This project was supported by the Centre for Global Eco-Innovation
and is part financed by the European Regional Development Fund.

Estimating Energy Losses of Utility-Scale Wind Turbines due to Blade Surface Damage.

By

Anna Cavazzini

In collaboration with

Helispeed Solutions Ltd.

I hereby declare that this thesis is the product of my own work and has not been submitted in substantially the same form for the award of a higher degree elsewhere. The research here presented was carried out at Lancaster University from October 2018 to September 2019.

Ideas, methods and analysis are, if not otherwise stated, the outcomes of meetings and discussions with Dr. M. S. Campobasso, Professor P. Angelov, Dr. E. Minisci and G. Packer, CEO of Helipseed Solutions Ltd.

I would like to highlight that excerpts of this thesis will be published in the following International Conference Proceedings :

Proceedings of IOWTC 2019
ASME 2019 2nd International Offshore Wind Technical Conference
November 3-6, 2019, St. Julian's, Malta
IOWTC2019-7578

Anna Cavazzini, Edmondo Minisci and M. Sergio Campobasso,
MACHINE LEARNING-AIDED ASSESSMENT OF WIND TURBINE ENERGY LOSSES DUE TO BLADE LEADING EDGE DAMAGE

Date....................... Signature of candidate.......................................

Contents

1 Introduction **10**

2 Literature review **13**

3 Framework of Energy Loss Prediction System **15**

3.1 System overview 15
3.2 Future operation of the geometry analysis system 16
3.3 Airfoil database 18
3.4 Artificial Neural Network 18
3.5 Wind turbine computational aerodynamics 18
3.6 Power control 18
3.7 Energy estimator and AEP 19

4 Parametrization and generation of leading edge delamination **22**

4.1 Parametrization of leading edge delamination damage 22
4.2 Generation of leading edge delamination damage 26
4.2.1 Coordinates of the damaged airfoil 26
4.2.2 MATLAB code to generate the damaged geometries 28

5 Wind turbine computational aerodynamics **31**

5.1 Mesh generation 31
5.1.1 Computational domain 31
5.1.2 Mesh 31
5.2 CFD analysis set-up 37
5.2.1 Governing equations 37
5.2.2 Numerical settings 39
5.3 Blade element momentum theory analysis 41

6 Power control **43**

6.1 Introduction 43
6.2 Typical speed and pitch control of utility-scale wind turbines 43
6.3 Adaptive control for a damaged turbine 45

7 System automation **53**

7.1 Overview . . . 53

7.1.1 Scripting ANSYS DesignModeler . . . 54

7.1.2 Scripting ANSYS Meshing . . . 56

7.1.3 Scripting ANSYS Workbench and FLUENT . . . 59

7.2 Automation of the system . . . 61

7.2.1 Original idea and its impracticability . . . 61

7.2.2 Alternative applied solution . . . 63

7.2.2.1 Creation of the original workbench project . . . 63

7.2.2.2 ANSYS Workbench script . . . 64

7.2.2.3 Running the simulations . . . 65

7.2.2.4 Extraction of the lift and drag curves . . . 67

7.3 Machine learning approach . . . 67

8 CFD validation **70**

8.1 NREL 5MW blade sections . . . 70

8.1.1 NACA 64-618 airfoil . . . 70

8.1.1.1 Mesh refinement study . . . 70

8.1.1.2 Validation against the experimental results . . . 72

8.1.2 DU 93-W-210 LM airfoil . . . 74

8.1.3 DU 91-W2-250 airfoil . . . 75

8.1.4 DU 97-W-300 LM airfoil . . . 75

8.1.5 DU 99-W-350 airfoil . . . 75

8.1.6 DU 99-W-405 LM airfoil . . . 76

8.2 Comparison between the power curves and loads obtained with CFD results and experimental data . . . 77

8.3 Nominal and damaged DU 96-W-180 airfoils . . . 78

8.3.0.1 Nominal DU 96-W-180 . . . 78

8.3.0.2 Damaged DU 96-W-180 . . . 80

9 Results **85**

9.1 Database generation . . . 85

9.2 Validation of ANN approach . . . 88

9.3 Damaged turbine, AEP loss and power control . . . 95

10 Conclusions and future work **101**

List of Figures

3.1 Definition of ALPS modules and their dependencies. 16

3.2 Idea behind the future development of the Geometry Analysis System (damaged blade from [21]). 17

3.3 Representation of a real damaged blade. 17

3.4 Rayleigh distributions corresponding to different mean wind speeds. 20

3.5 Qualitative representation of the numerical approximation of the AEP. 20

4.1 Parametrization of the leading edge delamination damage. 23

4.2 Example of mildly damaged airfoil, characterized by the parameters $s_u/c \times 100 = 0.6$, $s_l/c \times 100 = 0.52$ and $d/c \times 100 = 0.1$. 24

4.3 Example of moderately damaged airfoil, characterized by the parameters $s_u/c \times 100 = 2.5$, $s_l/c \times 100 = 2.08$ and $d/c \times 100 = 0.3$. 25

4.4 Example of severely damaged airfoil, characterized by the parameters $s_u/c \times 100 = 5.4$, $s_l/c \times 100 = 5.85$ and $d/c \times 100 = 1$. 25

4.5 Schematic illustration of the procedure followed to obtain the normal direction to the considered airfoil at each point P_i on the upper side. 27

4.6 Illustation of the procedure to obtain the x and y coordinates of the delaminated airfoil. 28

4.7 Curves generated for each one of the damages. 30

5.1 Computational domain for the CFD simulations: the farfield is divided into a velocity inlet and a pressure outlet. 32

5.2 Mesh around the nominal NACA 64-618 with close up views of the leading and trailing edges: special care was taken to assure a good orthogonal quality. 33

5.3 Wall y^+ around the nominal NACA 64-618 airfoil: the figure shows that its value is lower than 1 for all the values of x/c, thus the grid fulfils the constraints. 35

5.4 Mesh around a damaged NACA 64-618 airfoil with close up views of the leading and trailing edge: the face corresponding to the damage has been meshed separately in order to ensure a good orthogonal quality along all the airfoil surface. 36

5.5 Nondimensional wall y^+ around the damaged NACA 64-618 airfoil. 37

6.1 Nominal wind turbine power control. 44

6.2 Rotor torque as a function of the rotor speed currently adopted for speed and pitch regulated wind turbines. It is possible to notice the linear relation between the two quantities in regions 1.5 and 2.5, and the quadratic relation in region 2. 46
6.3 Damaged wind turbine power control. 47
6.4 Rotor power as a function of the wind speed for a particular case of damaged turbine at the fixed rotor speed $1.3\Omega_{c.i.}$. 48

7.1 Workbench file created for a each airfoil. 54
7.2 Curves generated for each one of the damages. 55
7.3 Detail of the computational domain around a damaged airfoil. 55
7.4 Division of the airfoil profile in the meshing process. 56
7.5 Edges on the boundary of the domain and internal lines. 57
7.6 Tasks performed by the ANSYS Workbench journal file; $N_{airfoils}$ refers to the number of different baseline geometries analysed, while N is the total number of damaged airfoils referring to the same baseline geometry.. 60
7.7 Work flow of the original idea: the definition of the tasks can be found in figure 7.6 or in subsection 7.1.3. $N_{airfoils}$ is the total number of different baseline airfoils considered, while N is the number of delaminated geometries corresponding to the same baseline airfoil. 62
7.8 Schematic explanation of the process described in this section. The complete Workbench project is created only for airfoil 1 and then copied inside all the damaged airfoils folders. 65
7.9 Schematic explanation of the process described in this section. 66
7.10 General schematic of one-layer MLP feed-forward ANN systems. 68

8.1 The figure shows the six airfoils composing the NREL 5 MW blade. 71
8.2 Blade of the NREL 5 MW turbine. 72
8.3 Medium grid around the NACA 64-618 airfoil, with a close-up view of the leading edge. 73
8.4 Results of the mesh refinement study for the NACA 64-618 airfoil. 73
8.5 Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 93-W-210 LM airfoil, CFD results versus experimental data. 74
8.6 Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 91-W2-250 airfoil, CFD results versus experimental data. 75
8.7 Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 97-W-300 LM airfoil, CFD results versus experimental data. 75
8.8 Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 99-W-350 airfoil, CFD results versus experimental data. 76
8.9 Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 99-W-405 LM airfoil, CFD results versus experimental data. 77
8.10 Power (P), torque (Q) and thrust (T) against wind speed (U). The solid line refers to the curves obtained with the experimental data, while the dashed line to those obtained using the CFD data. 77

8.11 Mesh for the DU 96-W-180 airfoil with a close-up view of the leading edge. 79
8.12 Results of the mesh refinement study for the nominal DU 96-W-180 airfoil. 79
8.13 Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 96-W-180 airfoil, CFD against experimental data. 80
8.14 Damage chosen for the validation of the DU damaged 96-W-180 airfoil. 81
8.15 Mesh for the DU 96-W-180 damaged airfoil with a close-up view of the leading edge. 82
8.16 Results of the mesh refinement study for the damaged DU 96-W-180 airfoil. 83
8.17 Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the damaged DU 96-W-180 airfoil, CFD against experimental data. 83
8.18 Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the nominal and damaged DU 96-W-180 airfoil, CFD against experimental data. . 84

9.1 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the NACA 64-618 airfoil. 89
9.2 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the NACA 64-618 airfoil. 89
9.3 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 93-W-210 LM airfoil. 90
9.4 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 93-W-210 LM airfoil. 90
9.5 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 91-W2-250 airfoil. 91
9.6 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 91-W2-250 airfoil. 91
9.7 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 97-W-300 LM airfoil. 92
9.8 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 97-W-300 LM airfoil. 92
9.9 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 99-W-350 airfoil. 93
9.10 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 99-W-350 airfoil. 93

9.11 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 99-W-405 LM airfoil. 94
9.12 Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 99-W-405 LM airfoil. 94
9.13 3D view of analyzed leading edge erosion damage. 96
9.14 Comparison of lift and drag forces of nominal and damaged airfoils at six radial positions. 97
9.15 Steady state performance and control curves. Solid, dotted and dashed lines refer respectively to analysis of turbine with nominal blade geometry, damaged blade surface using CFD, and damaged blade surface using ML. 100

10.1 Example of a damaged blade that could be analysed with the ALPS system developed in this work. 102

List of Tables

4.1 Damage parameter choice for ALPS database. 24

6.1 Comparison between the control of the nominal turbine and that of the damaged one. 52

7.1 Number of elements on edges 3 and 4 for the most severe delamination damage of each category. 59

8.1 Parameters for the mesh refinement study for the NACA 64-618 airfoil. 71
8.2 Parameters for the mesh refinement study for the DU 96-W-180 damaged airfoil. . 82

9.1 Successful simulations for each model of airfoil in the generation of the database. . 85
9.2 Percentage of failed CFD simulations for each considered angle of attack for the NACA 64-618 airfoil. 86
9.3 Percentage of failed CFD simulations for each considered angle of attack for the DU airfoils. 87
9.4 Main geometric values of analyzed blade damage. 95
9.5 Damage geometry parameters at six radial positions. 97

Chapter 1

Introduction

Blade surface damage can significantly impair the performance of wind turbines, leading to a non-negligible power loss over the entire range of operating wind speeds. The maintenance of wind farms is difficult and expensive, particularly for those at sea, and new solutions are being sought in order to reduce its cost and inconvenience. A recent Offshore Renewable Energy (ORE) CATAPULT project aims at developing a completely automated maintenance process of the offshore wind turbines, using drones and robots [25]. Up to now, periodic checks are planned at defined intervals of time. This means that the maintenance may not be necessary at the moment it is done, and that its cost could be higher than the economic loss associated with the power losses caused by blade surface damage. To help in overcoming these difficulties, this work introduces a new prototype able to link to the blade surface damage the associated power loss, in order to decide whether the maintenance is necessary or not. The interest in understanding and quantifying the dependence of wind turbine energy production to blade damage, particularly in the frequent case of damage due to leading edge erosion, is increasing. However, to the best of the author's knowledge, there is presently no research into developing a reliable and rapid system for assessing such losses or tackling the challenges of developing this type of system for industrial application. The work presented in this thesis aims at addressing this shortfall by presenting and demonstrating the modular wind turbine AEP loss prediction system (ALPS) framework, a novel data- , CFD- and engineering code-based technology for assessing wind turbine energy losses and load variations due to blade surface damage. The information made available by this technology can be used to *a*) quantify the revenue reductions due this type of AEP losses and help asset managers decide when it is most convenient to undertake maintenance, and *b*) assess the convenience of adopting safeguards such as leading edge protection [12, 27]. The rationale behind the development of ALPS is to provide a tool capable of rapidly and reliably assessing AEP losses due to real blade surface damage with a definition available in the form of images. Although ALPS has been designed to deal with fairly general surface damage pattern and severity, the demonstration provided in the study described in this thesis focuses on the AEP losses of a multi-megawatt offshore wind turbine caused by moderate to advanced erosion-induced leading edge delamination. The aim of this work is to explain the structure and capabilities of the ALPS system, and to describe in detail the implementation of the different modules composing it.

Prior to the effective implementation of the system, an extensive literature review has been carried out to analyse the state of the art of the subject. This is a fairly new field, therefore is was not possible to find many similar works in the literature. The analysis of previous relevant studies linked to this research is presented in chapter 2.

In chapter 3, an overview of the developed Annual Energy Production Loss Prediction System (ALPS) is presented. The main features of each module are described, in order to provide the reader with a general idea of the purpose of the system and its functionalities. Each component of the ALPS will then be illustrated in more detail in the following chapters.

In chapter 4, the leading edge delamination analysed within this work is described. First of all, the parameters defining the damage are presented and shown graphically. Then, the geometric procedure to obtain the coordinates of the damaged airfoils starting from those of the nominal one is explained. The damages have been generated automatically with the help of a MATLAB script, whose pseudo-code is presented in the final section of the chapter.

All the aspects regarding the aerodynamics of wind turbines are collected in chapter 5. First of all, the mesh generation process is described both for the nominal and for the damaged airfoils. Then, the equations governing the flow around the airfoils are recalled for the reader, and the numerical settings chosen for the simulations are described. Finally, the main characteristics of the blade element momentum theory code used in this work are stated.

Chapter 6 deals with the description of the power control strategy adopted both for the nominal and damaged turbines. The power control of the nominal turbine has not been implemented within this work: it was already part of the BEMT code used, developed by Jonkman *et al.* [18]. The control of the nominal turbine is unlikely to yield an optimal performance of the damaged turbine, due to altered aerodynamics properties of the eroded blades. For this reason, a slightly different control strategy has been implemented for the turbine characterized by the delamination damage. Both power controls are described in chapter 6: one of the nominal turbine is explained briefly, while more attention is paid to that of the damaged one.

An important part of the work presented herein consists in the generation and analysis of thousands of delaminated geometries. The analysis of one damaged airfoil requires, if done manually, about two hours of work. It was therefore necessary to find a way to automatise the process. The system automation is presented in chapter 7: first of all, an overview of the problem is provided to the reader, then the automation of each adopted software is described. Finally, the automation of the whole process is explained, together with the problems encountered and the solutions found. The chapter ends with the description of the last component of the automation process: the machine learning approach, which enables the lift and drag curves associated to every possible damage to be obtained by learning from a limited set of data.

The model turbine chosen for the work is introduced in chapter 8. The chapter starts with a brief description of the blade, then the CFD set-up presented in chapter 5 is validated against the wind tunnel data available for all the airfoils composing the blade. Since the grids generated for the six different airfoils are all characterized by the same features, a mesh refinement study has been carried out only for the airfoil closest to the tip of the blade, which is particularly involved in the power capture. In the last part of the chapter, a different validation was carried out for the DU 96-W-180 airfoil. This airfoil was chosen in the experimental work of Sareen *et al.* [26] to analyse the effect of leading edge delamination damage on wind turbines' blades. The wind tunnel measurements available both for the nominal and for the damaged airfoil have been compared to the results of the simulations in order to validate the damage generation procedure developed in this work.

Chapter 9 presents the results achieved within the project. First of all, the obtained database is described, together with a validation of the adopted ANN approach. Then, the capabilities of ALPS are demonstrated by applying it to the analysis of a damaged blade.

Finally, in chapter 10 an analysis of the work done is presented, together with the description of further ideas that will be object of future developments.

All the work described in this thesis was carried out by the author, under the constant supervision of M. S. Campobasso and P. Angelov, apart from the implementation of the machine learning algorithms. These were developed by E. Minisci as part of a collaboration with Strathclyde University in Glasgow.

Chapter 2

Literature review

Wind turbine blades often operate in harsh environmental conditions, being exposed to precipitation that occurs in a variety of forms. Over time, rain, hail and airborne abrasive particles such as sand, can erode blade surfaces, particularly at the leading edge. Recent experimental evidence indicates that leading edge erosion reduces significantly the aerodynamic performance of wind turbine blades and, thus, the power plant energy production, by more than 20 % [5, 26]. Significant energy losses due to geometry alterations of the leading edge surface region can also occur due to insect, dust [19] and ice accretion [33].

Previous research efforts to assess the impact of wind turbine in-service blade surface alterations on aerodynamic power losses have focused on the detrimental effect of ice accretion [16], dust [19, 6] and insect debris [9] accumulation, and leading edge roughness [31, 29, 11]. Leading edge erosion also results in significant reductions of the lift and large increments in the drag forces, and can thus dramatically reduce the blade performance, particularly in the outboard region, where the relative air speeds are high and most energy capture take place. Despite this, however, the study of the energy losses due to blade erosion has received fairly little attention thus far.

Sareen *et al.* [26] carried out a comprehensive campaign of wind tunnel measurements to assess the aerodynamic performance loss of the DU 96-W-180 wind turbine airfoil subject to geometry alterations due to different patterns (pits, gouges and delamination) and severity (moderate, medium and advanced) of leading edge erosion. Their eroded airfoil design relied on photographic records of wind turbine blades in operation and eroded blades undergoing repair provided by 3M. These photographic data sets collected from wind power plant operators covered a range of rotor blade sizes up to and including megawatt-scale rotors that had been in operation for 1 to 10+ years. Making use of their airfoil force data based on wind tunnel testing, and the PROPID wind turbine analysis and design code, the authors found that the loss of Annual Energy Production (AEP) of a 2.5 MW class turbine due to leading edge erosion can approach 25 % of the nominal value. Schramm *et al.* [27] used OpenFOAM [24], an open-source Navier-Stokes (NS) computational fluid dynamics (CFD) package, and the National Renewable Energy Laboratory (NREL) wind turbine engineering code FAST [17] to assess the impact of a particular leading edge delamination pattern on the AEP of the NREL 5 MW reference turbine [18]. With reference to wind frequency data measured with a met mast at Risø [30], they reported an AEP reduction of about 8 % for the leading edge damage they considered. Wang *et al.* [32] developed a pitting erosion geometric model for the NS CFD analysis of the aerodynamic performance of wind turbine airfoils suffering from this type of surface damage. Each erosion pit was modeled as a semi-circular cavity. Making use of two-dimensional (2D) flow simulations performed with the ANSYS FLUENT incompressible Reynolds-averaged Navier-Stokes (RANS) solver coupled to Menter's $k-\omega$ Shear Stress Transport (SST) turbulence model [23], the authors carried out comprehensive parametric analyses of the dependence of the lift reduction and drag increment of the S809 wind turbine airfoil [28] on the depth, surface density, surface extent, and location (distance from the leading edge) of the erosion pits. For all these parameters, critical values were determined above which the airfoil performance did not undergo any further significant reduction.

Chapter 3

Framework of Energy Loss Prediction System

3.1 System overview

ALPS is a modular analysis framework consisting of several interlinked modules. Its main function is to determine the annual energy production of a wind turbine whose blades feature surface damage of general pattern and severity. The modules of ALPS are indicated in the block diagram of figure 3.1, which also shows all module dependencies. The calculation of the wind turbine power and loads for the entire range of operating wind speeds is accomplished by means of an engineering code based on the blade element momentum theory (module labeled 'Wind Turbine Model'), which requires the lift and drag curves of the airfoils making up the cross sections of the damaged blade considered. The airfoil geometry and force database of ALPS (box labelled 'airfoil Database') contains geometry and force data of airfoils whose shape conforms to the nominal design intent or differs from that because of a blade surface damage developed and/or accrued during turbine operation. The airfoil data presently included in the ALPS database have been computed with Navier-Stokes Computational Fluid Dynamics, but hybridization methods aiming to make concurrent use of experimental and CFD airfoil force data will be investigated in forthcoming extensions.

The main input of ALPS is the current 3D geometry of the blades, including any surface damage (box with enlarged view of delaminated blade leading edge). This information is passed to a geometry analysis system (GAS), which provides a sectional representation of the damaged blade surface, *i.e.* nominal or damaged airfoil profiles corresponding to a user-given number of blade cross sections. In general, none of the damaged airfoil profiles will be contained in the current database. Therefore, a machine learning-based method is used to determine the lift and drag force data of each damaged airfoil making use of the knowledge repository of the airfoil database. In this study, an Artificial Neural Network (ANN) system is used to estimate the values of lift and drag for the whole operating range of the local angle of attack (AoA). The GAS module includes parametric representations of typical blade surface damage patterns, and, for each damaged cross section extracted from the monitored 3D blade geometry, returns the value of the parameters defining the damage of the associated airfoil. Such set of parameters, along with a set of operation parameters of the blade sections (*e.g.* AoA range) forms the input to the ANN module, which returns force data over the requested range of operating conditions.

The force data of the damaged blade sections are then used by the wind turbine module to determine the power curve and the aerodynamic loads of the turbine being assessed. To obtain a realistic estimate of the turbine with damaged blades, however, it is essential to modify the turbine power control strategy yielding the optimal values of rotor speed below rated wind speed and the value of the blade pitch limiting the power to its rated value above the rated wind speed. The modified power control algorithm is indicated by the box labeled 'Control' in figure 3.1, and the modified control strategy is discussed later in the thesis. The damaged turbine AEP is then computed by integrating the power curve of the damaged turbine against the wind frequency distribution at the turbine installation site.

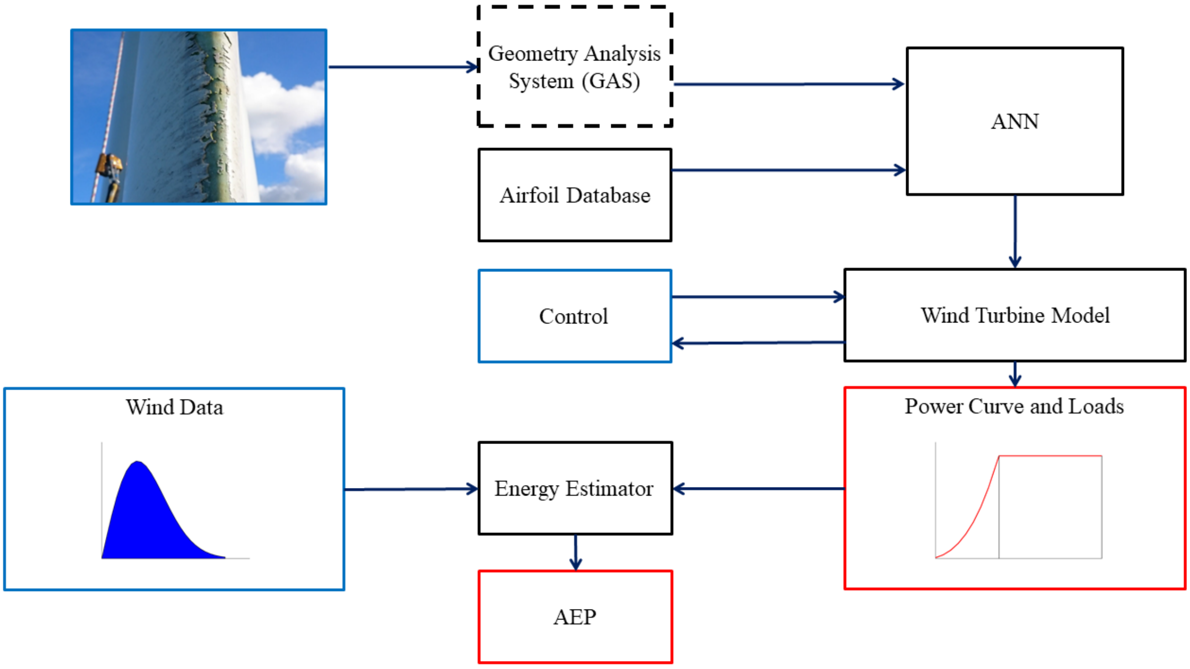


Figure 3.1: Definition of ALPS modules and their dependencies.

3.2 Future operation of the geometry analysis system

The first module of ALPS will be the Geometry Analysis System (GAS). As can be noticed from figure 3.1, the GAS box is dashed. The meaning of the dashed line is that the module has not been implemented yet, and it will be object of future work. Its purpose will be to provide a representation of the real damage observed on the blade of an operating wind turbine. The main idea of its future operation is summarized in figure 3.2:

- The input of the GAS system will be a set of pictures of a damaged blade from different angles and positions. In such a way, the system will be able to reconstruct the 3D geometry of blade and damage, and particularly its depth. In figure 3.2, only one picture of the blade is shown for simplicity.

- The damaged blade will be divided by the GAS system into a user-given number of slices.
- Each slice represents a damaged airfoil geometry, and it is characterized by three parameters: the curvilinear length of the damage on the upper side of the blade s_u, the curvilinear length on the lower side s_l and the depth d, as shown in figure 3.2.

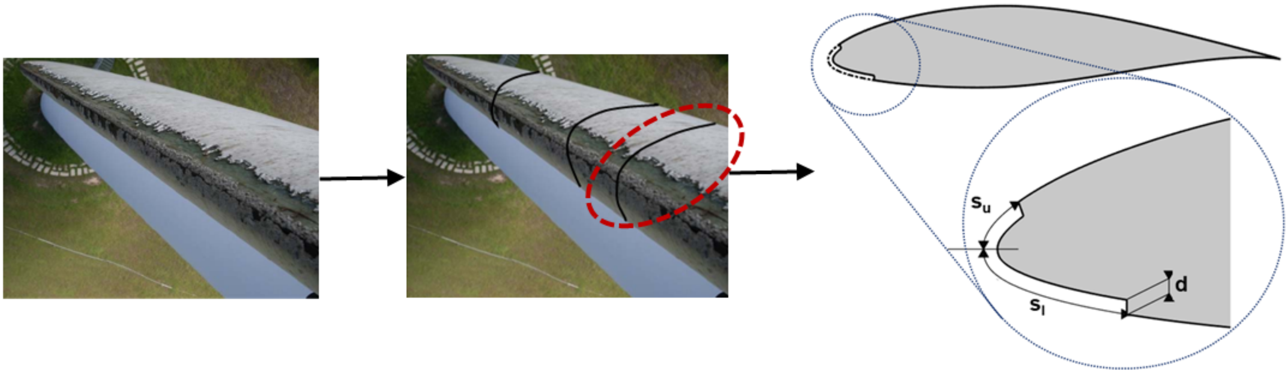


Figure 3.2: Idea behind the future development of the Geometry Analysis System (damaged blade from [21]).

Choosing enough slices, it will be possible to obtain a realistic representation of the observed damage. To give a qualitative representation of what could be achieved in the future, an example has been created applying a random distribution of the parameters s_u, s_l and d to 600 slices of the NREL 5 MW model turbine. The result can be seen in figure 3.3: the represented damage is not identical to the real one considered for reference, but this is nevertheless a promising result that encourages to develop and improve the technology.

Figure 3.3: Representation of a real damaged blade.

3.3 Airfoil database

The airfoil database contains the lift and drag curves of a selected set of damaged airfoils. For each nominal airfoil geometry, three levels of damage are considered: mild, moderate, and severe. These are characterized by three parameters: the curvilinear length on the upper side of the airfoil, the curvilinear length on the lower side, and the depth of the damage. Given a triad of values for these parameters, the geometry of the damaged airfoil is obtained using an in-house MATLAB program; while the lift and drag curves are acquired by means of Navier-Stokes CFD simulations. More detail on the creation of the airfoil database can be found in chapter 4.

3.4 Artificial Neural Network

In general, none of the damaged airfoils obtained with the GAS system from the picture of a real blade will be included in the database. Running CFD simulations for each one of the observed damages would increase enormously the computational time needed to obtain an estimate of the power loss of the turbines under examination. The Artificial Neural Network (ANN) procedure, learning from the limited set of data computed by means of NS CFD, enables the ALPS system to predict the unknown values of lift and drag for every possible damage without the need to run CFD simulations again. The input to the ANN system is represented by the parameters characterizing the damage, together with the considered angle of attack. Once the needed information is provided to the system, the output represented by the lift and drag curves can be obtained in a few seconds. This means that, once a database has been constructed in one or two weeks running NS CFD simulations for a limited number of damages, the energy loss of a wind farm consisting of hundreds of turbines can be determined in a few minutes thanks to the ANN procedure. More details on the implementation of the system will be provided in chapter 7.

3.5 Wind turbine computational aerodynamics

For each operating condition, namely set values of wind speed, rotor speed and blade pitch, the aerodynamic power and rotor loads are determined with a blade element momentum theory (BEMT) code using as input also the lift and drag force data of the nominal or damaged airfoils corresponding to the rotor blade sections. The force data for the airfoils making up the blade under analysis are determined by means of the ANN system, which, in turn, requires the force data for all airfoils in the ALPS database. The airfoil force data of the ALPS database are determined with Navier-Stokes CFD. The main features of the BEMT and CFD codes used in the analyses reported below are summarized in chapter 5.

3.6 Power control

The general strategy of the power control algorithm of utility-scale wind turbines aims at maximizing power by maximizing the power conversion efficiency between cut-in and rated wind speeds, and limiting power to the rated value between rated and cut-out wind speeds. In real applications there exists transitional regions before and between these two main regions. In this work, the general power control strategy developed for the NREL 5 MW reference turbine [18] is adopted. However, some alterations to this power control strategy are adopted for the turbine with damaged blades to avoid overestimating the AEP loss due to the blade leading edge damage. The power control strategies adopted for the nominal and damaged turbines are described in chapter 6.

3.7 Energy estimator and AEP

Of great interest for the wind farm owners is the annual energy production (AEP), the energy harvested by each wind turbine during one year of operation. Since the power produced by the turbine is a function of the wind speed, this quantity depends strongly on the wind speed distribution at the site of installation of the wind turbines. Clearly, it is not possible to know exactly the wind speed in every wind farm location, every hour of the day of every year, thus the wind speed distribution is approximated by a probability function. For this purpose, the most used probability distributions are the Weibull and the Rayleigh. The Weibull distribution depends on two parameters: a shape factor and a scale factor. The Rayleigh distribution, instead, depends on only one parameter: the mean wind speed at the particular site considered. This is usually obtained from meteorological mast measurements taken throughout a year or more in a location as close as possible to that of the wind farm. It might occur that the meteorological mast is not at the same height of the wind turbine hub. The wind speed varies as a function of the distance from the ground, so usually a power law is used to obtain the wind speed at the needed height, starting from the known wind speed at a different altitude:

$$\frac{U_1}{U_2} = \left(\frac{h_1}{h_2}\right)^{\alpha}, \tag{3.1}$$

where U_1 and U_2 are the wind speeds at height h_1 and h_2 respectively, and α is a parameter that depends on a variety of factors, such as the type of terrain, the concentration of buildings, etc.; usually $\alpha = 1/7$. Once known the mean wind speed of the chosen site at the altitude of the turbine's hub, $(\bar{U})$, the Rayleigh probability density function is given by:

$$prob(U) = \frac{\pi}{2}\left(\frac{U}{\bar{U}^2}\right)\exp\left\{-\frac{\pi}{4}\left(\frac{U}{\bar{U}}\right)^2\right\}. \tag{3.2}$$

Some Rayleigh distributions for different mean wind speeds are shown in figure 3.4. To obtain an estimate of the annual energy production, it is necessary to calculate the area underlying the curve $P(U) \times prob(U) \times h_{year}$ (the black curve in figure 3.5), where $P(U)$ is the power of the turbine as a function of the wind speed, $prob(U)$ is the probability to observe a wind speed U in the chosen site, and h_{year} is the number of hours in a year. Notice that $prob(U) \times h_{year}$ gives the expected number of hours in a year during which the wind speed is U. Multiplying this quantity by the power corresponding to U, and integrating over all the operational wind speeds of the turbine, one gets:

$$AEP = h_{year}\int_{U=U_{c.i}}^{U=U_{c.o.}} P(U)prob(U)\ dU, \tag{3.3}$$

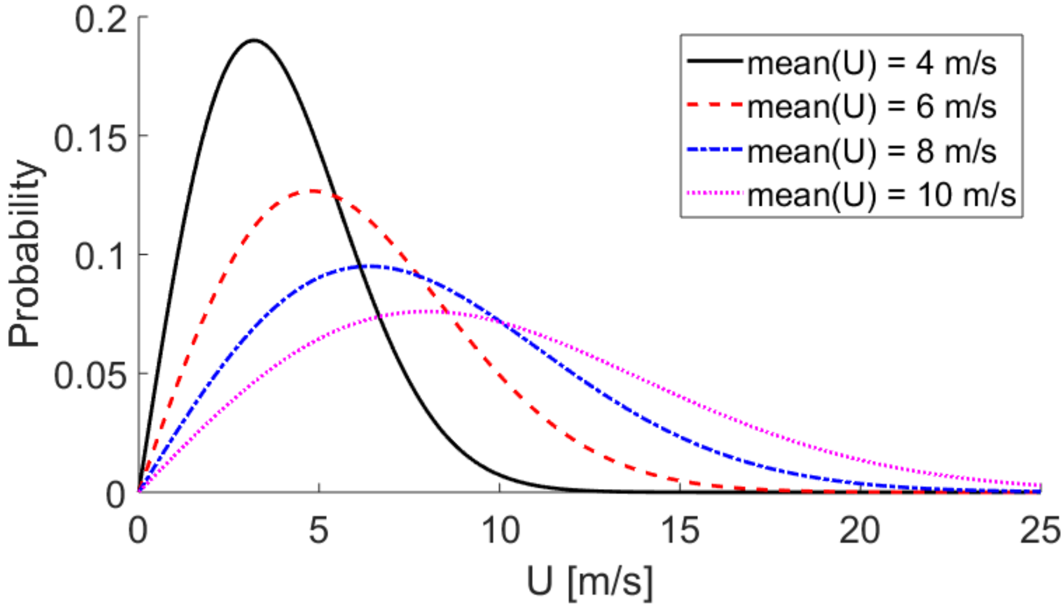


Figure 3.4: Rayleigh distributions corresponding to different mean wind speeds.

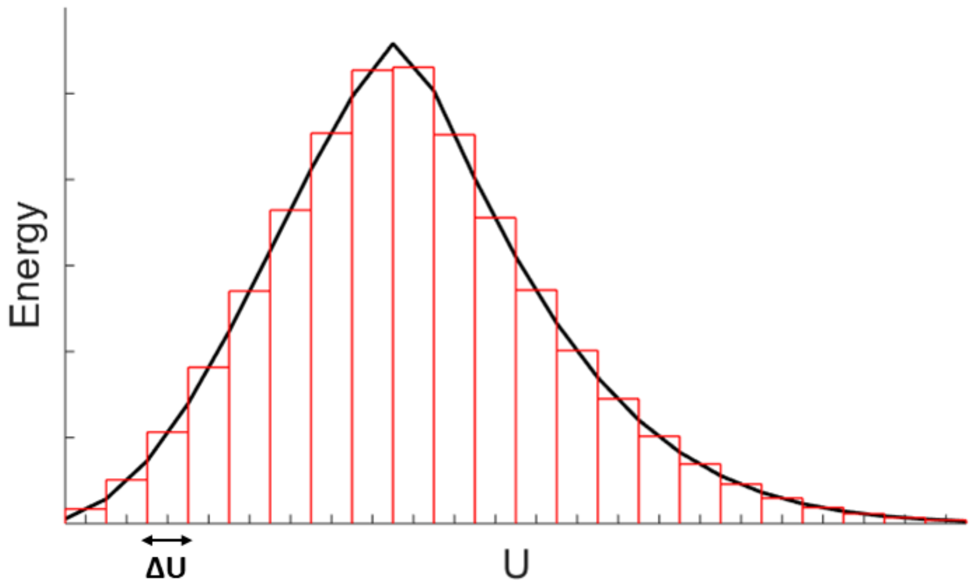


Figure 3.5: Qualitative representation of the numerical approximation of the AEP.

where $U_{c.i.}$ is the cut-in wind speed and $U_{c.o.}$ is the cut-out wind speed. By dividing the interval $[U_{c.i.}, U_{c.o.}]$ into N subintervals of length ΔU, and using the property of linearity of the integral:

$$AEP = h_{year} \sum_{i=1}^{i=N} \int_{U_i - \frac{\Delta U}{2}}^{U_i + \frac{\Delta U}{2}} P(U_\infty) prob(U_\infty) \ \ dU_\infty. \tag{3.4}$$

Approximating these integrals with the areas of the rectangles having width ΔU and height U_i (the red rectangles in figure 3.5), one gets:

$$AEP = \Delta U h_{year} \sum_{i=1}^{i=N} P(U_i) prob(U_i) \tag{3.5}$$

and this is a good estimate of the annual energy production of the wind turbine. More information on this subject can be found, for example, in [7].

