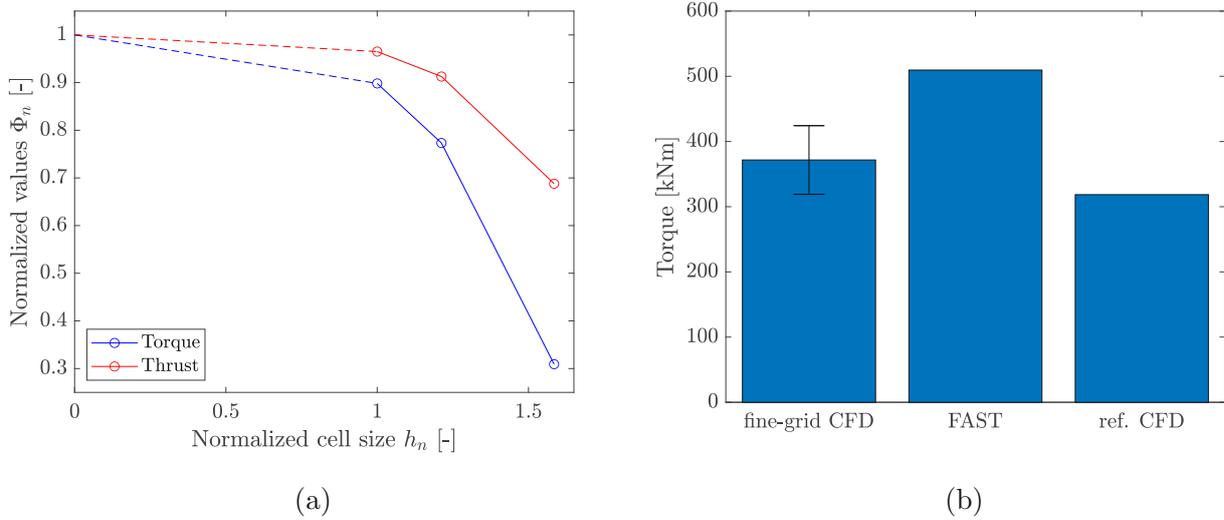


Figure 6.4: a) Normalized results of the grid independence study and b) Comparison of the fine-grid solution to results from FAST code [68] and reference CFD data [65].

Figure 6.4b shows reasonable agreement of the torque calculated on the fine grid with other numerical calculations. As a measure of uncertainty, the fine-grid GCI is presented as an error bar.

The high  $GCI^{21}$  values of 14% and 4.5% for torque and thrust indicate that the computation is not yet in the asymptotic range. It is believed that grid convergence improves with targeted grid refinement, especially concerning the number of chordwise nodes. The *coarse* and *medium* grids only have 200 and 420 nodes in the chord direction, while the two-dimensional airfoil grid studies presented in Sec. 5.2 suggest that 640 nodes are required to obtain accurate results at higher local angles of attack. However, for the comparative nature of the simulations presented in this chapter, the *medium* grid is considered sufficient and is therefore used for further analysis.

### 6.1.2 Setup of the Eroded Blade

The eroded section is assumed to be in the outermost 40% of the blade and stretches from the leading edge to  $10\%c$  in the chordwise direction (Fig. 6.5). Here, erosion is modeled using only the Amplification Roughness model, using the same input parameter  $k_s = 101 \mu\text{m}$  as in the two-dimensional simulations.

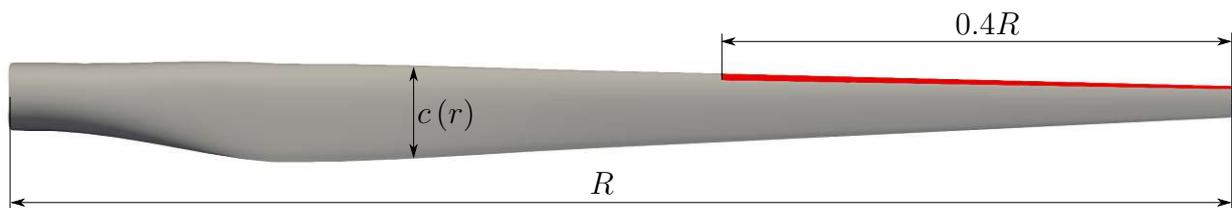


Figure 6.5: Smooth and eroded sections of the blade in grey and red, respectively.

A workflow has been developed that, starting from the existing mesh, makes it possible to easily modify the size of the eroded area (Fig. 6.6). A custom OpenFOAM utility has been programmed, which, after providing a patch name and the spanwise coordinates, writes a list of faces within that range. The source file of that utility *leadingEdge.C* is included in Appendix B. Then, a Matlab script uses the list to determine the faces which lie in the specified range from the leading edge ( $< 10\%c$ ). The *topoSet* and *createPatch* utilities are used to split the original patch and create the eroded and smooth patches of the blade.

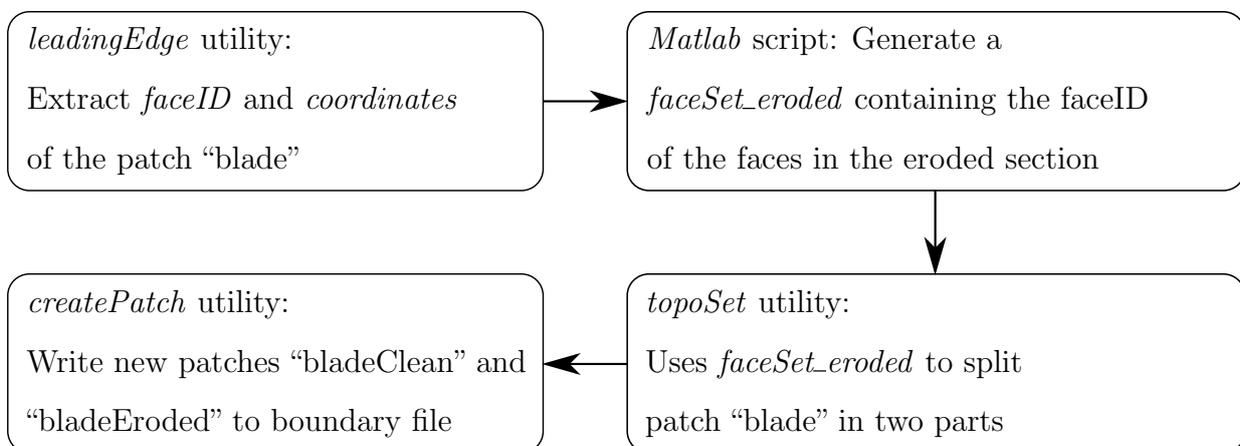


Figure 6.6: Workflow to split the patch *blade* into smooth and eroded parts.

## 6.2 Results

The results presented in this section were obtained on the *medium* grid. Steady-state and transient simulations were performed with the solution algorithms *SRFSimpleFoam* and *SRFPimpleFoam*, respectively. As for the eroded rotor blade, LEE is modeled using the Amplification Roughness model with an input of  $k_s = 101 \mu\text{m}$ . The extent of the eroded section is chosen as defined above.

### 6.2.1 Steady State Simulations

Figure 6.7 shows the flow field around the eroded blade at  $r/R = 65\%$  in terms of the relative velocity  $\vec{u}_R$ . The general flow direction along the y-axis is indicated by arrows representing the absolute velocity  $\vec{u}_I$ .

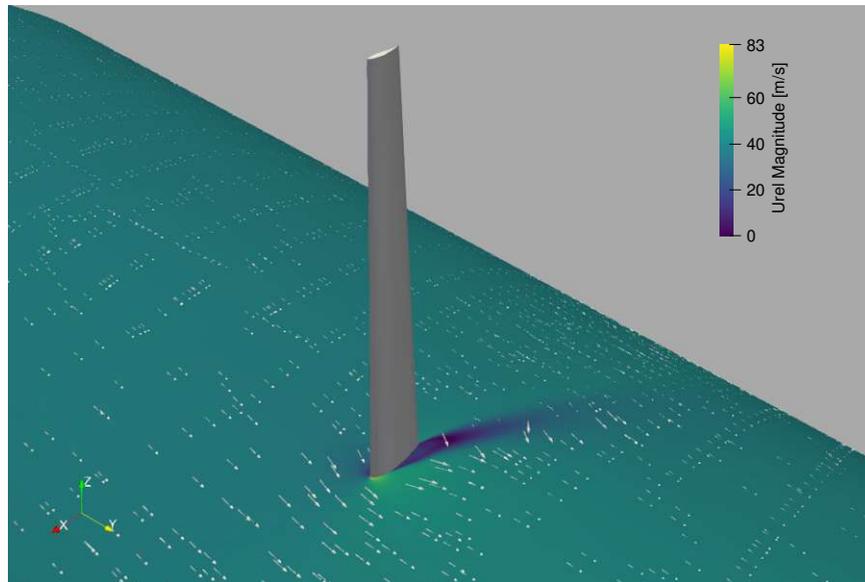


Figure 6.7: Flow field at  $r/R = 65\%$  around the eroded blade.

