

UPC

CTTC

**Towards a virtual platform
for aerodynamic design,
performance assessment
and optimization of
Horizontal Axis Wind
Turbines**

Centre Tecnològic de Transferència de Calor
Departament de Màquines i Motors Tèrmics
Universitat Politècnica de Catalunya

Daniel Martínez Valdivieso
Doctoral Thesis

Towards a virtual platform for aerodynamic design, performance assessment and optimization of Horizontal Axis Wind Turbines

Daniel Martínez Valdivieso

TESI DOCTORAL

presentada al

Departament de Màquines i Motors Tèrmics
E.T.S.E.I.A.T.
Universitat Politècnica de Catalunya

per a l'obtenció del grau de

Doctor per la Universitat Politècnica de Catalunya

Terrassa, March 2017

Towards a virtual platform for aerodynamic design, performance assessment and optimization of Horizontal Axis Wind Turbines

Daniel Martínez Valdivieso

Directors de la Tesi

Dr. Carlos David Pérez Segarra

Dr. Assensi Oliva Llena

Tribunal Qualificador

Dr. Cristóbal Cortés Gracia

Universidad de Zaragoza

Dr. Antonio Pascau Benito

Universidad de Zaragoza

Dr. Joaquim Rigola Serrano

Universitat Politècnica de Catalunya

Every three seconds someone dies from extreme hunger (poverty), while at the same time, the 1 % of population possesses the same wealth as the remaining 99 %. There is no point on doing rocket science if we still have such strong social gradients. For how long are we going to turn a blind eye on this topic? Why don't we rescue our siblings and do science together? We will progress faster and it will be more ethical.

In my free time, I teach Critical Thinking, Emotional Intelligence and Practical Philosophy to teenagers, youth and sometimes adults. To me, Critical Thinking is the ability to analyze and understand the key concepts on a certain topic in order to apply this knowledge to improve our performance. I believe that this ability is shaped by three pillars: philosophy, psychology and science. In this specific order. Without exception, the first block is the one that systematically shocks the audience the most. I always begin by asking the next questions adjusting the depth and the pace to the public in front: What is the universe? What is the purpose of life? What is the origin of life? What is the actual difference between the alive and inert organisms and objects? What is the purpose of human race? To the best of my knowledge we are at an embryonic stage to answer those questions and many like-minded ones. The first questions that we can really *look into their eyes* are the next: What's the purpose of my life? How am I going to use it? Where do we want to bring humanity?

I must confess that, I am fascinated by those questions and so they are the multiple audiences to whom I have addressed this Critical Thinking speech. Definitely, I would like to be part of a *Type I civilization*¹. The work done in this thesis is just a tiny dot contributing to that objective. Maths and physics are extremely useful, they are crucial to understand the world we live. The comprehension of complexity fuels my motivation and I truly flow when I learn patterns in these and other fields. It makes me deeply happy and I think I am not alone in these believes and feelings. Could you imagine to

¹the one able to use and store the energy received from its closest star, according to the Kardashev Scale

fully understand the underlying fundamental laws that govern the universe? It will help us to understand how did it all began. I contend that it would be an extreme pleasure for human curiosity, specially for those having the honor to be in the front line of the research. Those who look at the eyes of the mistery to unravel it.

On the other hand, I deeply believe in the win-win scenarios. I inconditionally support the concept of the *Intelligent egoism* that the XIV Dalai Lama fiercely defences: it is great to look after yourself, but there are always strategies to benefit all the directly or indirectly involved parties. In brief, the idea is: *I grow, and you grow with me*. In other words, we always behave with self-interest. Nothing is wrong with that. The point is to find out ways to benefit yourself while improving the lives of those around you. Believe it or not, but according to the United Nations, there are still 663 million people in the world without access to potable water, half of them in Sub Saharan Africa. Additionally, statistics reveal that roughly every three seconds² someone dies of extreme poverty while, for example, in Spain, the average citizen spends 700 €/year in clothes. Is it the type of society we want? Is it fair?