Chapter 4

Parametrization and generation of leading edge delamination

This chapter describes the procedure for obtaining the geometry of an airfoil with leading edge delamination damage. First of all, the parametrization chosen to characterize the damage is described. Then, the chosen set of values for the parameters are summarized in a table, and three graphic examples of damage are provided: a mild, a moderate, and a severe delamination affecting the leading edge. Finally, the procedure to obtain the coordinates of the damaged airfoils making use of the software MATLAB is described in great detail.

4.1 Parametrization of leading edge delamination damage

The parameters chosen to describe the delamination damage on the leading edge of an airfoil are, as shown in figure 4.1:

- s_u: the curvilinear length of the damage on the upper side of the airfoil (suction side);
- s_l: the curvilinear length of the damage on the lower side of the airfoil (pressure side);
- d: the uniform depth of the damage.

The delamination damage has been divided into three categories: mild, moderate, and severe, each one characterized by a range of values of the parameters. It is not easy to find in the literature reference values for the parameters characterizing the delamination. The most exhaustive source of information is the paper by Sareen *et al.* [26], in which wind tunnel experiments are carried out for the DU 96-W-180 airfoil with different types of damage: pits, gauges, and delamination. In these experiments, the depth of the delamination is kept constant to a value of 3.81 mm. Since the chord of the airfoil used for the experiments is 0.457 m, this means that $d/c \times 100 = 0.834$. The curvilinear length of the damage is instead differentiated by Sareen *et al.* depending on the severity of the damage. For a mild damage, $s_u/c \times 100 = 1$, and $s_l/c \times 100 = 1.3$. The damage on the lower side is considered to be 1.3 times the one on the upper surface because the angle of attack is most of the time positive, and this causes a greater damage on the pressure

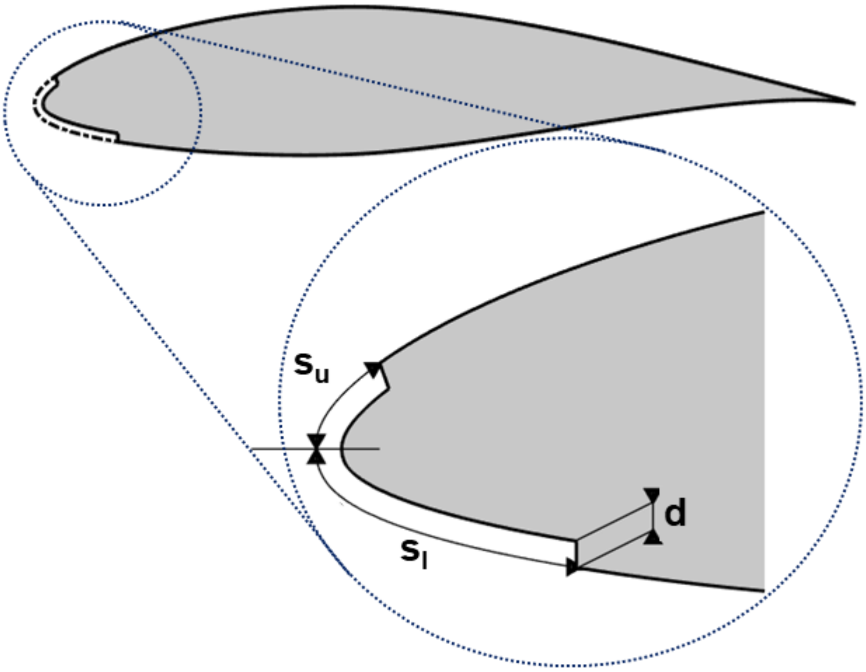


Figure 4.1: Parametrization of the leading edge delamination damage.

side compared to the suction side (see [26]). For moderate damage, following the same criterion, Sareen *et al.* set the curvilinear extension on the upper side is $su/c \times 100 = 2$, and on the lower side $s_l/c \times 100 = 2.6$. Finally, for a severe damage, the values chosen by the authors are $s_u/c \times 100 = 3$ and $s_l/c \times 100 = 3.9$. The curvilinear extension of the leading edge delamination considered in the work presented herein is instead characterized by a wider range. Following the idea of Sareen *et al.*, the values used to characterized the delamination on the lower side of the airfoil are 1.3 times the values used for the upper side. This means that, if the mild delamination on the upper side of the airfoil goes from $s_u/c \times 100 = 0.2$ to $s_u/c \times 100 = 1.0$ with a step of 0.2, then the values for the lower side start from $s_l/c \times 100 = 0.26$ and end with $s_l/c \times 100 = 1.3$ with a step of 0.26. The depth of the damage is not kept constant in this work, but varied depending on the delamination level. The constant depth considered by Sareen *et al.* in [26] of $d/c \times 100 = 0.834$ corresponds in our parametrization to a severe delamination damage. For each delamination stage (mild, moderate and severe), the 3 parameters s_u, s_l and d are varied independently between the minimum and maximum values, leading to the creation of a large number of different damaged geometries for each airfoil. The geometry of the nominal airfoils is perturbed according to the value range of the delamination parameters reported in table 4.1. The number of cases considered are 36 for the mild delamination, 245 for the moderate delamination, and 726 cases for the severe delamination, leading to a total of 1007 cases. Some examples of delaminated geometries are shown in figures 4.2, 4.3, 4.4.

The mild delamination in figure 4.2 is characterized by $s_u/c \times 100 = 0.6$, $s_l/c \times 100 = 0.52$

	Mild			Moderate			Severe		
	min	max	Δ	min	max	Δ	min	max	Δ
$s_u/c \times 100$	0.5	1.0	0.2	1	2.8	0.3	3	6	0.3
$s_l/c \times 100$	0.65	1.3	0.26	1.3	3.6	0.39	3.9	7.8	0.39
$d/c \times 100$	0.1	0.1	-	0.2	0.6	0.1	0.7	1.2	0.1

Table 4.1: Damage parameter choice for ALPS database.

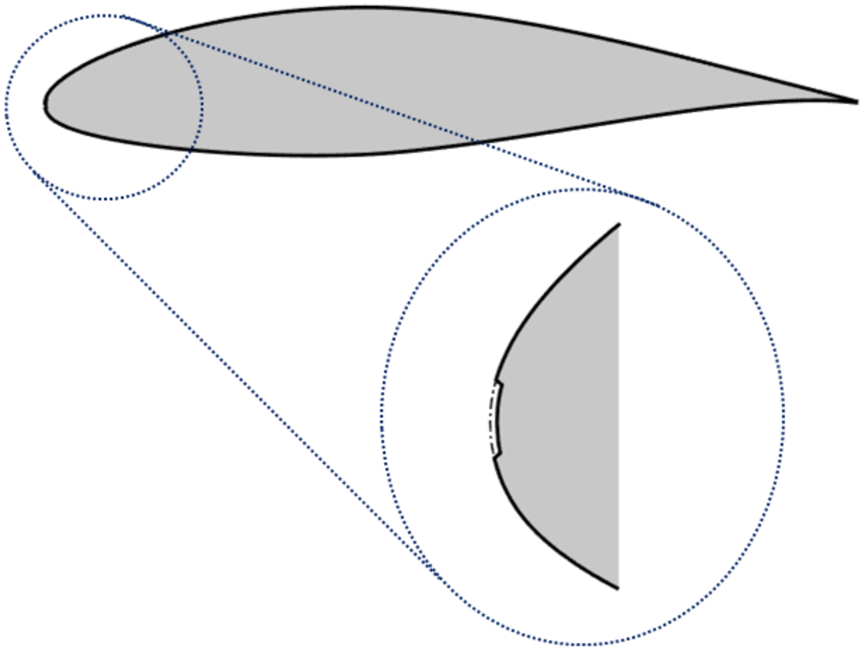


Figure 4.2: Example of mildly damaged airfoil, characterized by the parameters $s_u/c \times 100 = 0.6$, $s_l/c \times 100 = 0.52$ and $d/c \times 100 = 0.1$.

and $d/c \times 100 = 0.1$, and it is noticeable only considering a close-up view of the leading edge. The moderate erosion damage of figure 4.3 is characterized by $s_u/c \times 100 = 2.5$, $s_l/c \times 100 = 2.08$ and $d/c \times 100 = 0.3$, and it is more evident. Finally, an example of severe erosion damage is shown in figure 4.4: here $s_u/c \times 100 = 5.4$, $s_l/c \times 100 = 5.85$ and $d/c \times 100 = 1$ and the damage is clearly noticeable even without taking a close-up view of the leading edge.

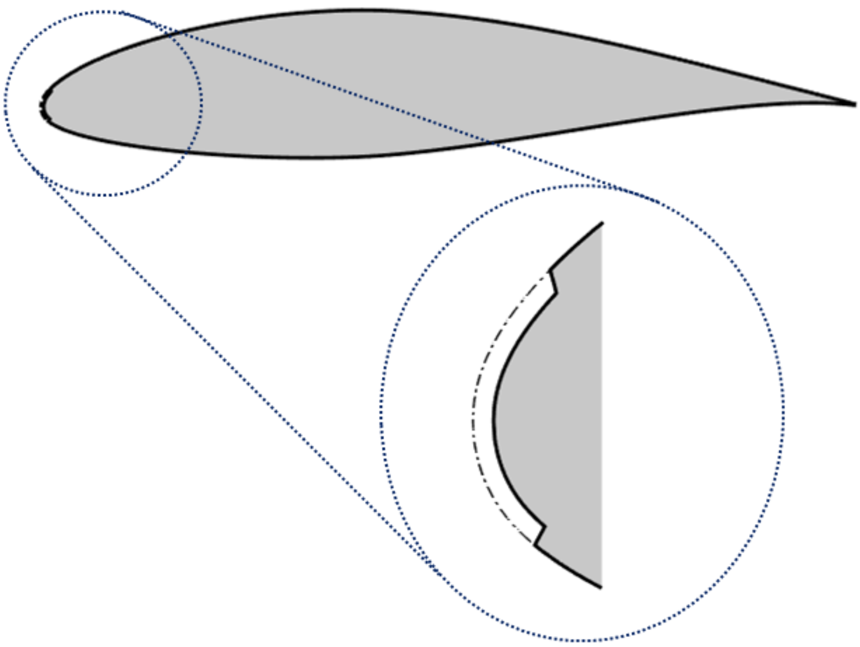


Figure 4.3: Example of moderately damaged airfoil, characterized by the parameters $s_u/c \times 100 = 2.5$, $s_l/c \times 100 = 2.08$ and $d/c \times 100 = 0.3$.

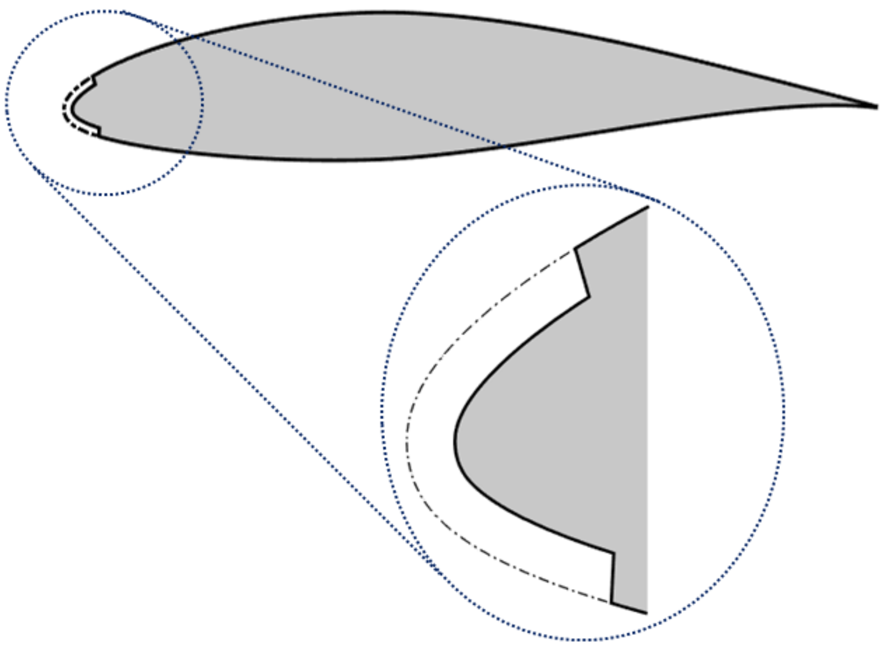


Figure 4.4: Example of severely damaged airfoil, characterized by the parameters $s_u/c \times 100 = 5.4$, $s_l/c \times 100 = 5.85$ and $d/c \times 100 = 1$.

4.2 Generation of leading edge delamination damage

This section deals with the automated generation of the damaged airfoils database using MATLAB. In the first part, the geometrical procedure followed to obtain the coordinates of the delaminated airfoils from those of the nominal ones is explained. In the second part, the structure of the program is described and a pseudo-code of the algorithm is presented.

4.2.1 Coordinates of the damaged airfoil

The coordinates of the damaged geometries are obtained from those of the nominal airfoils, which can be found on the National Renewable Energy Laboratory (NREL) site [2]. The files downloaded from here contain the x and y coordinates of 400 points defining the airfoil geometry, starting and ending on the trailing edge and moving counter-clockwise. This means that the coordinates from 1 to 200 belongs to the upper side, while the coordinates between 201 and 400 to the lower side. To generate the delaminated geometry, first of all it is necessary to determine the curvilinear length on the pressure side and on the suction side of the airfoil, denoted as s_u and s_l respectively. To do so, the coordinates are divided into the ones belonging to the suction side and the ones belonging to the pressure side, and they are ordered so that the first point corresponds to the leading edge and the last one to the trailing edge. Then, each component i of the vectors s_u and s_l contains the curvilinear length from the leading edge of the airfoil to the point with coordinates (x_i, y_i), namely:

$$\begin{aligned} s_u(i) &= \sum_{j=1}^{j=i} \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2} \\ s_l(i) &= \sum_{j=1}^{j=i} \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2} \end{aligned} \tag{4.1}$$

Consider now a delamination damage characterized by the parameters s_u, s_l and d. First of all, the coordinates of the points better approximating a curvilinear length s_u on the upper side and s_l on the lower side of the airfoil are obtained. They are denoted by x_{s_u} and x_{s_l} respectively. Then, the program generates separately the damages on the upper and lower sides. Consider, first of all, the upper side of the airfoil. All the points characterized by $x_i > x_{s_u}$ are the same for the damaged and nominal airfoil, so if $x_i > x_{s_u}$, then $x'_i = x_i$ and $y'_i = y_i$, where x' and y' denote the coordinates of the damaged airfoil. If $x_i < x_{s_u}$, instead, the coordinates of the damaged airfoil are different from those of the nominal one because they correspond to a damaged section. The delamination damage is considered as an erosion of the airfoil material of depth d in the normal direction, which is approximated following the procedure explained below (see figure 4.5):

- Excluding the leading edge, the angular coefficient of the tangent line to the airfoil at point P_i is approximated by the one of the line connecting the points P_{i-1} and P_{i+1}, namely:

$$m = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} \tag{4.2}$$

- The angular coefficient of the normal line is then given by:

$$m_n = -\frac{1}{m} \tag{4.3}$$

- The angle between the x axis and the normal line is given by:

$$\alpha = \arctan(m_n) \tag{4.4}$$

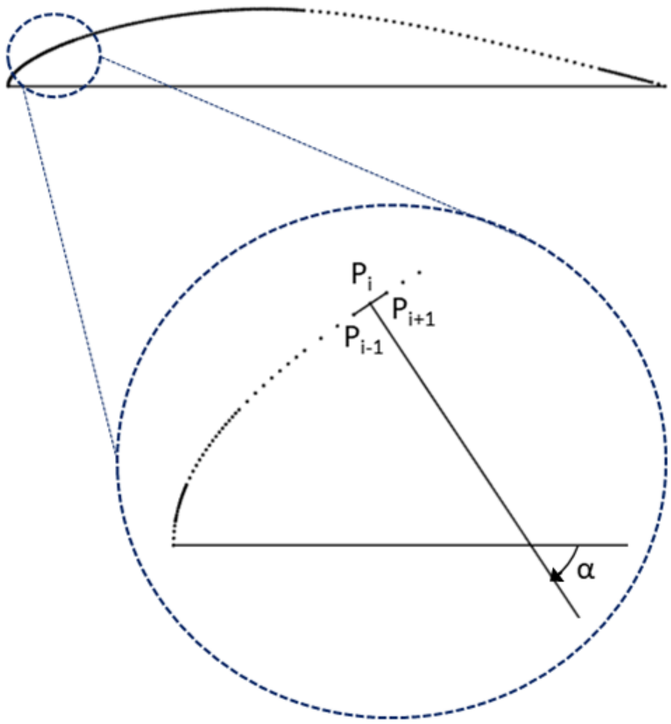


Figure 4.5: Schematic illustration of the procedure followed to obtain the normal direction to the considered airfoil at each point P_i on the upper side.

The normal direction to the leading edge is simply the direction of the x-axis. Each point of the nominal airfoil P_i of coordinates (x_i, y_i) becomes, in the eroded part of the airfoil, the point P_i' of coordinates (x_i', y_i') given by (see figure 4.6):

$$\begin{aligned} x_i' &= x_i + d\cos(\alpha) \\ y_i' &= y_i + d\sin(\alpha) \end{aligned} \tag{4.5}$$

It can be noticed that, at some point on the airfoil, the curvature changes. To obtain the coordinates of the damage in the case the angle α becomes positive, the expression for y_i' should be changed to:

$$y_i' = y_i - d\sin(\alpha), \tag{4.6}$$

but this never happened in the case considered.

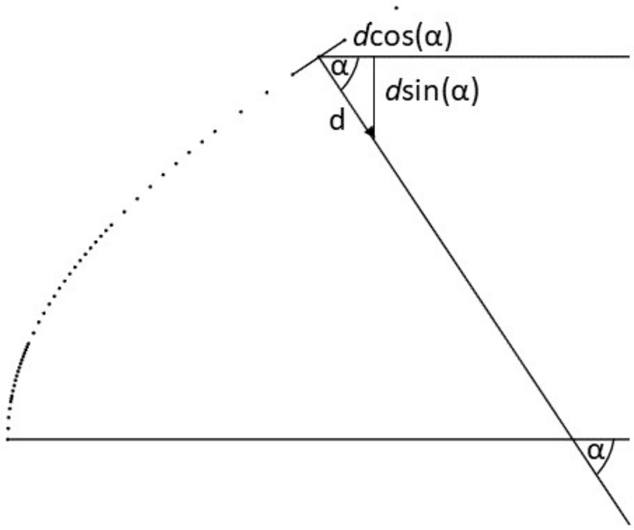


Figure 4.6: Illustation of the procedure to obtain the x and y coordinates of the delaminated airfoil.

Some difficulties may arise when obtaining the coordinates of the damaged airfoil. Two subsequent points could interchange, leading to a non-smooth profile, or two points could be too close to one another, leading to problems in the geometry creation. To avoid these situations, two controls on the newly obtained points have been implemented:

1. **Control 1**: Consider the leading edge of the airfoil, as depicted in figure 4.6. The unknown function connecting all these points is decreasing going towards the leading edge. If the points are enumerated so that the index is increasing in the direction of the leading edge, it is expected that $y_{i+1} < y_i$. The first implemented control eliminates the point y_{i+1} if it happens to be greater than y_i.

2. **Control 2** Another problem that commonly arises is that two points are too close to one another. If the distance between the x coordinates of two subsequent points happens to be lower that 5×10^{-5}, one of them is eliminated by the second implemented control.

The controls are applied until there are no more problematic points. The creation of the damage on the lower side of the airfoil is analogous and will not therefore be repeated here.

4.2.2 MATLAB code to generate the damaged geometries

The MATLAB code that generates the damaged geometries is based on *for* cycles. The most external cycle is on the N_{aero} airfoils composing the considered model turbine. Then, for each airfoil, there is a cycle on the type of damage: mild, moderate and severe. For each type of damage, the parameters s_u, s_l and d vary in a range of values, giving rise to another three *for* loops, one for each variable. The pseudo-code of the algorithm has been written in algorithm 1.

Algorithm 1 Coordinates of the damaged airfoils.

for airfoil = 1:N_{aero} **do**
  $N \leftarrow 000000$
  Load airfoil coordinates (x, y)
  for damage type in {mild, moderate, severe } **do**
    for $s_u = s_u^{start}$:s_u^{end} **do**
      for $s_l = s_l^{start}$:s_l^{end} **do**
        for $d = d^{start}$:d^{end} **do**
          $N \leftarrow N + 1$
          if $\lfloor(\frac{N}{100})\rfloor \leq \lfloor(\frac{N-1}{100})\rfloor$ **then**
            Create folder $\lfloor(\frac{N}{100})\rfloor$.
          else
            Continue
          end if
          Find x coordinate corresponding to a curvilinear length s_u: x_{s_u}.
          for $i = 2 : 200$ **do** ▷ The coordinates of the points on the upper side of the airfoils go from 1 to 200.
            if $x(i) > x_{s_u}$ **then**
              $x'(i) = x(i),\ y'(i) = y(i)$
            else if $x(i) <= x_{s_u}$ **then**
              $m = (y_{i+1} - y_{i-1})/(x_{i+1} - x_{i-1})$.
              $\alpha = \arctan\left(-\frac{1}{m}\right)$.
              $x'(i) = x(i) + d\cos(\alpha),\ y'(i) = y(i) + d\sin(\alpha)$
            end if
          end for
          Find x coordinate corresponding to a curvilinear s_l: x_{s_l}.
          for $i = 201 : length(x)$ **do** ▷ The coordinates of the points on the lower side of the airfoils go from 201 to the end of the vector.
            if $x(i) > x_{s_l}$ **then**
              $x'(i) = x(i),\ y'(i) = y(i)$
            else if $x(i) <= x_{s_l}$ **then**
              $m = (y_{i+1} - y_{i-1})/(x_{i+1} - x_{i-1})$.
              $alpha = \arctan\left(-\frac{1}{m}\right)$.
              $x'(i) = x(i) + d\cos(\alpha),\ y'(i) = y(i) + d\sin(\alpha)$
            end if
          end for
          Apply control 1
          Apply control 2
          Create folder N inside the main folder $\lfloor(\frac{N}{100})\rfloor$ and save the coordinates there .
        end for
      end for
    end for
  end for
end for

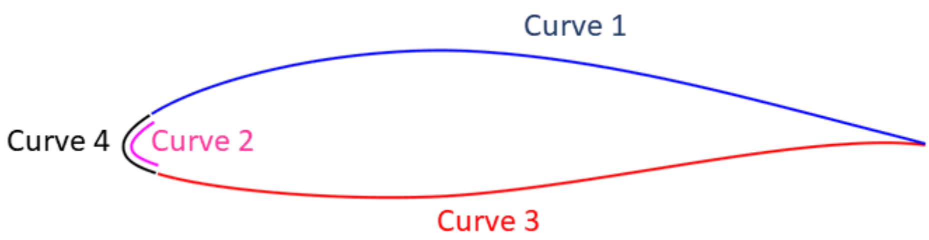


Figure 4.7: Curves generated for each one of the damages.

For each type of airfoil composing the considered turbine, N folders are created: one folder for each one of the damaged geometries. The coordinates of the delaminated airfoils are divided into four curves (see figure 4.7):

- **Curve 1**: The section of the upper side that has not been affected by the delamination damage;
- **Curve 2**: The curve corresponding to the delaminated part;
- **Curve 3**: The section on the lower side that has not been affected by the delamination damage;
- **Curve 4**: The curve corresponding to the original airfoil that has been eroded. This curve is needed in the mesh generation process.

The N folders are then grouped into $\lfloor(\frac{N}{100})\rfloor + 1$ main folders. For example, if $N = 296$ damages are created, the 296 folders are grouped into $\lfloor(\frac{296}{100})\rfloor + 1 = 3$ main folders: 000000,000001,000002. The main folder 000000 contains the folders from 000001 to 000099, the main folder 000001 contains the folders from 000100 to 000199, and finally 000002 contains the folders from 000200 to 000296. Finally, a table is created for each type of airfoil, containing the number from 000001 to N identifying the damage and the values of the corresponding three parameters s_u, s_l and d.

Chapter 5

Wind turbine computational aerodynamics

The airfoils force data of the ALPS database are obtained by means of NS CFD simulations. The power curve and the loads acting on the nominal and damaged turbines are then reconstructed with the help of a blade element momentum theory code, using the lift and drag curves provided by the CFD simulations and the ANN procedure. This chapter explains all the aspects regarding the computational aerodynamics of the turbine, from the NS CFD simulations to the Blade Element Momentum Theory (BEMT) code used in this work.

5.1 Mesh generation

5.1.1 Computational domain

The computational domain has been created with Design Modeler, a geometry creation software included in ANSYS Workbench 19.0. A representation of the domain used for the simulations is shown in figure 5.1. The pressure outlet and the velocity inlet are both situated at a distance of 45 chords with respect to the airfoil, in order to avoid instabilities in the solution process. In the case of the damaged airfoils, the computational domain has been divided into two parts: a first part corresponding to the domain around the nominal airfoil, and a second part corresponding to the damage. The process of creation of the geometries has been accelerated with a JavaScript that enables the necessary commands to run in a few seconds instead of creating all the features by hand. More details on the scripting procedure will be provided in chapter 7.

5.1.2 Mesh

The meshes for the simulations have been created using the software ANSYS Meshing, which is part of ANSYS Workbench. As already done for the generation of the computational domain, the meshing process has been automated using a script. Since ANSYS Meshing allows the instructions given on the Graphical User Interface to be registered, the scripting process is easier and faster

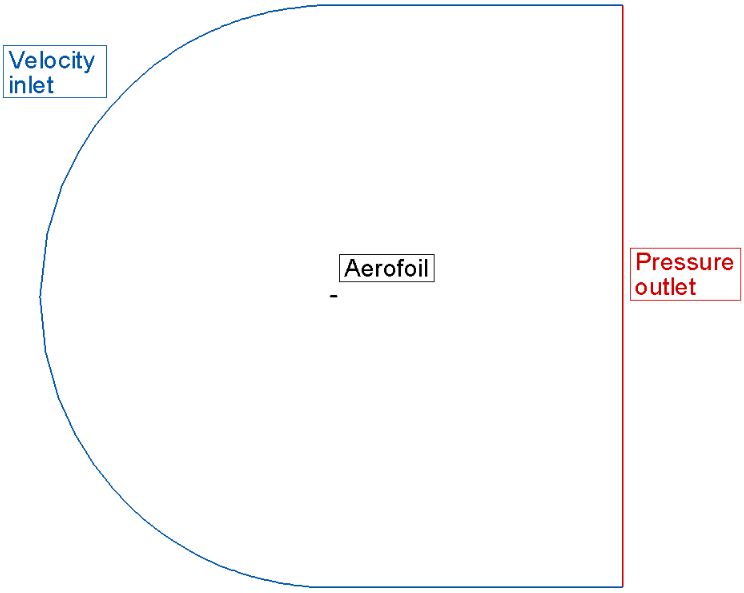


Figure 5.1: Computational domain for the CFD simulations: the farfield is divided into a velocity inlet and a pressure outlet.

than in the case of Design Modeler. The grids are structured and C-shaped both for the nominal and for the damaged airfoils.

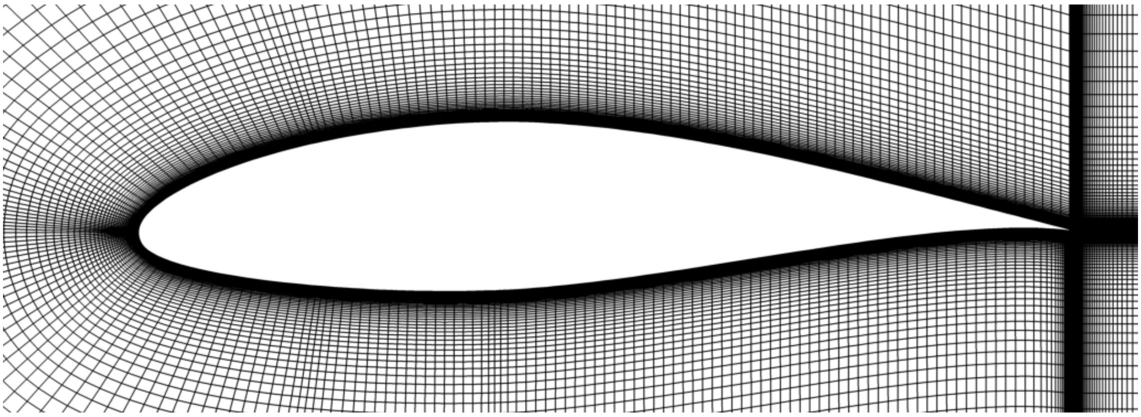
The nominal airfoils grids are characterized by 326 elements on the airfoil, 150 elements along the C-cut, and 150 elements in the normal direction, for a total of approximately 94 000 elements (see figure 5.2). A mesh refinement study has been carried out in order to establish the independence of the solution on the mesh characterized by these parameters, and the details can be found in chapter 8. When creating the mesh for the simulations, it is necessary to ensure that the distance of the first layer from the wall of the airfoil guarantees a y^+ value of order 1. The y^+ value is defined as:

$$y^+ = \frac{\rho U_\tau \Delta y}{\mu}, \tag{5.1}$$

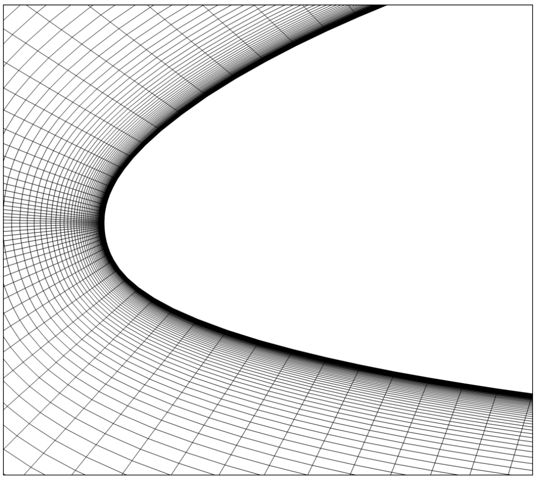
where:

- ρ is the density of air;
- μ is its dynamic viscosity;
- U_τ is the frictional velocity, defined as:

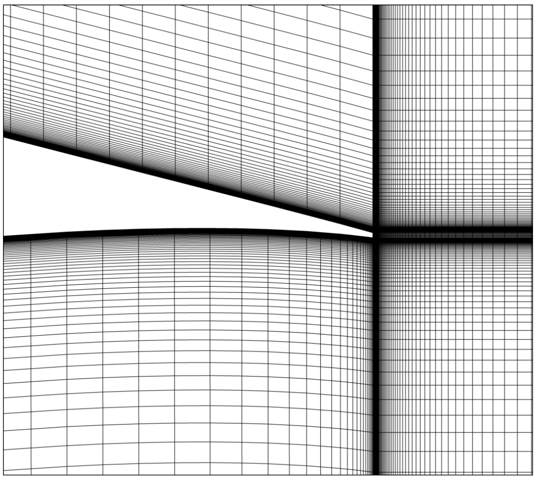
$$U_\tau = \sqrt{\frac{\tau_w}{\rho}}, \tag{5.2}$$



(a) Mesh around the nominal NACA 64-618 airfoil.



(b) Close-up view of the leading edge of the nominal NACA 64-618 airfoil.



(c) Close-up view of the trailing edge of the nominal NACA 64-618 airfoil.

Figure 5.2: Mesh around the nominal NACA 64-618 with close up views of the leading and trailing edges: special care was taken to assure a good orthogonal quality.

where τ_w is the wall shear stress, given by

$$\tau_w = \mu \left(\frac{\partial U}{\partial y} \right)_{y=0}, \tag{5.3}$$

U being the flow speed and y the direction normal to the airfoil. The wall shear stress can be obtained from the skin friction coefficient C_f as:

$$\tau_w = \frac{1}{2} C_f \rho U^2. \tag{5.4}$$

- Δy is the thickness of the considered layer.

The y^+ value defines the different regions of the turbulent boundary layer in turbulent flows. In particular:

- For $y^+ < 5$, the flow is almost completely dominated by viscous shear, and there exists a linear relation between the dimensionless velocity u^+ and the value of y^+:

$$u^+ = y^+$$

. This region is called linear sub layer.

- For $30 < y^+ < 500$ both the viscous and turbulent effects are important, and there exists a logarithmic relation between u^+ and y^+: the region is therefore called log-law layer.
- For $y^+ > 500$ the inertia forces are more important than the viscous forces, this is the outer region.
- For $5 < y^+ < 30$, neither the linear law nor the logarithmic law hold. However, for $5 < y^+ < 11$ the linear law is more accurate, and for $y^+ > 11$ the logarithmic law gives a better approximation.

The suitable value of y^+ to impose when generating the mesh depends on the modelling approach:

- If a wall function treatment is adopted, the value of y^+ should be greater than 11.
- If, instead, the turbulence equations are integrated all the way down to the airfoil wall, it is necessary to obtain a value of y^+ of order 1 in order to capture all the boundary layer, including the viscous sub layer.

The modelling approach chosen in this work is the second one, therefore is it important to obtain a value of $y^+ \approx 1$ when generating the grids. Recalling the definition of y^+ (5.1) the desired value can be obtained by imposing the first layer thickness to be:

$$\Delta y = \frac{y^+ \mu}{\rho U_\tau}. \tag{5.5}$$

U_τ can be easily obtained from its definition 5.2. The adopted minimum wall distance is different for the NACA and the DU airfoils because the Reynolds numbers of the simulations are not the same. The Reynolds number is a dimensionless quantity characterizing the flow, defined as:

$$Re = \frac{\rho u l}{\mu} \tag{5.6}$$

where:

- ρ is the density of air.
- u is the freestream velocity.
- l is the reference length.
- μ is the dynamic viscosity of air.