The evaluation of convergence is based firstly on the residuals of the field variables and secondly on the behavior of parameters of interest such as torque. However, it proved challenging to demonstrate convergence as both torque and the residuals oscillate (see Fig. 6.8). Adjustments to the solver settings and tolerances did not result in significant improvements.

This behavior leads to the assumption that unsteady flow phenomena are present that cannot be captured by the steady-state approach.

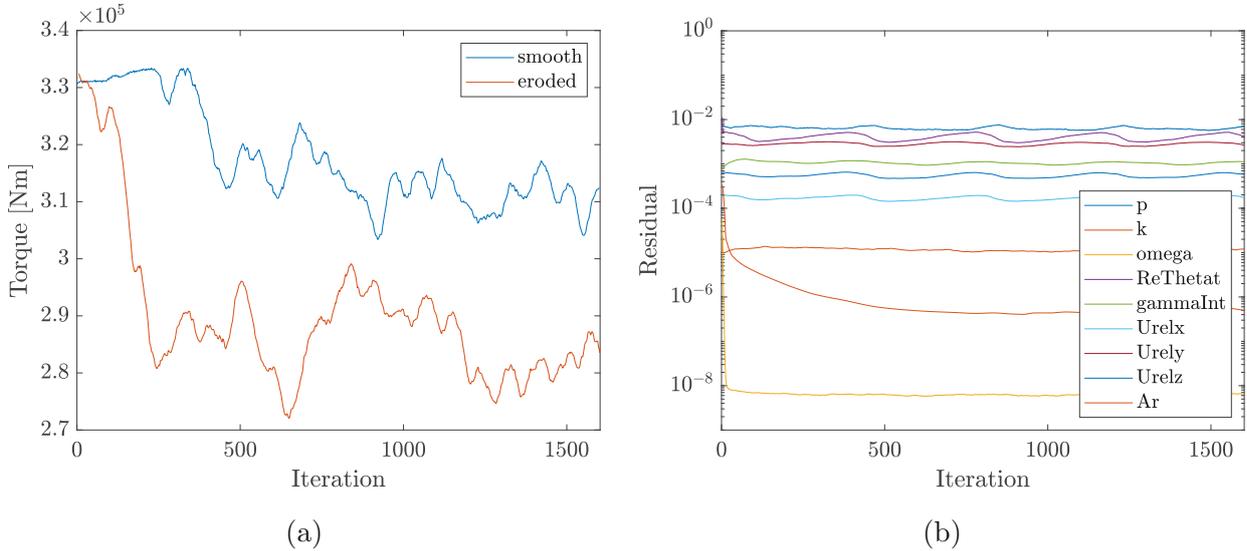


Figure 6.8: Results of the steady-state computation: a) Comparison of torque for both configurations and b) Residuals of the simulation with an eroded blade.

Nevertheless, a comparison can be made between smooth and eroded rotor blades, which is shown in Fig. 6.8a. From a solution for the smooth blade as a starting point, both simulations were run for 1600 iterations. One can see the decrease in torque due to the added roughness. Some global parameters of interest are averaged and stated in Tab. 6.5. With this configuration of LEE, the simulated torque is reduced by 11.4% compared to the smooth blade.

Parameter	Smooth	Eroded
Torque $M_y$ [Nm]	$3.20 \cdot 10^5$	$2.83 \cdot 10^5$
Thrust $F_y$ [N]	$5.80 \cdot 10^4$	$5.51 \cdot 10^4$
Flapwise moment $M_x$ [Nm]	$-1.26 \cdot 10^6$	$-1.19 \cdot 10^6$
Torsional moment $M_z$ [Nm]	$-2.68 \cdot 10^3$	$-3.04 \cdot 10^3$

Table 6.5: Comparison of global parameters of a rotor blade, smooth and eroded.

The pressure distribution on both sides of the smooth and eroded rotor blade is shown in Fig. 6.9. In addition, the relative velocity  $\vec{u}_R$  is visualized as streamlines. Since it is difficult to draw conclusions from the figure below, two sections of each blade are examined in more detail. The selected spanwise locations are at  $r/R = 65\%$  and  $r/R = 95\%$  with chord lengths of 1.77 m and 1.003 m, respectively.

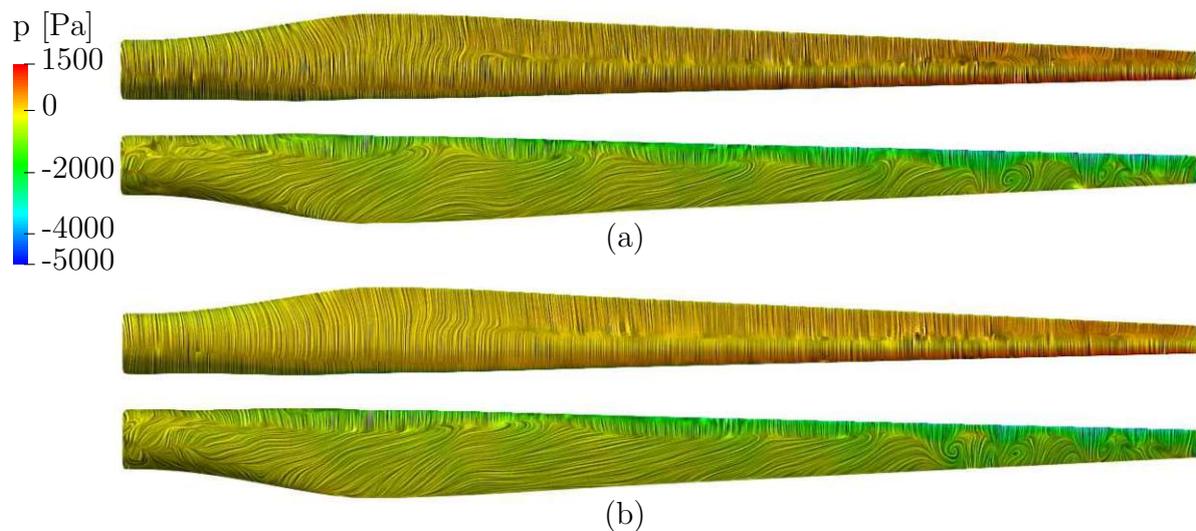


Figure 6.9: Pressure distribution on the a) smooth and b) eroded blade along with surface streamlines in terms of  $\vec{u}_R$ .

Figure 6.10 shows the pressure coefficient distribution for each blade at both blade sections. In the case of the eroded blade, the pressure is generally higher in the front part of the upper surface<sup>3</sup>. An adverse pressure gradient is present towards the trailing edge; the simulated  $c_p$  in this region is slightly lower for the eroded blade. Note that at  $r/R = 65\%$ , the adverse pressure gradient spans over two-thirds of the chord due to a high local angle of attack of  $21.5^\circ$ . The adverse pressure gradient region of the eroded blade is larger in both sections. Overall, the lower pressure on the lower surface and the higher pressure on the upper surface lead to a reduced performance of the eroded blade.

<sup>3</sup>The blade surface, referred to as the “upper surface”, is the area that experiences negative pressure (=suction side) on the side facing away from the absolute flow direction.