Gandhi said: *there is no they, only us; all of us*. I support it and I am willing to bring it one step further. *They are me. I am you. We are the same entity. I am* many people: the bullied, Aylan Kurdi³ and all of those without a decent healthcare system. I did not decide where and when I was supposed to be born. Nobody did. Consequently, I *am* also the raped woman in DR of Congo, the Asian teenager exploited by sewing for (mostly) occidental brands in exchange of miserable salaries and without the right for association. I am not only a 27-year-old Spanish white and male engineer. I cannot satisfy a single need alone. I am just a tiny gear of bigger machine. Humanity is the machine. We are an ultra-social species. We need each other and we must relentlessly pursue the true equality of opportunities for all of us, which at least must guarantee access to: air, water, food, shelter, human rights, duties, energy, education and information.

Do not get me wrong. Technology and science are fundamental parts of humanity and I give them full credit. I am passionate about research, which is propelled by curiosity, an in-built human feature. What's more, I declare my love to answer those questions at the beginning of this piece of writting. Sooner or later, (I hope) we will find the answer to them, but I contend that it is much faster and, without discussion, much more ethical, to first attain a scenario where social equality is a fact rather than a utopy. Once there, we can walk to that aim as a team. A team of the whole humanity. We are all players, nobody is out of the task.

²www.everythreeseconds.net

³the three-year-old Syrian boy drowned in the Mediterranean sea in September 2015.

From my little square meter and taking advantage of this chance in the presentation of this humble scientific contribution, I encourage everyone on Earth to shift their eyes to this social equality. In this sense, Viktor Frankl's metaphore is lovely: humans are like eyes, if they *see* themselves they are ill, because it is not the task for which one they were conceived to. According to Logotherapy, being human implies self-transcendence: the more you deliver yourself to a cause or to a person, the more human you become. The more you focus in self-actualization, the more it escapes from you.

I think that we must fight against our egos, become as humble as dust, cultivate self-compassion to enable constructive self-criticism, connect with the rest and commit to personal growth. We must delete this huge social gradients. We have to eradicate the current inequality. And we have to do it now.

Dream. Believe. Decide. Commit. Work. Serve. And die for the truth.

Ninguna acción de otro ser humano, aunque me perjudique o me hiera, me avasalle y me ataque frontalmente, la percibo contra mi ser. Incluso cuando tenga intención directa de buscar mi punto más débil para destruirme. La interpreto únicamente como signo de debilidad o reclamo de atención. Pero nunca como maldad explícita y gratuita.

Dani M. Valdivieso

Acknowledgements / Agradecimientos / Agraïments

Mohandas Karamchand Gandhi said: There is no **they**, only **us**, all of us. I totally agree with this expression. I believe that we⁴ are all interconnected and whatever occurs to someone in the opposite side of the Earth it automatically involves me in. In this sense, my first words of gratitude should be aligned to this idea. To me, it is not a vague concept but very specific idea. I must probably thank every single person, because everyone is part of this job: the fruit recollectors back in tropical countries that perform a crucial step in the chain that delivers pinapples in the backstreet supermarket; the intermediate rank police(wo)man fighting against organized crime in developing countries; the worker controlling the automatized chain where the keyboard used to type this document was manufactured; etc. I want to make visible here the complex, but extremely efficient work and accumulated wisdom, that makes possible this thesis. Additionally and more concretely, thousands of pieces of software and hardware enable me to add my contribution on the top of this *giant* and (hopefully) provide some new insight in the scientific world through the content of this thesis.

The next block of acknowledgments goes to the Spanish and Catalan citizens. I feel that they are my real bosses. It is true that the scholarships that financially supported me are from *El gobierno de España* and *La Generalitat de Catalunya*, and I am grateful to them too. However, the ultimate promoter and *stakeholder* of my work is the citizen (my cousin, my neighbor, the old lady buying the bread at the bakery and the lorry driver paying the highway toll). All of them have been in my mind every single day I had to come to my workplace.