It is strictly connected to the y^+ value because the skin friction coefficient C_f depends on it. In the case of the NACA 64-618 airfoil, a wall distance of $4.4 \times 10^{-6} c$ guarantees a nondimensionalized minimum wall distance $y^+ \leq 1$ for all the angles of attack, while in the case of the DU airfoils the needed wall distance is $3.9 \times 10^{-6}c$. The plot of the wall y^+ for the NACA 64-618 airfoil can be seen in figure 5.3; the graphs obtained for the DU airfoils are analogous. The transitional SST turbulence model has been adopted. This means that the flow starts as laminar and then, as the Reynolds number increases and reaches the critical value, it becomes turbulent. The transition between laminar and turbulent flow can be seen in figure 5.3 at about $x/c = 0.45$, where an abrupt jump of the y^+ value can be observed. This is due to the fact that τ_w, and thus C_f, has higher value in the case of turbulent flows with respect to laminar ones. The mesh of the

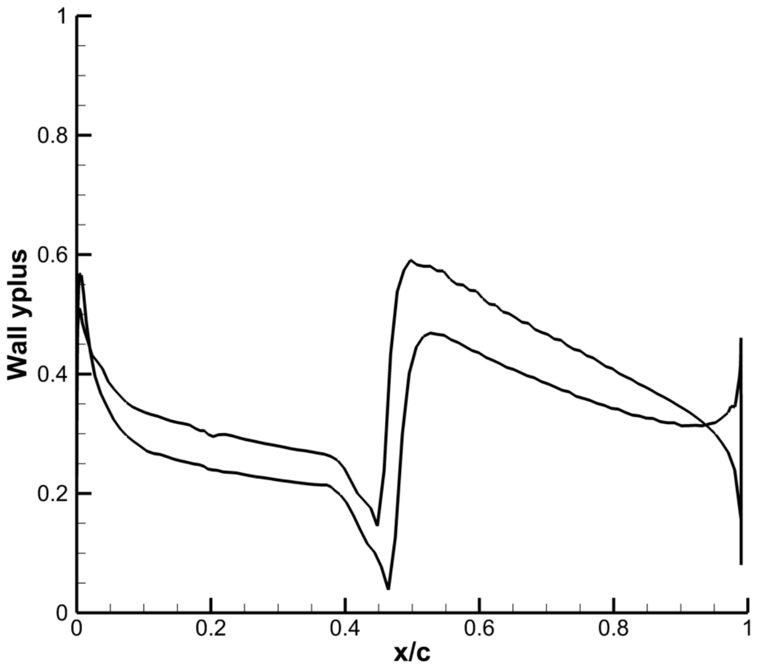
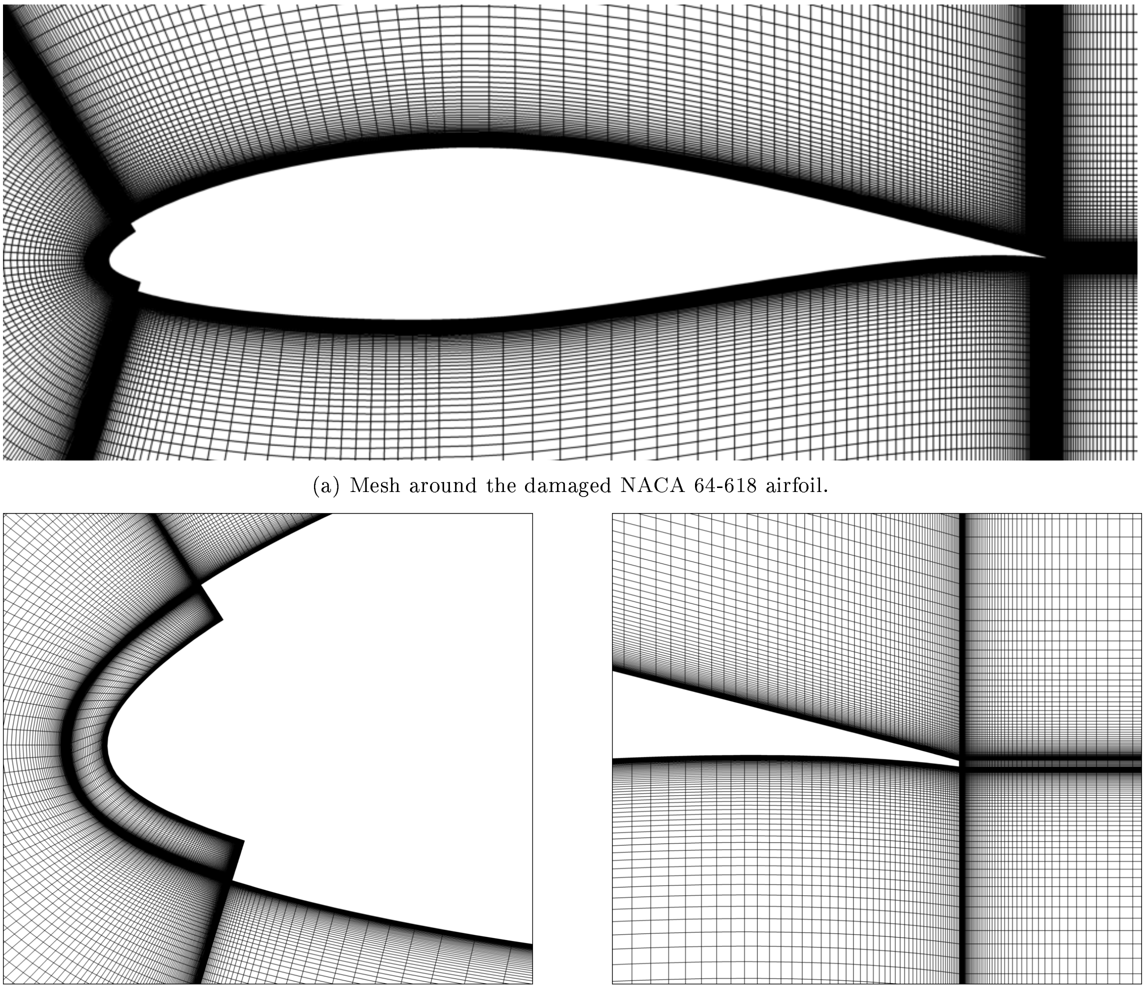


Figure 5.3: Wall y^+ around the nominal NACA 64-618 airfoil: the figure shows that its value is lower than 1 for all the values of x/c, thus the grid fulfils the constraints.

damaged airfoils is instead characterized by 612 elements around the airfoil, 150 elements along the C-cut and 150 elements in the normal direction. The total number of elements is 126 000. The distribution of the elements around the leading edge changes with the severity of the damage, in order to adapt the grid to the considered geometry. The decision about the number of points along each section of the airfoil is automated through a MATLAB script, and it will be described in chapter 7.

A grid for a particular damage of the NACA 64-618 airfoil, characterized by the parameters $s_u = 5.4\%c$, $s_l = 5.85\%c$ and $d = 1\%c$, is shown in figure 5.4. The Reynolds number is the same for all the damaged airfoils, so the minimum wall distance does not change as it did in the nominal case. The adopted minimum wall distance of $3.9 \times 10^{-6}c$ guarantees a nondimensionalized wall distance $y^+ \leq 1.5$ for all angles of attack. It is very difficult to obtain a value of y^+ lower or at least equal to one because of the complex geometry of the damaged airfoil. Using a wall distance smaller than the one considered, instability problems arose. The nondimensional y^+ obtained is



(a) Mesh around the damaged NACA 64-618 airfoil.

(b) Close-up view of the leading edge of the damaged NACA 64-618 airfoil.

(c) Close-up view of the trailing edge of the damaged NACA 64-618 airfoil.

Figure 5.4: Mesh around a damaged NACA 64-618 airfoil with close up views of the leading and trailing edge: the face corresponding to the damage has been meshed separately in order to ensure a good orthogonal quality along all the airfoil surface.

considered acceptable and a good compromise between accuracy and stability. The y^+ plot for the particular damaged airfoil considered herein is shown in figure 5.5 as an example. In this case, the fully turbulent $k-\omega$ SST model was used, and the flow becomes immediately turbulent. It can be noticed from picture 5.5 that the value of wall y^+ jumps up immediately, without the transition observed in the case of the nominal airfoil. The peaks observable between $x/c = 0$ and $x/c = 0.1$ are located in correspondence of the edges of the damage, where there is a discontinuity in the domain and the flow is therefore perturbed.

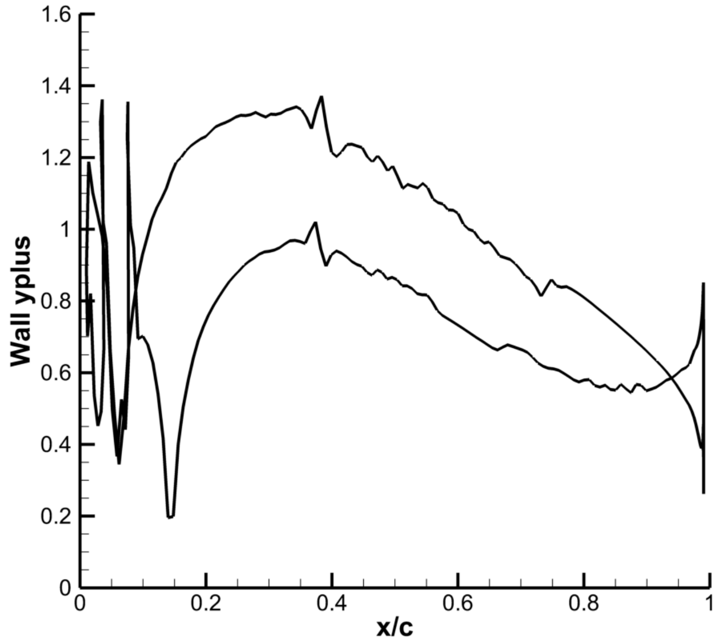


Figure 5.5: Nondimensional wall y^+ around the damaged NACA 64-618 airfoil.

5.2 CFD analysis set-up

5.2.1 Governing equations

The computational part of this work is based on the solution of the 2D Navier-Stokes equations, which express the continuity of mass and the conservation of momentum. For an incompressible flow in a 2D Cartesian system the equations read:

$$\begin{aligned} \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} &= 0 \\ \frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_j u_i) &= -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j} \qquad i = 1, 2 \end{aligned} \tag{5.7}$$

where:

- u_1 u_2 are the Cartesian components of velocity in the x and y directions respectively;
- p is the pressure;
- τ_{ij}, $i, j = 1, 2$ are the components of the molecular stress tensor. For a Newtonian fluid, they are given by:

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right] \tag{5.8}$$

  where μ is the molecular dynamic viscosity.

The equations can be written in compact form as:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial(\mathbf{E}_c - \mathbf{E}_d)}{\partial x} + \frac{\partial(\mathbf{F}_c - \mathbf{F}_d)}{\partial y} = 0 \tag{5.9}$$

where:

$$\mathbf{U} = \begin{bmatrix} 0 \\ u_1 \\ u_2 \end{bmatrix} \qquad \mathbf{E}_c = \begin{bmatrix} u_1 \\ u_1^2 + p \\ u_1 u_2 \end{bmatrix} \qquad \mathbf{F}_c = \begin{bmatrix} u_2 \\ u_1 u_2 \\ u_2^2 + p \end{bmatrix} \qquad \mathbf{E}_d = \begin{bmatrix} 0 \\ \tau_{11} \\ \tau_{12} \end{bmatrix} \qquad \mathbf{F}_d = \begin{bmatrix} 0 \\ \tau_{12} \\ \tau_{22} \end{bmatrix} \tag{5.10}$$

When dealing with turbulent flows, the Navier-Stokes equations have to be averaged over the turbulence time scales, leading to the appearance of a new term: the Reynolds stress tensor. The Reynolds-averaged Navier-Stokes equations can be written as:

$$\begin{aligned} \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} &= 0 \\ \frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_j u_i) &= -\frac{\partial p}{\partial x_i} + \frac{\partial \hat{\tau}_{ji}}{\partial x_j} \qquad i = 1, 2 \end{aligned} \tag{5.11}$$

where all the terms already described are the same as before, and $\hat{\tau}_{ij}$ is the sum of the molecular stress tensor and the Reynolds stress tensor:

$$\hat{\tau}_{ij} = \tau_{ij} + \tau_{ij}^R \tag{5.12}$$

The expression for τ_{ij}^R is:

$$\tau_{ij}^R = \mu_T \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right] - \frac{2}{3} k \delta_{ij} \tag{5.13}$$

where μ_T represents the eddy viscosity and k is the turbulent kinetic energy. Due to the addition of these new terms, now the number of unknowns has risen to 5: p, u_1, u_2, μ_T and k. The equations are, however, still three. This means that, to close the system, at least two more equations are needed, providing that these two new equations do not bring other unknowns with them. The closing equations are provided by the turbulence model, which describes the behaviour of the turbulent variables. A variety of turbulence model can be found in the literature, each one adapting better to a particular physical problem. In this work, two different turbulence models were used: Menter's Shear Stress Transport (SST) [23], and Langtry-Menter 4-equation transitional SST model, also knows as gamma-Retheta-SST [20].

- **Menter's Shear Stress Transport turbulence model** Menter's SST turbulence model consists of two equations: one for the turbulent kinetic energy k and the other for the specific dissipation rate ω, which is an additional unknown:

$$\begin{aligned} \frac{\partial k}{\partial t} + \frac{\partial}{\partial x_j}(u_j k) &= \frac{\tau_{ij}^R}{\rho} \frac{\partial u_i}{\partial x_j} - \beta^* \omega k + \frac{1}{\rho} \frac{\partial}{\partial x_j} \left[(\mu + \sigma^* \mu_T) \frac{\partial k}{\partial x_j} \right] \\ \frac{\partial \omega}{\partial t} + \frac{\partial}{\partial x_j}(u_j \omega) &= \frac{\gamma \omega}{k} \frac{\tau_{ij}^R}{\rho} \frac{\partial u_i}{\partial x_j} - \beta \omega^2 + \frac{1}{\rho} \frac{\partial}{\partial x_j} \left[(\mu + \sigma \mu_T) \frac{\partial \omega}{\partial x_j} \right] \end{aligned} \tag{5.14}$$

To close the system of equation, we need a relation between k, ω and μ_T:

$$\mu_T = \gamma^* \frac{\rho k}{\omega} \tag{5.15}$$

Finally, the values for the parameters are:

$$\beta = \frac{3}{40}, \quad \beta^* = \frac{9}{100}, \quad \gamma = \frac{5}{9}, \quad \gamma^* = 1, \quad \sigma = \frac{1}{2}, \quad \sigma^* = \frac{1}{2} \tag{5.16}$$

- **Langtry-Menter 4-equation Transitional SST Model** The Langtry-Menter 4-equation Transitional SST Model is made up of four equations: the two equations of Menter's SST model, and two more equations, one for the intermittency coefficient γ and the other one for the local transition asset momentum thickness Reynolds number $\hat{Re}_{\theta t}$. The four equations that make up the model read:

$$\begin{aligned}
\frac{\partial k}{\partial t} + \frac{\partial}{\partial x_j}(u_j k) &= \frac{\tau_{ij}^R}{\rho}\frac{\partial u_i}{\partial x_j} - \beta^* \omega k + \frac{1}{\rho}\frac{\partial}{\partial x_j}\left[(\mu + \sigma^* \mu_T)\frac{\partial k}{\partial x_j}\right] \\
\frac{\partial \omega}{\partial t} + \frac{\partial}{\partial x_j}(u_j \omega) &= \frac{\gamma\omega}{k}\frac{\tau_{ij}^R}{\rho}\frac{\partial u_i}{\partial x_j} - \beta\omega^2 + \frac{1}{\rho}\frac{\partial}{\partial x_j}\left[(\mu + \sigma \mu_T)\frac{\partial \omega}{\partial x_j}\right] \\
\frac{\partial \gamma}{\partial t} + \frac{\partial}{\partial x_j}(u_j \gamma) &= P_\gamma - E_\gamma + \frac{1}{\rho}\frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_T}{\sigma_\gamma}\right)\frac{\partial \gamma}{\partial x_j}\right] \\
\frac{\partial \hat{Re}_{\theta t}}{\partial t} + \frac{\partial}{\partial x_j}(u_j \hat{Re}_{\theta t}) &= P_{\theta t} + \frac{1}{\rho}\frac{\partial}{\partial x_j}\left[\sigma_{\theta t}(\mu + \mu_T))\frac{\partial \hat{Re}_{\theta t}}{\partial x_j}\right]
\end{aligned} \tag{5.17}$$

An exhaustive description of the model goes beyond the purpose of this work, so the definitions of the parameters σ_γ, P_γ, E_γ, $P_{\theta t}$, $\sigma_{\theta t}$ will not be provided here. The details about the derivation of the equations and all the quantities involved can be found in [20].

5.2.2 Numerical settings

In this work, the aerodynamic forces acting on the nominal and damaged airfoils are determined using incompressible 2D Navier-Stokes CFD simulations with the ANSYS FLUENT software. For each airfoil, 16 values of the angle of attack are considered, ranging from $-10°$ to $20°$ with a step of $2°$. The selected solver is the pressure based, with an absolute velocity formulation. All the simulations are steady. The chosen turbulence model varies depending on the case:

- For the nominal airfoils, the transitional SST model has been adopted. The lift and drag curves of these airfoils were obtained experimentally in a wind tunnel. These facilities are characterized by a low level of turbulence intensity, and the flow experienced by the airfoil is therefore transitional. The most coherent choice for the simulations is that to reproduce the conditions of the experiments, hence the decision to apply the transitional SST turbulent model in this case.

- For the damaged airfoils, the applied turbulence model was Menter's $k - \omega$ SST. The behaviour of the flow around the delaminated airfoils is harder to predict. Wind tunnel measurements were obtained by Sareen *et al.* [26] for different cases of damaged geometries, and these are the only experimental data available in the literature. Working on the validation of a damaged airfoil similar to that used for the experiments, both the fully turbulent $k-\omega$ SST and the transitional SST models were tested. The first model gave a good agreement for the lift coefficient, but it over-predicted the drag. Using the transitional model, instead, the overall agreement with the experimental data was better. The behaviour of the flow observed in the transitional simulations is as follows:

- the flow starts as laminar at the leading edge,
- corresponding to the edges of the damage, separation occurs for all angles of attack and the flow becomes turbulent;
- At this point, it is difficult to establish if the flow recovers as laminar and soon after becomes turbulent again, or if it becomes turbulent once it reaches the damage without any recovery. Aerodynamic studies and further investigations of the consequences of the presence of the damage will be object of future work. It is not possible to know exactly the behaviour of the flow around an operating damaged blade, so it is difficult to establish which turbulence model would describe in the best way possible the real aerodynamics of the case under analysis. The airfoil surface, in the simulations, was considered smooth. This is not likely to happen for real blades: after years of operation, when the delamination occurs, they will almost certainly be characterized by increased roughness. In this case, the flow will become turbulent as soon as it meets the edges of the delamination damage. Therefore, it was decided in this work to opt for the fully turbulent $k-\omega$ SST, considered more suitable to represent the real aerodynamics of the damaged airfoils. Another possibility would be that to force the flow to be laminar before the edges of the damage, and become turbulent after that, rather than impose a turbulent flow everywhere. However, this would not make a big difference in terms of results, and it would increase the computational time by a factor 1.4.

Recalling the definition of the Reynolds number

$$Re = \frac{\rho u l}{\mu} \tag{5.18}$$

where:

- ρ is the density of air, $\rho = 1.225\ kg/m^3$.
- u is the freestream velocity, which was set to 10 $m \cdot s$ for all the simulations.
- l is the reference length. In our case, it is represented by the chord length, which is 1 m for all the airfoils used for the simulations.
- μ is the dynamic viscosity of air, whose value is $1.81 \times 10^{-5}\ kg/(m \cdot s)$.

The Reynolds numbers used for the validation of the nominal airfoils matches those of the experiments, while for all the damaged airfoils the adopted Reynolds number is 7 M. The chosen value is similar to the span-wise averaged Reynolds number of a utility scale wind turbine at rated wind speed. To obtain the desired Reynolds, having set the freestream velocity, the reference length and the density, it was chosen to impose a modified dynamic viscosity, given by:

$$\mu = \frac{\rho u l}{Re}. \tag{5.19}$$

A velocity inlet far field boundary condition (BC) is applied on the front, lower and upper parts of the far field boundary, whereas a pressure outlet BC is applied on the rear part. The far field BC data for the SST model consists of a freestream turbulence intensity of 5 % and a turbulent-to-laminar viscosity ratio of 1. At the airfoil surface, a no-slip condition is enforced. No wall functions are used and the equations of the turbulence model are integrated all the way down to the wall boundary. The coupled scheme is used for the pressure-velocity coupling, and the spatial discretization is of second order upwind for all the equations. The default value of 200 for the flow Courant number is changed into 100, while for the explicit relaxation factors and the under-relaxation factors the default values of FLUENT have been retained. In particular, for the explicit relaxation factors of momentum and pressure the default value is 0.5. Regarding the under-relaxation factors: a value of 1 is used for the density and for the body forces, 0.8 for the quantities related to turbulence model: the turbulent kinetic energy, the specific dissipation rate, and also the intermittency and momentum thickness Reynolds if the transition model is used. Finally, the under-relaxation factor for the turbulent viscosity is set to 1 by default. Convergence is considered to be achieved when a reduction of 6 orders of magnitude is observed in the root mean square of the residuals of both the flow and turbulence model equations, or when the maximum number of iterations (2000) is reached. In the vast majority of cases, all 2000 iterations were performed by the solver, achieving a residuals reduction of 4 to 5 orders of magnitude. ANSYS FLUENT allows a journal to be recorded containing all the instructions given to the Graphical User Interface (GUI), so the CFD set-up can be automated easily. More details will be provided in chapter 7.

5.3 Blade element momentum theory analysis

The aerodynamic power and loads of turbines with nominal and damaged blade geometry for given values of wind speed, rotor angular speed and blade pitch are determined with AeroDyn [15], the NREL time-domain wind turbine aerodynamics module. Aerodynamic calculations within AeroDyn are based on the principles of actuator lines, whereby the 3D flow field around a slender body is approximated with a local 2D flow model past cross sections, and the distributed pressure and viscous stresses are approximated by lift forces, drag forces, and pitching moments lumped at the node associated with the considered 2D cross section. Using this approach, the code can compute aerodynamic loads on both rotor blades and turbine tower. Analysis nodes are distributed along the length of each blade and the tower, the 2D forces and moment at each node are computed as distributed loads per unit length, and the total 3D aerodynamic loads are found by integrating the 2D distributed loads along the length of the considered blade or the tower.

AeroDyn consists of four submodels: *i*) rotor wake induction, *ii*) blade airfoil aerodynamics, *iii*) tower influence on blade aerodynamics, and *iv*) tower drag, and all four modules are used in the AeroDyn analyses reported in this study. For operating (*i.e.* not parked or idle) wind turbine rotors, AeroDyn calculates the influence of the rotor wake on blade aerodynamics via axial and circumferential induction factors based on the quasi-steady BEMT, which requires an iterative nonlinear solve coupling modules *i*) and *ii*). By quasi-steady, it is meant that the induction reacts instantaneously to loading changes. The calculation of the induction factors and resulting inflow velocities and angles relative to the moving blade are based on the local flow field around each analysis node of the considered blade. The effects of local inflow skew, wind shear, turbulence, tower flow disturbances can also be included in the analysis. The Glauert's empirical correction with Buhl's modification replaces the linear momentum balance at high axial induction factors. In AeroDyn, 3D flow features are either neglected or captured through corrections inherent in the model (*e.g.* Prandtl tip and hub loss corrections or skewed wake corrections) or the input data (*e.g.* rotational augmentation corrections applied to the airfoil force coefficients). The AeroDyn analyses reported below use steady uniform incoming wind aligned with the rotor axis, *i.e.* no wind shear and zero yaw error; the airfoil aerodynamics is assumed steady (*e.g.* no dynamic stall model is used), and the static pressure perturbation of the downwind tower on the blade aerodynamics is included.

Chapter 6

Power control

This chapter focuses on the power control of a speed and pitch regulated wind turbine. In the first part, the typical control of an operating wind turbine is described, based on the work of Jonkman *et al.* [18]. Analysing its features, it has been observed by the author that this control could overestimate the power losses in case the turbine began to show signs of damage. It has thus been decided to implement a slightly different control for a damaged turbine, whose description concludes the chapter.

6.1 Introduction

The behaviour of operating wind turbines for the complete range of operating wind speeds is governed by different types of control, which are a fundamental part of the turbine itself. Some examples are provided by:

- **The yaw control**, whose aim is that to align the turbine to the wind speed in order to maximise the power and to reduce the non-symmetrical loads;
- **The electrical braking**, which converts some of the generator energy into heat and helps to reduce the speed of the turbine in high wind conditions.
- **The mechanical braking**, which stops the turbine in case of emergency;
- **The power control**, whose aim is to capture the maximum possible energy before reaching rated power, and to keep the power to the rated value once the value has been reached.

Only the power control has been analysed within this work.

6.2 Typical speed and pitch control of utility-scale wind turbines

The typical control strategy that will be described in this section has not been implemented by the author: it was already part of the BEMT code adopted, *i.e.* NREL FAST [17] and AeroDyn [15].

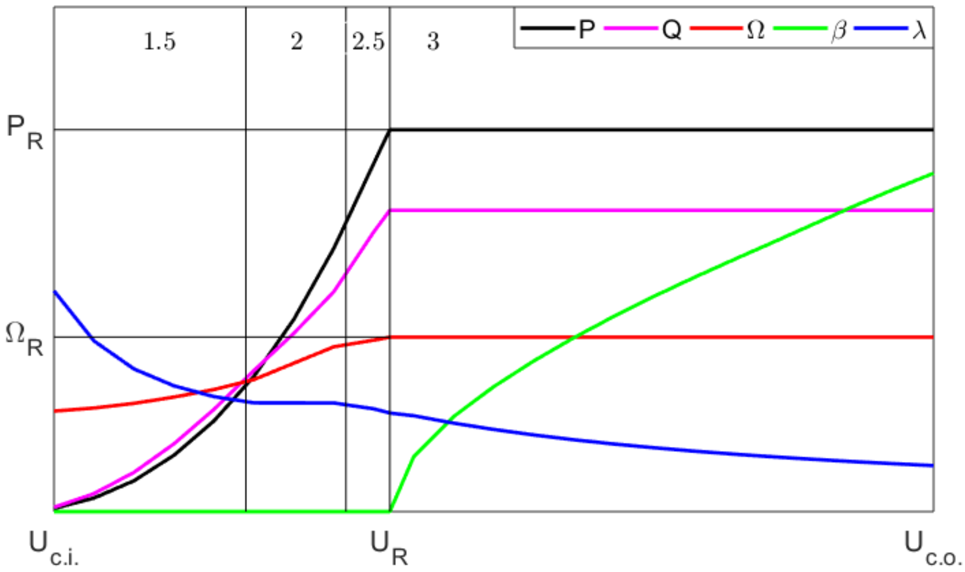


Figure 6.1: Nominal wind turbine power control.

Detailed information about its effective implementation in the software can be found in the report by Jonkman *et al.*[18], which is the main source for this chapter. The control described herein has been specifically designed for the NREL 5 MW model turbine, but the variable generator speed variable blade-pitch-to-feather power control strategy is widely adopted for a large number of operating wind turbines. The power control is divided into five regions:

- Region 1, which is a start-up region. Here all the energy harvested from the wind is used for the start-up of the turbine;
- Region 1.5, a transition region where a linear relation exists between the rotor speed Ω and the rotor torque Q;
- Region 2, where the rotor speed is modified in order to harvest the maximum possible power.
- Region 2.5, a transition region analogous to region 1.5;
- Region 3, where the turbine has reached rated power and the pitch is varied in order to control the rotational speed of the turbine, with the final aim of maintaining the power to the rated value.

The only input of the control is the generator speed Ω. The profiles of turbine power P, low-speed shaft torque Q, angular speed Ω, tip-speed ratio (TSR) λ and blade pitch β against the wind speed U for a turbine with nominal blade shapes are reported in Fig. 6.1, in which the subscripts *c.i.*, R and *c.o.* denote respectively cut-in, rated and cut-out speeds. The turbine starts producing power once the wind speed reaches $U_{c.i.}$. In region 2, starting shortly after wind speed $U_{c.i.}$, the TSR λ is kept constant and equal to the value yielding the maximum power coefficient, thus maximizing the harvested power. Therefore, Ω increases linearly with U, and the aerodynamic torque increases quadratically with both Ω and U in region 2. The maximum power coefficient was found by Jonkman *et al.* by running simulations for different rotor speeds and blade pitch at a fixed wind speed of 8 m/s. The optimal pitch was found to be $0°$, while the value of rotor speed associated to the maximum power was used to obtain the optimum tip-speed ratio. The power coefficient strictly depends on the lift and drag curves of the airfoils composing the turbine. Since for a damaged blade these curves will necessarily be different from those of a nominal one, also the maximum extractable power will be different, and the same holds true for the rotor speed at which this maximum will be reached. This means that the optimal tip-speed ratio of a nominal and of a damaged turbine is not necessarily the same. This is the first reason that suggested the implementation of an adaptive control for the damaged turbine be analysed. Region 2 is preceded by the transition region 1.5, in which the aerodynamic torque varies linearly with Ω, and Ω increases more slowly with U with respect to what occurring in region 2. This region extends from the cut-in generator speed until 30% above this value. Given the relation between the generator speed and the rotor speed, this is analogous to consider the corresponding rotor speeds. Similarly, before the power reaches its rated value, the rotor speed starts increasing more slowly with the wind speed (region 2.5) than in region 2. This is because, also in region 2.5, the control algorithm enforces a linear relationship between Q and Ω. The region extends from $0.9\Omega_R$ to $0.99\Omega_R$. Until the rated wind U_R, the right boundary of region 2.5, is reached, the blade pitch β remains constant. As the wind speed increases above U_R, however, Ω remains constant, and the blades are actively pitched to feather by the amount required to reduce the lift and its projection on the rotor plane and maintain constant torque and power until the wind speed reaches $U_{c.o.}$, where the turbine is shut down.

The rotor torque as a function of the rotor speed is displayed in figure 6.2. It is possible to notice the linear relation between the two quantities in regions 1.5 and 2.5, and the quadratic relation in region 2. After region 2.5, when the rated torque is reached, the rotor speed is kept constant. This is why region 3 is not included in the figure.

6.3 Adaptive control for a damaged turbine

As highlighted in the results chapter, the control of the nominal turbine is unlikely to yield an optimal performance of the damaged turbine, due to altered aerodynamic properties of the eroded blades. For this reason, a modified power control strategy for a turbine with eroded blade edges is proposed below, based on some assumptions:

- The cut-in and the rated rotor speeds, $\Omega_{c.i.}$ and Ω_R, of the damaged turbine are considered to be the same as those of the nominal turbine.
- The rated power P_R of the damaged turbine is the same as that of the nominal turbine.
- The cut-out wind speed $U_{c.o.}$ of the damaged turbine is the same as the one of the nominal turbine.

These assumptions have been made only to explain the complete set-up of the control strategy. In reality, if the loads on the tower are too high due to the presence of the damaged blades, the turbine would be shut down well before the cut-out wind speed.

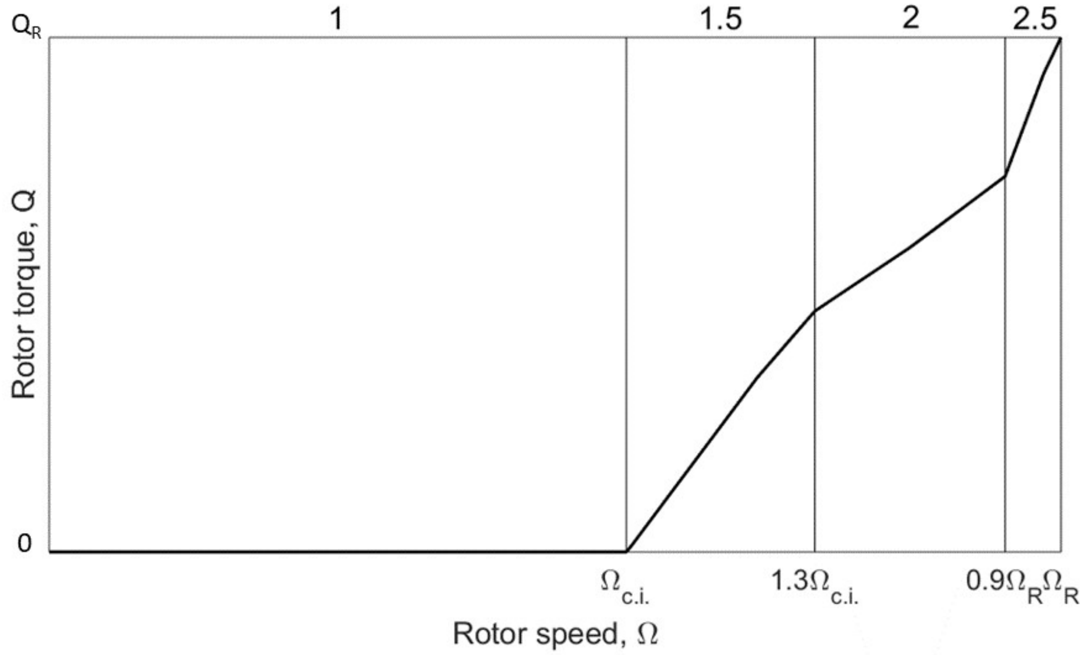


Figure 6.2: Rotor torque as a function of the rotor speed currently adopted for speed and pitch regulated wind turbines. It is possible to notice the linear relation between the two quantities in regions 1.5 and 2.5, and the quadratic relation in region 2.

Region 1.5 is characterized, as in the case of the nominal turbine, by a linear relation between the rotor speed Ω and the rotor torque Q. The aim of the damaged turbine control in region 2 and 3.1 is to maximise the power extraction by changing, respectively, the rotor speed and the blade pitch. Finally, in region 3.2 the blade pitch is modified in order to keep the power constant to the rated value. The values of rotor speed Ω and blade pitch β that define the behaviour of the damaged turbine according to the control have been found by using a MATLAB program together with AeroDyn, the BEMT code used in this work. The final input needed to compute the power curve of the turbine is a table containing the selected wind speeds, and the associated rotor speed and blade pitch. For each control region, and for each point in the control region, it is therefore necessary to obtain this triplet. The rotor power P, rotor torque Q, rotor speed Ω, blade pitch β and tip speed ratio λ of the damaged turbine as a function of the wind speed U in the various control regions are depicted in figure 6.3.

Region 1.5 is, for the damaged turbine, conceptually the same as it is for the nominal turbine. This means that it is characterized by a linear relation between the torque Q and the rotor speed Ω, but in general the constant characterizing this relation will be different. Region 1.5 starts, according to our assumptions, in correspondence of the cut-in rotor speed, $\Omega_{c.i.}$., and it ends when the rotor speed reaches 30% above the cut-in value. This is in agreement with the control of the nominal turbine. The values needed to the BEMT code to obtain the power and loads of the considered turbine are the wind speed, the rotor speed, and the blade pitch. In this region, the blade pitch does not change, so its value is always $\beta = 0°$. The rotor speed goes from $\Omega_{c.i.}$ at

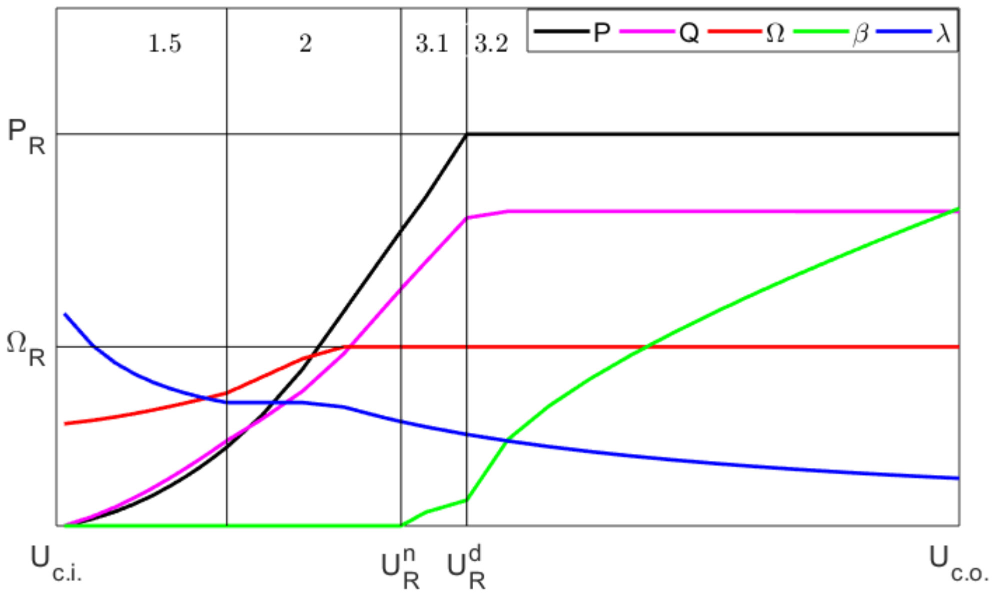


Figure 6.3: Damaged wind turbine power control.

the beginning of region 1.5 to $1.3\Omega_{c.i.}$ at the end of the same region. The wind speeds associated to each rotor speed are instead unknown and need to be determined. We know that there is a linear relation between the rotor torque and the rotor speed. This means that, once the constant of proportionality is determined, it will be possible to associate to each rotor speed between $\Omega_{c.i.}$ and $1.3\Omega_{c.i.}$ the torque produced by the wind turbine. Once known the expected torque $\bar{Q}$ for each rotor speed, and consequently the power $\bar{P}$, the associated wind speed can be found by solving the equation $P(U) - \bar{P} = 0$ with a numerical method. First of all, the value of the proportionality constant needs to be found. At the cut-in rotor speed, the rotor torque is 0. The point corresponding to $1.3\Omega_R$ is in common between regions 1.5 and 2, thus, at this rotor speed, the tip speed ratio, and consequently the wind speed, must be the one which maximises the power capture. The wind speed corresponding to the maximum power at the rotor speed $1.3\Omega_{c.i.}$ is found by means of the golden section search method, which allows to find the maximum and minimum of uni-modal functions in an interval. The extremes on the considered interval are the cut-in wind speed of the nominal turbine, $U_{c.i.}$, and the rated wind speed of the nominal turbine, U_R^n. We are sure to find the maximum in this interval because, in general, to harvest the maximum possible power, at each wind speed before rated, the rotational speed of the damaged turbine will need to be higher than that of the nominal turbine. This means that the rated rotational speed Ω_R will be reached by the damaged turbine at a lower wind speed than that characterizing the nominal turbine. Since $1.3\Omega_{c.i.} \leq \Omega_R$, the chosen interval guarantees the presence of a maximum. Figure 6.4 shows the rotor power as a function of the wind speed between $U_{c.i.}$ and U_R at the rotor speed of $1.3\Omega_{c.i.}$ for a particular case of a damaged turbine, but a similar behaviour should be observed in all cases.