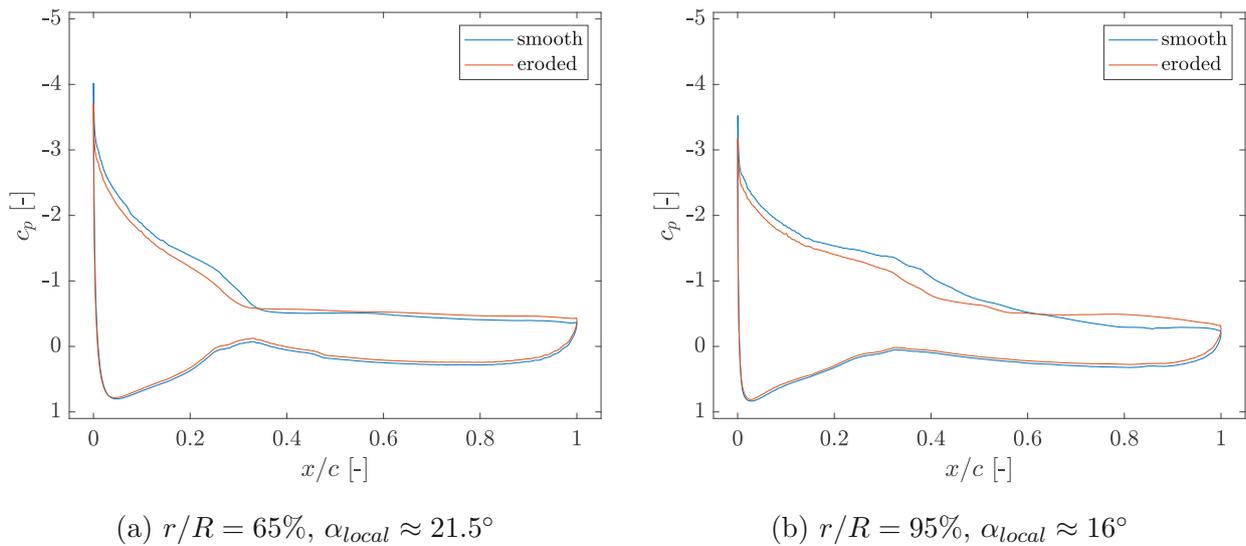


Figure 6.10: Distribution of the pressure coefficient  $c_p$  for the smooth and eroded blade at two sections of the blade.

Regarding the laminar-turbulent transition behavior, the smooth and eroded rotor blades are compared based on the intermittency and skin friction coefficients. Figure 6.11 shows the distribution of the intermittency factor  $\gamma$  on the suction side of the blades. The laminar parts of the boundary layer flow are shown in blue ( $\gamma = 0$ ), while the red regions represent the turbulent parts ( $\gamma = 1$ ). Overall, the eroded blade shows higher values of  $\gamma$ , indicating a more turbulent flow. The leading edge of the eroded blade is particularly affected, meaning that no laminar boundary layer develops there.

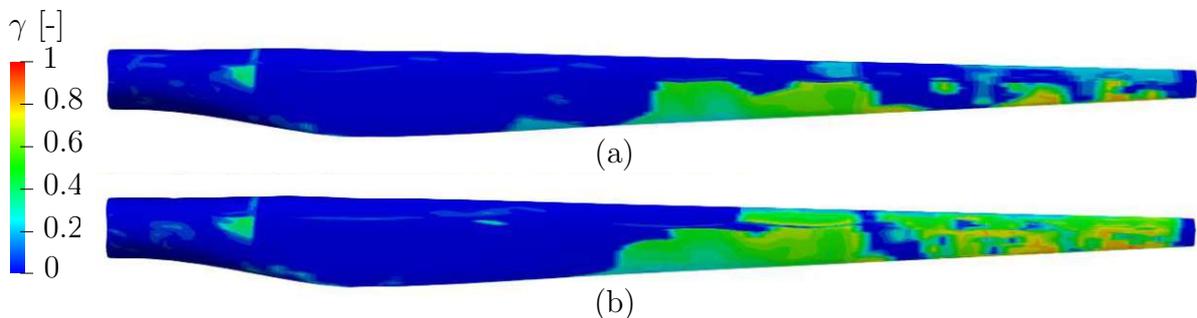


Figure 6.11: Distribution of the intermittency factor  $\gamma$  on the suction side of the a) smooth and b) eroded rotor blade.

The skin friction coefficient  $c_F$  is depicted in Fig. 6.12 for the suction side of each blade at the two sections defined above. As defined previously, the damaged part of the eroded blade stretches until  $10\%c$ , which the shaded grey area indicates. In general,  $c_F$  of the eroded blade is higher in this area than in the corresponding area of the smooth blade. After the eroded section, the skin friction factor drops and approaches the values of the smooth blade again.

The trend of  $c_F$  at  $r/R = 65\%$  suggests that at the leading edge of the smooth blade, a laminar boundary layer is present. However, in the case of an eroded rotor blade, no laminar boundary layer can form.

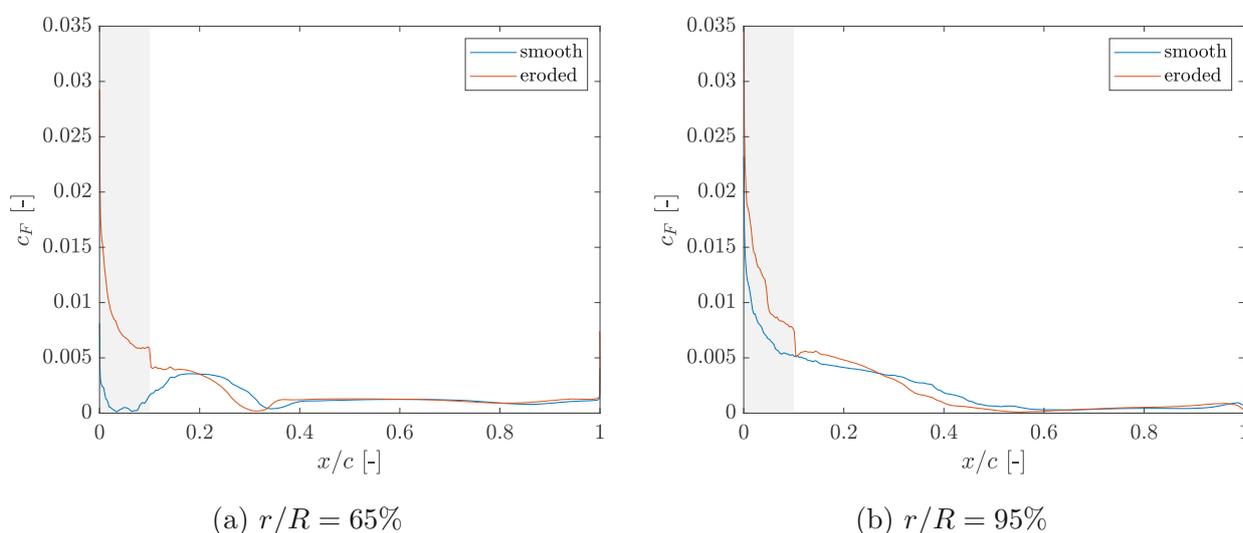


Figure 6.12: The skin friction coefficient  $c_F$  on the upper surface of the blade for the smooth and eroded blade at two stations of the blade.

### 6.2.2 Transient Simulations

Transient simulations with the steady-state solution as the initial condition were performed. Relevant settings for the time-marching are listed in Tab. 6.6. *SRFPimpleFoam* is the transient variant of the *SRFSimpleFoam* solver; it is a solver for incompressible flow in a single reference frame based on the PIMPLE algorithm. It is controlled by a corresponding configuration section in the *fvSolution* file (see List. 6.3). This section also contains the residual control; if the tolerances specified here are met, the outer loop is considered converged, and the simulation advances to the next time step. The rest of the settings remain unchanged.

Subject	Property
Algorithm/Solver	SRFPimpleFoam
Time discretization scheme	Euler
Time step	$1 \cdot 10^{-6}$ s
max. Courant number $C$	$\approx 25$
Simulation time	1.25 ms

Table 6.6: Time-related simulation settings.

It should be noted that the stability of the simulation could be improved by successively reducing the time step from  $1 \cdot 10^{-3}$  s to  $1 \cdot 10^{-6}$  s in the course of the first few iterations. This approach avoids the development of high-pressure peaks within the domain, which significantly increases the speed at which a solution is achieved.

The final time step was chosen by trial and error, and it was found that a Courant number of about 25 is the limit at which a stable simulation is possible.

Figure 6.13 shows the computed torque  $M_y$  and thrust  $F_y$  over time after the final time-step-reduction to  $1 \cdot 10^{-6}$  s. The simulations were run for  $t = 1.25$  ms; each computation took about 150h on 24 compute nodes. That illustrates the immense computational effort that transient simulations of this magnitude require.

```

1 PIMPLE
2 {
3     momentumPredictor          yes ;
4     nNonOrthogonalCorrectors   2;
5     nOuterCorrectors           4;
6     nCorrectors                 3;
7     consistent                  yes ;
8     pRefCell                    0;
9     pRefValue                   0;
10    turbOnFinalIterOnly         false ;
11    correctPhi                   true ;
12    residualControl
13    {
14        p          { relTol    0;      tolerance  1e-4;    }
15        ".*"       { relTol    0;      tolerance  1e-6;    }
16    }
17 }

```

Listing 6.3: Settings for the transient time marching algorithm specified in the *fvSolution* file.