The *Universitat Politècnica de Catalunya* (UPC) deserves my deep respect for the high-quality education that the institution provided me during my Bachelor, Master

⁴humans, but ideally all living beings

and Ph.D. degrees. I could realize about the good job that was being done on me when I participated in the ERASMUS program in DTU (Technical University of Denmark). I experienced there a different way to proceed and I coworked with mates from all over the world. I was well-equipped to develop myself in such environment, being able to add some value in every project I was asked to participate in my stay. That is, in part, thanks to UPC.

I had the first contact with the Centre Tecnològic de Transferència de Calor i Massa (CTTC) when I was studying my university degree back in 2009. I still remember my feeling of pleasure when I knew I could apply in CTTC most of the engineering concepts learnt. To me it was the first time to be *useful* as an engineer. Carlos David Pérez-Segarra was the one to blow my mind with the outstanding way he exposed numerical methods. Immediately after, I decided to collaborate with the research group, and Assensi Oliva Llena offered me a scholarship to reward my interest. I am very grateful to both of them for this initial trust on me. Then, Xavi Trias⁵ helped me to understand the apparently unattainable Fractional Step Method during my Driven Cavity initial exercise. He supported me in those highly-advanced mathematical issues. He was my pillar then. During the first three years of my thesis I received the support from Oriol Lehmkhul and I cordially thank him for the guidance. On the professional level, I must also greet Octavi Pavón and Jordi Chiva for their unconditional support in front of the hardware, software and algorithmical problems that I faced during this thesis.

I also want to clearly state here that the core technical concepts employed along this thesis cannot be seen as my sole contribution. Many co-workers must be acknowledged about their autorship. Federico Favre is the person who has worked the most in the topic of the Immersed Boundary Method. Olga Estruch brought the Dynamic Meshes to a new stage. Jordi Muela is the developer of the Sliding Meshes. Oscar Antepara has designed the code referent to Adaptive Mesh Refinement. Joan Calafell is the person in charge for the Wall Model LES. Aleix Báez was the person to implement the first version of the Blade Element Momentum code.

Once again, I want to thank Carlos David Pérez Segarra because of the thorough read of this thesis. He carefully examined every comma of this text and his contributions have been of great help to enhance the quality of this dissertation.

Also, at the CTTC level, I must show my gratitude to Imanol Aizpurúa, Néstor Balcázar, Aleix Báez, Carles Oliet, Nico Valle and Xavi Trias. Doing a thesis is a long-term project with ups and downs. Thus, having powerful friendships is indispensable to get enough support to get by and forget about giving up when the idea strikes your

⁵or F.X. Trias as he would put at the back of his football t-shirt

mind. The frequent conversations I shared with them, in both technical and personal issues, were of extreme help to me during the whole period when the thesis stretched.

Still in the technical field, I have to explicitly thank two entities. First, I want to thank Enair S.L., specifically Carlos Díez, Román Dato, Juanan Villa and Antonio Sáez, for their collaborative work during the project of designing the 10 and 40 kW wind turbines with Termo Fluids S.L. and their agreement to include this information in this thesis. Second, I appreciate the work done by ECN in the MEXICO project, and I thank them the will to share the data of the project. As Gerard Schepers asked me in the e-mail when disclosing the information, I publicly acknowledge their work using the sentence they provide me: *The data used have been supplied by the consortium which carried out the EU FP5 project Mexico: 'Model rotor EXperiments In Controlled conditions'*.

I must mention here the contribution of Jonay Roda and Gemma Clofent, my former teachers in physics and maths back in highschool. Thanks to them I got a solid education background that strongly helped me during the first year of engineering. Special thanks to Jonay Roda, who is the teacher that has made, by large, the best contributions to my personal development, till the point that I cannot consider this achievement (or any other) anything else but a common merit. His guidance, corrections and support during *El Treball de Recerca* about *Electromagnetic contamination* made me flourish as very few events have procured. Nowadays, I still have to thank him for the opportunity that he has given me to develop my volunteering program on *Critical Thinking, Emotional Intelligence and Practical Philosophy* at INS Jonqueres (my former high-school) where he is now the director. Jonay, no he perdut mai el neguit de volar, i me n'alegro de poder donar-li sentit a la teva feina pel simple fet ser com sóc: un ésser humà que mantè viva la seva curiositat. Gràcies per semblar en mi i per cultivar-me de la forma que ho has fet i que encara avui dia continues fent. Gràcies!