The golden section search algorithm narrows the interval enclosing the maximum until the

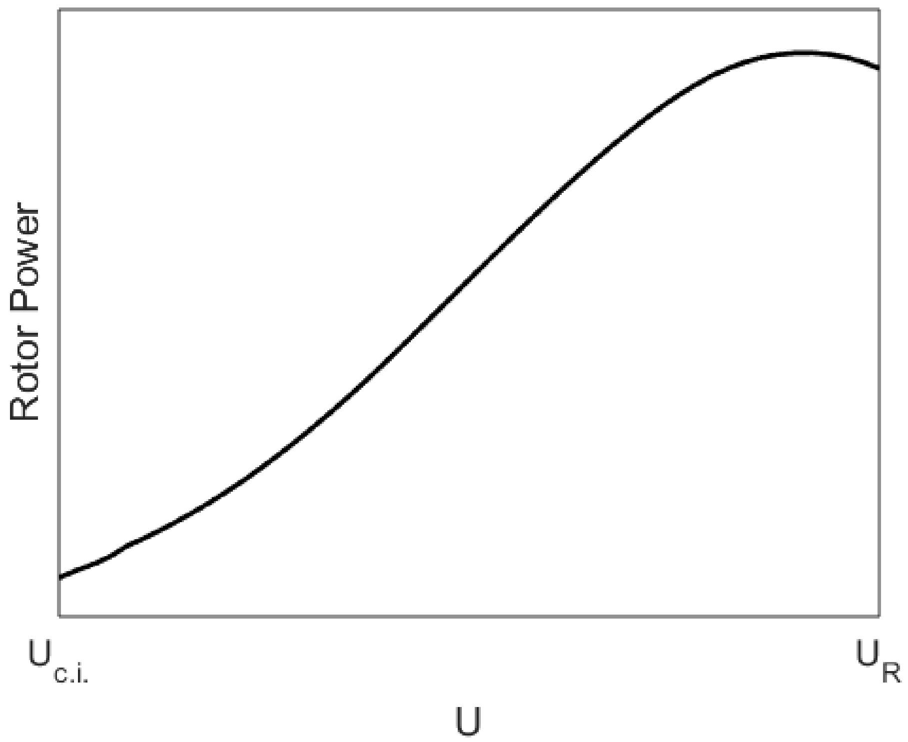


Figure 6.4: Rotor power as a function of the wind speed for a particular case of damaged turbine at the fixed rotor speed $1.3\Omega_{c.i.}$.

desired tolerance tol or the maximum number of iterations $maxIt$ is reached. The pseudo-code of the golden section search algorithm implemented is shown in algorithm 2.

Once the wind speed $U_{1.3\Omega_{c.i.}}$ is obtained, the power produced by the turbine at the final point belonging to region 1.5 is obtained running AeroDyn with the input $[U_{1.3\Omega_{c.i.}}, 1.3\Omega_{c.i.}, 0]$:

$$P^{\max}_{1.3\Omega_{c.i.}} = P(U = U_{1.3\Omega_{c.i.}}, \Omega = 1.3\Omega_{c.i.}, \beta = 0°). \tag{6.1}$$

Now, the torque can be obtained from the power as:

$$Q_{1.3\Omega_{c.i.}} = \frac{P_{1.3\Omega_{c.i.}}}{1.3\Omega_{c.i.}} \tag{6.2}$$

and the slope k of the straight line defining the torque as a function of the rotor speed is:

$$k = \frac{Q_{1.3\Omega_{c.i.}}}{1.3\Omega_{c.i.} - \Omega_{c.i.}} \tag{6.3}$$

Once obtained this relation, it is possible to determine the torque associated to each rotor speed in region 1.5, and consequently the power. The range of rotor speeds from $\Omega_{c.i.}$ to $1.3\Omega_{c.i.}$ has been discretized into N equidistant points. Then, the torque associated to each one of them

Algorithm 2 Golden section search.

$\phi = \frac{(\sqrt{5}+1)}{2}$
Read a, b, $maxIt$, tol
Obtain $fa = P(U = a,\ \Omega = 1.3\Omega_{c.i.},\ \beta = 0°)$ and $fb = P(U = b,\ \Omega = 1.3\Omega_{c.i.},\ \beta = 0°)$ with the BEMT code
for $(k = 1 : maxIt)$ **do**
  $k \leftarrow k + 1$
  $c = a + (b - a)/\phi$
  $fc = P(U = c,\ \Omega = 1.3\Omega_{c.i.},\ \beta = 0°)$ ▷ The power associated to the wind speed c, with no pitch and rotor speed $1.3\Omega_{c.i.}$ is obtained with the BEMT program.
  $d = b - (b - a)/\phi$
  $fd = P(U = d,\ \Omega = 1.3\Omega_{c.i.},\ \beta = 0°)$ ▷ The power associated to the wind speed d, with no pitch and rotor speed $1.3\Omega_{c.i.}$ is obtained with the BEMT program.
  if $(fc \geq fd)$ **then**
    $a = d;\quad d = c,\quad fd = fc$
    $c = a + \phi(b - a),\quad fc = P(U = c,\ \Omega = 1.3\Omega_{c.i.},\ \beta = 0°)$
  else
    $b = c;\quad c = d,\quad fc = fd$
    $d = b - (b - a)/\phi,\quad fd = P(U = d,\ \Omega = 1.3\Omega_{c.i.},\ \beta = 0°)$
  end if
  if $(|b - c| \leq tol)$ **then**
    STOP
  end if
end for
$U_{1.3\Omega_{c.i.}} = \frac{c+d}{2}$ ▷ The wind speed that gives maximum power at a fixed rotor speed $\Omega = 1.3\Omega_{c.i.}$ and no blade pitch is obtained as the medium point of the interval of length tol enclosing the maximum of the power

is given by:

$$Q_i = k(\Omega_i - \Omega_{c.i.}), \qquad i = 1, 2, \ldots, N \tag{6.4}$$

and the corresponding rotor power is:

$$P_i = \Omega_i Q_i, \qquad i = 1, 2, \ldots, N \tag{6.5}$$

Finally, it is necessary to determine the wind speed corresponding to each rotor speed, solving the equations:

$$f(U) = P(U) - P_i = 0, \qquad i = 1, 2, \ldots, N. \tag{6.6}$$

In this case, the power is considered as a function of the wind speed only, because the rotor speed and the blade pitch are assigned. The roots can be found by means of the secant method, a quasi-Newton method derived from Newton's method by replacing the analytical expression of the derivative with a numerical approximation. The Newton's iteration would read:

$$U^{n+1} = U^n - \frac{f(U^n)}{f'(U^n)} \tag{6.7}$$

where $f'(U)$ is the analytical expression of the derivative of f. The dependence of the power on the wind speed is complex, so the derivative $f'(U)$ cannot be determined analytically, and needs instead to be approximated numerically:

$$f'(U) = P'(U) \approx \frac{P(U^n) - P(U^{n-1})}{U^n - U^{n-1}}, \tag{6.8}$$

This represents one iteration of the secant method, characterized by a convergence rate of $\phi = \frac{\sqrt{5}+1}{2}$. The implementation of the method is shown in algorithm 3.

Algorithm 3 Secant method.

for $i = 1 : N$ **do**
  Read a, b, $maxIt$, tol ▷ a and b are the first two guesses needed from the program to start the iterations
  Obtain $Pa = P(U = a,\ \Omega = \Omega_i,\ \beta = 0°)$ and $Pb = P(U = b,\ \Omega = \Omega_i,\ \beta = 0°)$ with the BEMT code
  for $(k = 2 : maxIt)$ **do**
    $k \leftarrow k + 1$ ▷ Update iterations count
    $dP = \dfrac{P^n - P^{n-1}}{U^n - U^{n-1}}$
    $U^{n+1} = U^n - \dfrac{P^n - P_i}{dP}$
    $P^{n+1} = P(U^{n+1}, \Omega_i, 0°)$
    if $|P^{n+1} - P_i| \leq tol$ **then**
      STOP
    end if
  end for
  $U_i = U^{n+1}$
end for

At this point, all the quantities needed have been obtained in region 1.5, including the cut-in wind speed, which will in general be higher than that of the nominal turbine.

Region 2 for the damaged turbine extends from the wind speed corresponding to a rotor speed $\Omega = 1.3\Omega_{c.i.}$ to the rated wind speed of the nominal turbine, U_R^n. In this region, the goal is to maximise the power extraction. Since the blade pitch is still zero, for each chosen value of wind speed U in the interval, only the rotor speed has to be determined. This is done by using the golden section search method already described in the previous section, substituting the wind speed with the rotor speed in the formulas. In general, the rated rotor speed will be reached by the damaged turbine well before the rated wind speed of the nominal turbine. From this moment until the end of region 2, the rotor speed is kept constant and equal to its maximum value.

Region 3 starts when the wind speed reaches the rated value of the nominal turbine U_R^n, and it is divided into two subregions: 3.1 and 3.2. The first one goes from the rated wind speed of the nominal turbine U_R^n to the rated wind speed of the damaged turbine, U_R^d. Here, the rotor speed is kept constant to the rated value, and the pitch is changed in order to maximise the power capture. The optimal value of the pitch is determined by means of the golden section search algorithm described in algorithm 2. The only difference is that, in this case, the values of wind speed and rotor speed are known, and the pitch is the unknown quantity. Intuitively, it might be thought that the maximum power at a fixed wind and rotor speed is obtained with zero pitch, since the angle of attack would be higher and more lift would be generated. This is not always the case: in fact, it can happen that the increase in the lift is accompanied by a non negligible increase of the drag, lowering the harvested power. An example will be shown in chapter 9. Once the turbine has reached the rated power, the pitch has to be regulated in order to maintain the power constant to its maximum value. Thus, from the rated wind speed of the damaged turbine U_R^d, to the cut-out wind speed $U_{c.o.}$, the secant method is used to find the value of the pitch that, at each wind speed, guarantees to maintain the rated power. The algorithm is the same described in algorithm 3, with the difference that the unknown is no longer the wind speed, but the blade pitch.

The comparison between the nominal and damaged turbine controls is illustrated in table 6.1.

	Nominal turbine	**Damaged turbine**
Region 1: Until $\Omega_{c.i.}$	The turbine is not producing power and the wind speed is used to accelerate the rotor.	Same as for the nominal turbine
Region 1.5: from $\Omega_{c.i.}$ to $1.3\Omega_{c.i.}$	Linear relation between the rotor torque Q and the rotor speed Ω: $$Q = k_{nom}\Omega$$.	Linear relation between the rotor torque Q and the rotor speed Ω: $$Q = k_{dam}\Omega$$
Region 2	from $1.3\Omega_{c.i.}$ to $0.9\Omega_R$: Constant and optimum tip speed ratio for the power capture	from $U_{1.3\Omega_{c.i.}}$ to U_R^n: Maximum possible power capture.
Region 2.5: from $0.9\Omega_R$ to Ω_R	Linear relation between the rotor torque and the rotor speed	Not defined
Region 3: from Ω_R to cut-out	Change the blade pitch in order to maintain the rated power	Until U_R^d: maximise the power capture by changing the pitch. After U_R^d: Change the blade pitch in order to maintain the rated power.

Table 6.1: Comparison between the control of the nominal turbine and that of the damaged one.

Chapter 7

System automation

This chapter deals with the automation of the process of generation of the database. After a brief overview of the problem, the automation of the single components via JavaScript and scheme codes is described. Then, two possible solutions for the complete automation of the system are presented, with the respective pros and cons. The selected solution and the reasons for the choice are provided to the reader. Finally, the last component of the automation process, the machine learning approach, is described in detail.

7.1 Overview

The automation process involves mainly the generation of the database of damaged airfoils, as well as its extension to all possible damages by means of a machine learning approach. In order to make ALPS a reliable system, it is necessary to run thousands of NS CFD simulations, each one requiring the generation of the computational domain and structured mesh, and the setting of the CFD set-up. Despite the numerical settings being always the same for all the damaged geometries, the computational domain and the mesh change with the damage. It is therefore necessary to find a way to automatise the process in order to obtain the needed quantities faster. The software used is ANSYS Workbench, a simulation platform that allows to build the computational domain, to generate the mesh and to run the CFD simulations all in the same worksheet. A sample Workbench file similar to those generated within this project is shown in figure 7.1. The first box, labelled "Geometry", refers to the creation of the computational domain. By double clicking on it, the graphical user interface of the software Design Modeler opens. The geometry and the domain can be directly created by the user, or by running a script. The first option takes about half an hour, while running a script requires no more than a minute. The geometry box is connected to the mesh box, which opens ANSYS Meshing. Also in the case of the mesh creation, the instructions can be given via a script. Both ANSYS Design Modeler and ANSYS Meshing work with JavaScript files. Finally, connected to the mesh boxes there are the FLUENT boxes, one for each angle of attack, containing the set-up for the simulations. In this particular example, only two FLUENT boxes have been displayed, however, the Workbench projects created for this work contain 16 FLUENT boxes, one for each angle of attack ranging between $-10°$ and $+20°$

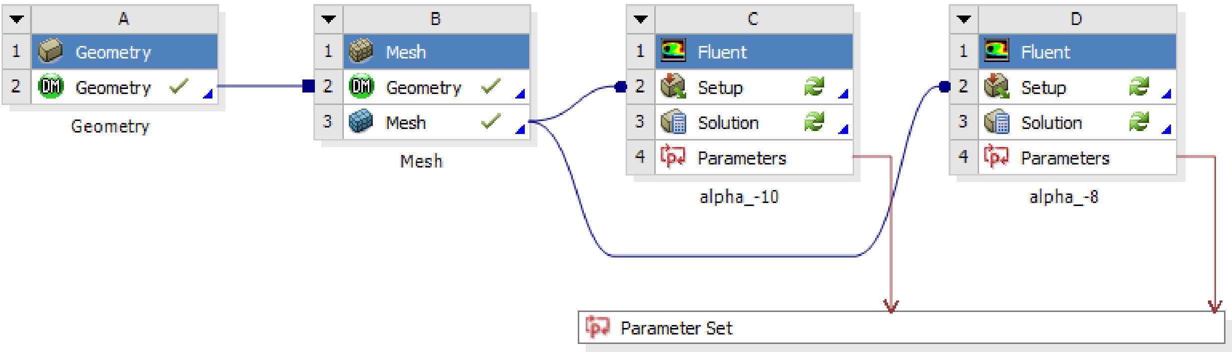


Figure 7.1: Workbench file created for a each airfoil.

with a step of 2°. The set-up for the FLUENT simulations can be scriptised with a FLUENT journal file, automatically generated by the program. Finally, a Workbench script is needed to direct these three software and make them perform their tasks in the correct order. Summarizing, the ingredient pieces needed for the generation of a Workbench project are: a Workbench script, a Design Modeler JavaScript, an ANSYS Meshing JavaScript and a FLUENT Journal file. In the following sections, the scripting process of each one of these software will be described.

7.1.1 Scripting ANSYS DesignModeler

ANSYS Design Modeler, the software used for the geometry creation, does not allow to register all the instructions given during this process. As a matter of fact, the script has to be written almost completely from scratch, leading to some difficulties because of the lack of documentation. Useful hints have been found in the Scripting Application Program Interface (Scripting API, [4]), but they refer essentially to basic commands. The main source of information regarding the creation of the scripts was the ANSYS installation folder, where the needed instructions were found by searching for key words or by analysing the Java scripts contained there. Documentation has been found also on the website CFD online [1]. The structure of the JavaScript code written to automatise the creation of the domain for the damaged airfoils with Design Modeler is as follows:

1. First of all, it loads the files containing the four curves that define the damage (see figure 7.2). The full path of the folder where the files are contained is needed by the script.

2. Then, it connects the starting and/or ending points of the curves in order to close the geometry (it can be seen from figure 7.2 that the curves are not connected);

3. The following step is that to create the inlet and outlet boundaries. These are considered by Design Modeler as sketches of the active plane and their creation can be registered, simplifying the work.

4. Then, it creates the guiding lines on the upper and lower edges of the damage, giving as

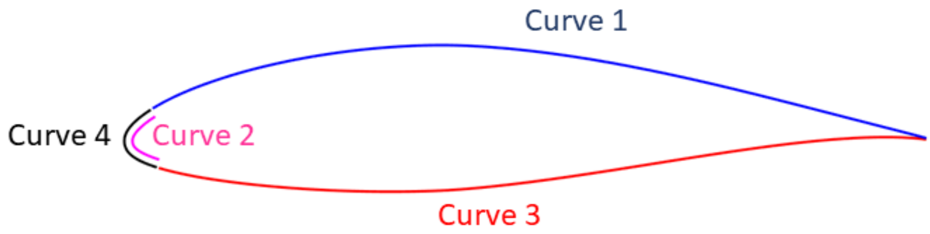


Figure 7.2: Curves generated for each one of the damages.

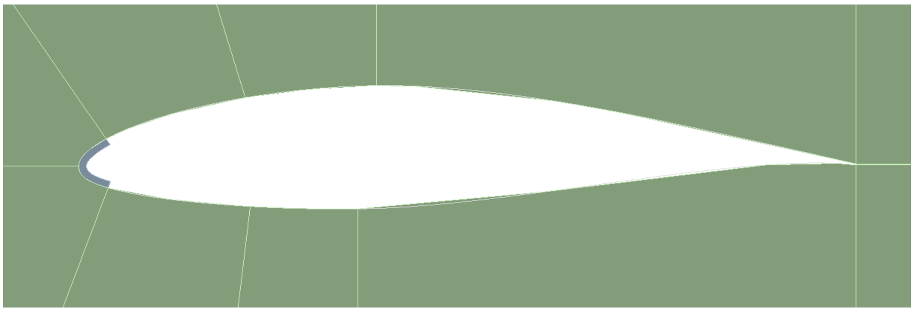


Figure 7.3: Detail of the computational domain around a damaged airfoil.

input to the program their starting points, identified as the first and last elements of the array of points "Curve 4" respectively, and their directions. In this way, the script is general and works for every possible damaged geometries. These guiding lines are needed to obtain a mesh with good orthogonal quality.

5. Finally, it creates all the other guiding lines. They start from fixed coordinates on the airfoil surface, and they have a fixed direction. The values for these parameters are determined manually, and they are the same for all the delaminated geometries referring to the same baseline airfoil. The guiding lines for a damaged geometry can be seen in figure 7.3. When the domain has been completed, it results in a division into 12 faces: one for the damage, and the other 11 delimited by the lines visible in figure 7.3. It could seem that the faces are 10, but the trailing edge of the airfoil is truncated, so the horizontal lines starting from the back of the airfoil are two, and they enclose a very thin face.

The only thing that needs to be changed in the script, from one damaged airfoil to another referring to the same baseline geometry, is the path in which to find the four coordinates files. A MATLAB program has been implemented to copy the general Design Modeler script into each one of the folders containing the geometries of the damaged airfoils, every time changing the path. For each damaged airfoil referring to the same baseline geometries, the scripts will be all equal apart from the lines containing the path of the coordinates file. For damaged airfoils referring to different baseline geometries, the scripts will be different in the position and direction of the guiding lines on the airfoil.

7.1.2 Scripting ANSYS Meshing

ANSYS Meshing allows the instructions given to the GUI to be registered while creating the mesh, making the scripting process easier. For this work, however, since the needed commands were few and easily found on the same sources already cited for DesignModeler, the script has been created manually. The structure of the JavaScript read by ANSYS meshing is as follows:

1. First of all, it sets all the general options, such as the solver preference, the physics preference (CFD in our case), the smoothing of the mesh, the maximum element size and so on.

2. Then, it sets the sizing and the bias for all the edges in the computational domain.

3. Finally, it imposes the quadrilateral face meshing on all the 12 faces, in order to obtain a structured mesh.

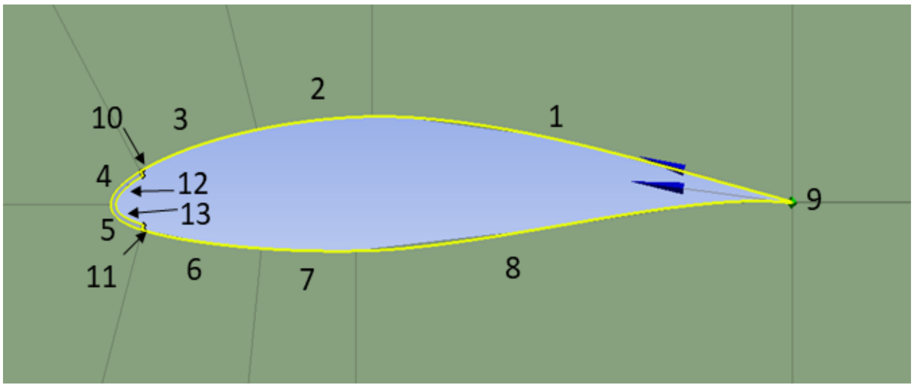


Figure 7.4: Division of the airfoil profile in the meshing process.

Regarding the sizing and the bias of the edges, let us consider the damaged airfoil profile shown in figure 7.4. The airfoil surface is divided into 13 segments:

- **9: Back edge of the airfoil**. The trailing edge of all the airfoils is truncated, so at the back there is an edge. The number of divisions for this short edge is set to 4 for all the grids.

- **1, 2, 7, 8: Fixed length segments on the pressure and suction side**. These segments do not change in length when the damaged geometry changes, because they are not involved in the delamination process. Since their lengths is quite similar for all the types of airfoil considered in this work, they have been discretized with the same number of elements in all cases. Edges 1 and 8 are characterized by 94 elements, biased towards the trailing edge, while edges 2 and 7 are covered by 10 elements with no bias.

- **3, 6: Variable length segments on the pressure and suction side**. The length of these two segments changes with the delaminated geometry because they have one fixed end (the vertices in common with segments 2 and 7 respectively), and one mobile end (the vertices in common with segments 4 and 5 respectively), that depends on the curvilinear length of the delamination damage. The number of elements on these segments changes with the eroded geometry, and they are biased towards their mobile end.

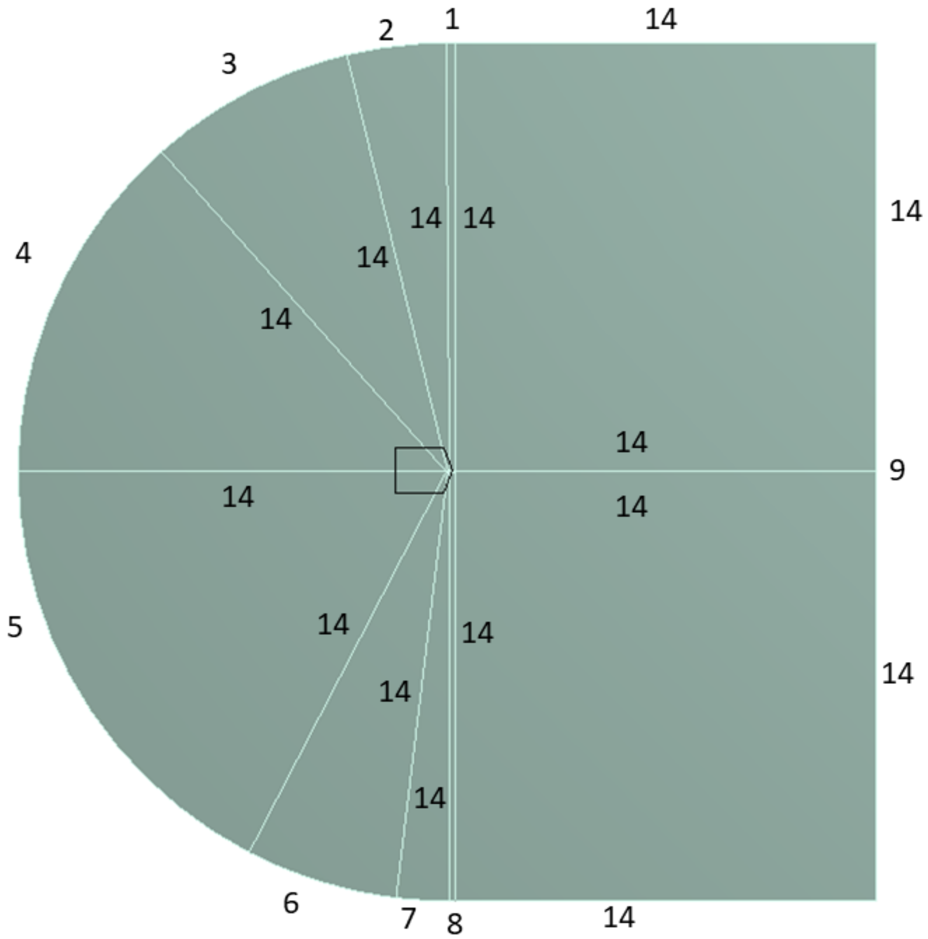


Figure 7.5: Edges on the boundary of the domain and internal lines.

- **4, 5, 12, 13: Upper and lower edges of the delamination damage**. The length of these segments changes with the delamination damage, and therefore the number of elements on them is not kept constant. Edges 4 and 5 are considered as internal lines: they are only needed for the meshing process of the face corresponding to the damage, but they do not contribute to the airfoil boundary. Edges 4,12 and 5,13 are biased towards edges 3 and 6 respectively.

- **10, 11: Depth of the delamination damage**. The length of these two edges changes with the damage, and consequently also their sizing changes.

Almost all the edges around the airfoil have their counterpart on the boundary of the domain, as shown in figure 7.5. These edges are characterized by the same number of elements and the same bias of those around the airfoil. The edges labelled 14, are instead all discretized by 150 elements, biased towards the airfoil.
The decision about the number of elements assigned to each variable length segment is based on

the definition of the quantities characterizing the sizing. The software ANSYS Meshing allows to set, for each edge, the number of elements, the type of bias and the bias factor. The bias factor is defined as the length of the longest element divided by the length of the shortest elements characterizing the discretization:

$$bias\ factor = L/l, \tag{7.1}$$

where L is the length of the longest element and l that of the shortest one. The bias factor, in turn, defines the growth rate of the elements:

$$growth\ rate = (bias\ factor)^{\frac{1}{N_{el}-1}}, \tag{7.2}$$

where N_{el} is the total number of elements on the edge. This means that, if the shortest element of the discretization has length l, the following one has length $(growth\ rate) \cdot l$, the next one $(growth\ rate)^2 \cdot l$ and so on. The sum of the lengths of all the elements must give the length of the segment, namely:

$$\sum_{i=0}^{N_{el}-1} (growth\ rate)^i l = L_{edge} \tag{7.3}$$

This is a geometric series of common ratio *growth rate*, so its finite sum from $i = 0$ to $i = N_{el} - 1$ is given by:

$$\frac{1 - (growth\ rate)^{N_{el}}}{1 - (growth\ rate)}. \tag{7.4}$$

Recalling the definition of the growth rate, 7.3 becomes:

$$\frac{1 - (bias\ factor)^{\frac{N_{el}}{N_{el}-1}}}{1 - (bias\ factor)^{\frac{1}{N_{el}-1}}} = \frac{L_{edge}}{l}. \tag{7.5}$$

Solving for N_{el} and denoting the bias factor with the symbol b, one obtains:

$$N_{el} = \frac{\log_b(L_{edge}/(bl) - 1) - \log_b(L_{edge}/l - 1)}{1 + \log_b(L_{edge}/(bl) - 1) - \log_b(L_{edge}/l - 1)} \tag{7.6}$$

The formula gives the number of elements needed to obtain a smallest element of length l with a bias factor b. The most important constraint regarding the sizing is that, towards the airfoil, the minimum wall distance guarantees a maximum $y^+ \leq 1$, so l should be lower or equal than the minimum wall distance. Let us consider edges 10 and 11: they are biased on both ends, with a bias of the type

$$\rule{1cm}{0.4pt}\quad\rule{2cm}{0.4pt}\quad\rule{4cm}{0.4pt}\quad\rule{2cm}{0.4pt}\quad\rule{1cm}{0.4pt}. \tag{7.7}$$

Since the bias is symmetric with respect to the midpoint of the segment, it is possible to consider only half a segment with bias of type:

$$\rule{1cm}{0.4pt}\quad\rule{2cm}{0.4pt}\quad\rule{4cm}{0.4pt}, \tag{7.8}$$

or:

$$\rule{4cm}{0.4pt}\quad\rule{2cm}{0.4pt}\quad\rule{1cm}{0.4pt} \tag{7.9}$$

Edge	Max number of elements		
	Mild	**Moderate**	**Severe**
4	15	35	56
5	20	46	73

Table 7.1: Number of elements on edges 3 and 4 for the most severe delamination damage of each category.

and then double up the number of elements needed to obtain the desired value of l with bias b. The bias factor b is set to $b = 10$ for the mild damage, $b = 100$ for the moderate, and $b = 200$ for the severe. These values have been chosen after some attempts because they gave good results for all the cases considered. The value of N_{el} for these edges is between 70 and 80 in all cases. Consider now edge 4, which is biased towards edge 3 (see figure 7.4) with a bias of type:

$$\text{——————} \quad \text{———} \quad \text{—} \tag{7.10}$$

The number of elements on this edge is obtained using equation 7.3, imposing that l is equal to the minimum wall distance needed to obtain a y^+ value less or equal than 1 and $L_{edge} = s_u$, the curvilinear length of the delamination damage on the upper side of the airfoil. The bias factor is set to $b = 1000$ for the mild damage, $b = 1500$ for the moderate damage, and $b = 2000$ for the severe damage. The number of divisions on edge 3, which is biased towards edge 4, is then determined as $100 - N_4$, where N_4 is the number of elements on edge 4. In this way, the total number of elements on the segment obtained joining edges 3 and 4 is constant for all the damaged geometries. The same bias factor adopted for edge 4 is imposed on edge 3. In this case, the minimum wall distance does not need to be as small as on edge 4, so there is more elasticity about the sizing. The same exact procedure is repeated for edges 5 and 6, changing the value for L_{edge} with s_l, the curvilinear length of the damage on the lower side of the airfoil. To make sure that the number of elements on edges 4 and 5 was never greater that 100, the values of N_{el} obtained for the most severe damage of each category have been checked (see table 7.1). One ANSYS Meshing script has been created for each damaged airfoil with the help of a MATLAB program. First of all, one general script has been prepared with labels in place of the number of elements and bias factor of the edges with variable length. Then the MATLAB program, for each damaged geometry, computes the needed values as explained above, and substitutes the labels with the correct numbers. Finally, it saves the JavaScript in the folder relative to the damage considered.

7.1.3 Scripting ANSYS Workbench and FLUENT

The ANSYS Workbench script, called Workbench journal file, is needed to connect all the different software. Moreover, it is strictly linked to the scripting of FLUENT, which is based on the scheme language. Selecting

$$File \rightarrow Scripting \rightarrow Record\ \ Journal...$$

on ANSYS Workbench, in fact, it is possible to register all the interactions with the Workbench itself and all the interactions with FLUENT, but not with Design Modeler or ANSYS Meshing.

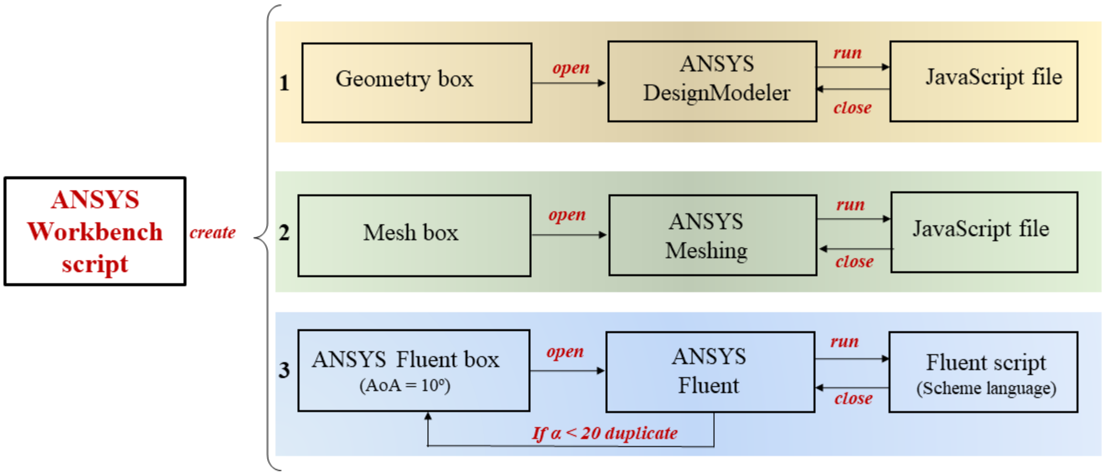


Figure 7.6: Tasks performed by the ANSYS Workbench journal file; $N_{airfoils}$ refers to the number of different baseline geometries analysed, while N is the total number of damaged airfoils referring to the same baseline geometry..

Explained in detail:

- If one starts recording from the Workbench and creates a geometry box, this action will be recorded and written on a file, whose location is selected by the user.
- If one opens DesignModeler from the geometry box and starts creating features, these will not be recorded. The same happens for ANSYS Meshing, which has, however, another internal scripting tool.
- Finally, if one creates and opens a FLUENT box from the Workbench, both its creation and the actions performed within FLUENT will be recorded. The FLUENT scripting is, therefore, strictly connected to the ANSYS Workbench scripting.

Figure 7.6 displays schematically the role of the ANSYS Workbench script:

1. The script gives the instruction to create a geometry box and to open the software Design Modeler. Once inside DesignModeler, it directs the program to run the JavaScript file that creates the desired geometry and the computational domain. Finally, when the creation of the features is complete, DesignModeler is closed.
2. The second step is that to create a mesh box and to connect it to the geometry box. The mesh box allows to open ANSYS Meshing with the geometry and computational domain created before already loaded. Once inside the program, the script gives the instruction to run the JavaScript file which defines the mesh features and generates the mesh. Finally, ANSYS Meshing is closed.
3. The final task is the creation of the 16 FLUENT boxes, one for each angle of attack from $-10°$ to $20°$ with a step of $2°$. First of all, the script connects a FLUENT box to the mesh box. After this operation, and only for the first FLUENT box, the mesh box needs to be

updated in order to transfer its information to FLUENT. Then, the Workbench journal opens ANSYS FLUENT from the FLUENT box and it configures the numerical settings of the simulation for the first angle of attack. Once everything is set correctly, FLUENT is closed. Finally, the same procedure is repeated for all the other 15 FLUENT boxes referring to the remaining angles of attack. For what concerns running the simulations, two options are available:

- The first one is to run the simulation for each angle of attack as soon as the set up is completed. In this case, the procedure would be to create the first FLUENT box, set up and run the simulation, close the FLUENT box, and do the same for the remaining 15 cases.
- The second option is that to update the project once set up all the FLUENT boxes: this instruction runs all the simulations for the different angles of attack one after the other.