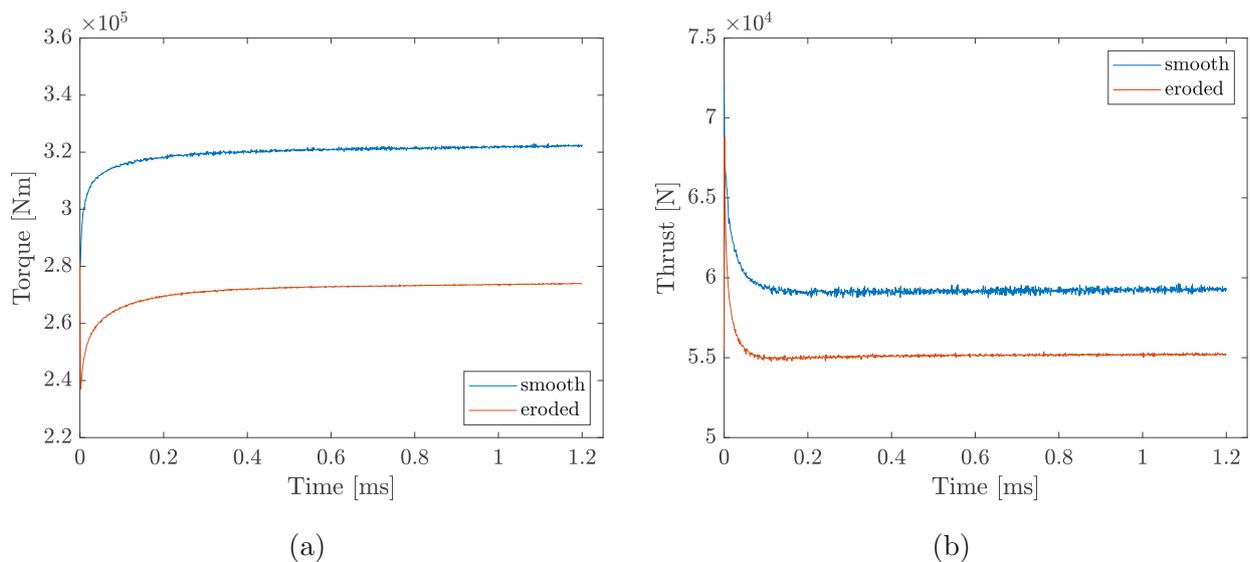


Figure 6.13: Transient results in terms of a) torque  $M_y$  and b) thrust  $F_y$  for smooth and eroded blades.

The results show that  $M_y$  and  $F_y$  approach an asymptotic value close to the averaged steady-state solution. Periodic oscillations are expected to occur since they were also observed in the two-dimensional cases. Here, such oscillations cannot be seen - most likely due to the short simulation duration. These are presumably lower-frequency oscillations that cannot be captured in only 1.25 ms of simulation time. Thus, for more detailed investigations of transient phenomena in the order of seconds, the computing power must be significantly increased, and a longer project duration must be expected. Measures to speed up the simulations include the optimization of the solution algorithms and decreasing the grid size.

Nevertheless, this section shows that a transient simulation with the present setup is possible and provides satisfactory results. The case setup can therefore serve as a starting point for future investigations, e.g., to simulate fluid-structure interaction. Such an approach is outlined in the following section.

Parameter	Smooth	Eroded
Torque $M_y$ [ $Nm$ ]	$3.23 \cdot 10^5$	$2.76 \cdot 10^5$
Thrust $F_y$ [ $N$ ]	$5.92 \cdot 10^4$	$5.52 \cdot 10^4$
Flapwise moment $M_x$ [ $Nm$ ]	$-1.29 \cdot 10^6$	$-1.19 \cdot 10^6$
Torsional moment $M_z$ [ $Nm$ ]	$-2.83 \cdot 10^3$	$-3.18 \cdot 10^3$

Table 6.7: Comparison of global parameters of a rotor blade, smooth and eroded.

Table 6.7 shows the relevant averaged global parameters. All values are within 5% of the steady-state solution. It can be concluded that, for determining global parameters such as torque, if transient effects are neglected, steady-state computations provide appropriate results. Finally, it can also be said that the Amplification Roughness model is applicable to large-scale simulations as well and is able to predict blade performance losses due to LEE.

### 6.3 Coupling the Simulation with FEA - Outline

Although, in reality, aeroelastic effects occur in a rotor blade, they have been neglected in the model so far. To also take these effects caused by fluid-structure interaction (FSI) into account, the fluid simulation must be coupled with structural analysis. Finite element analysis (FEA) is used to determine the structural response of the blade, while the CFD model already created is used to determine the aerodynamic loads. As with the fluid simulation, open-source software is used for the FEA too - namely the three-dimensional structural finite element program CalculiX [69]. The coupling of the fluid side and the structure side is achieved with the preCICE coupling library. This library allows for a partitioned coupling of multi-physics simulations [70], and its basic concept is shown schematically in Fig. 6.14. Information between the two solvers, i.e., forces and displacement, is passed at the interface between the fluid and the solid domain.

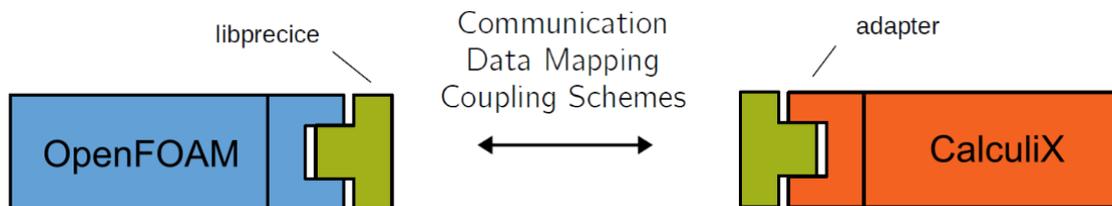


Figure 6.14: FSI approach using OpenFOAM and CalculiX with the preCICE library, adapted from [71].

Since the complete implementation of an FSI simulation is beyond the scope of this paper, only a suggested workflow is partially outlined below. That includes the translation of an already existing FEA model to the appropriate file format and the basic setup of the preCICE coupling library. However, it is intended that this outline, although being only a condensed version, will be helpful regarding the implementation of an FSI simulation in subsequent research.

### 6.3.1 Translation of the FEA Model

The rotor blade from the WindPACT 1.5MW turbine is included as a template model in the NuMAD tool [72]. With this tool, the FEA model properties, including mesh size and element types, are defined. An overview regarding the structural properties, e.g., the definition of the materials and composite stacks, is given in Appendix D. Unfortunately, NuMAD is unable to generate the model directly for use with CalculiX. However, it is possible to generate a text file *shell7.src* which can be used to create an Ansys Mechanical APDL model.

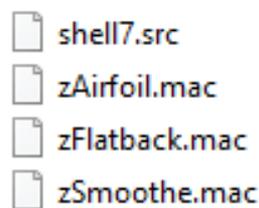


Figure 6.15: The APDL input file *shell7.src* and some macros *.mac* as written by NuMAD.

Then, the APDL model must be converted to the Abaqus input format *.inp*, which is also used by the CalculiX solver. One way to convert the model is to use the Ansys Workbench and FE Modeler. After importing the input file *shell7.src*, the model is saved as an APDL project (see Fig. 6.16a). Then it is added to a Workbench project (see Fig. 6.16b) to which the FE modeling tool is connected.

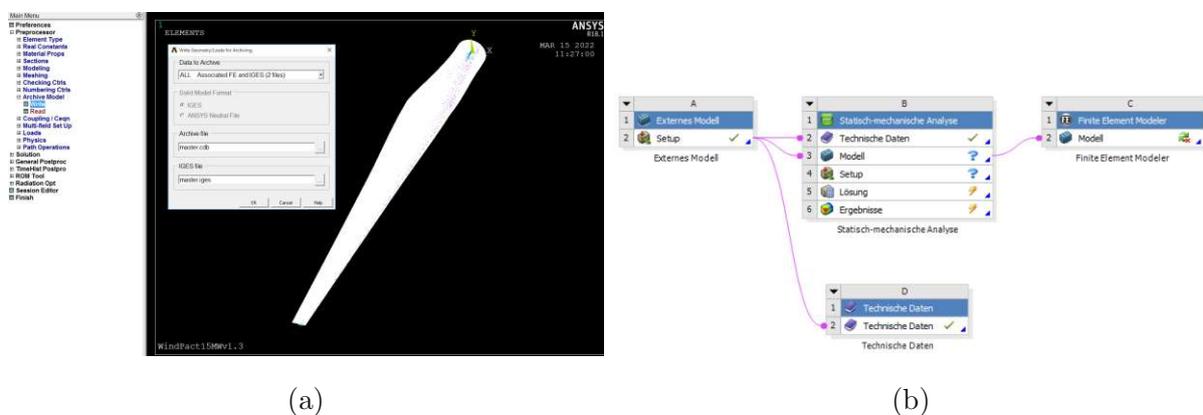


Figure 6.16: The process of converting the model from an APDL project to the *.inp* format.