The only ANSYS Workbench tasks that are not registered are the instructions to read the JavaScript files, but their format can be easily found with an on-line search. The creation of the Fluent script for all the 16 FLUENT boxes by hand is time consuming: it takes almost an hour to complete. To speed up the scripting process, the complete set up has been registered only for the first FLUENT simulation, the one referring to the angle of attack of $-10°$. The FLUENT box for the angle of attack of $-8°$ has been obtained by duplicating the first one, and modifying inside FLUENT only the settings that needed to be changed from one angle to another, that is the components of the freestream velocity and the vectors along which to calculate the lift and drag forces. Then, the lines of the code referring to the duplicated box have been isolated, and the numbers varying with the angle of attack have been substituted with labels. Finally, a MATLAB program duplicates these lines of code for each remaining angle of attack, replaces the labels with the correct values for the angle considered and adds these lines to the Workbench script. In this way, the FLUENT section of the code can be obtained much faster than by creating all the boxes by hand while recording the operations. The FLUENT script is general and works for each possible damaged airfoil. The Workbench script differs from one airfoil to another only in the JavaScript files loaded for the geometry and mesh creation.

7.2 Automation of the system

Once scripted all the single components of the system, it is necessary to automatise the whole process. In this section, the initial idea will be presented, along with the reasons why it was later on abandoned. Then, the solution that has actually been applied will be described in detail.

7.2.1 Original idea and its impracticability

The original idea for the generation of the database was to run all the scripts remotely on a big cluster. For each damaged airfoil, a bash file should have run in batch mode the ANSYS Workbench script, which in turn would have generated the geometry, the mesh, and the set-up for all the simulations. Then, the plan was to update the project running all the simulations for the 16 angles of attack, save the Workbench file, and go on with the next airfoil, until completing the database. During the development of the system, it turned out that it was not possible to run the Design Modeler and ANSYS Meshing JavaScript files in batch mode: the implemented instructions were based on the presence of the Graphical User Interface. Moreover, the GUI of Design Modeler and ANSYS Meshing was not working on the cluster for security reasons. It was therefore not possible to proceed in this way and an alternative solution needed to be found. With reference to figure 7.6 for the definition of the task numbers, the work flow explained above

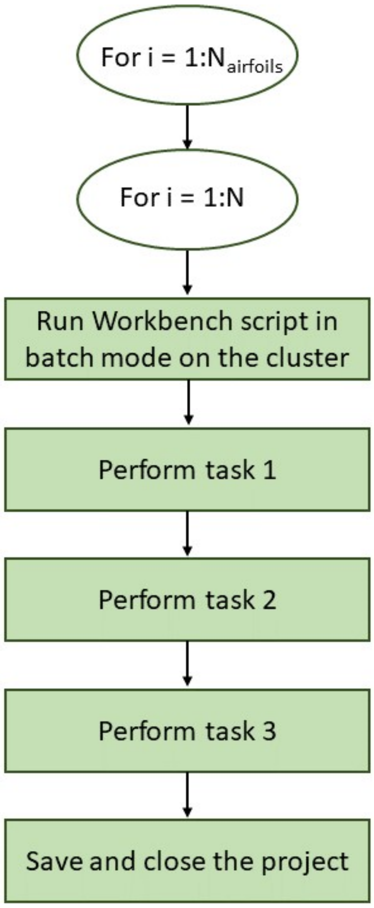


Figure 7.7: Work flow of the original idea: the definition of the tasks can be found in figure 7.6 or in subsection 7.1.3. $N_{airfoils}$ is the total number of different baseline airfoils considered, while N is the number of delaminated geometries corresponding to the same baseline airfoil.

is illustrated in figure 7.7. In the figure, $N_{airfoils}$ refers to the total number of different baseline airfoils considered, while N is the number of delaminated geometries corresponding to the same baseline airfoil. This option can be defined as serial because the tasks are performed one after the other. The positive aspect of this implementation is that, once run the bash script on the cluster, the process is completed automatically without the need of any further input from the user. The downside is that a great number of different controls should be implemented in order to monitor the process correctly. If the process should fail, in fact, without controls it would be difficult and time consuming to determine the cause of the failure and to restart the process from where it stopped.

7.2.2 Alternative applied solution

The main problem regarding the first idea tried was the absence of the GUI on the cluster. This meant that all the tasks involving Design Modeler and ANSYS Meshing needed to be performed on a local PC. Another possible solution was then to run these two programs on a PC for all the damaged geometries, load the incomplete Workbench files on the cluster and run there only the part of the script involving the FLUENT simulations. This option was particularly time consuming: the set up of the FLUENT simulations for the 16 angles of attack takes about 10 minutes for each damaged airfoil. Summing this up to the 6000+ airfoils considered, it leads to about 1000 hours of computational time. At best, 20 airfoils could be considered simultaneously on the cluster, and this gives 2 days of work just for the set-up of the simulations, to be added to the time required to generate the geometries and the meshes on a local PC. A solution to this problem is provided by the fact that, once having a complete project (with the geometry, the mesh, and the set up for all the simulations), ANSYS Workbench is able to automatically change the geometry and the mesh inside all the FLUENT boxes if they are changed upstream in the geometry box and in the mesh box. Therefore, since the settings for the simulations are the same for all the airfoils, it is possible to start from a complete Workbench project, and to substitute, for each damaged airfoil, the geometry and the mesh of the original project with the correct ones. This could be defined as a "block implementation", because the tasks are not performed all together one after the other, but in separate processes. The positive aspect of this implementation is that the single tasks are easier to monitor: if a program crashes it is straightforward to identify the problem and the cause. For this reason, it is not necessary to implement further controls. Moreover, the failure of a single task does not stop the entire process. The only negative aspect is that some input from the user is needed. It must be pointed out, however, that this refers only to running the Windows batch scripts and loading the projects on the cluster. All things considered, the original idea and the implemented one are simply two different possible ways to address the same problem. Once solved the issues encountered in the implementation of the first idea, one or the other can be chosen at the users' discretion, depending on their needs and preferences. The alternative applied solution will be described in detail in the following sections.

7.2.2.1 Creation of the original workbench project

As said in chapter 4, the folders containing the files related to each damaged airfoil are identified by a six digits number ranging from 000001 to N, N being the total number of damaged airfoils referring to the same baseline geometry. The idea is to create an original complete Workbench project, with the geometry, the mesh, and the FLUENT boxes, and to copy this complete project into each one of the folders containing the coordinates of the damaged airfoils. The original Workbench project is created using the geometry of the first damaged airfoil, the one contained in the folder 000001. For this airfoil, the generation of the Workbench project follows the procedure which was intended to be used for all of them, not remotely but using a local PC supporting the GUI. Only for this airfoil, therefore, the Workbench script is run from the Workbench, generating the geometry, the mesh, and the set up for all the simulations. The whole process takes about 15 minutes. Once having the complete Workbench project, ANSYS Meshing is opened again manually, and only the features which are common to all the damaged geometries are kept. In this way, when updating the mesh for a new geometry, it will be possible to add only the sizing and bias for the edges that have changed in length, reducing the computational time. Reading a complete JavaScript code for the creation of a mesh takes about 10 seconds, while reading a script that adds only the sizing of the variable length edges requires no more that 3 seconds. It could seem a negligible difference, given the time scales we are talking about, but when generating thousands of meshes it makes a difference. Moreover, it has been observed within this project that often reading a long script causes ANSYS Meshing to crush, while this never happened with the shorter scripts used. All the Workbench projects are characterized by a graphical interface with extension .wbpj, and a folder containing the files read by the project: the geometry file, the mesh file, and the files containing the set up for each simulation. Once created the original Workbench project, all the files are copied with the help of a Windows batch script into each one of the damaged airfoil folders, that now contain:

- The coordinates of the damaged airfoil;
- The DesignModeler script which creates the geometry and the computational domain for the simulations;
- The ANSYS Meshing script which sets the sizing for the edges that vary in length;
- The original Workbench project;

The process is summarized in figure 7.8.

7.2.2.2 ANSYS Workbench script

Since the strategy for the generation of the database has changed, the part of the ANSYS Workbench script that generates the FLUENT boxes and the set ups for all the simulations is no longer necessary. What is still necessary, instead, is the part of the script which generates geometry and mesh. The Workbench script contained into each one of the damaged airfoils folders has the following structure:

- First of all, it opens DesignModeler from the geometry box, and it runs the DesignModeler script which replaces the geometry used to create the original Workbench project with the current one. When the geometry has been substituted, DesignModeler is closed;
- Then, it opens ANSYS Meshing from the mesh box and it runs the script which adds the missing sizing features When all the settings for the mesh creation have been set, the program is closed.
- Finally, it generates the mesh associated to the new geometry selecting "Update" from the options of the mesh box. This operation updates the mesh also in all the FLUENT boxes. Once everything is done, the project is saved and closed.

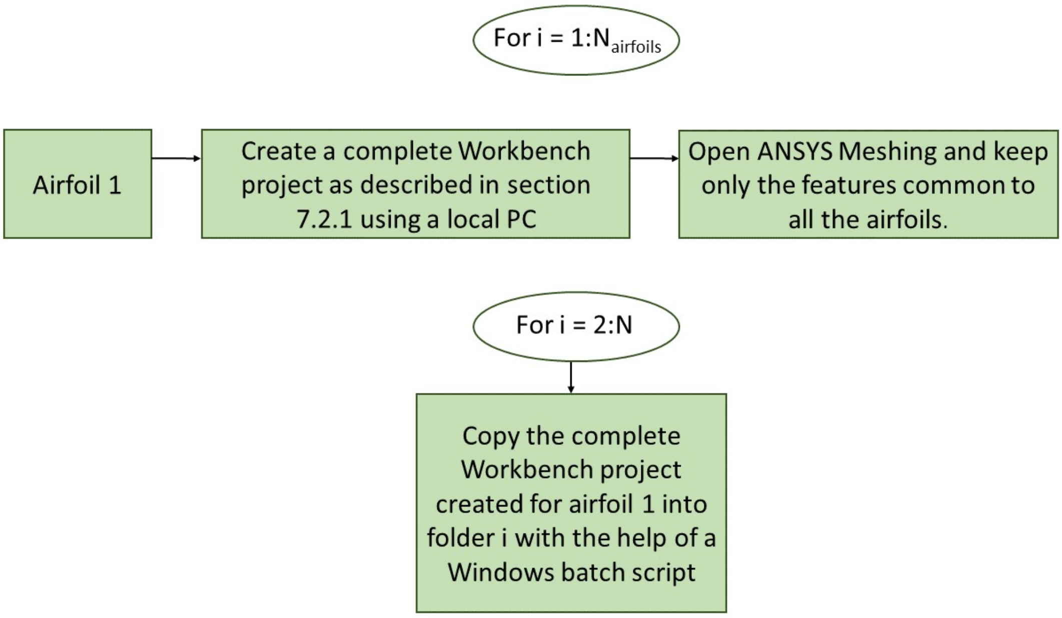


Figure 7.8: Schematic explanation of the process described in this section. The complete Workbench project is created only for airfoil 1 and then copied inside all the damaged airfoils folders.

The process is summarized in figure 7.9. All these operations must be done on a PC that supports the GUI of ANSYS: in our case, it was a local computer. The time required to update the project for each damaged geometry is about 5 minutes. For this project, about 1000 damaged geometries were created for each nominal airfoil. Updating one project after the other would require about 83 hours of computational time. To speed up the process, since the PC used had four cores, an equal amount of work has been assigned to each core. For this purpose, four Windows batch script have been written: each one of them opens a project, runs the ANSYS Workbench script, and when the project is closed it moves on to the next one. If there are N projects to update, the first script updates those from 1 to $N/4$, the second one those from $N/4+1$ to $N/2$, and so on with the other scripts. This solution allows to reduce the computational time to about 20 hours.

7.2.2.3 Running the simulations

After having updated all the Workbench projects, they were load on the High End Computing cluster of Lancaster University in order to run all the simulations. For each damaged airfoil, a bash script gives the instruction to read a further Workbench script, which in turn runs the simulations for all the angles of attack. This Workbench script was recorded as described in section 7.1.3 and is the same for all the damaged airfoils. All the simulations were launched together: some of them started immediately, and others were queued until a cluster node became available. The simulations for the sweep of 16 angles of attack for each damaged airfoil required about 64 minutes to run using a 16 cores node. The availability of the nodes on the cluster is variable, but usually 20 of them could be used together, allowing to complete all the simulations

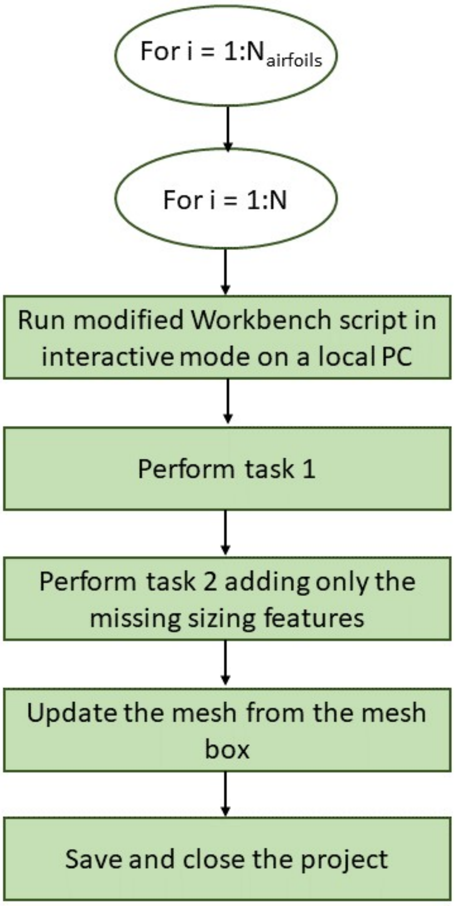


Figure 7.9: Schematic explanation of the process described in this section.

for a type of airfoil in about 2 days and a half. Obtaining all the results for the generation of the CFD database required about 2 weeks.

7.2.2.4 Extraction of the lift and drag curves

Once all the simulations have been run, the history of convergence of the lift and drag curves generated by FLUENT are stored into the folder of the Workbench project files relative to the angle of attack to which they refer. The vast majority of the simulations reached convergence, while a small number of them diverged or kept oscillating. To obtain the lift and drag curves characterizing the damaged airfoils, it is necessary to distinguish between the cases. This analysis is done with the help of a Matlab program, which checks the convergence and classifies the results in the following way:

- If the solution diverged, the simulation is classified as failed.
- If the solution converged; the difference between one approximation of the solution and the next one is computed for the last 100 iterations. If one or more of these differences are greater than 10^{-3} for the lift and 10^{-4} for the drag, the simulation is classified as unsteady. This means that the value keeps oscillating, and it is often due to the fact that a steady simulation is not able to capture the correct aerodynamic behaviour of the flow, and an unsteady simulation would probably be needed. This happens especially for the highest angles of attack. If, instead, the convergence criterion is respected, the solution is classified as converged and the last value of the lift and drag coefficients are extracted and collected into a table.

A list of the failed and unsteady simulations is also generated by the program. The reason for the fail is usually some issue regarding the mesh, so the failed simulations are corrected and run again. The unsteady simulations are instead discarded. Finally, for each airfoil, the MATLAB program creates a table with six columns and as many rows as the number of simulations converged. The first three columns contain the parameters of the damage: s_u/c, s_l/c and d/c, the fourth column the angle of attack, and the last two columns the computed list and drag. These tables are fundamental for the last step of the automation process: the machine learning approach.

7.3 Machine learning approach

The machine learning algorithms described in this section have been developed by Dr. Edmondo Minisci as part of a collaboration with Strathclyde University.

Machine learning consists of using algorithms and statistical models to develop computer systems that effectively perform a specific task without using explicit instructions, but rather relying on data patterns and inference. ML algorithms build a mathematical model based on given data sets (training data) to make predictions without being explicitly programmed to perform the task. In this study, ML is used to develop a mathematical model providing rapid estimates of the eroded airfoil force coefficients without solving the Navier-Stokes equations by means of time-consuming CFD simulations. Generic multi-layer perceptron (MLP) feed-forward Artificial

Neural Network with one hidden layer are used in ALPS to learn from the lift coefficient c_l and the drag coefficient c_d data patterns in the ALPS database of damaged airfoils, and infer the aerodynamic forces of eroded airfoils not contained in the database. One c_l and one c_d ANN model is contructed for each of the 6 airfoils making up the blade of the NREL 5 MW turbine.

Single layer MLPs like those used herein consist of universal function approximators $\mathbf{f_g}(\mathbf{x}) : \mathbb{R}^{n_d} \to \mathbb{R}^{n_o}$ [10], where n_d is the size of the input vector $\mathbf{x}$, and n_o is the size of the output vector function $\mathbf{f_g}$. In the present ALPS version, $\mathbf{x} = (s_u/c, s_l/c, d/c, \alpha)^T$ and $n_d = 4$, whereas $n_o = 1$ since the output function is scalar, corresponding to either c_l or c_d. The general matrix-vector definition of $\mathbf{f_g}$ is:

$$\mathbf{f_g}(\mathbf{x}) = A2(\mathbf{b}^{(2)} + \mathbf{W}^{(2)}(A1(\mathbf{b}^{(1)} + \mathbf{W}^{(1)}\mathbf{x}))) \tag{7.11}$$

where $\mathbf{W}^{(1)}$ is a weight matrix of size $(N \times n_d)$ and N is the number of *neurons* on the hidden layer, $\mathbf{b}^{(1)}$ is a bias (column) vector of length N, $\mathbf{W}^{(2)}$ is a weight matrix of size $(n_o \times N)$, $\mathbf{b}^{(2)}$ is a bias (column) vector of length n_o, and $A1$ and $A2$ are the activation functions of the hidden layer and the output layer respectively. The general schematic of one-layer MLP feed-forward ANN systems, like that used in the present ALPS release, is depicted in Fig. 7.10.

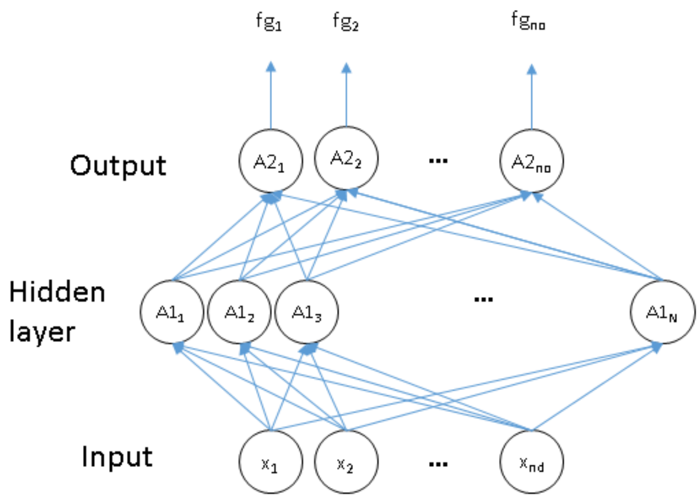


Figure 7.10: General schematic of one-layer MLP feed-forward ANN systems.

Typically, $A1$ is the hyperbolic tangent sigmoid function defined as $tanh(a) = (e^a - e^{-a})/(e^a + e^{-a})$, and $A2$ is linear. With this choice, function $\mathbf{f_g}(\mathbf{x})$ becomes:

$$\mathbf{f_g}(\mathbf{x}) = \mathbf{b}^{(2)} + \mathbf{W}^{(2)}(tanh(\mathbf{b}^{(1)} + \mathbf{W}^{(1)}\mathbf{x})) \tag{7.12}$$

Given a set of N_s training samples $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_{N_s}, \mathbf{y}_{N_s})\}$, where $\mathbf{y}_i$ is the observed response to the input $\mathbf{x}_i$, a learning algorithm seeks the values of the weight matrices and bias vectors that minimise the difference between part or all of the observed N_s sample responses $\mathbf{y}_i$ and the N_s responses given by $\mathbf{f_g}(\mathbf{x}_i)$. Gradient-based optimization is often used to determine the weight matrices and the bias vectors by minimizing the aforementioned difference. In this circumstance, since the responses are assumed to be smooth functions of the inputs $\mathbf{x}_i$ and the internal weights and biases, the gradients with respect to weights and biases of the difference between sample responses and ML approximations can be computed using the so-called backpropagation method, which essentially relies on applying the chain rule for derivation. The backpropagation approach can be applied repeatedly to propagate gradients through all layers starting from the output at the top, where the network gives its response, and working all the way down to the bottom (hidden layer), where the input is provided. Once the gradients with respect to weights and biases for each layer are computed, the objective function expressing the level of fitting of the training data can be optimised.

If the ANN system has a sufficiently high number of neurons or degrees of freedom in the hidden layer, a parameter proportional to the chosen number of neurons, the fitting error can be reduced to machine zero; this results in the system *overfitting* the training data, and being possibly unable to generalize adequately its predictions to regions of the input space where there are no or insufficient training data. To mitigate this risk, two different approaches have been adopted in this study. One approach consists of subdividing the available data set into a training set and a validation set. At each step of the gradient-based optimization via backpropagation (training), the fitting error based on the validation set is also computed. The sysyem training is then stopped when the fitting error based on the validation set starts increasing. This typically happens before the fitting error based on the training set achieves machine zero. The alternative approach introduces one or more *regularisation* terms that modify the fitting function so that the obtained ANN system has good generalization properties. One of the most widely used regularisation approaches is Bayesian regularisation, in which a modified linear combination of fitting errors, weights and biases is minimised [22].

Using either learning approaches to prevent overfitting, the learning process via backpropagation consists of a local gradient-based optimisation problem, which is solved with the Levenberg-Marquardt (LM) method [14]. Because of the local nature of the method, however, the learning outcome can be affected by the process initialisation. To reduce the likelihood of this occurrence, 100 training runs with different random initializations are carried out both with the training and validation set approach, and the Bayesian regularization approach. Moreover, this is done with different values of N in both cases to determine the value of this parameter that maximizes the predicitive capabilities of the ANN system. In this study, the size of the training set is 95 % of the randomised database when using Bayesian regularization, and the remaining 5 % forms the test set, used to test the generalization strength of the ANN system. The size of the training set is instead 80 % of the randomised database and that of the validation set is 15 % of the same database when using the approach without regularization, and the remaining 5 % forms the test set. The ANN system selected for the force coefficients of each of the 6 sets of damaged airfoils is that with high fitting of the training set and good generalization on the test test. The typical optimal value of N for both the lift and force coefficients of the ANN system of the 6 airfoils is found to vary between 20 and 30.

Chapter 8

CFD validation

This chapter deals with the validation of the CFD set-up described in chapter 5. First of all, the model turbine adopted for the work is presented, together with a description of the blade and of the six airfoils composing it. Since the grids generated for all the nominal airfoils are characterized by the same features, a mesh refinement study has been carried out only for the most external one. The lift and drag curves obtained with CFD simulations for the six airfoils are then validated against experimental data. Finally, the validation of the nominal and damaged DU 96-W-180 is carried out in order to assess the capabilities of the developed damage representation.

8.1 NREL 5MW blade sections

The model turbine used for the demonstration of the capabilities of ALPS is the NREL 5 MW (see [18]), a three bladed variable speed and variable pitch-to-feather turbine. It is made up of six different airfoils: the NACA 64-618, the DU 93-W-210 LM, the DU 91-W2-250, the DU 97-W-300 LM, the DU 99-W-350 and the DU 99-W-405 LM. A graphical representation of the blade is shown in figure 8.2, while the shapes of the six airfoils can be seen in figure 8.1. The two cylinders are characterized by zero lift and a constant drag of 0.35 for cylinder 1 and 0.5 for cylinder 2. The lift and drag curves of the six airfoils, instead, have been obtained through wind tunnel experiments and are available inside the certified tests folder of the FAST software [17]. In the following sections, a validation of the CFD set-up described in chapter 5 against the experimental data has bee carried out for all the airfoils.

8.1.1 NACA 64-618 airfoil

8.1.1.1 Mesh refinement study

Since the grids used for all the airfoils of the NREL 5 MW turbine are characterized by the same parameters, a mesh refinement study has been carried out only for the NACA 64-618. Three grids were considered in the analysis: a coarse grid, a medium grid, and a fine grid. The coarse grid is characterized by 196 elements on the airfoil, 75 elements along the C-cut and 75 elements in the normal direction, for a total of 26 000 quadrilateral elements. The adopted wall distance

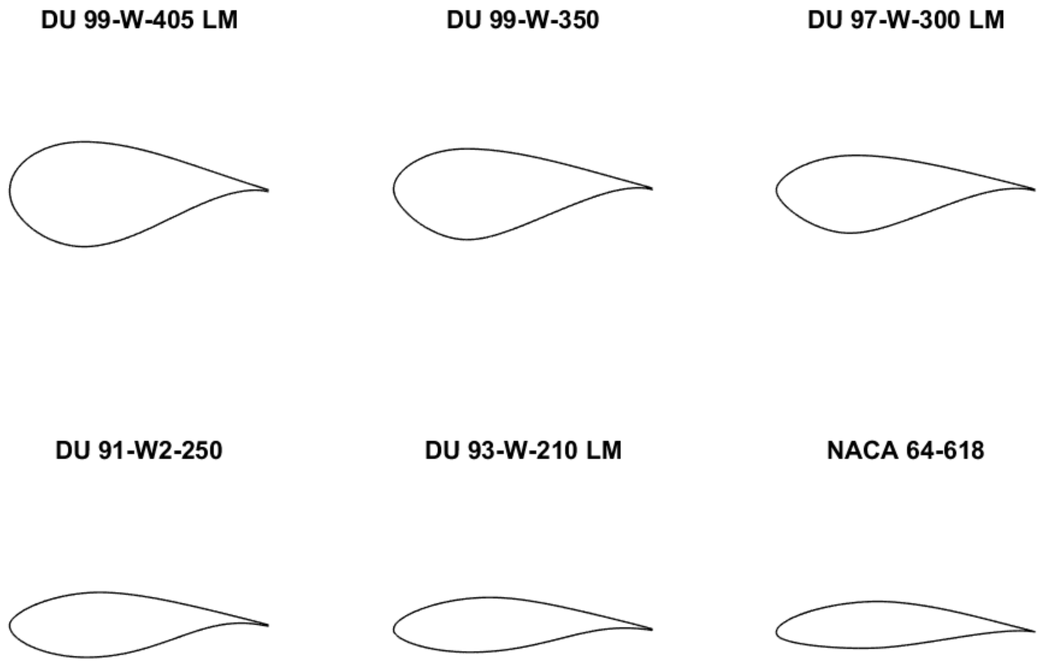


Figure 8.1: The figure shows the six airfoils composing the NREL 5 MW blade.

Grid	El. on the airfoil	El. along the C-cut	El. normal dir.	Total elements	Wall distance
Coarse	196	75	75	26 000	8.8×10^{-6}
Medium	326	150	150	94 000	4.4×10^{-6}
Fine	680	250	250	320 000	2.2×10^{-6}

Table 8.1: Parameters for the mesh refinement study for the NACA 64-618 airfoil.

of $8.8 \times 10^{-6}c$ guarantees a y^+ value of about 2. The medium grid features 326 elements on the airfoil, 150 elements on the C-cut and 150 elements in the normal direction and has about 94 000 elements total. The wall distance adopted in this case is $4.4 \times 10^{-6}c$, and it guarantees a maximum value of y^+ of about 1. A picture of the medium grid around the airfoil, with a close-up view on the leading edge, can be seen in figure 8.3. Finally, the fine grid is characterized by 680 elements on the airfoil, 250 elements along the C-cut, and 250 elements in the normal direction, and has a total of 320 000 elements. In this case, the adopted wall distance is $2.2 \times 10^{-6}c$, and it guarantees a maximum y^+ of about 0.5. The parameters characterizing the three grids have been summarized in table 8.1 for best clarity. The lift coefficient as a function of the angle of attack and the polar curve obtained with the three grids are shown in figure 8.4. The results obtained with the three grids for the prediction of the lift coefficient are almost superimposed (see the left picture of figure 8.4). A small difference between the coarse grid and the medium and fine grids can be noticed for angles of attack higher than 6°. For the 20° angle of attack, some differences can be noticed also between the fine and medium grids. This is due to the fact that for such high angles of attack the stall has a big influence and the aerodynamic behaviour of the airfoil could

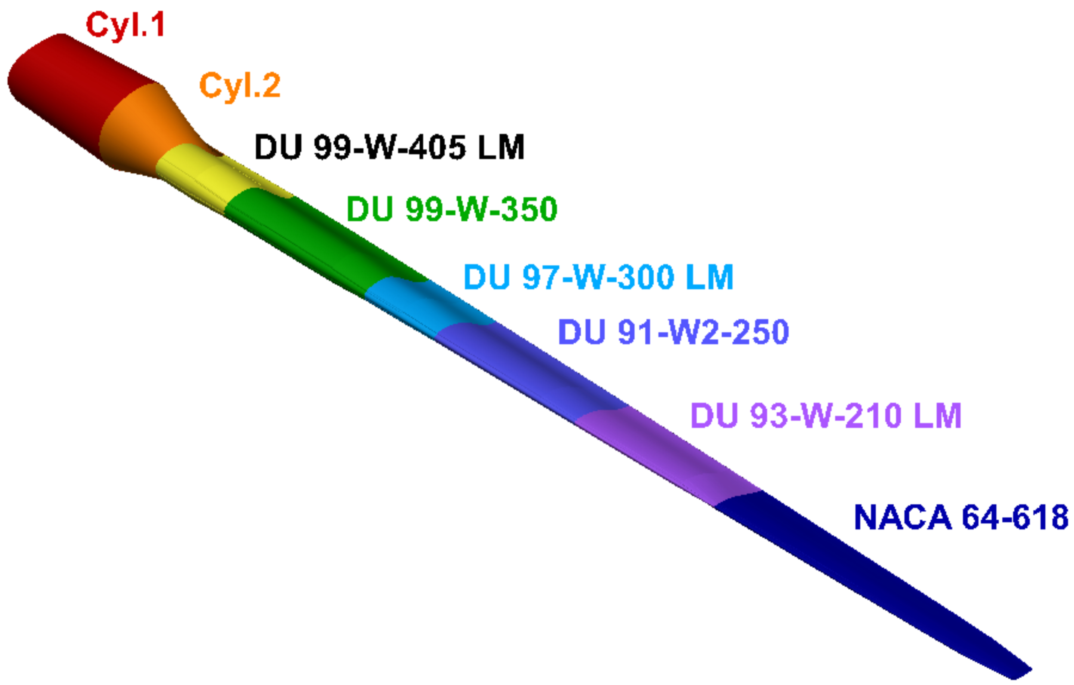


Figure 8.2: Blade of the NREL 5 MW turbine.

be better captured by a time-dependent simulation. In the presented work, all the simulations were steady and in some cases such as this one, for the high angles of attack, the solution kept oscillating between a small range of values. The situation is different in the case of the polar curve (see the right picture of figure 8.4). Here the differences between the coarse and medium grids are clearly noticeable, while the medium and fine grids give almost the same result. The solution obtained with the medium grid can therefore be considered mesh independent, and this grid has been used for all the simulations of this section.

8.1.1.2 Validation against the experimental results

To validate the adopted approach, the values of the lift and drag coefficients obtained with the CFD simulations have been compared to the experimental data. They are wind tunnel measurements obtained for angles of attack between $-10°$ and $20°$ at a Reynolds number $Re = 6 \times 10^6$. These data can be found on the NREL's National Wind Technology Center forum, where they have been posted by Jason Jonkman [3]. The CFD simulations have been run with the incompressible 2D solver of ANSYS FLUENT for the same angles of attack, between $-10°$ and $20°$, with a step of $2°$. The adopted Reynolds number, based on the freestream velocity, the density of air, the dynamic viscosity, and the reference length, is the same used for the simulations. Since the chosen value for the freestream velocity is 10 m/s, the density of air is 1.225 kg/m^3), the dynamic viscosity of air 1.81×10^{-5} $kg/(m \cdot s)$, and the reference length is 1 m, this would lead to a Reynolds number $Re = 6.77 \times 10^5$. To obtain the correct Reynolds, the adopted approach is

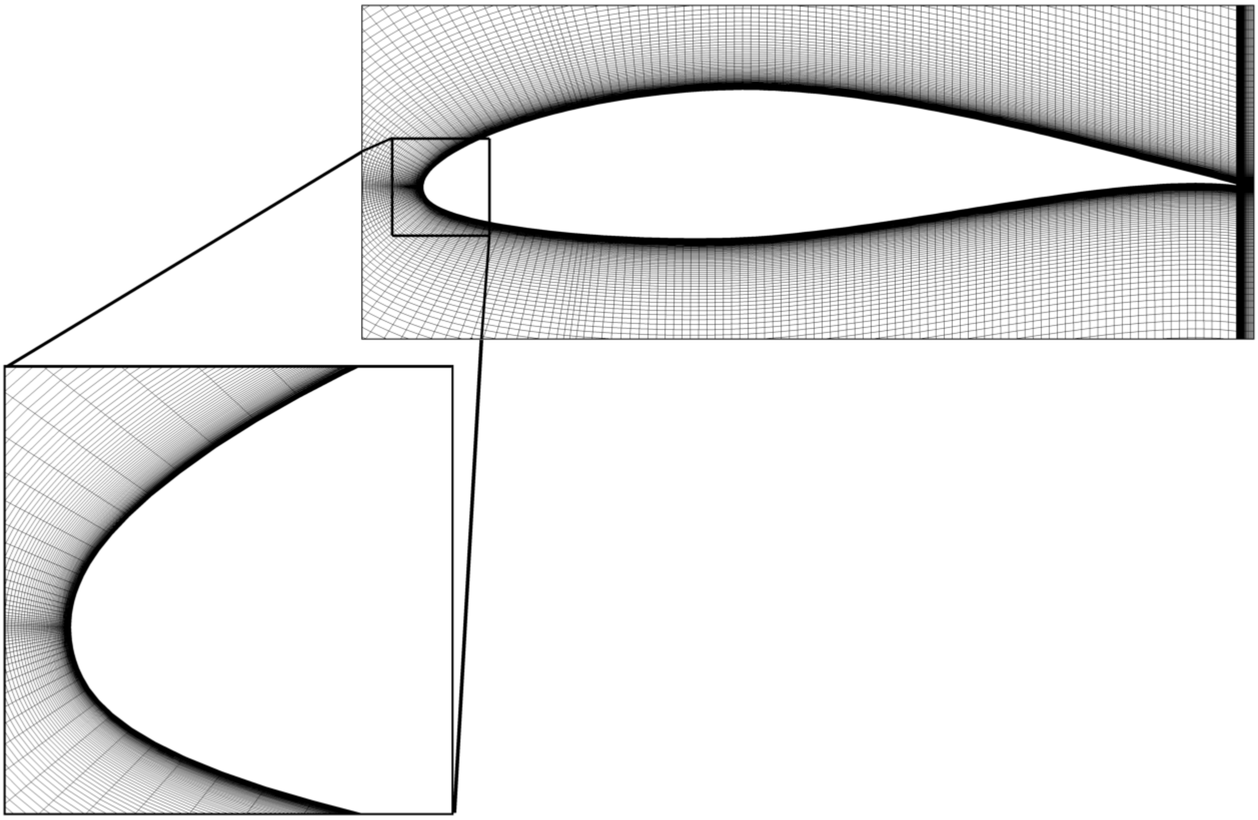


Figure 8.3: Medium grid around the NACA 64-618 airfoil, with a close-up view of the leading edge.