Finally, the FE Modeler creates an output file in the format specified in the “Target System” drop-down menu by clicking “Generate Data”. Then the conversion process is completed. Alternatively, one could use the *abaqus fromansys translator* that is implemented in Abaqus [73]. However, the use of commercial software (Ansys/Abaqus) is not exactly ideal; after all, both solvers and the coupling library are open-source software.

### 6.3.2 Setting up preCICE

The main preCICE configuration file *precice-config.xml* specifies, among other things, which solvers are participating and which numerical methods are used for data mapping. In general, the configuration file consists of five parts which are shown in List. 6.4. For more details about the configuration of preCICE, the reader is referred to preCICE’s online documentation [71].

```

1 <precice-configuration >
2   <solver-interface dimensions="3">
3     <data .../>           // data values the solvers want to exchange
4     <mesh .../>          // meshes at coupling interface
5     <participant .../>   // definition of the participants
6     <m2n .../>           // communication setup
7     <coupling-scheme .../> // definition of the coupling schemes
8   </solver-interface >
9 </precice-configuration >

```

Listing 6.4: Main structure of the preCICE configuration file *precice-config.xml*.

preCICE is a coupling library that is called by each solver. To call preCICE, each solver needs an adapter - in this case, for both OpenFOAM and CalculiX, an adapter has to be configured. The OpenFOAM adapter is loaded through an entry in *system/controlDict* shown in (List. 6.5).

```

1 functions
2 {
3     preCICE_Adapter
4     {
5         type preciceAdapterFunctionObject;
6         libs ("libpreciceAdapterFunctionObject.so");
7     }
8 }

```

Listing 6.5: *controlDict* entry to load the OpenFOAM adapter.

The dictionary *system/preciceDict* is the configuration file for the OpenFOAM adapter. Among other things, the patches that participate in the coupled simulation are specified here. In the present case, the configuration file would look something like the one shown in List. 6.6.

```

1 FoamFile
2 {
3     ...
4 }
5 preciceConfig "precice-config.xml";
6 participant Fluid;
7 modules (FSI);
8
9 interfaces
10 {
11     Interface1
12     {
13         mesh             Fluid_Mesh;
14         patches          (blade);
15         locations        faceCenters;
16         readData
17         ( Displacement );
18         writeData
19         ( Force );
20     };

```

```

21 };
22
23 FSI
24 {
25     rho rho [1 -3 0 0 0 0 0] 1.225;
26 }

```

Listing 6.6: The configuration file *preciceDict* for the OpenFOAM adapter.

Furthermore, the BCs of the interface patch (*blade*) must be adjusted to allow for a moving mesh. That also requires a file *constant/dynamicMeshDict*. Further details concerning the OpenFOAM adapter can be found in [74].

The CalculiX adapter needs a configuration file *config.yml* which is shown in List. 6.7.

```

1 participants :
2     Calculix :
3         interfaces :
4             - nodes-mesh: Solid_Mesh
5             patch: blade
6             read-data: [Forces]
7             write-data: [Displacement]
8
9 precice-config-file: ../precice-config.xml

```

Listing 6.7: The configuration file *config.yml* for the CalculiX adapter.

While the fluid simulation with OpenFOAM can be started, as usual, CalculiX has to be launched using an adapted executable, as shown in List. 6.8.

```

1 ccx_preCICE -i rotorBlade -precice-participant Calculix

```

Listing 6.8: Running the adapted CalculiX executable with the input file *rotorBlade.inp*.

# Chapter 7

## Conclusion

LEE on wind turbine rotor blades, and ways to numerically model it, were reviewed in this work. Two methodologies, i.e., geometrical and mathematical representation of LEE, were applied in numerical simulations predicting the flow around the damaged wind turbine airfoils. An existing roughness model proposed by Langel et al. [14] was implemented into OpenFOAM and validated using experimental and numerical reference data. Two-dimensional airfoil simulations revealed satisfactory results for both methodologies, confirming their capability to model LEE and thus demonstrating that it is possible to model erosion numerically. In addition, the Amplification Roughness model was applied to a full-scale rotor blade simulation. The high computational effort, in particular for transient simulations, gives reason for a discussion on the practicability of this approach.

At the beginning of this thesis, the theoretical background necessary to understand LEE and the issues raised by it was presented. Modeling a range of LEE intensities for numerical purposes is done using the Amplification Roughness model and an approach that directly modifies the geometry of the airfoil. The Amplification Roughness model, an extension of the Langtry-Menter transition model, has been implemented in OpenFOAM. It generally performs well, but some specific features that could be improved were identified. The model may benefit from a more robust implementation, i.e., implicit treatment of the sink terms. Moreover, extending the model to compressible cases could significantly expand its scope of application.

When applied to two-dimensional numerical airfoil simulations, it delivered satisfactory results in terms of aerodynamic performance prediction. Suggestions to further improve the results include the application of the Kato-Launder limiter and calibration of the roughness input parameters to actual experimental data. While the case of an airfoil with a geometric representation of the delaminated leading edge also provided consistent results that matched the experimental data, the combination of the two models did not work as expected. It is suspected that the irregular geometry leads to unfavorable behavior of the Amplification Roughness model. Thus further investigation into the influence of non-standard airfoil geometries, e.g., damaged airfoils or airfoils equipped with flaps, on the performance of the Amplification Roughness model is recommended for future work. Incorporating an automatic approach to mesh generation may be of interest to efficiently investigate the varying extents of a delaminated leading-edge. To sum up, the roughness and delamination models efficiently predict an airfoil's performance penalty, with some potential for improvement identified.

Unsteady simulations showed that a delaminated airfoil experiences the highest amplitudes of surface pressure oscillations. However, owing to a lack of time-dependent experimental data, definitive conclusions are difficult to draw.

In addition to the two-dimensional investigations, a full-scale model of a single wind turbine rotor blade was developed. The idea was to establish a base model for FSI simulations in future projects to study the dynamic behavior of a blade with a partially eroded leading edge. In this work, a mesh was also created, and steady-state simulations were carried out successfully. The eroded parts were modeled using the Amplification Roughness model. With the assumed extent of erosion, an 11% torque reduction was recorded compared to a clean blade. A transient simulation was set up, and 1.2 ms of flow were simulated. The calculated values for torque and thrust are within 3% of the averaged steady-state results. For this reason, and due to the lower computational effort, the steady-state setup is preferable if only global parameters are required, and the precise temporal evolution is irrelevant. However, for capturing potential blade oscillations, where larger time scales in the order of seconds are expected, the duration of the simulation is too short. To simulate longer time scales with the current setup, as is necessary for the investigation of the dynamic behavior through FSI simulations, the computing power has to be increased, and longer simulation

## CONCLUSION

---

times are to be expected. The process for coupling the unsteady flow simulation of the WindPACT 1.5MW rotor blade with structural analysis using open source software was also outlined.

In general, this work shows that the simulation setup performs well and that the Amplification Roughness model can be used to model the effects of LEE on the performance of a rotor blade. For analyzing rotor blade vibrations, the necessary transient simulations are extremely resource-demanding.

# Bibliography

- [1] Global Wind Report 2021. Technical report, Global Wind Energy Council, March 2021. <https://gwec.net/global-wind-report-2021>, accessed 24.05.2021.
- [2] Mark Keegan, David Nash, and Margaret Stack. On erosion issues associated with the leading edge of wind turbine blades. *Journal of Physics D: Applied Physics*, 46(38):3001, September 2013.
- [3] Lisa Rempel. Rotor blade leading edge erosion - real life experiences. *Wind Systems Magazine October*, 2012.
- [4] Casey Martinez, Festus Asare Yeboah, Scott Herford, Matt Brzezinski, and Viswanath Puttagunta. Predicting wind turbine blade erosion using machine learning. *SMU Data Science Review*, 2(2):17, September 2019.
- [5] Woobeom Han, Jonghwa Kim, and Bumsuk Kim. Effects of contamination and erosion at the leading edge of blade tip airfoils on the annual energy production of wind turbines. *Renewable Energy*, 115:817–823, 2018.
- [6] Alessio Castorrini, Lorenzo Cappugi, Aldo Bonfiglioli, and Sergio Campobasso. Assessing wind turbine energy losses due to blade leading edge erosion cavities with parametric CAD and 3D CFD. *Journal of Physics: Conference Series*, 1618(5):052015, September 2020.
- [7] Matthias Schramm, Hamid Rahimi, Bernhard Stoevesandt, and Kim Tangager. The influence of eroded blades on wind turbine performance using numerical simulations. *Energies*, 10(9):1420, 2017.

- [8] Emil Krog Kruse. *A Method for Quantifying Wind Turbine Leading Edge Roughness and its Influence on Energy Production: LER2AEP*. DTU Wind Energy PhD. DTU Wind Energy, Denmark, 2019.
- [9] Erich Hau. *Windkraftanlagen - Grundlagen, Technik, Einsatz, Wirtschaftlichkeit*. Springer Vieweg, 5th edition, June 2014.
- [10] Jakob I. Bech, Charlotte B. Hasager, and Christian Bak. Extending the life of wind turbine blade leading edges by reducing the tip speed during extreme precipitation events. *Wind Energy Science*, 3(2):729–748, 2018.
- [11] Olivier Cleynen. Wing profile nomenclature, April 2011. [https://commons.wikimedia.org/wiki/File:Wing\\_profile\\_nomenclature.svg](https://commons.wikimedia.org/wiki/File:Wing_profile_nomenclature.svg), accessed 01.02.2022.
- [12] Robert E. Mayle. The Role of Laminar-Turbulent Transition in Gas Turbine Engines. *Turbo Expo: Power for Land, Sea, and Air*, 5, June 1991.
- [13] Robin B. Langtry and Florian R. Menter. Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes. *AIAA Journal*, 47(12):2894–2906, 2009.
- [14] Christopher M. Langel, Raymond C. Chow, C. P. van Dam, and David C. Maniaci. Rans based methodology for predicting the influence of leading edge erosion on airfoil performance. Technical Report SAND2017-11289, Sandia National Laboratories, October 2017.
- [15] Hermann Schlichting. *Boundary Layer Theory*. McGraw-Hill, Inc., 1979.
- [16] Kieran Pugh, Ghulam Rasool, and Margaret Stack. Some thoughts on mapping tribological issues of wind turbine blades due to effects of onshore and offshore raindrop erosion. *Journal of Bio- and Tribo-Corrosion*, 4(50), July 2018.
- [17] Agrim Sareen, Chinmay A. Sapre, and Michael S. Selig. Effects of leading edge erosion on wind turbine blade performance. *Wind Energy*, 17(10):1531–1542, 2014.

- [18] Behrooz Amirzadeh, Arghavan Louhghalam, Mehdi Raessi, and Mazdak Tootkaboni. A computational framework for the analysis of rain-induced erosion in wind turbine blades, part I: Stochastic rain texture model and drop impact simulations. *Journal of Wind Engineering and Industrial Aerodynamics*, 163:33–43, 2017.
- [19] Emil Krog Kruse, Christian Bak, and Anders Smærup Olsen. Wind tunnel experiments on a NACA 63<sub>3</sub>-418 airfoil with different types of leading edge roughness. *Wind Energy*, 24(11):1263–1274, 2021.
- [20] Yan Wang, Ruifeng Hu, and Xiaojing Zheng. Aerodynamic analysis of an airfoil with leading edge pitting erosion. *Journal of Solar Energy Engineering*, 139(6), August 2017.
- [21] Emil Krog Kruse, Niels Sørensen, Christian Bak, and Mikkel Nielsen. CFD simulations and evaluation of applicability of a wall roughness model applied on a NACA 63<sub>3</sub>-418 airfoil. *Wind Energy*, 23(11):2056–2067, November 2020.
- [22] Tobias Knopp, Bernhard Eisfeld, and Javier Bartolome Calvo. A new extension for  $k$ - $\omega$  turbulence models to account for wall roughness. *International Journal of Heat and Fluid Flow*, 30(1):54–65, 2009.
- [23] Kostas Latoufis, Vasilis Riziotis, Spyros Voutsinas, and Nikos Hatzigargyriou. Effects of leading edge erosion on the power performance and acoustic noise emissions of locally manufactured small wind turbine blades. *Journal of Physics: Conference Series*, 1222(1):012010, May 2019.
- [24] Josué Enríquez Zárata, María de los Ángeles Gómez López, Javier Alberto Carmona Troyo, and Leonardo Trujillo. Analysis and detection of erosion in wind turbine blades. *Mathematical and Computational Applications*, 27(1), 2022.
- [25] Thomas C. Corke, A. Bar-Sever, and Mark V. Morkovin. Experiments on transition enhancement by distributed roughness. *The Physics of Fluids*, 29(10):3199–3213, 1986.
- [26] Hermann Schlichting. Experimental investigation of the problem of surface roughness. *National Advisory Committee on Aeronautics, NACA TM-832*, 1936.

- [27] Johann Nikuradse. Laws of flow in rough pipes. *VDI Forschungsheft*, 361, 1933.
- [28] Ernst G. Feindt. Untersuchungen über die Abhängigkeit des Umschlages laminar-turbulent von der Oberflächenrauigkeit und der Druckverteilung. *DFL Bericht*, 43, 1956.
- [29] Michael F. Kerho and Michael B. Bragg. Airfoil boundary-layer development and transition with large leading-edge roughness. *AIAA Journal*, 35(1):75–84, 1997.
- [30] Jeffrey P. Bons. A review of surface roughness effects in gas turbines. *Journal of Turbomachinery*, 132(2), January 2010.
- [31] Florian R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, 1994.
- [32] Florian R. Menter, M. Kuntz, and Robin B. Langtry. Ten years of industrial experience with the SST turbulence model. *Heat and Mass Transfer*, 4, January 2003.
- [33] Bernhard Semlitsch. Numerische Strömungssimulation für ingenieurmäßige Anwendungen [Vorlesungsfolien], TU Wien, 2020. <https://tuwel.tuwien.ac.at/mod/resource/view.php?id=1021699>, accessed 06.09.2021.
- [34] Christopher L. Rumsey. The Menter Shear Stress Transport Turbulence Model. NASA Langley Research Center, 2021. <https://turbmodels.larc.nasa.gov/sst.html>, accessed 06.09.2021.
- [35] Jared A. Carnes and James G. Coder. Analyzing the near-wall behavior of the Langtry–Menter Transition Model. *Flow, Turbulence and Combustion*, 108:683–715, August 2021.
- [36] Philippe R. Spalart and Christopher L. Rumsey. Effective inflow conditions for turbulence models in aerodynamic calculations. *AIAA Journal*, 45(10):2544–2553, 2007.
- [37] Christopher L. Rumsey. The Langtry-Menter 4-equation Transitional SST Model. NASA Langley Research Center, 2021. [https://turbmodels.larc.nasa.gov/langtrymenter\\_4eqn.html](https://turbmodels.larc.nasa.gov/langtrymenter_4eqn.html), accessed 14.01.2022.

- [38] Patrick Dassler, Dragan Kozulovic, and Andreas Fiala. Modelling of roughness-induced transition using local variables. In *V European Conference on Computational Fluid Dynamic, Lisbon, Portugal, 14-17 June 2010*.
- [39] Karen A. Flack and Michael P. Schultz. Review of hydraulic roughness scales in the fully rough regime. *Journal of Fluids Engineering*, 132(4), April 2010.
- [40] Antti Hellsten and Seppo Laine. Extension of the k-omega-SST turbulence model for flows over rough surfaces. In *AIAA Atmospheric Flight Mechanics Conference, New Orleans, Louisiana, August 11-13, 1997*, pages 252–260.
- [41] Hakan Nilsson. Proceedings of CFD with OpenSource Software, Implement a turbulence model, 2020. [http://dx.doi.org/10.17196/OS\\_CFD#YEAR\\_2020](http://dx.doi.org/10.17196/OS_CFD#YEAR_2020), accessed 04.06.2021.
- [42] Gerhard Holzinger. *OpenFOAM - A little User Manual*. March 2020.
- [43] OpenCFD Limited. *OpenFOAM-v2012 Programmer's Guide*, 2020.
- [44] Chenzhou Lian, Guoping Xia, and Charles L. Merkle. Impact of source terms on reliability of CFD algorithms. *Computers and Fluids*, 39(10):1909–1922, December 2010.
- [45] Gerhard Holzinger. Error: Request from objectRegistry failed. CFD Online Discussion Forums, OpenFOAM Programming Development Forum, September 2021. <https://www.cfd-online.com/Forums/openfoam-programming-development/238689-error-request-objectregistry-failed.html>, accessed 27.09.2021.
- [46] W.A. Timmer. An overview of NACA 6-digit airfoil series characteristics with reference to airfoils for large wind turbine blades. *47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 5-8 January 2009, Orlando, Florida; AIAA 2009-268*, January 2009.
- [47] Jason Jonkman, S. Butterfield, Walter Musial, and George Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, February 2009.