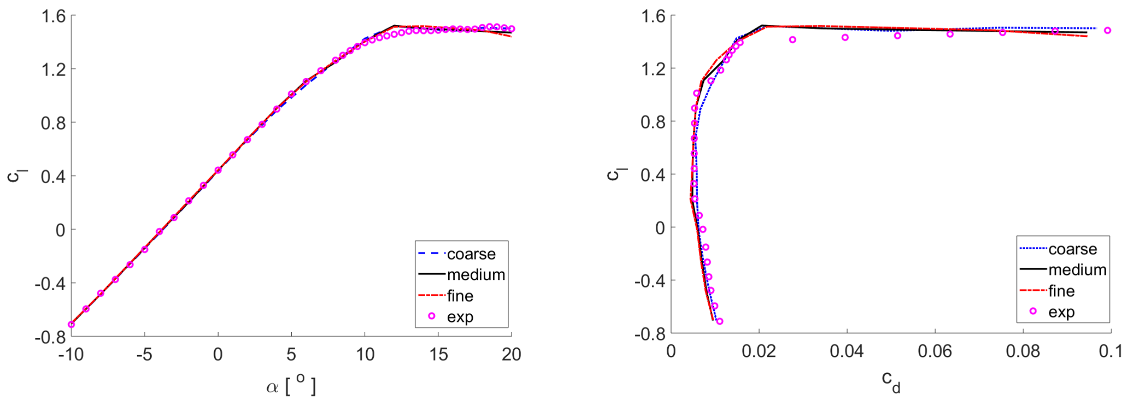


Figure 8.4: Results of the mesh refinement study for the NACA 64-618 airfoil.

to use a modified dynamic viscosity of air, namely $\mu = 2.042 \times 10^{-6}\ kg/(m \cdot s)$. The simulations are run with the transitional SST turbulence model, with a modified a_1 coefficient $a_1 = 0.29$, as suggested by Zanon *et al.* in [33]. The scheme used is the coupled, and the spatial discretization is second order upwind. The solution is considered to be converged with a residual drop of six orders , or when the maximum number of iterations (4000) has been reached. The comparison between the CFD and experimental results can be seen in figure 8.4, and it is characterized by an excellent agreement.

8.1.2 DU 93-W-210 LM airfoil

The same type of validation described in the previous section has been carried out for the DU 93-W-210 LM airfoil. The experimental data have been found in [3], and they are based in this case on a Reynolds number $Re = 7 \times 10^6$. To obtain the same Reynolds number for the simulations, the dynamic viscosity of air has been changed in this case to $1.75 \times 10^{-6}\ kg/(m \cdot s)$. The adopted CFD set-up is the same used for the NACA 64-618, namely transitional SST with $a_1 = 0.29$ for the turbulence, pressure-velocity coupling and second order upwind spatial discretization. The convergence criteria is also the same used in the previous case. The set-up described here is adopted for all the DU airfoils, so it won't be repeated in the following sections. The results of the validation study for the DU 93-W-210 LM are shown in figure 8.5: the agreement is excellent in the linear part of the lift curve, while some differences can be noticed for angles of attack higher than 10°, where the CFD simulations tend to over-predict the experimental data. This is due to the fact that for high angles of attack stall may occur, and CFD simulations are not able to accurately capture the real aerodynamics of the airfoil. Nevertheless, the agreement between CFD and experimental data is good and the results are comparable to the ones found in literature (see, for example, [33]). Moreover, the high angles of attack are not experienced by this airfoil, which is located in the middle part of the blade, due to the high blade pitch angle.

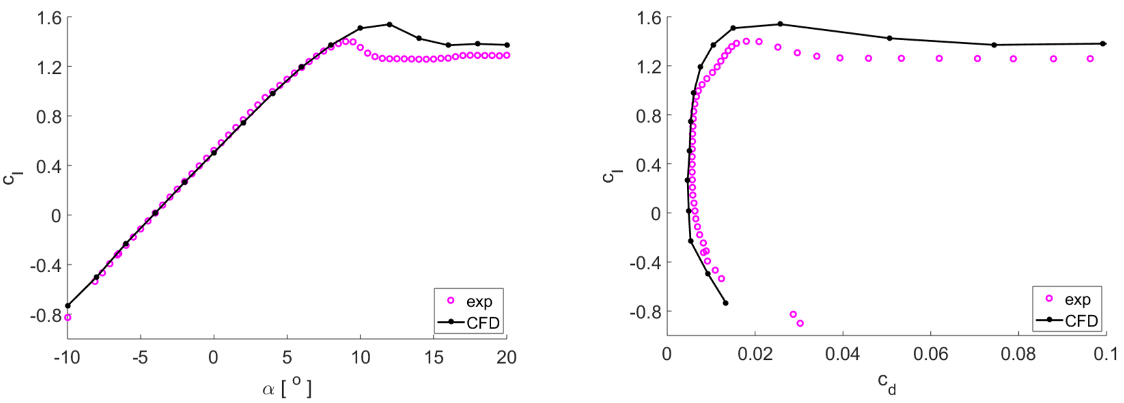


Figure 8.5: Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 93-W-210 LM airfoil, CFD results versus experimental data.

8.1.3 DU 91-W2-250 airfoil

The comparison between the CFD simulations and the experimental data for the DU 91-W2-250 airfoil is shown in figure 8.6, and the same remarks of the previous section apply.

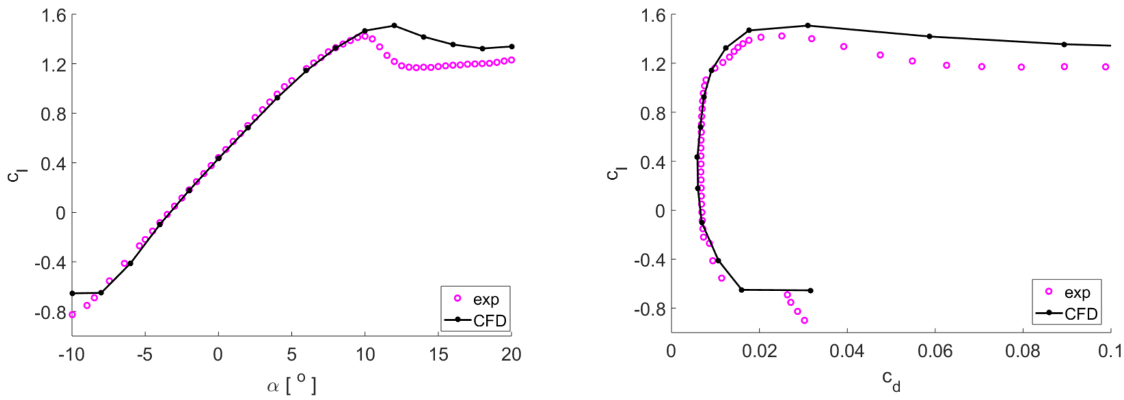


Figure 8.6: Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 91-W2-250 airfoil, CFD results versus experimental data.

8.1.4 DU 97-W-300 LM airfoil

The comparison between experimental and CFD data for the DU 97-W-300 LM airfoil can be seen in figure 8.7: a very good agreement has been obtained.

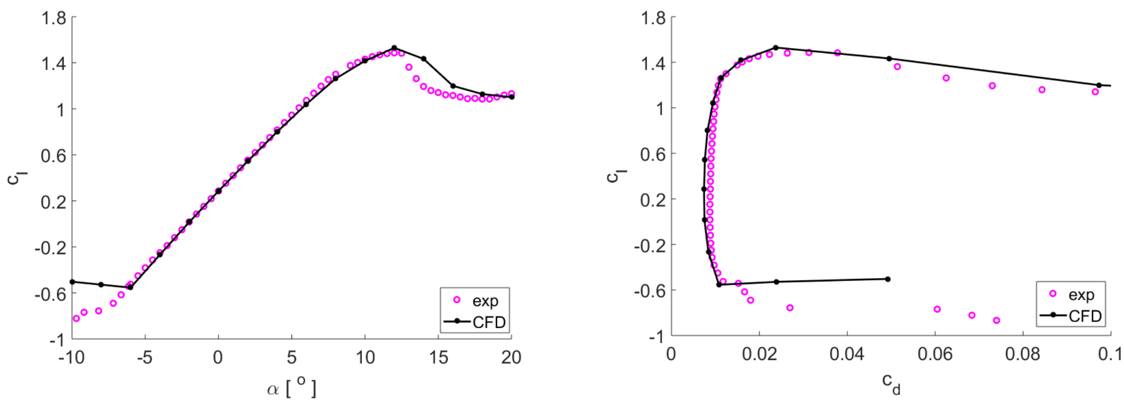


Figure 8.7: Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 97-W-300 LM airfoil, CFD results versus experimental data.

8.1.5 DU 99-W-350 airfoil

In the case of the DU 99-W-350 airfoil, the agreement between CFD and experimental data is worst than in the other cases, probably because this airfoil is thicker and the steady CFD simulations are not able to capture well its real aerodynamic behaviour, especially for very high and negative angles of attack (see figure 8.8). The results obtained with unsteady simulations could be different, but they have not been analysed within this work, whose purpose was to validate the ANN predictions rather than the CFD procedure. This airfoil, being close to the root of the blade, does not contribute substantially to the power production of the turbine, therefore it is not fundamental to obtain a perfect agreement. The improvement of this results will nonetheless be object of future work.

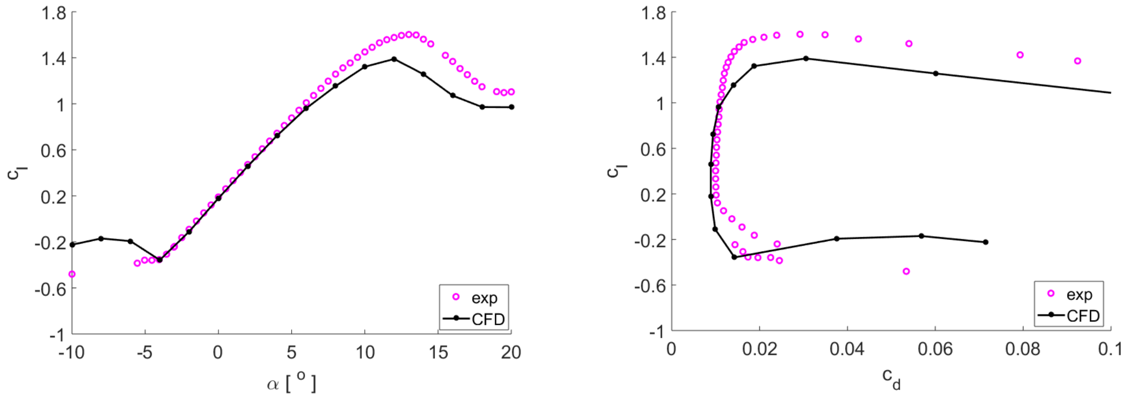


Figure 8.8: Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 99-W-350 airfoil, CFD results versus experimental data.

8.1.6 DU 99-W-405 LM airfoil

For the DU 99-W-405 LM airfoil, the agreement between CFD and experimental data is not perfect, especially for the drag coefficient, which is over-predicted by the CFD simulations (see figure 8.9). The reasons for that are the thicker shape of the airfoil and the difficulties in the prediction of the stall, which could be better captured by unsteady simulations. The improvement of these results will be object of future work. Once again, the airfoil is close to the root of the blade, almost not contributing to the power production.

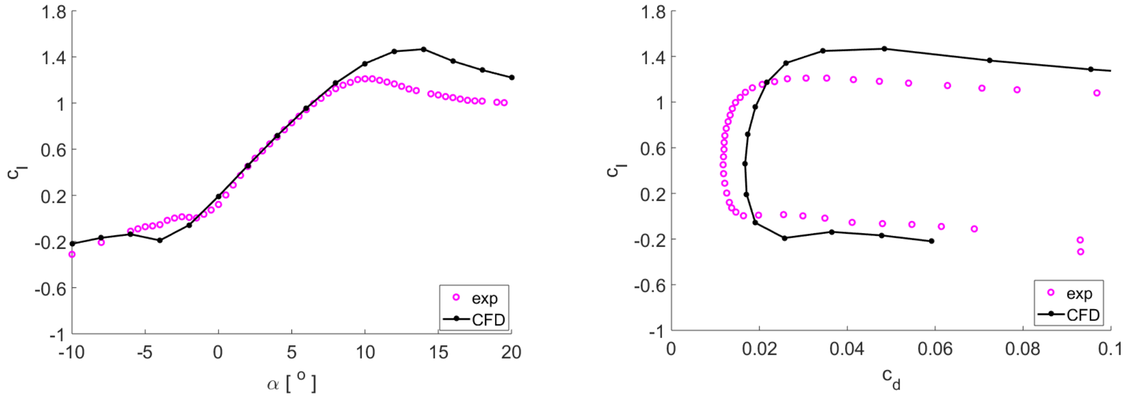


Figure 8.9: Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 99-W-405 LM airfoil, CFD results versus experimental data.

8.2 Comparison between the power curves and loads obtained with CFD results and experimental data

The final step of the validation for the NREL 5 MW turbine consists in the comparison of the power curve obtained using experimental data with that obtained using CFD data. Since some differences are noticeable between the experimental lift and drag curves and the computational ones, it is important to make sure that the obtained power curve and the loads are the same. For this purpose, the BEMT code NREL FAST has been run first with the experimental lift and drag curves, and then using the CFD results. The comparison is shown in figure 8.10: the agreement

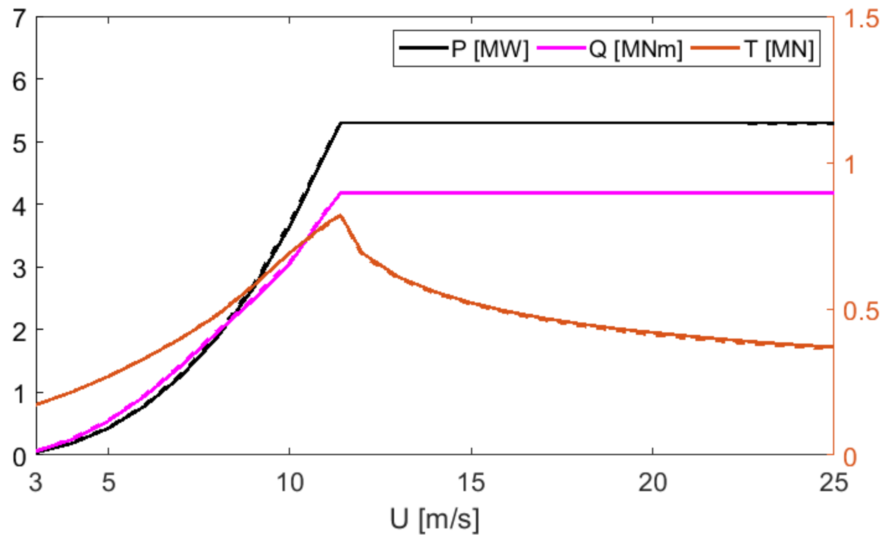


Figure 8.10: Power (P), torque (Q) and thrust (T) against wind speed (U). The solid line refers to the curves obtained with the experimental data, while the dashed line to those obtained using the CFD data.

is very good for all the quantities considered. Having established that the CFD results for the nominal turbine are reliable, it will be possible to use these data for all the subsequent analysis.

8.3 Nominal and damaged DU 96-W-180 airfoils

This section treats the validation of the clean and damaged DU 96-W-180 airfoil. This airfoil is not included in the NREL 5 MW model turbine used in this work, but it was chosen by Sareen *et al.* to carry out wind tunnel experiments on the impact of leading edge erosion on the airfoil performance (see [26]). This airfoil is usually situated in the middle part of the blade or close to the tip. In the work presented by Sareen *et al.*, a variety of photographs of turbines of different sizes, in operation from 1 to more than 10 years, have been analysed. They investigated the stages of the erosion process, identifying three major features: pits, gauges, and erosion. The damage was classified into five stages and three types. In this section, the results obtained with the method adopted in this work to reproduce the damages are compared to the results obtained by Sareen *et al.* in their experiments, in order to check if our method is reliable. First of all, the validation of the clean airfoil is considered, and a mesh refinement study is carried out to verify the independence of the solution on the grid. Then, the chosen damage is described in detail, and a mesh refinement study and validation are presented also for the damaged airfoil. Finally, the lift coefficient as a function of the angle of attack and the polar curve of clean and damaged airfoil are compared.

8.3.0.1 Nominal DU 96-W-180

Here, the validation of the clean DU 96-W-180 is considered. The results obtained with CFD simulations for the lift and drag coefficients are compared with the experimental data obtained by Sareen *et al.* in [26]. The experiments were carried out for three different Reynolds number: 1×10^6, 1.5×10^6 and 1.85×10^6. In this work, a Reynolds number of 1.5×10^6 was chosen for the validation.

Mesh refinement study First of all, a mesh refinement study has been carried out. The parameters chosen are the same already used for the mesh refinement of the NACA 64-618 airfoil, listed in table 8.1. The only difference is in the value for the wall distance, since the Reynolds number used in this case is different. This means that the distribution of the quadrilateral elements in the grid of the DU 96-W-180 will be different from that of the grid for the NACA 64-618. The wall distance adopted in the coarse grid is $3.2 \times 10^{-5}c$, for the medium grid $1.6 \times 10^{-5}c$ and for the fine grid $8 \times 10^{-6}c$. These values guarantee a maximum y^+ of 2 for the coarse grid, 1 for the medium grid and 0.5 for the fine grid. A picture of the medium grid with a close-up view of the trailing edge can be seen in figure 8.11. The lift coefficient as a function of the angle of attack and the polar curve obtained with the three grids are shown in figure 8.12. It can be noticed that the solutions obtained with the medium and fine grid are almost superimposed, while the result given by the coarse grid is very different. It was not possible to obtain a converged solution with the fine grid for the 20° angle of attack because the lift and drag values kept oscillating in a small interval. The results given by the coarse grid are closer to the experimental data, but they are

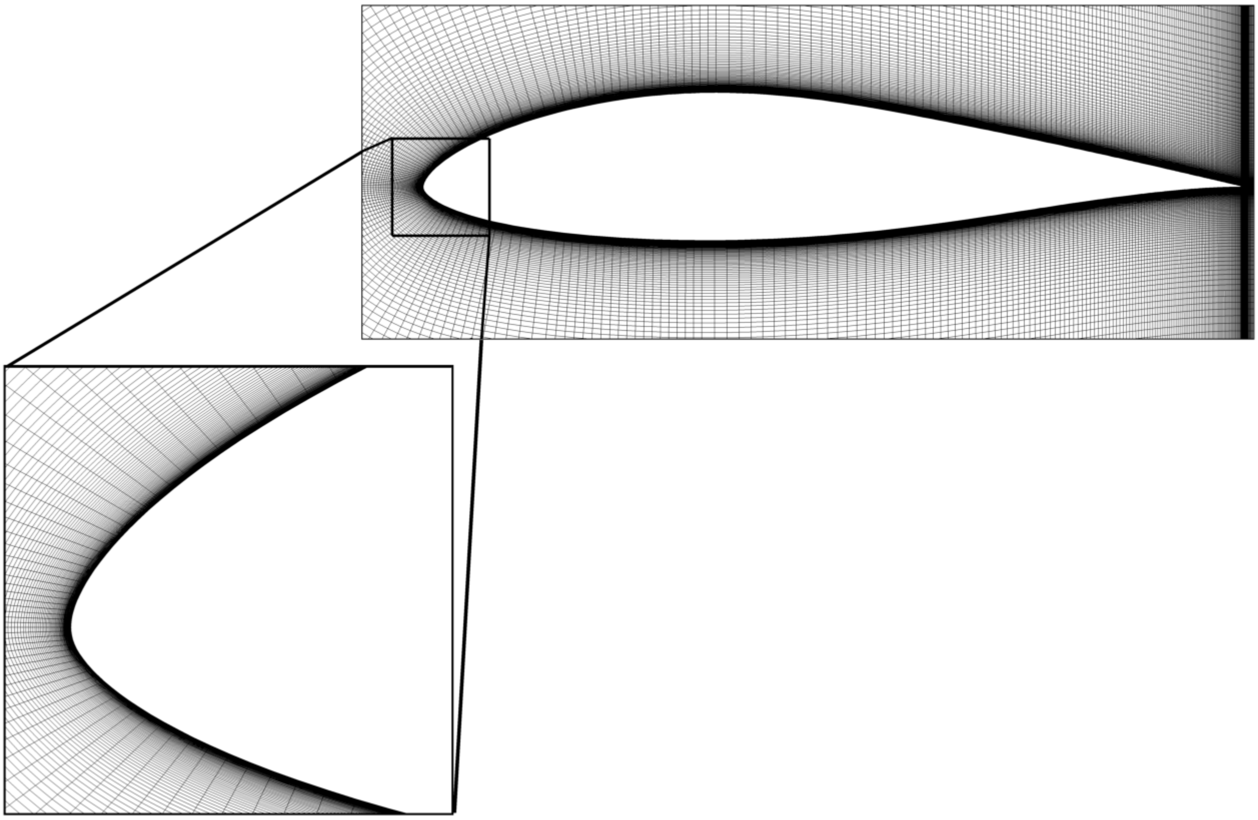


Figure 8.11: Mesh for the DU 96-W-180 airfoil with a close-up view of the leading edge.

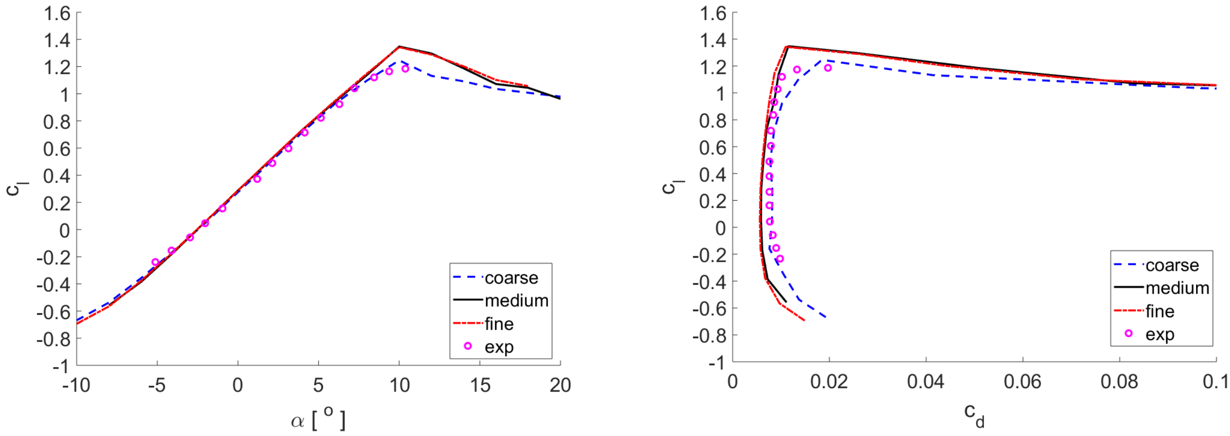


Figure 8.12: Results of the mesh refinement study for the nominal DU 96-W-180 airfoil.

not mesh independent and therefore not reliable.

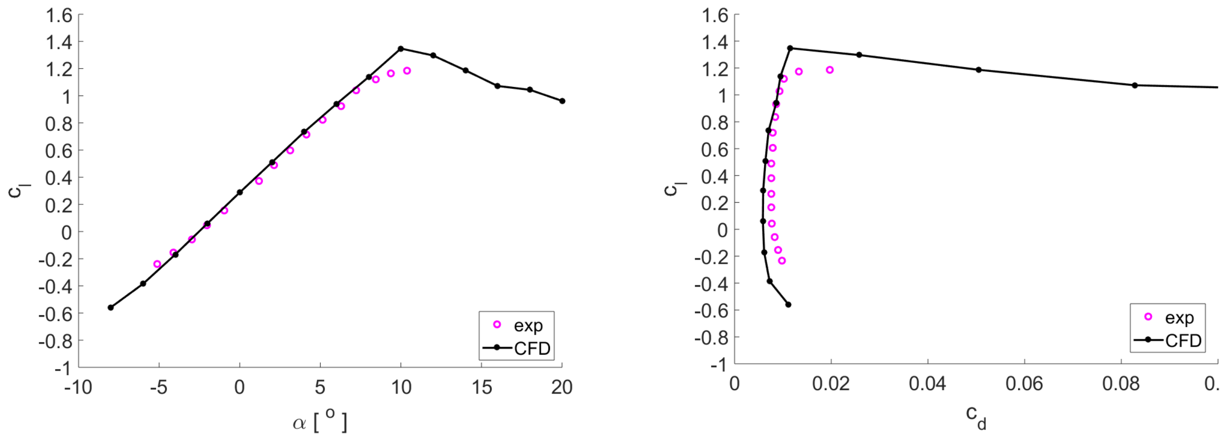


Figure 8.13: Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the DU 96-W-180 airfoil, CFD against experimental data.

Validation against the experimental results The results obtained with the medium grid have been compared to the experimental data. The CFD simulations were run using ANSYS FLUENT version 19.0. The turbulence model used is the transitional SST, with a modified a_1 coefficient of 0.29. The Reynolds number, based on the freestream velocity, the density of air, the dynamic viscosity of air and the reference length, is 1.5×10^6. The chosen freestream velocity is 10 m/s to guarantee incompressibility of the flow, the reference length is 1 m, the density of air 1.225 kg/m^3 and the dynamic viscosity has been modified to 8.17×10^{-6} $kg/(m \cdot s)$ to obtained the desired Reynolds. The comparison between the CFD results obtained with the medium grid and experimental data is shown in figure 8.13. An overall good agreement can be noticed, especially in the linear part of the lift coefficient. For high angles of attack the CFD simulations tend to overpredict both the lift and drag coefficients, but the results are in good agreement with other works found in the literature (see [27]).

8.3.0.2 Damaged DU 96-W-180

Here we consider the validation of the damaged DU 96-W-180 airfoil. The results of the CFD simulations are validated against the experimental results obtained by Sareen *et al.* in [26]. The Reynolds number chosen for the simulations is the same used for the validation of the nominal DU 96-W-180, namely 1.5×10^6. This section is organized in the following way: first of all, a detailed description of the selected damage is presented, then a mesh refinement study is performed, and finally the obtained results are validated against the experimental data, and compared to the results obtained for the nominal airfoil.

Damage definition The damage considered here for the validation is that labelled in [26] as Stage 5 Type C, characterized by 1600 pits, 800 gauges and the highest level of delamination. Our method does not enable us to consider damages such as pits and gauges, for which a 3D approach would be needed, so we considered only the delamination damage. Denoting by c the chord of the airfoil, the highest level of delamination in the work of Sareen *et al.* is characterized by $s_u = 3\%c$, $s_l = 3.9\%c$, and $d = 3.81\ mm$ (see figure 8.14). The chord of the airfoil used for

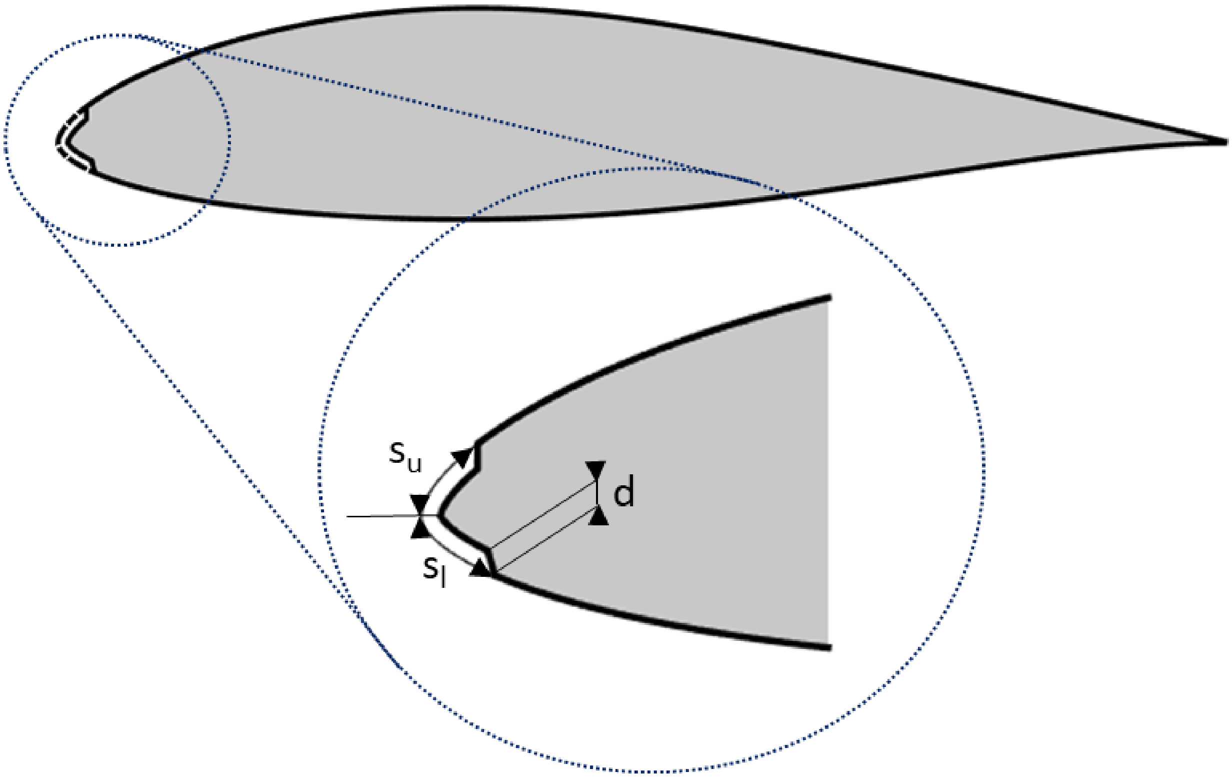


Figure 8.14: Damage chosen for the validation of the DU damaged 96-W-180 airfoil.

the experiments is 0.457 m, while the chord of the airfoil used for the simulations is 1 m. For this reason, the depth of the damage has been scaled using the proportion

$$d_{exp} : c_{exp} = d_{CFD} : c_{CFD} \tag{8.1}$$

which gives:

$$d_{CFD} = \frac{d_{exp} c_{CFD}}{c_{exp}} = 8.34mm \tag{8.2}$$

This is not exactly the same damage considered in the experiments, so we cannot expect to obtain the exact same results. The aim of this test is that to see if the modelling technique used in this work is able to capture the real behaviour of a damaged airfoil. The goal is that to obtain similar results to the experimental ones, having considered a similar damage.

Mesh refinement study First of all, a mesh refinement study has been performed also for the damaged airfoil. As for the other cases, three meshes have been considered: a coarse grid, a medium grid, and a fine grid. If the results obtained with the medium and fine grid are close enough, the solution obtained with the medium grid id considered to be mesh independent and thus reliable. The coarse grid is characterized by 306 elements around the airfoil, 75 along the

Grid	El. on the airfoil	El. along the C-cut	El. normal dir.	Total elements	Wall distance
Coarse	306	75	75	30 000	3.2×10^{-5}
Medium	612	150	150	126 000	1.6×10^{-5}
Fine	1224	250	250	400 000	8×10^{-6}

Table 8.2: Parameters for the mesh refinement study for the DU 96-W-180 damaged airfoil.

C-cut and 75 in the normal direction, for a total of 30 000 quadrilateral elements. The wall distance of $3.2 \times 10^{-5}c$ guarantees a maximum y^+ value of approximately 2. The medium grid has 612 elements on the airfoil, 150 along the C-cut and 150 in normal direction, and counts a total of 126 000 elements. The wall distance of $1.6 \times 10^{-5}c$ guarantees a maximum y^+ value of 1. Finally, the fine grid has 1224 elements on the airfoil, 250 along the C-cut and 250 in normal direction, for a total of 400 000 elements. In this case the wall distance is $8 \times 10^{-6}c$, and the maximum y^+ is about 0.5. The parameters for the three grids are collected in table 8.2 for better clarity. A picture of the medium grid for the DU 96-W-180 airfoil with a close-up view of the leading edge is shown in figure 8.15. The results of the mesh refinement stuy are shown in figure

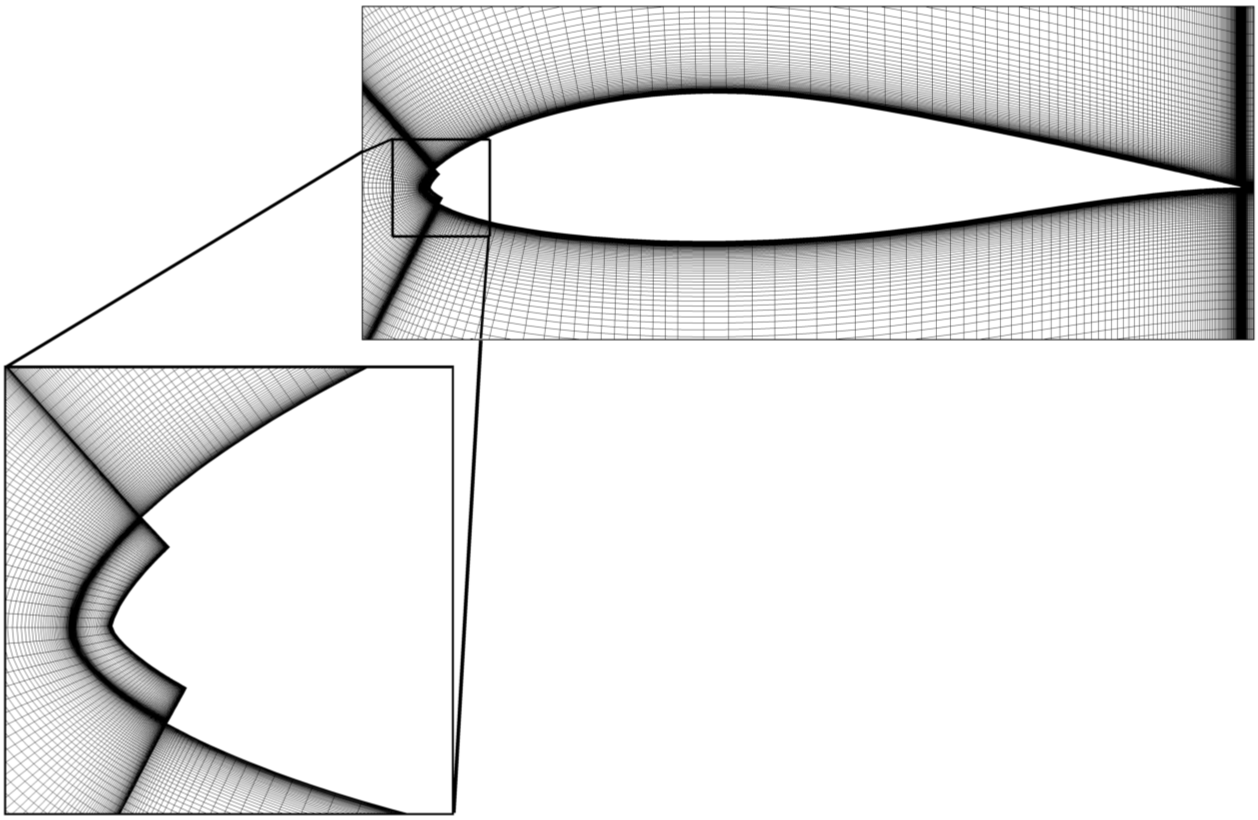


Figure 8.15: Mesh for the DU 96-W-180 damaged airfoil with a close-up view of the leading edge.

8.16: in this case there are not great differences between the three grids. Only for high angles of attack the coarse grid differs slightly from the medium and fine meshes. The medium grid is therefore considered suitable for the simulations.

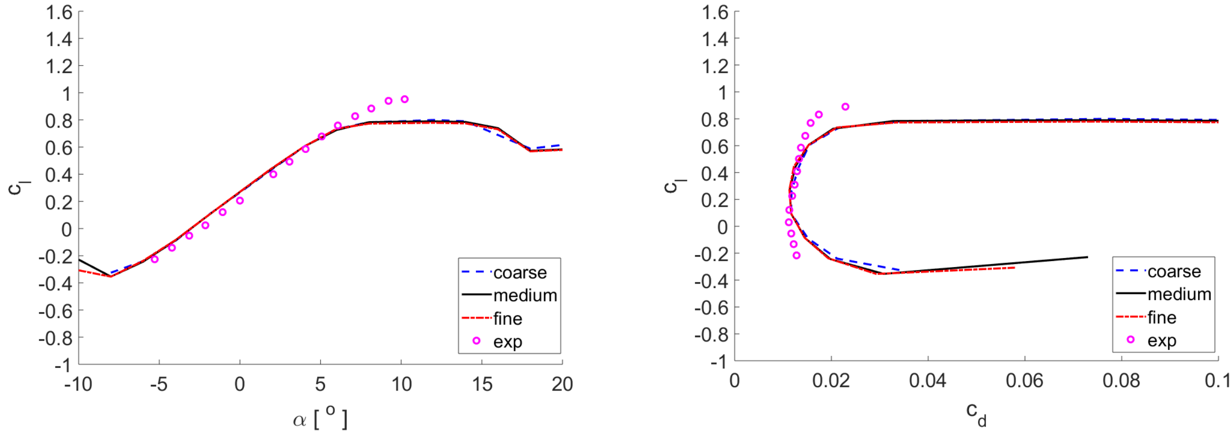


Figure 8.16: Results of the mesh refinement study for the damaged DU 96-W-180 airfoil.

Validation against the experimental results The CFD set-up used for the damaged airfoil is exactly the same already used for the nominal airfoil described in the previous section, so it will not be repeated here. The comparison of the results obtained with the CFD simulations using the medium grid and the experimental data is shown in figure 8.17. The agreement between

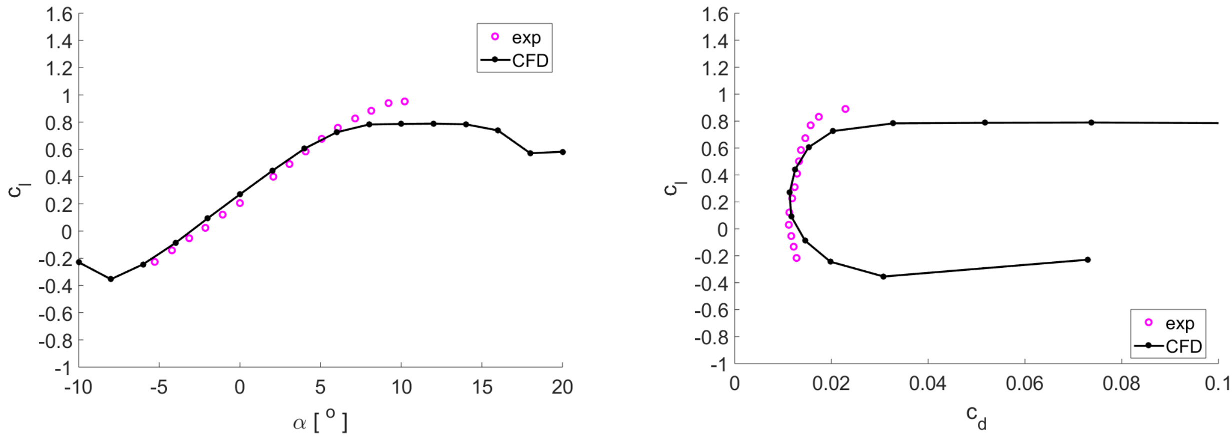


Figure 8.17: Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the damaged DU 96-W-180 airfoil, CFD against experimental data.

CFD and experimental data is good in the linear part of the lift coefficient, while the lift is under predicted for high angles of attack. As said before, this was expected because the two damages are not identical. The damaged airfoil used for the experiments was produced by a model maker following a pattern created with simulations by the authors of the article [26]. It is likely that the delamination geometry was slightly different on the real airfoil compared to the one showed in picture 8.14. We must also take into account the fact that pits and gauges have not been considered here. Considering all the factors, we can conclude that the results are satisfying and the technique used is able to reproduce a real damage.

Comparison between the nominal and damaged airfoils Finally, it is interesting to compare the results obtained for the nominal and for the damaged airfoils, both with CFD and experiments. From picture 8.18, it can be noticed that the lift coefficient decreases for the damaged

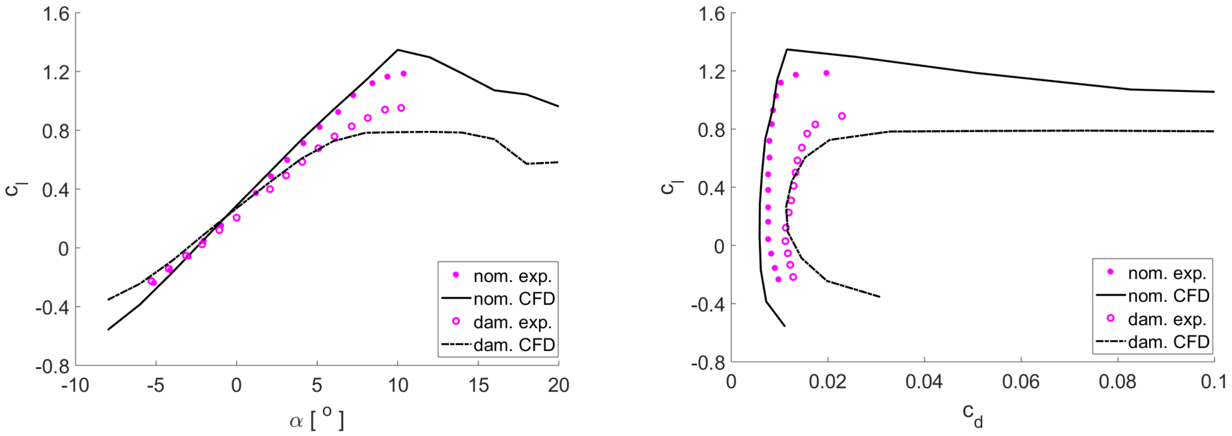


Figure 8.18: Lift coefficient as a function of the angle of attack (left) and polar curve (right) for the nominal and damaged DU 96-W-180 airfoil, CFD against experimental data.

airfoil, while the drag coefficient increases as expected. The CFD simulations tend to over-predict the lift of the nominal airfoil and to under-predict that of the damaged one, but they are able to capture the change of relative sign between the lift of the nominal and damaged airfoils. Before $-1°$, in fact, the lift coefficient of the damaged airfoil is higher that than of the nominal airfoil, while this situation reverses for angles higher than $1°$. From the picture it can be noticed that the same happens for the data obtained with the CFD simulations. To obtain a correct estimate of the power loss associated to a particular damage, it is important that the differences between the lift and drag coefficients of the nominal and damaged airfoil are well captured. Figure 8.18 highlights that this is the case, at least for the angles of the attack between $-1°$ and $8°$, which are the most frequently observed for an airfoil close to the tip of the blade, such as the DU 96-W-180.

Chapter 9

Results

In this chapter the results obtained are described. First of all, the obtained database is presented. Then, a validation of the ANN approach is carried out in order to verify the reliability of the implemented system. Finally, a damaged turbine is analysed in great detail.

9.1 Database generation

The first outcome of the project is the creation of a database of damaged airfoils. For each one of the six nominal geometries composing the NREL 5MW turbine, a total of 1007 delaminated geometries were created and analysed. For each damaged airfoil, 16 FLUENT simulations needed to be run, leading to a total of 96672 simulations, 16112 for each type of airfoil. The entire process required about two weeks, and, as expected, not all the calculations were successful. The failed simulations could be effectively detected thanks to the MATLAB program described in paragraph 7.2.2.4. If the failure was caused by problems in the grid, the set-up has been fixed and the simulations re-run. If the failure was instead due to other reasons, such as oscillatory results or convergence problems, the results have been neglected. The number of successful simulations for each model of airfoil and their percentage of the total are reported in table 9.1. The airfoils characterized by the greatest percentage of fails are the NACA 64-618 and the DU 93-W-210 LM. It has been observed that the thin shape of these two airfoils led to instabilities in the simulations, especially for high angles of attack.

A deeper insight of the failures distribution is provided in table 9.2 for the NACA airfoils

Airfoil	Successful simulations	Percentage of the total
NACA 64-618	14744	92%
DU 93-W-210 LM	14734	91%
DU 91-W2-250	15578	97%
DU 97-W-300 LM	15390	96%
DU 99-W-350	15623	97%
DU 99-W-405 LM	15494	96%

Table 9.1: Successful simulations for each model of airfoil in the generation of the database.

NACA 64-618

AoA	**Failures**
$-10°$	14%
$-8°$	10%
-6°	4%
-4°	4%
-2°	4%
0°	4%
2°	5%
4°	5%
6°	5%
8°	6%
10°	6%
12°	12%
14°	15%
16°	15%
18°	14%
20°	11%

Table 9.2: Percentage of failed CFD simulations for each considered angle of attack for the NACA 64-618 airfoil.

and 9.3 for the DU airfoils. In these tables, for each airfoil, the percentage of failed simulations is displayed as a function of the angle of attack. In the case of the NACA 64-618 and the DU 93-W-210 LM, an high percentage of failures is observed for angles of attack lower than 8° and higher than 10°. Due to the shape of the airfoils, the simulations often gave oscillatory results. As these two models of airfoil are situated towards the tip of the blade, the operational angles of attack always fall into the interval $[-6°, 10°]$, where the number of successful simulations is very high. In the future developments of the work, two options could be considered:

- the simulations could be run for these two airfoils only for the operational angles of attack, saving computational time;
- for the very high and very low angles of attack the steady simulations could be replaced by time-dependent ones.

The other airfoils (DU 91-W2-250, DU 97-W-300 LM, DU 99-350, DU 99-W-405 LM) work instead in a wider interval of angles of attack. From the respective tables it can be noticed that the percentage of failures is very low, apart from the last two angles of attack. Once again, these two angles could be discarded, or time-dependent simulations could be considered.

DU 93-W-210 LM

AoA	Failures
-10°	11%
-8°	1%
-6°	1%
-4°	2%
-2°	3%
0°	3%
2°	4%
4°	4%
6°	4%
8°	5%
10°	5%
12°	11%
14°	18%
16°	23%
18°	23%
20°	19%

DU 91-W2-250

AoA	Failures
-10°	1%
-8°	1%
-6°	1%
-4°	1%
-2°	2%
0°	2%
2°	2%
4°	2%
6°	3%
8°	3%
10°	3%
12°	3%
14°	3%
16°	5%
18°	9%
20°	12%

DU 97-W-300 LM

AoA	Failures
-10°	2%
-8°	2%
-6°	3%
-4°	3%
-2°	3%
0°	3%
2°	3%
4°	3%
6°	4%
8°	4%
10°	5%
12°	6%
14°	6%
16°	6%
18°	7%
20°	12%

DU 99-W-350

AoA	Failures
-10°	1%
-8°	2%
-6°	2%
-4°	2%
2°	2%
0°	2%
2°	2%
4°	2%
6°	2%
8°	3%
10°	4%
12°	5%
14°	4%
16°	4%
18°	3%
20°	8%

DU 99-W-405 LM

AoA	Failures
-10°	0.5%
-8°	1%
-6°	1%
-4°	2%
-2°	3%
0°	3%
2°	3%
4°	4%
6°	4%
8°	4%
10°	5%
12°	5%
14°	6%
16°	6%
18°	7%
20°	7%

Table 9.3: Percentage of failed CFD simulations for each considered angle of attack for the DU airfoils.

9.2 Validation of ANN approach

The database generated with the CFD simulations comprises a great number of different delamination damages, but not all of them. Learning from this limited amount of data, the machine learning approach allows to extend the database and to obtain the lift and drag curves associated to all the possible delaminated geometries. As explained in chapter 7, the data computed by means of CFD are divided in two sets: the training set and the test set. The purpose of the training set is that to create the ANN system, while the test set is needed for the validation of the implemented algorithm. Even is the system is learning from the CFD data, it does not mean that all of them will necessarily be approximated well. In fact, the implemented algorithm tries to find the best fit of the data while using the lowest number of neurons possible. This means that some, or many, of the data could be approximated poorly. To make sure that the fit is satisfactory, all the dataset obtained with the CFD simulations have been compared to that given by the machine learning approach. The damaged geometries have been divided into groups corresponding to the same baseline airfoil. For each damaged airfoil of each group, and for the angles of attack for which it was possible to obtain the lift and drag coefficients by means of CFD, the values of c_l and c_d have been computed also with the ANN system. Then, the root mean square of the approximation error between CFD and ANN has been obtained for each damaged airfoil of the database. The root mean square error is defined as:

$$RMSE_{c_l} = \sqrt{mean(c_l^{CFD} - c_l^{ANN})^2}, \qquad RMSE_{c_d} = \sqrt{mean(c_d^{CFD} - c_d^{ANN})^2}. \tag{9.1}$$

The results of the validation are shown in figures 9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8, 9.9, 9.10, 9.11 ans 9.12. For each baseline airfoil composing the NREL 5MW turbine, two histograms and two pie charts (one for the lift coefficient and one for the drag coefficient) are displayed. The histogram reports the number of damaged airfoils characterized by a RMSE error included in a defined interval, while the pie chart shows in a more compact way the same results. It can be noticed that, for all the airfoils, the agreement between CFD and ANN is very good, and the percentage of cases characterized by a RMSE higher than 2% for the lift and 0.2% for the drag is small. More difficulties in the generations of the ANN models were encountered with the thinner airfoils, such as the NACA 64-618, while an almost perfect agreement can be observed in the case of the thicker DU 99-W-405 LM.

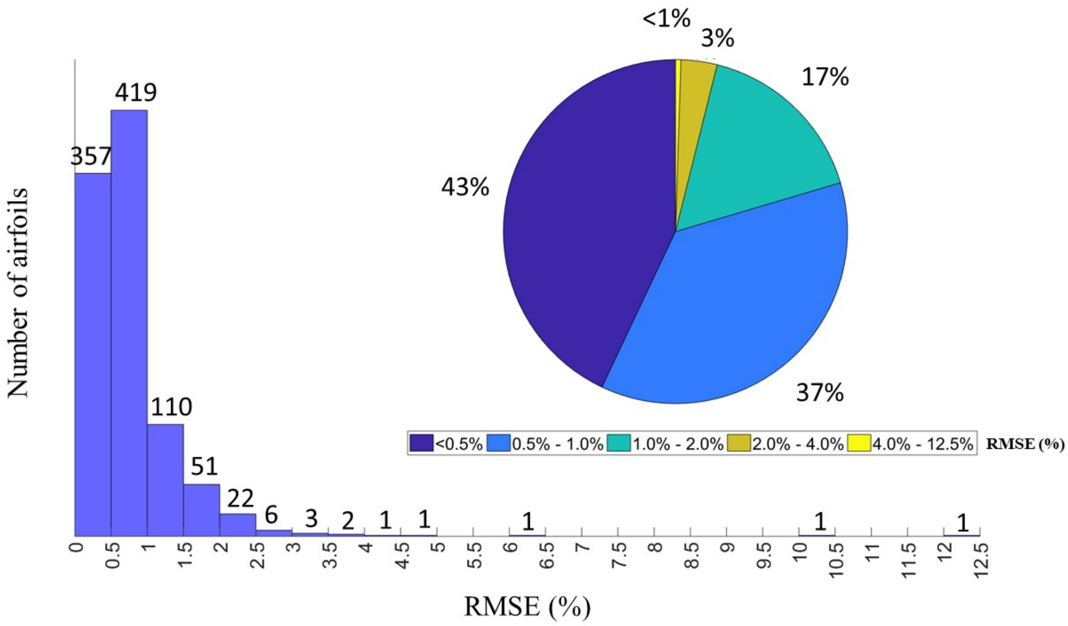


Figure 9.1: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the NACA 64-618 airfoil.

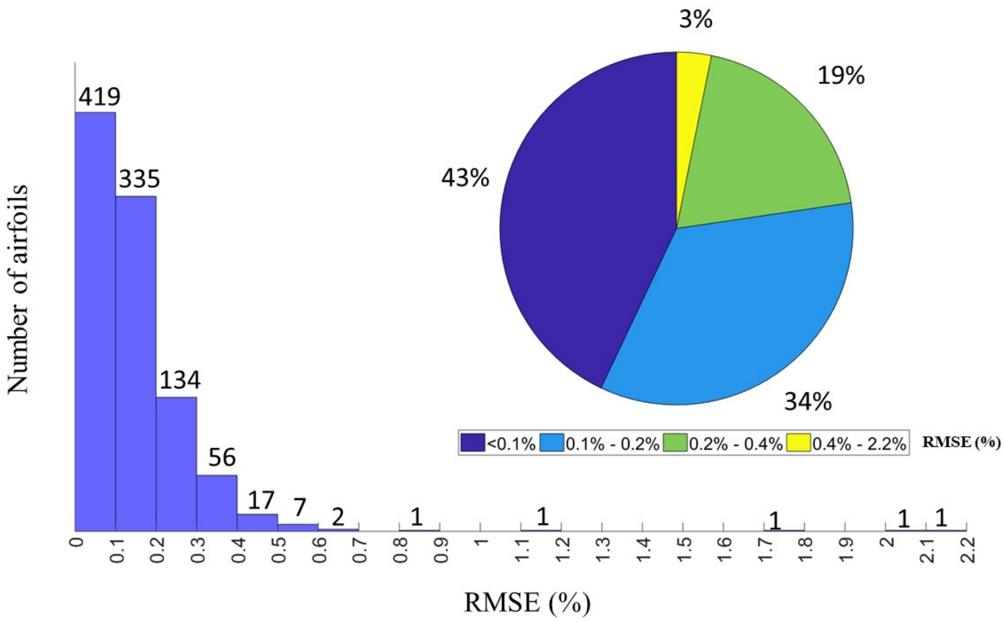


Figure 9.2: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the NACA 64-618 airfoil.

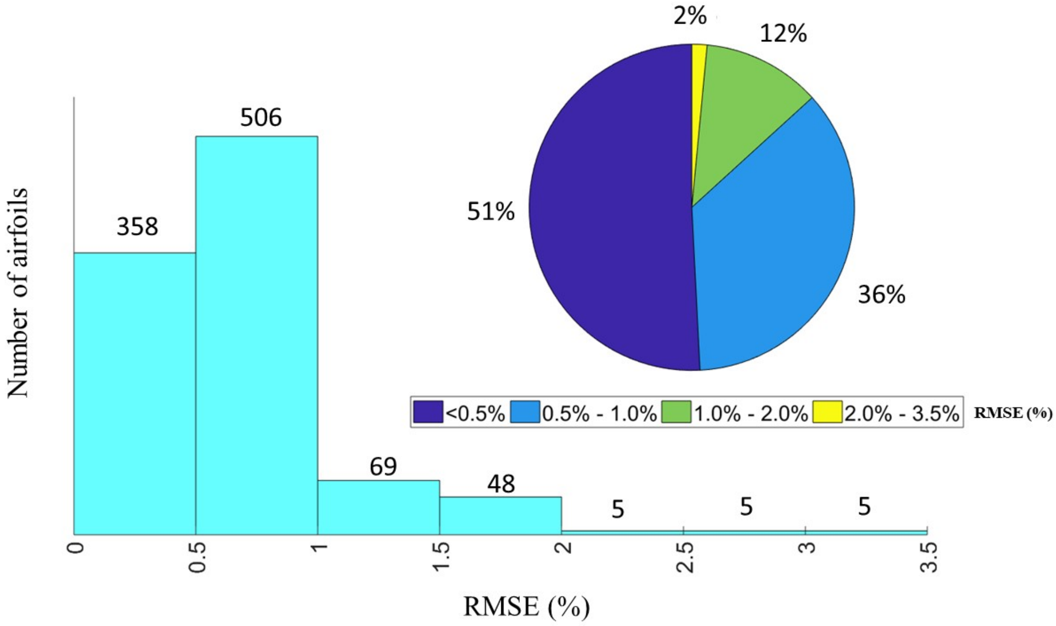


Figure 9.3: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 93-W-210 LM airfoil.

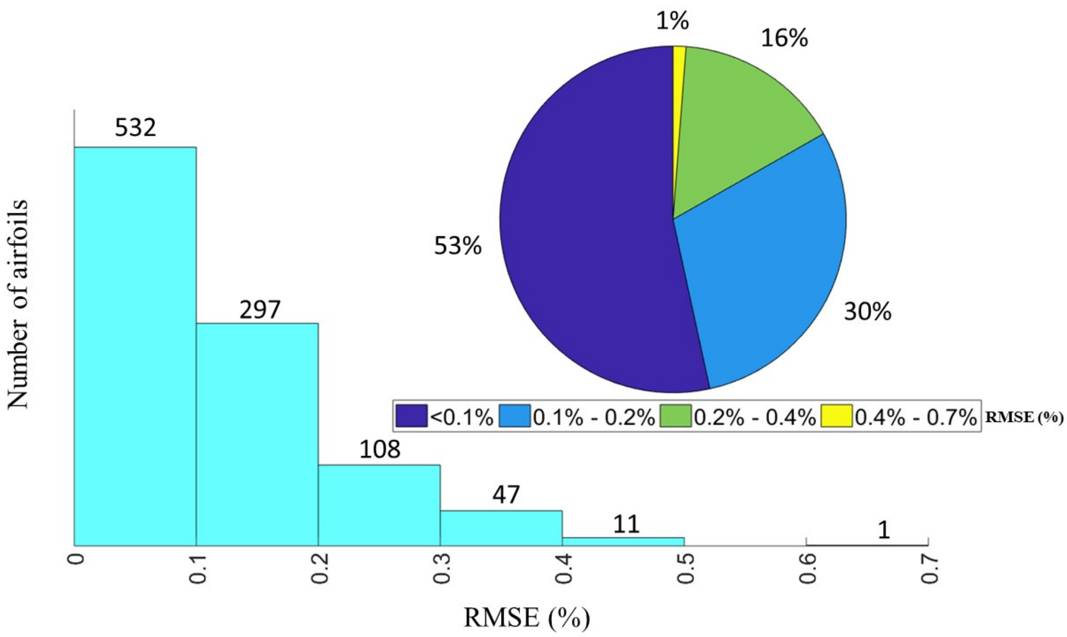


Figure 9.4: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 93-W-210 LM airfoil.

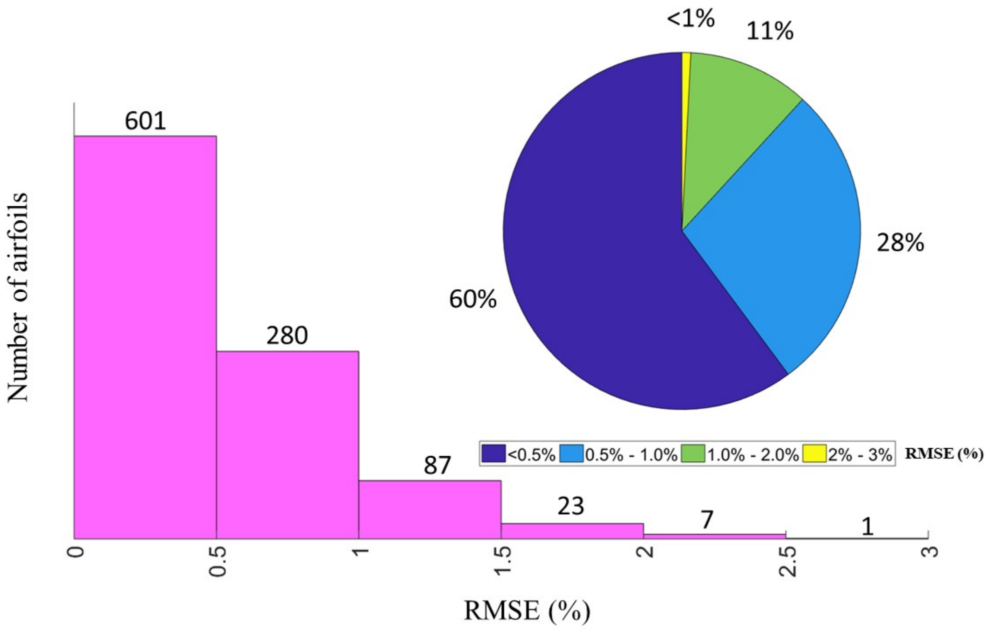


Figure 9.5: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 91-W2-250 airfoil.

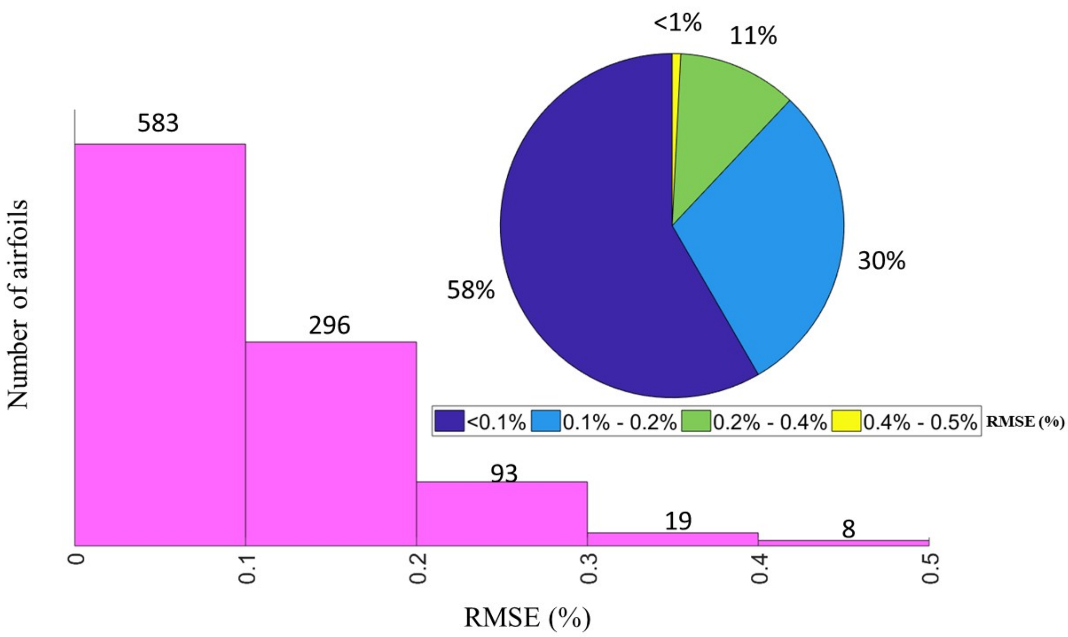


Figure 9.6: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 91-W2-250 airfoil.

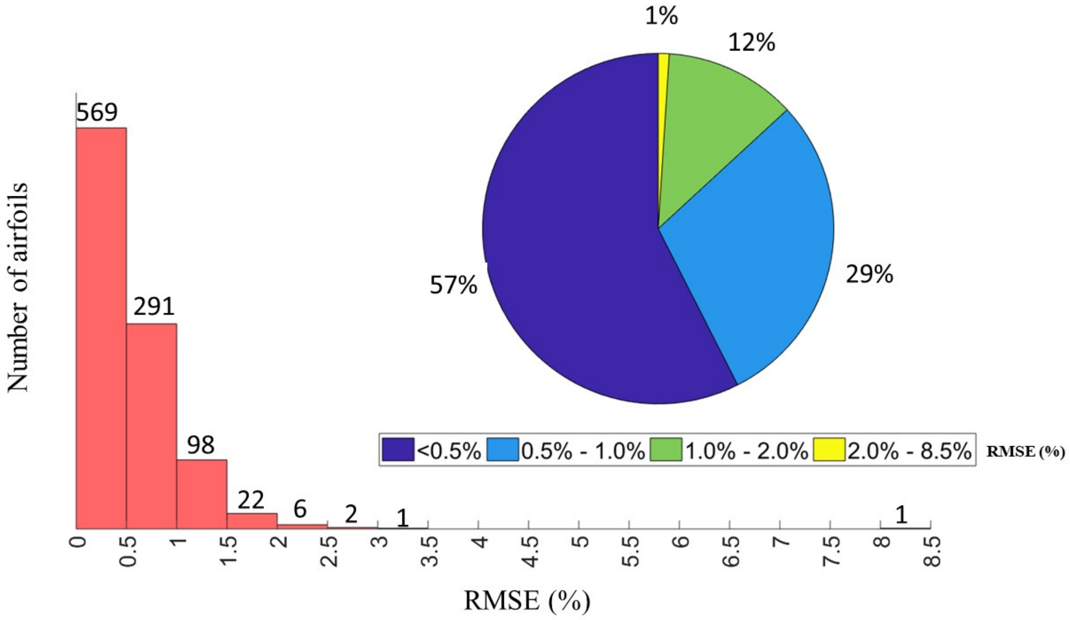


Figure 9.7: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 97-W-300 LM airfoil.

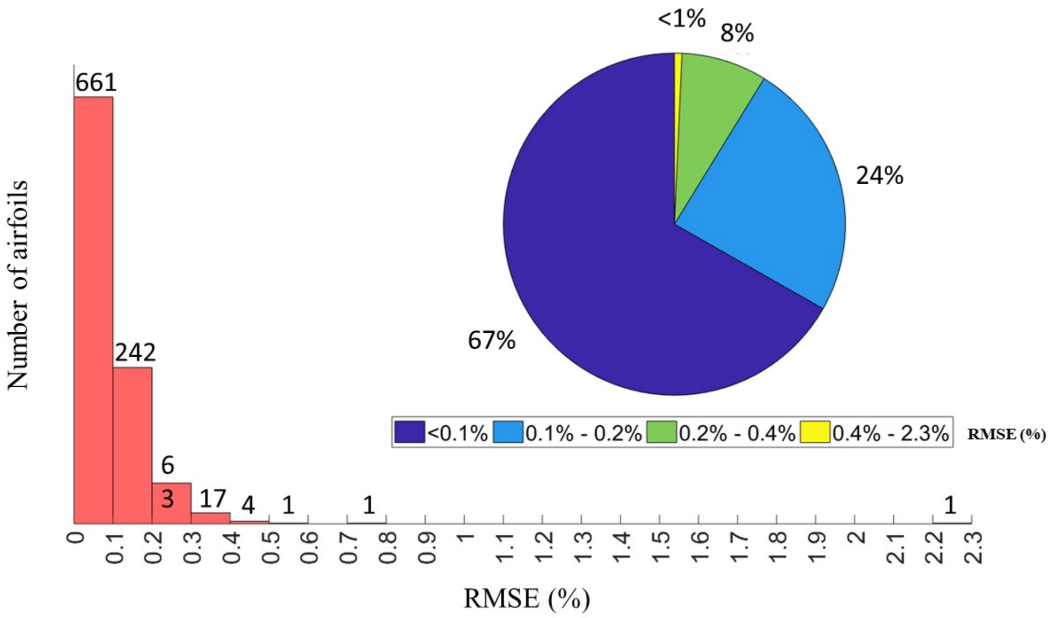


Figure 9.8: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 97-W-300 LM airfoil.

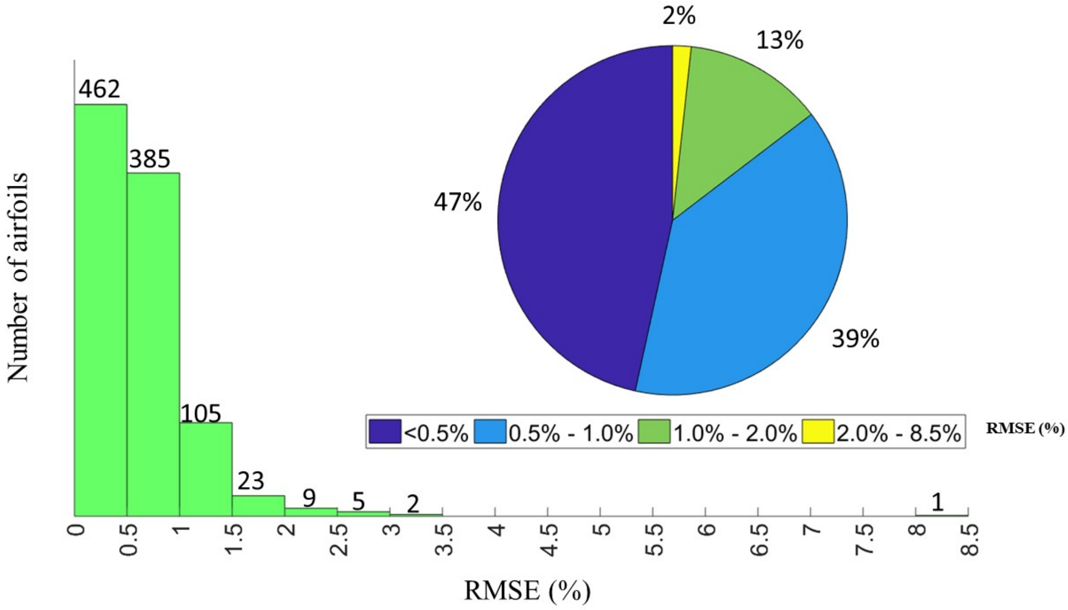


Figure 9.9: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 99-W-350 airfoil.

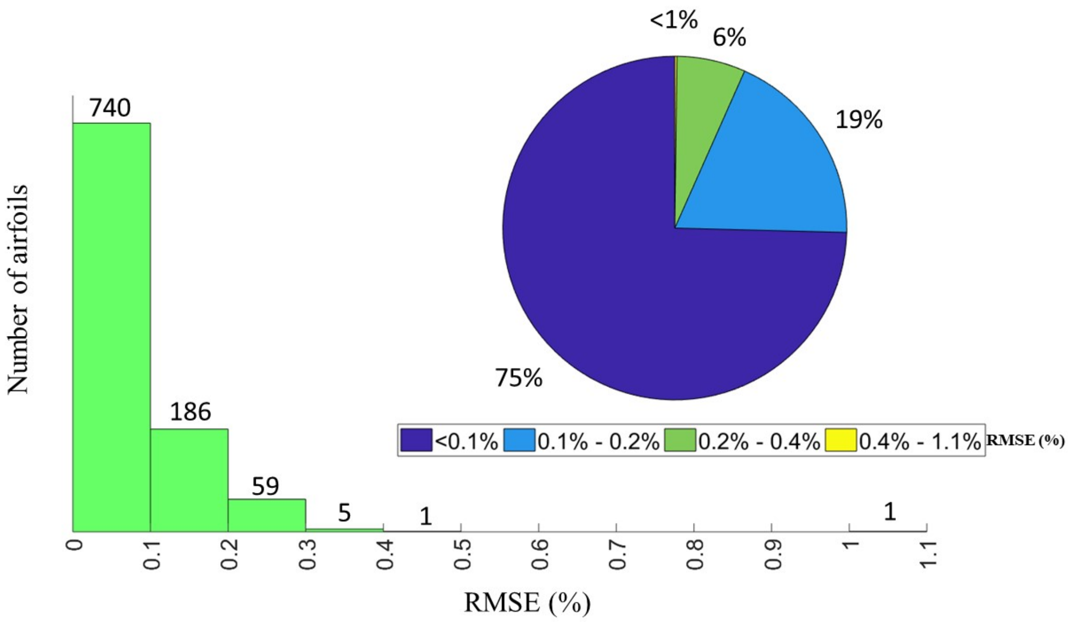


Figure 9.10: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 99-W-350 airfoil.

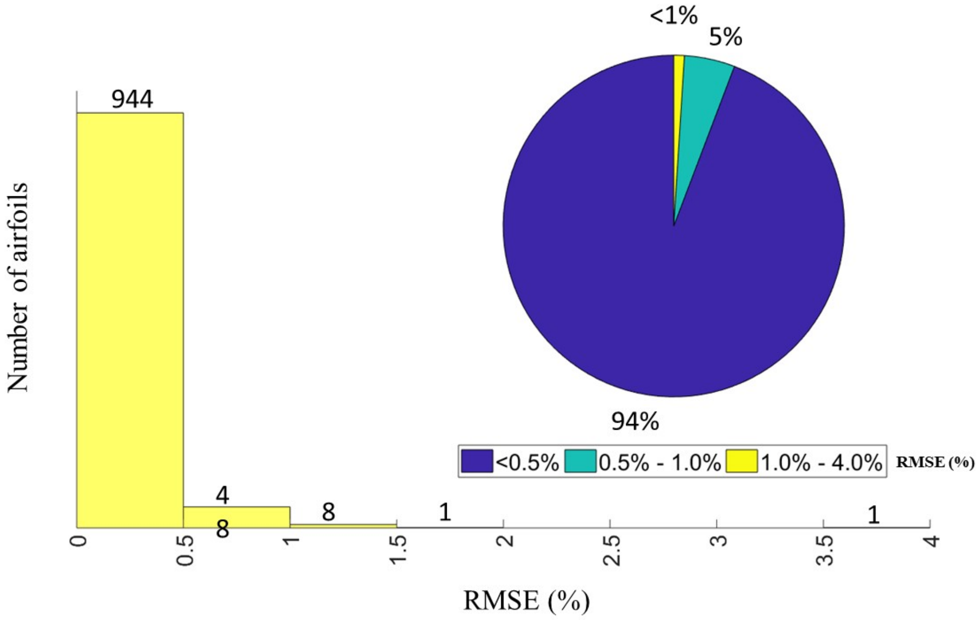


Figure 9.11: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the lift coefficient of the DU 99-W-405 LM airfoil.