- [48] Robert S. Ehrmann, Benjamin Wilcox, Edward B. White, and David C. Maniaci. Effect of surface roughness on wind turbine performance. Technical Report SAND2017-10669, Sandia National Laboratories, October 2017.
- [49] David C. Maniaci, Edward B. White, Benjamin Wilcox, Christopher M. Langel, C.P. van Dam, and Joshua A. Paquette. Experimental measurement and CFD model development of thick wind turbine airfoils with leading edge erosion. *Journal of Physics: Conference Series*, 753(2):022013, September 2016.
- [50] Jason Jonkman. NREL 5MW rotor geometry. NREL’s National Wind Technology Center Forum, June 2012. <https://forums.nrel.gov/t/nrel-5mw-rotor-geometry/330/10>, accessed 16.09.2021.
- [51] Michael Selig. UIUC airfoil coordinates database - NACA 63<sub>3</sub>-418, 2021. <https://m-selig.ae.illinois.edu/ads/coord/naca633418.dat>, accessed 01.09.2021.
- [52] Andreas Ennemoser. PyAero Version 1.2, 2020. <https://github.com/chiefenne/PyAero>, accessed 01.09.2021.
- [53] Andreas Ennemoser. PyAero Version 1.2 - Documentation: 5. Splining and refining airfoil contours, 2021. [https://pyaero.readthedocs.io/en/latest/spline\\_refine.html](https://pyaero.readthedocs.io/en/latest/spline_refine.html), accessed 01.09.2021.
- [54] OpenCFD Limited. *OpenFOAM-v2112 User Guide*, 2021.
- [55] Alessandro Zanon, Michele De Gennaro, and Helmut Kühnelt. Wind energy harnessing of the NREL 5 MW reference wind turbine in icing conditions under different operational strategies. *Renewable Energy*, 115:760–772, 2018.
- [56] American Institute Of Aeronautics and Astronautics (AIAA). *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*. 1998. AIAA G-077-1998.
- [57] Ismail B. Celik, Urmila Ghia, Patrick J. Roache, Christopher J. Freitas, Hugh Coleman, and Peter E. Raad. Procedure of estimation and reporting of uncertainty due to

- discretization in CFD applications. *Journal of Fluids Engineering*, 130(7):078001, July 2008.
- [58] Sanjana Sanjay, Senthilkumar Sundararaj, and Kannan Budda Thiagarajan. Numerical simulation of flat plate boundary layer transition using OpenFOAM®. *AIP Conference Proceedings*, 2112(1):020134, 2019.
- [59] OpenCFD Limited. *OpenFOAM-v2112 User Guide, Turbulence transition T3A*, 2021. <https://www.openfoam.com/documentation/guides/latest/doc/verification-validation-turbulent-t3a.html>, accessed 18.11.2021.
- [60] John Coupland. Flat Plate Transitional Boundary Layers. ERCOFTAC Classic Collection Database, 1990. <http://cfd.mace.manchester.ac.uk/ercoftac/doku.php?id=cases:case020#references>, accessed 16.09.2021.
- [61] M. Kato and Brian Launder. The modelling of turbulent flow around stationary and vibrating square cylinders. *9th Symposium on Turbulent Shear Flows, Kyoto, Japan*, January 1993.
- [62] D. J. Malcolm and A. Craig Hansen. Windpact turbine rotor design study: June 2000–june 2002 (revised). Technical Report NREL/SR-500-32495, National Renewable Energy Laboratory, April 2006.
- [63] Jennifer Rinker and Katherine Dykes. Windpact reference wind turbines. Technical Report NREL/TP-5000-67667, National Renewable Energy Laboratory, April 2018.
- [64] Brian R. Resor and Tyler Bushnell. A 1.5 MW NuMAD blade model. Technical report, Wind Energy Technologies Department Sandia National Laboratories, 2012.
- [65] Lin Wang, Robin Quant, and Athanasios Kolios. Fluid structure interaction modelling of horizontal-axis wind turbine blades based on CFD and FEA. *Journal of Wind Engineering and Industrial Aerodynamics*, 158:11–25, November 2016.

- [66] Javier Martinez Suarez, Piotr Doerffer, Oskar Szulc, and Fernando Tejero. Aerodynamic analysis of wind turbine rotor blades. *TASK Quarterly*, 19(2):129–140, April 2015.
- [67] Fumiya Nozaki. CFD for Rotating Machinery, 2015. <https://www.slideshare.net/fumiyanozaki96/cfd-for-rotating-machinery-using-openfoam>, accessed 04.02.2022.
- [68] Patricio A.L. Gallardo. Static and fatigue analysis of wind turbine blades subject to cold weather conditions using finite element analysis. Master’s thesis, University of Victoria, 2011.
- [69] Guido Dhondt. *CalculiX: The Finite Element Method for Three-Dimensional Thermomechanical Applications*. Wiley, 2004.
- [70] Hans-Joachim Bungartz, Florian Lindner, Bernhard Gatzhammer, Miriam Mehl, Klaudius Scheufele, Alexander Shukaev, and Benjamin Uekermann. preCICE – a fully parallel library for multi-physics surface coupling. *Computers and Fluids*, 141:250–258, 2016. Advances in Fluid-Structure Interaction.
- [71] Benjamin Uekermann. The preCICE documentation, 2022. <https://precice.org/docs.html>, accessed 28.04.2022.
- [72] Ernesto Camarena, Evan Anderson, Kelley Ruehl, Ryan J. Clarke, Joshua Paquette, and Brandon L. Ennis. NuMAD v3.0. Zenodo. <https://doi.org/10.5281/zenodo.5851606Q>, 2022.
- [73] Simulia. Abaqus Analysis User’s Guide, 2022. <http://130.149.89.49:2080/v6.13/books/usb/default.htm?startat=pt01ch03s02abx28.html>, accessed 28.05.2022.
- [74] Gerasimos Chourdakis. A general OpenFOAM adapter for the coupling library preCICE. Masterarbeit, Technical University of Munich, October 2017.
- [75] Patrick J. Roache. Perspective: A method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering*, 116(3):405–413, September 1994.

- [76] John W. Slater. Examining Spatial (Grid) Convergence. NPARC Alliance CFD Verification and Validation Web Site, 2021. <https://www.grc.nasa.gov/www/wind/valid/tutorial/spatconv.html>, accessed 14.03.2022.
- [77] Mohamed Sukri Mat Ali, Con Doolan, and Vincent Wheatley. Grid convergence study for a two-dimensional simulation of flow around a square cylinder at a low Reynolds number. In *Seventh International Conference on CFD in the Minerals and Process Industries CSIRO, Melbourne, Australia*, pages 1–6, November 2009.
- [78] Luís Eça and Martin Hoekstra. A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies. *Journal of Computational Physics*, 262:104–130, 2014.

# Appendix A

## Estimation of the Discretization Error

The Grid Convergence Index (GCI) method according to the approach of Celik et al. [57] was chosen to quantify uncertainty due to discretization. The method is based on the Richardson extrapolation and provides a uniform way of reporting convergence for grid refinement studies [75].

First, a representative grid size  $h$  has to be defined for each mesh:

$$h = \left[ \frac{1}{N} \sum_{n=0}^N (\Delta V_i) \right]^{\left(\frac{1}{3}\right)}. \quad (\text{A.1})$$

For two-dimensional calculations Eq. A.1 reduces to:

$$h = \left[ \frac{1}{N} \sum_{n=0}^N (\Delta A_i) \right]^{\left(\frac{1}{2}\right)}, \quad (\text{A.2})$$

where  $N$  is the total number of cells.  $\Delta V_i$  and  $\Delta A_i$  are the volume and area of the  $i^{th}$  cell, respectively.

After simulations were run on three different grids<sup>1</sup>, variables  $\Phi$  critical to the conclusions of the study have to be reported, e.g., lift and drag coefficients. A grid refinement factor:

---

<sup>1</sup>where grid size  $h_1$  refers to the finest grid:  $h_1 < h_2 < h_3$

$$r_{ij} = \frac{h_{coarse}}{h_{fine}} > 1.3, \quad (\text{A.3})$$

is desirable. The apparent order  $p$  of the method is calculated by solving an implicit expression using fixed-point iteration. The extrapolated values can be calculated from:

$$\Phi_{ext}^{21} = \frac{r_{21}^p \cdot \Phi_1 - \Phi_2}{r_{21}^p - 1}, \quad (\text{A.4})$$

and in a similar way  $\Phi_{ext}^{32}$ . Approximate and extrapolated relative errors  $e_a^{21}$  and  $e_{ext}^{21}$  are calculated next:

$$e_a^{21} = \left| \frac{\Phi_1 - \Phi_2}{\Phi_1} \right|, \quad (\text{A.5})$$

$$e_{ext}^{21} = \left| \frac{\Phi_{ext}^{12} - \Phi_1}{\Phi_{ext}^{12}} \right|. \quad (\text{A.6})$$

Finally, the fine-grid convergence index:

$$GCI^{21} = \frac{1.25 \cdot e_a^{21}}{r_{21}^p - 1}, \quad (\text{A.7})$$

with a safety factor of 1.25 can be reported for each variable of interest  $\Phi$ . The  $GCI^{21}$  “indicates an error band on how far the solution is from the asymptotic value and how much the solution would change with further grid-refinement” [76]. As a measure of uncertainty, the GCI is usually presented as error bars. The convergence ratio:

$$R = \frac{\epsilon_{21}}{\epsilon_{32}} = \frac{\Phi_2 - \Phi_1}{\Phi_3 - \Phi_2}, \quad (\text{A.8})$$

compares the three different meshes and determines the convergence behaviour of the system. Ali et al. [77] and Eça and Hoekstra [78] suggest three possible convergence conditions:

- $0 < R < 1$ : monotonic convergence
- $R < 0$ : oscillatory convergence
- $R > 1$ : divergence

# Appendix B

## Leading Edge Utility

```
1 /*-----*\
2 ===== |
3 \\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4 \\      /  O p e r a t i o n |
5 \\      /  A n d          | Copyright (C) 2011-2016 OpenFOAM Foundation
6  \\/      M a n i p u l a t i o n |
7 -----
8 Copyright (C) 2021 2021 Christoph Holzinger, TU Wien
9 -----
10 Application
11     leadingEdge.C
12 Group
13     grpPreProcessingUtilities
14 Description
15     Computes all the faces of a patch that are within an area.
16     Specifically for airfoil-type geometries, in this case a rotor
17     blade.
18     z-direction: spanwise
19 Usage
20     leadingEdge     nameOfThePatch     zmin     zmax
21 Compile
22     compile with "wmake"
23 \*-----*/
24 #include "fvCFD.H"
25 #include "argList.H"
```



```
59     const label patchI = mesh.boundaryMesh().findPatchID(
60         patchName);
61     if (patchI < 0)
62     {
63         FatalError
64             << "Unable to find patch " <<
65             patchName << nl
66             << exit(FatalError);
67     }
68
69     // Print patch information
70     Info << "Number of cells of the chosen patch: ";
71     Info << mesh.Cf().boundaryField()[patchI].size() << endl;
72     Info << "Patch " << patchI << ": " << mesh.boundary()[patchI
73         ].name() << " with "
74         << mesh.boundary()[patchI].Cf().size() << " faces.
75         Starts at total face "
76         << mesh.boundary()[patchI].start() << endl;
77
78     // Output File for Coordinates and FaceIndex
79     fileName outputFile("patchCoordinates.txt");
80     OFstream os(outputFile);
81
82     // Iterate over all faces of patchI
83     forAll(mesh.boundary()[patchI], faceI)
84     {
85         // face = global faceID, faceI = local faceID
86         const label& face = mesh.boundary()[patchI].start() + faceI;
87         // Check if z-coordinate is between zmin and zmax:
88         if
89         (
90             (mesh.Cf().boundaryField()[patchI][faceI][2] > zmin) &&
91             (mesh.Cf().boundaryField()[patchI][faceI][2] < zmax)
92         )
93         {
94             Info << "Face: " << faceI << endl;
95         }
96     }
```



# Appendix C

## Boundary Conditions OpenFOAM

This section contains some BC files for the 2D simulations in OpenFOAM. Included are the files for  $U$  and  $p$ , as well as the turbulence fields  $\nu_t$ ,  $k$ ,  $\omega$ . Furthermore, an input file *angleOfAttack* has been created, which is used to initialize and calculate the velocity components.

```

1  /*----- C++ -----*\
2  | ===== |
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: v1912 |
5  | \ \ / A n d | Website: www.openfoam.com |
6  | \ \ / M a n i p u l a t i o n | |
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         IOobject;
13     location      "0";
14     object        angleOfAttack;
15 }
16 // * * * * * //
17 aoa              6; // Enter angle of attack
18 U_mag           75.3; // Magnitude velocity for Re = 5M
19
  
```

```
20 // Compute velocity components
21 U_x #eval "$U_mag * cos(degToRad($aoa))";
22 U_y #eval "$U_mag * sin(degToRad($aoa))";
23
24 // Lift Direction
25 lift_dir_x #eval "-sin(degToRad($aoa))";
26 lift_dir_y #eval "cos(degToRad($aoa))";
27 lift_dir ($lift_dir_x $lift_dir_y 0);
28
29 // Drag Direction
30 drag_dir_x #eval "cos(degToRad($aoa))";
31 drag_dir_y #eval "sin(degToRad($aoa))";
32 drag_dir ($drag_dir_x $drag_dir_y 0);
```

Listing C.1: Initialization and calculation of velocity components in the file *angleOfAttack*.



```
1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        volScalarField;
6     location     "0";
7     object       p;
8 }
9 // * * * * *
10
11 dimensions      [0 2 -2 0 0 0 0];
12 internalField   uniform 0;
13
14 boundaryField
15 {
16     airfoil
17     {
18         type          zeroGradient;
19     }
20
21     ".*"          // Inlet, Outlet, Top, Bottom
22     {
23         type          freestreamPressure;
24         freestreamValue uniform 0;
25         value         uniform 0;
26     }
27
28     frontAndBack
29     {
30         type          empty;
31     }
32 }
33 // *****
```

Listing C.3: Boundary Condition file *p*.

```
1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        volScalarField;
6     location     "0";
7     object       nut;
8 }
9 // * * * * *
10
11 dimensions      [0 2 -1 0 0 0 0];
12 internalField   uniform 0;
13
14 boundaryField
15 {
16     airfoil
17     {
18         type          nutLowReWallFunction;
19         value         uniform 0;
20     }
21
22     ".*"         // Inlet, Outlet, Top, Bottom
23     {
24         type          calculated;
25         value         uniform 0;
26     }
27
28     frontAndBack
29     {
30         type          empty;
31     }
32 }
33 // *****
```

Listing C.4: Boundary Condition file *nut*.

```
1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        volScalarField;
6     location     "0";
7     object       k;
8 }
9 // * * * * *
10
11 dimensions      [0 2 -2 0 0 0 0];
12 internalField   uniform 8.505135e-03;
13
14 boundaryField
15 {
16     airfoil
17     {
18         type          fixedValue;
19         value         uniform 0;
20     }
21
22     ".*"        // Inlet, Outlet, Top, Bottom
23     {
24         type          inletOutlet;
25         inletValue    $internalField;
26         value         $internalField;
27     }
28
29     frontAndBack
30     {
31         type          empty;
32     }
33 }
34 // * * * * *
```

Listing C.5: Boundary Condition file *k*.

```
1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        volScalarField;
6     location     "0";
7     object       omega;
8 }
9 // * * * * *
10
11 dimensions      [0 0 -1 0 0 0 0];
12 internalField   uniform 376.5;
13
14 boundaryField
15 {
16     airfoil
17     {
18         type          fixedValue;
19         value         uniform 5.354e+09;
20     }
21
22     ".*"         // Inlet, Outlet, Top, Bottom
23     {
24         type          inletOutlet;
25         inletValue    $internalField;
26         value         $internalField;
27     }
28
29     frontAndBack
30     {
31         type          empty;
32     }
33 }
34 // ***** //
```

Listing C.6: Boundary Condition file *omega*.

# Appendix D

## Properties of the Rotor Blade

Material	$\rho$ [kg/m <sup>3</sup> ]	$E_x$ [GPa]	$E_y$ [GPa]	$G_{xy}$ [GPa]	$\nu_{xy}$ [-]
Gel coat	1230	3.44	3.44	1.38	0.3
Random mat	1670	9.65	9.65	3.86	0.3
CDB340 triaxial fabric	1700	24.2	8.97	4.97	0.39
Balsa	144	2.07	2.07	0.14	0.22
Spar cap mixture	1700	27.1	8.35	4.7	0.37

Table D.1: Properties of the materials used for the WindPACT 1.5MW blade [64].

Layer	Material	Thickness [mm]
1	Gel coat	0.51
2	Random mat	0.38
3	Triaxial fabric	0.89
4	0% - 15% <i>c</i>	Balsa
	15% - 50% <i>c</i>	Spar cap mixture
	50% - 85% <i>c</i>	Balsa
5	Spar cap mixture	0.89

Table D.2: Composite definition of the blade shell [64].