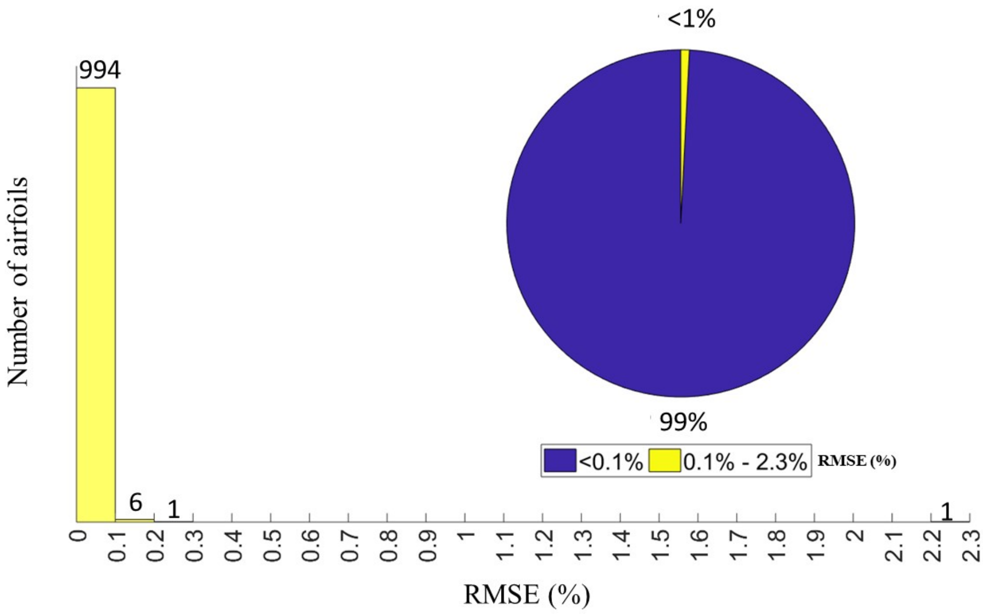


Figure 9.12: Histogram and pie chart illustrating the root mean square error between the solutions obtained with CFD and ANN for the drag coefficient of the DU 99-W-405 LM airfoil.

9.3 Damaged turbine, AEP loss and power control

To demonstrate the use of ALPS for industrial problems, a moderate to severe leading edge delamination affecting the blades of the NREL 5 MW reference turbine is considered. The damage is assumed to affect the entire length of the blade, and to have irregular edges (variable s_u and s_l) and variable depth d. It is also assumed that the geometry of the delamination damage is the same for all the three blades composing the turbine. Different delamination patterns on each blade, however, could be analyzed by means of the same approach adopted in [8] to analyze the impact on the mean and standard deviation of the turbine power of normally distributed deviations of the blade outer shape from the nominal geometry due to finite manufacturing tolerances. In that study, the performance of a three-blade rotor whose blades are not identical is obtained by taking the arithmetic average of the performance (power and loads) of three different fictitious rotors, each with identical blades affected by a different pattern of blade geometry errors.

The leading edge damage considered herein has different values of delamination parameters s_u/c, s_l/c and d/c over 25 radial segments or strips of the blade. The minimum, maximum and mean (μ) values and the standard deviation (σ) of the delamination parameters s_u, s_l and d of the damage analyzed below are reported in columns 2-4 of Tab. 9.4, in which all values are normalized by the airfoil chord c. A 3D view of the considered delamination pattern is provided in Fig. 9.13, in which US denotes the view of the blade upper side, and LS that of the blade lower side.

Table 9.4: Main geometric values of analyzed blade damage.

	min	max	μ	σ
$s_u/c \times 100$	2.71	6.4	4.44	1.02
$s_l/c \times 100$	2.76	7.86	5.63	1.24
$d/c \times 100$	0.4	0.97	0.61	0.2

To assess the reliability of the ANN system based on the ALPS database, the comparison of the lift and drag coefficient curves obtained using CFD and the selected ML approach at six radial positions is considered next. The value of the radius r of the 6 blade cross sections normalized by the rotor tip radius R is reported in the second row of Tab. 9.5, and the chord-normalized values of s_u, s_l and d of each section are reported below the corresponding value of r/R. The curves of lift coefficient c_l and drag coefficient c_d from $-10°$ to $20°$ of the six sections are depicted in the twelve subplots of Fig. 9.14. Each subplot provides 3 curves for the considered force coefficient, namely that of the delaminated section obtained using a CFD analysis (curve labeled CFD-d), that of the delaminated section obtained using the ANN system (curve labeled ANN-d), and that of the nominal section obtained using a CFD analysis (curve labeled CFD-n). With regard to the ANN prediction reliability, an overall excellent agreement between the CFD and ANN predictions of the force coefficients is observed. Some small discrepancies between the two predictions are observed only at the highest AoA values of the DU 97-W-300 LM and DU 99-W-350 airfoils. The most likely cause for this is the need for further optimization of the ANN set-up, which may require altering the number of ANN neurons and/or recalibrating the weights determining the amount of training

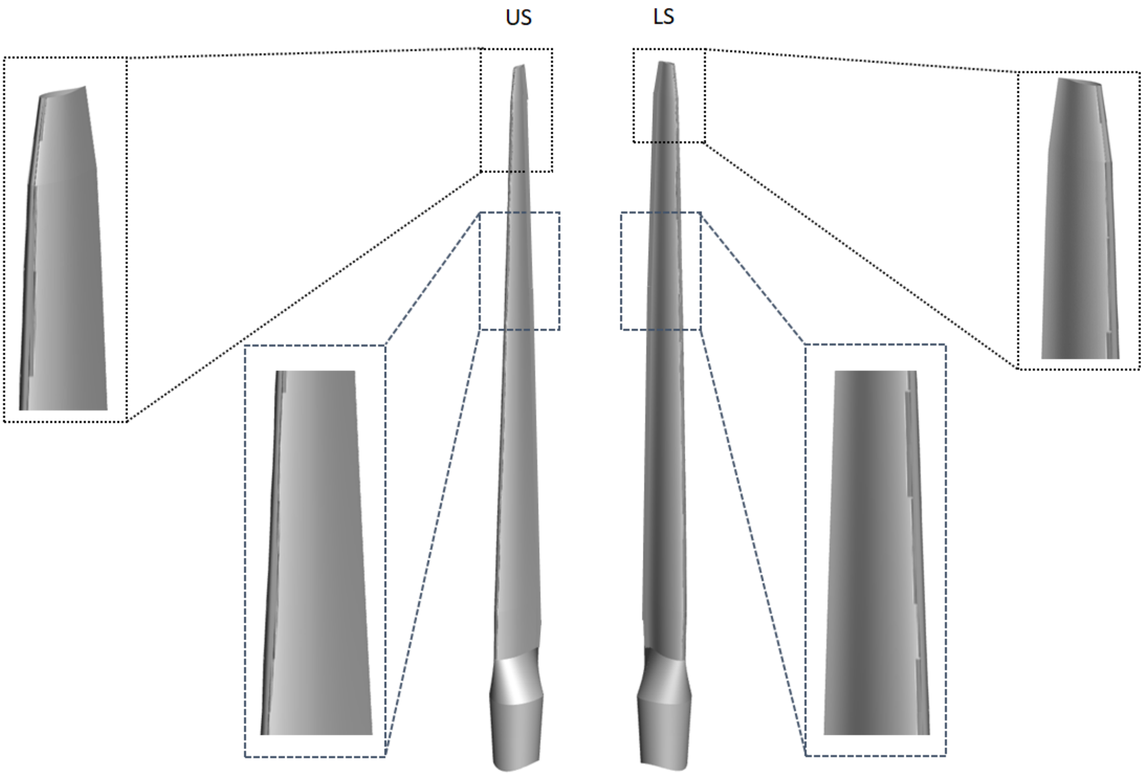


Figure 9.13: 3D view of analyzed leading edge erosion damage.

using the central and end regions of the considered AoA interval. In practice, the aforementioned small discrepancies do not affect the analyses below, since they occur for damaged airfoils which do not operate in the highlighted AoA range. With regard to computational cost, it is noted that obtaining the six pairs of c_l and c_d curves by means of ML, starting from the three delamination parameters of the six damaged airfoils, requires just a few seconds. Conversely, obtaining the same force data using CFD requires about 65 minutes of wall-clock time, resulting from 1 minute to generate 6 grids using concurrently 6 processor cores, and 64 minutes for determining 6 lift and drag curves with 2D CFD simulations using concurrently 6 16-core HEC nodes.

Cross-comparing the force coefficient curves of the damaged and nominal airfoils also shows that for all cross sections *a*) the magnitude of the lift coefficient of the eroded airfoils at low and high AoA is lower than that of their nominal counterparts, but is comparable to that of the nominal airfoils for AoA α between about $-5°$ and $5°$, and *b*) a qualitatively similar pattern is observed for the drag coefficients, and the difference between the mean drag coefficient of the damaged and nominal airfoils increases as the blade thickness decreases, *i.e.* as the blade radius increases.

The curves of the turbine power P, torque Q and thrust T against the freestream wind speed U of the damaged and nominal turbines are compared in Fig. 9.15(a), whereas those of the rotor speed Ω, the blade pitch β and the TSR λ of the same turbines are compared in Fig. 9.15(b).

Table 9.5: Damage geometry parameters at six radial positions.

	r/R					
	0.19	0.32	0.39	0.56	0.66	0.95
$s_u/c \times 100$	4.9	4.3	3.8	3.7	3.0	6.4
$s_l/c \times 100$	5.3	4.7	4.3	4.3	6.6	6.3
$d/c \times 100$	0.4	0.4	0.4	0.52	0.61	0.85

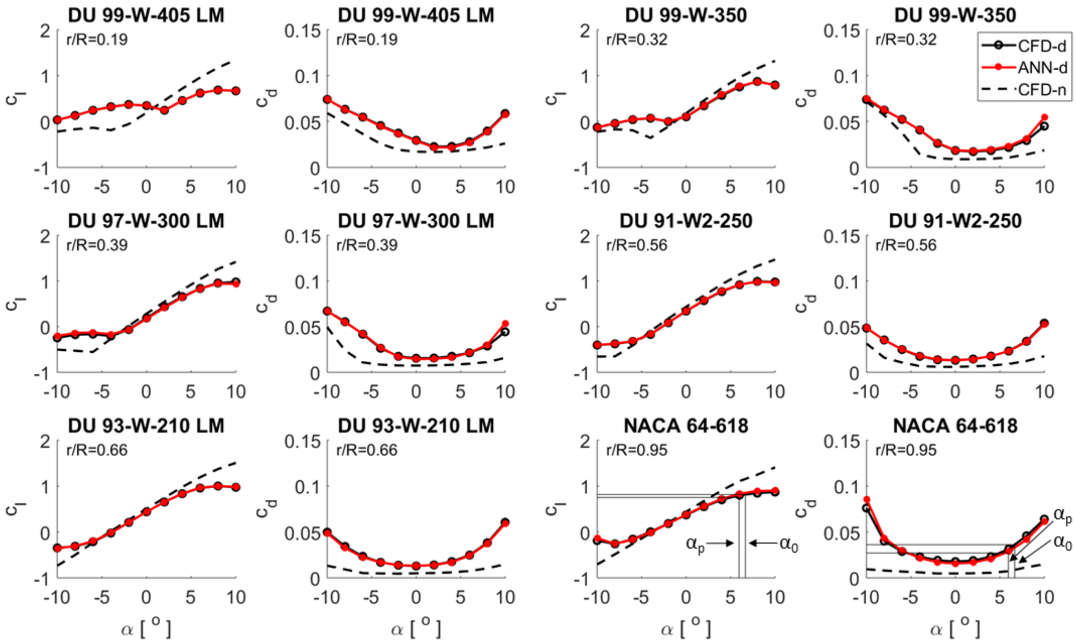


Figure 9.14: Comparison of lift and drag forces of nominal and damaged airfoils at six radial positions.

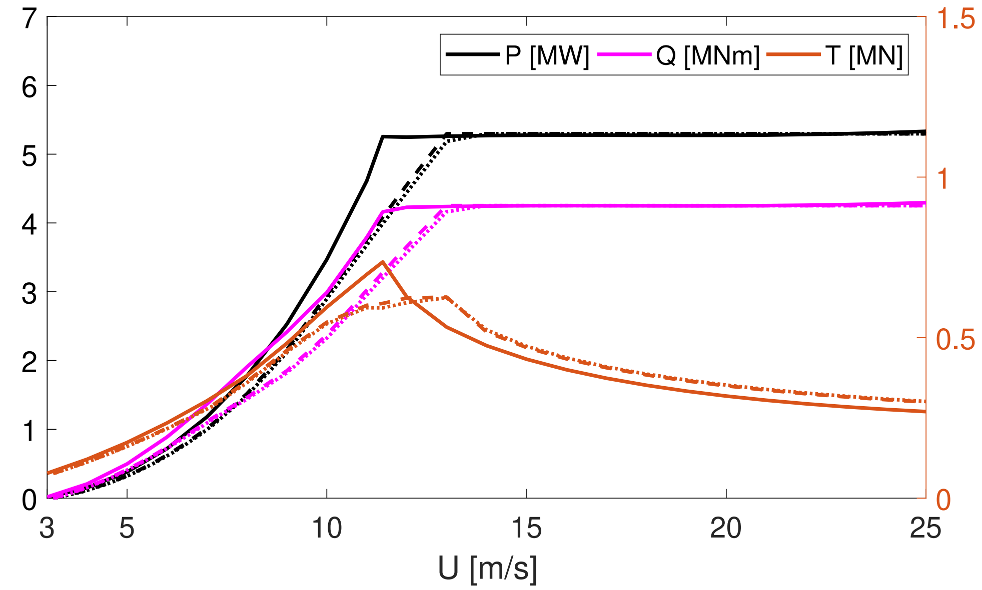
An excellent agreement between the power, load and regulation curves of the turbine with blade leading edge delamination obtained using either CFD or ANN airfoil data is observed. It is noted that, once the damaged airfoil lift and drag data are determined using the output of relatively lengthy direct CFD simulations or the substantially faster ML approach, the generation of both curve sets requires just a few seconds. For determining the regulation curves of the turbine affected by the given leading edge damage, AeroDyn requires lift and drag data for 27 airfoils, corresponding to the airfoils at the mean radius of the 25 blade strips, the aerodynamic section at the lowest blade radius and the tip section. The direct CFD method for determining the force data of the 27 airfoils requires a wall-clock time of 129 minutes, resulting from 1 minute for generating the CFD grids and 128 minutes to run 27 CFD analyses, each for 16 values of α, using concurrently 20 16-core HEC nodes. Using the ANN system, conversely, requires just a few seconds.

The power control strategy adopted for the turbine with damaged blades is similar to that used for the turbine with nominal blade geometry, but its target settings are adapted to the new rotor aerodynamic characteristics resulting from the sectional force alterations of the blades. To determine the adaptive control settings, the rated aerodynamic power of 5.30 MW, the minimum rotor speed of 6.9 RPM, the maximum rotor speed of 12.1 RPM and the cut-off wind speed of 25 m/s of the nominal turbine are adopted also for the turbine with eroded leading edges. Because of the differences of the lift and drag curves of the nominal and eroded airfoils highlighted above, however, the cut-in and rated wind speeds of the damaged and nominal turbines are slightly different: the cut-in and rated wind speeds of the nominal turbine are 3.0 and 11.4 m/s respectively, whereas those of the damaged turbine are 3.2 m/s and 13 m/s respectively. For both turbines, a linear relation between torque and rotor speed is enforced in region 1.5, and such linear relation is maintained until Ω increases by 30 % over the cut-in value. At the end of region 1.5, Ω=9.0 RPM for both turbines, but U=7.7 m/s for the nominal turbine and U=7.2 m/s for the damaged turbine. For the nominal turbine, the rotor speed in the following region 2 is set to the value corresponding to the known optimal value of λ=7.55 [18], and, once Ω achieves 90 % of its maximum value (Ω=11.6 RPM) at U=10.3 m/s, the control uses again a linear relation between torque and rotor speed (region 2.5) until the rated power and maximum rotor speed are achieved at the rated wind speed of 11.4 m/s. Once the rated power has been achieved, the blade pitch starts increasing to reduce AoAs and maintain the power equal to its rated value as the wind speed increases until the cut-off wind speed.

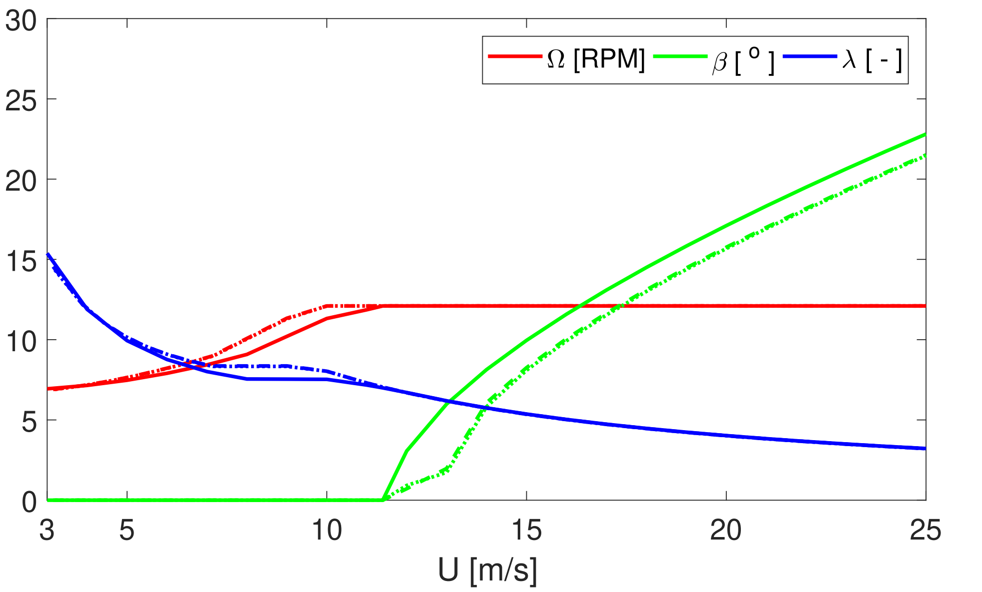
For the damaged turbine, the rotor speed in region 2 is set to the value that maximizes the aerodynamic power for each wind speed. This is achieved by wrapping AeroDyn with a MATLAB script that for each wind speed uses the golden section search [13] for gradient-based optimization to determine the optimal value of the rotor speed. This operation is required because the values of the optimal TSR (and also that of the corresponding maximum power coefficient) of the damaged and nominal turbines are different. The optimal operation parameters of the former turbine are unknown, and they also vary during operation as erosion progresses. Tracking the optimal TSR of the damaged turbine as U increases, one finds that this turbine achieves the maximum rotor speed of 12.1 RPM at U=10.0 m/s, well before the rated wind speed of the nominal turbine. Therefore, from U=10.0 m/s to U=11.4 m/s, the rotor speed of the damaged turbine is kept constant and equal to its maximum value. At 11.4 m/s, however, the aerodynamic power of the damaged turbine is still significantly lower than the rated power. To achieve more rapidly the rated aerodynamic power as the wind speed increases above 11.4 m/s, it is found that it is beneficial to start pitching to feather the eroded blades. At a first glance, this seems counterintuitive. The reason for this occurrence is that reducing the AoA at the outer sections of the eroded blade results in the consequent drag reduction outweighing the corresponding lift reduction, and thus in slightly larger torque at lower AoA. This is illustrated in the lift and drag curves of the blade section at r/R=0.95 in Fig. 9.14. The two values of the AoA labeled α_p and α_0 in these subplots are respectively the values computed by AeroDyn using the blade pitch that maximizes power and no pitch at U=12 m/s. It is seen that at $\alpha = \alpha_p$ a significantly lower drag than at $\alpha = \alpha_0$ can be obtained. As mentioned, the lift coefficient at $\alpha = \alpha_p$ is lower than at $\alpha = \alpha_0$, but due to low twist of the outboard blade sections the reduction of the drag outweighs that of the lift, resulting in an overall higher torque. At 13 m/s, the damaged turbine achieves the rated aerodynamic power. For U varying between 11.4 to 13.0 m/s, the blade pitch that maximizes the aerodynamic power is obtained by wrapping AeroDyn with a MATLAB script that for each wind speed uses the golden section search to determine the optimal blade pitch. Increasing the wind speed above the rated value of 13 m/s requires more significant increments of the blade pitch to maintain the power to its rated level. Between the rated and the cut-out wind speed, AeroDyn is wrapped by a MATLAB script that uses the secant method to determine the blade pitch yielding the rated aerodynamic power.

The AEP loss due to the considered damage is assessed by considering a site with mean wind speed of 9.36 m/s over one year, and featuring a Rayleigh wind frequency distribution. These wind data are representative of offshore sites on the north-western coast of England. Integrating the power curve of the nominal and damaged turbines against this wind frequency distribution and ignoring down-times of both turbines, gives AEP values of 22163 MWh and 20190 MWh. This corresponds to an AEP loss of 8.9 %, a value of the same order of magnitude of those reported in the literature. It has also been verified that the AEP loss of the damaged turbine obtained by ignoring its optimal TSR alterations (*i.e.* adopting the nominal dependence of the rotational speed on the wind speed from cut-in to cut-out but allowing the pitch controller to limit the aerodynamic power to a constant value of 5.3 MW above rated wind speed) are 16.7 % of the nominal AEP, twice the loss obtained with the adaptive power control discussed above.

With regard to structural loads, it is noted that the rotor thrust of the nominal turbine reaches a maximum at the rated operating point ($U = 11.4$ m/s) before dropping again. This peak is typical of variable generator speed variable blade-pitch-to-feather wind turbines because of the transition that occurs in the control system at rated speed between the active generator torque and the active blade pitch control regions. On the other hand, for the considered power control strategy of the damaged turbine, one notes that *a*) a fairly flat region of the thrust between the wind speed where the blade pitch starts increasing ($U = 11.4$ m/s) and that at which the rated power is achieved ($U = 13.0$ m/s) exists, and *b*) the thrust of the damaged turbine is higher than that of the nominal turbine even before the former reaches rated power, and remains such until the cut-off wind speed.



(a) Power (P), torque (Q) and thrust (T) against wind speed (U).



(b) Rotor speed (Ω), blade pitch (β) and TSR (λ) against wind speed (U).

Figure 9.15: Steady state performance and control curves. Solid, dotted and dashed lines refer respectively to analysis of turbine with nominal blade geometry, damaged blade surface using CFD, and damaged blade surface using ML.

Chapter 10

Conclusions and future work

This study has presented ALPS, a novel modular technology for assessing wind turbine energy losses due to blade surface damage. The wind turbine power curve is determined by using engineering codes based on BEMT, such as the NREL FAST and AeroDyn codes. The force coefficients of the damaged airfoils associated with the cross sections of the blades with general leading edge delamination due to erosion are determined rapidly using an artificial neural network system trained using a pre-existing database of damaged airfoil shapes containing airfoil force coefficients for the whole working range of AoAs. The presented demonstration focused on a radially irregular leading edge delamination of the blades of the NREL 5 MW turbine, and the analyses revealed that the considered damage of moderate to severe level, reduces the turbine AEP by about 8.9 %, a considerable amount which is also in line with other measured and computed estimates in the literature. The power control strategy of the damaged turbine was modified between rated and cut-in wind speeds to ensure maximum power also for the damaged turbine. This alteration is needed due to the different optimal TSR of the nominal and damaged turbines. In the case of power controllers lacking this adaptive strategy, thus not accounting for the reduction of the blade aerodynamic efficiency as wear and erosion progress during operation, the actual AEP loss is expected to be higher, up to twice the value recorded using adaptive control. With respect to this, the presented study provides some guidelines on how to make power control more robust to the time-dependent deterioration of the aerodynamic efficiency of the blades.

The damage analysed in this work is characterized by a slow variation in the span-wise direction. However, the ALPS system implemented is already capable of analysing more complicated damage patterns, such as that shown in figure 10.1. The only component missing in the ALPS system is the GAS module, which is currently under development and will be introduced in future publications.

In the presented demonstration, the ALPS database is generated using 2D Navier-Stokes incompressible flow simulations of the ANSYS FLUENT CFD code. By automating the geometry and mesh generation phases, the execution of the CFD analyses and the collection of the airfoil force data by means of MATLAB, JAVA, LINUX and ANSYS WorkBench scripting, and by using 20 16-cores nodes of the HEC cluster, the 96672 simulations required to determine the lift and drag curves of 6042 damaged airfoils have been carried out in about two weeks of wall-clock time. The failed simulations have been efficiently detected by means of a MATLAB code, and analysed

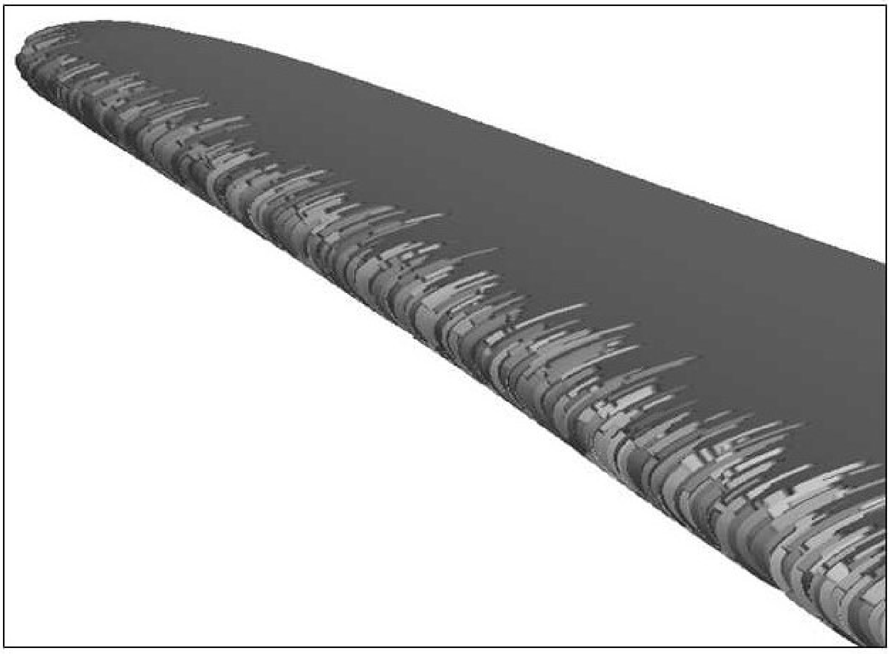


Figure 10.1: Example of a damaged blade that could be analysed with the ALPS system developed in this work.

to determine the cause of the failure. If it was due to the mesh, or to errors in the set-up, they have been corrected and the simulations re-run. Some of the results were oscillating around a mean value, especially for high or low angles of attack. In these cases, the steady simulations are not able to capture the complex aerodynamics behaviour of the airfoils, and time-dependent simulations would be needed. Further investigations on this matter will be object of future work. In section 9.1 the failed simulations have been grouped depending on the angle of attack, showing that the largest percentages are located in correspondence of the highest and lowest values of the angle of attack. Since these angles are usually not included in the operational range of the blade, they could also not been considered in future works. Once the database has been completed, the energy loss of wind farms consisting of hundreds of turbines featuring general blade surface damages not included in the database can be determined in a few minutes at any time of the 25-year lifetime of the wind farm thanks to the extremely low computational cost of the ML estimates and the BEMT-based engineering analyses. The ML approach has been validated in section 2, where the results obtained with the ANN system have been compared to those given by CFD simulations for all the airfoils in the database. The results show a very good agreement between the two sets of data, with the RMSE square error being in the vast majority of cases lower than the 0.2% for the drag coefficient and 2% for the lift coefficient. A practical issue in the ALPS applicability may arise when the nominal blade geometry is unavailable to the wind farm operator and/or turbine maintenance provider, since this may hinder the generation of the damaged airfoil geometries required to build the airfoil force database. This difficulty may be circumvented by basing the entire damage analysis on a turbine of known geometry, and overall design, control and rated power broadly comparable to the operational turbine under consideration, assuming that the percentage AEP loss is comparable. A second alternative consists of using reverse engineering to obtain an approximation to the nominal blade geometry. It is also noted, however, that the trend of wind turbine manufacturers also providing servicing to wind farm operators is rapidly increasing. This occurrence removes the geometry accessibility issue, since these service providers can use ALPS on behalf of wind farm operators to better inform cost analyses and blade material-related choices. In all cases, once the nominal blade geometry is available, the damaged airfoil database generation is a fairly straightforward and inexpensive process, which can be made even more accessible by using open-source software and cloud computing.

The presented study considered only the delamination pattern of the leading edge damage caused by erosion. In the generation of the ALPS database, the impact of earlier-stage pit and gouge damage patterns can be also analysed by replacing the 2D NS CFD simulations with 3D simulations of radially thin constant-chord blade slices featuring different sizes and distributions of these 3D damage patterns. The impact of variable-roughness along damaged blade surfaces can also be accounted for using custom-tailored wall boundary conditions.

Object of future work will also be the improvement of the CFD lift and drag curves referring to the nominal airfoils, presented in chapter 8. Some discrepancies were observed between the experimental and CFD data, especially for the ticker airfoils located towards the root of the blade. However, these airfoil almost do not contribute to the power production of the blade, and the agreement between the power curve obtained with experimental and CFD data is very good (see figure 8.10). For the purposes of this work, therefore, the CFD results are considered reliable.

Bibliography

[1] https://www.cfd-online.com. Accessed on 08 July 2019.

[2] National Wind Technology Center information portal.

[3] Nrel's national wind technology centre forum. https://wind.nrel.gov/forum/wind/viewtopic.php?t=858. Accessed on 21 July 2019.

[4] Scripting API. https://www.sharcnet.ca/Software/Ansys/17.0/en-us/help/wb_dm/dmscriptapi.html, 2019. Accessed on 08 July 2019.

[5] 3M. A 3M Study Is the First to Show the Effects of Erosion on Wind Turbine Efficiency. https://www.pressebox.com/pressrelease/3m-deutschland-gmbh/a-3m-study-is-the-first-to-show-the-effects-of-erosion-on-wind-turbine-efficiency/boxid/445007, 2011. Accessed on 6 May 2019.

[6] C. Bak, Andersen, P.B., Madsen, H.A., and M. Gaunaa. Design and Verification of Airfoils Resistant to Surface Contamination and Turbulence Intensity. AIAA paper 2008-7050, August 2008. 26th AIAA Applied Aerodynamics Conference, Honolulu, Hawaii.

[7] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi. *Wind Energy Handbook*. Wiley, Chichester, West Sussex, UK, 2011.

[8] Campobasso, M.S., E. Minisci, and M. Caboni. Aerodynamic design optimization of wind turbine rotors under geometric uncertainty. *Wind Energy*, 19(1):51–65, 2016.

[9] Corten, G.P. and Veldcamp, H.F. Aerodynamics: Insects can halve wind-turbine powers. *Nature Brief Communications*, 412:41–42, 2001.

[10] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[11] P. Fuglsang and C. Bak. Development of the Risø wind turbine airfoils. *Wind Energy*, 7(2):145–162, 2004.

[12] P. Giguere and Selig, M.S. Aerodynamic Effects of Leading-edge Tape on Airfoils at Low Reynolds Numbers. *Wind Energy*, 2:125–136, 1999.

[13] Giordano, F.R., Fox, W.P., Horton, S.B., and Weir, M.D. *A First Course in Mathematical Modeling*. Brooks/Cole, 511 Forest Lodge Rd Pacific Grove, USA, 2009.

[14] Hagan, M.T. and M. Menhaj. Training feed-forward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5:989–993, 1994.

[15] G. Heyman, B. Jonkman, R. Murray, R. Damiani, and J. Jonkman. AeroDyn: a time-domain wind and MHK turbine aerodynamics module. `https://nwtc.nrel.gov/AeroDyn`. Accessed on 6 May 2019.

[16] Jasinski, W.J., Noe, S.C., Selig, M.S., and Bragg, M.B. Wind turbine performance under icing conditions. *Journal of Solar Energy Engineering,*, 120(1):60–65, 1998.

[17] J. Jonkman. FAST: An aeroelastic computer-aided engineering (CAE) tool for horizontal axis wind turbines. `https://nwtc.nrel.gov/FAST`. Accessed on 6 May 2019.

[18] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Technical Report NREL/TP-500-38060, NREL, Golden, CO, USA, 2009.

[19] Khalfallah, M.G. and Koliub, A.M. Effect of dust on the performance of wind turbines. *Desalination*, 209:209–220, 2007.

[20] R.B. Langtry and F.R. Menter. Correlation Based Transition Modeling for Unstructured Parallelized Computational Fluid Dynamics Codes. *AIAA Journal*, 47:2894–2906, December 2009.

[21] J Liersch and J Michael. Investigation of the impact of rain and particle erosion on rotor blade aerodynamics with an erosion test facility to enhancing the rotor blade performance and durability. *Journal of Physics: Conference Series*, 524:012023, 06 2014.

[22] D. MacKay. Bayesian Interpolation. *Neural Computation*, 4:415–447, 1992.

[23] Menter, F.R. Two-Equation Eddy-Viscosity Turbulence-Models for Engineering Applications. *AIAA Journal*, 32(8):1598–1605, August 1994.

[24] OpenCFD Ltd. OpenFOAM - The Open Source Computational Fluid Dynamics (CFD) Toolbox. `www.openfoam.org`. Accessed on 6 May 2019.

[25] ORE CATAPULT. Autonomous vessels, drones and crawling robots – welcome to the future of offshore wind farm inspection and repair. `https://ore.catapult.org.uk/press-releases/mimree-inspection-repair-solution/`. Accessed on 25 Sep 2019.

[26] A. Sareen, Sapre, C.A., and Selig, M.S. Effects of leading edge erosion on wind turbine blade performance. *Wind Energy*, 17(10):1531–1542, 2014.

[27] M. Schramm, H. Rahimi, B. Stoevesandt, and K. Tamgager. The Influence of Eroded Blades on Wind Turbine Performance Using Numerical Simulations. *Energies*, 10(9):1–15, 2017.

[28] Somers, D.M. Design and Experimental Results for the S809 Airfoil. Technical Report NREL/SR-440-6918, NREL, Golden, CO, USA, 1997.

[29] Somers, D.M. The S814 and S815 Airfoils. Technical Report NREL/SR-500-36292, NREL, Golden, CO, USA, December 2004.

[30] I. Troen and E. Petersen. European Wind Atlas. Technical Report, ISBN 87-550-1482-8, Risø National Laboratoy, Roskilde, Denmark, 1989.

[31] van Rooij, R.P.J.O.M. and Timmer, W.A. Roughness Sensitivity Considerations for Thick Rotor Blade Airfoils. *Journal of Solar Energy Engineering*, 125(4), November 2003.

[32] Y. Wang, R. Hu, and X. Zheng. Aerodynamic Analysis of an Airfoil with Leading Edge Pitting Erosion Evolutionary Computation. *Journal of Solar Energy Engineering*, 139(6), 2017.

[33] A. Zanon, M. De Gennaro, and H. Kuehnelt. Wind energy harnessing of the NREL 5 MW reference wind turbine in icing conditions under different operational strategies. *Renewable Energy*, 115:760–772, 2018.