I cannot forget about Juan López Villarán, he is my very special friend since 2005, already for 12 years. Our long and deep conversations have had a huge impact on me. We have also been rock climbing mates and colleagues in the project of building a quadcopter from scratch. I am eternally in debt with him for the mutual support that we have delivered to each other, in both happy and hard times. Juan, uno puede ser fuerte solo, pero tener un compañero de *cordada* de tu calidad hace que el camino sea más fácil: los problemas se dividen y las alegrías se multiplican. ¡Gracias!

Lorena Castarnado Padilla was my girlfriend for four magnificent years. We shared trips, volunteering programs and nice discussions. I will always be grateful for the degree in which she trusted and supported me during the first years of my thesis.

Lorena, aunque ya no estemos en contacto, te sigo deseando lo mejor. Te lo mereces. Todo lo que probé fue por lo que yo consideraba tu bien. Eres una gran persona. Te quiero y te querré siempre.

The best romantic relationship that I have ever had is the one with Pau Salvador Mateo. She is a powerful woman that conquered my heart in the very first hour we met. With her I have explored and learnt without limits. Thanks to the degree that she has trusted me and opened herself with her vulnerability, I am now much more sensitive and warm-hearted. Gràcies per confiar en mi de la forma en la que ho has fet. Ojalá i pugui ajudar-te a eclosionar per complet i treure el teu millor-jo, que sabem que encara dorm. Ets formidable. T'estimo i t'estimaré per sempre.

Now, it is turn to mention my family. It is a pleasure to see the power of a large family. ¡Gracias a todxs lxs Martínez! Additionally, my most appreciated regards to my close family, ¡Gracias Valdivieso! Siempre tendré en el corazón el magnífico tiempo que pasamos en las navidades. The group of my *direct* cousins with Mónica, David, Miguel, Cristina, Vane, Dani, María, Águeda have played an important role in my personal development. Muchas gracias por aceptarme tan joven en el grupo, ¡convivir con vosotros en los viajes de primos me hizo crecer rápidamente!

Finally, it is time to be grateful to my core family:

Pedro, mi padre, nunca olvidaré tu apoyo incondicional, especialmente en el baloncesto. Aunque ahora no juegue un papel importante en mi vida, ha contribuido a mi autoestima de una forma en la que no puedo cuantificar. Gracias por apoyarme siempre y por darme lo mejor de ti. Gracias por ser capaz de perdonar tan rápidamente.

Paqui, mi madre, de no haber sido por tu perseverancia por mi integridad mental, estoy seguro que los momentos duros de mi infancia y adolescencia habrían roto mi estabilidad, haciendo imposible que hoy escriba estas palabras en calidad de doctor. Gracias por estar dispuesta a sacrificarte hasta tal punto por los demás y en concreto por mí.

Javi, mi hermano del alma, has sido mi referente durante muchos años. Has sido mi padre. Me has protegido, me has mimado, me has criticado y me has cuestionado como nadie lo ha hecho. En definitiva, me enderezaste el tronco para favorecerme a ser quien soy ahora. Gracias por entregarte de esa manera conmigo. Lo que soy, te lo debo a ti. Lo que llegue a ser, lo serás conmigo.

Gracias a todxs. Gràcies a tothom. Thanks to all of you. I would not have arrived that far without all your contributions.

Foreword

I have considered a good idea to add this preface on my thesis in order to help the reader to understand the context in which this thesis has been developed as well as some side elements involved on that process that, in my opinion, should not be part of the abstract nor the introduction. I guess that the following comment could be mentioned by any Ph.D. candidate but I feel that “this thesis has been special” and the purpose of this section is to provide some details about the word “special”.

This thesis, officially speaking, has been developed since September 2012, but extra-officially speaking since January 2011. At that early stage, I started to work and study different first-step applications and cases as the motivated student that I was lead by the professors of the CTTC (Heat and Mass Transfer Technological Centre) group at UPC (Polytechnic University of Catalonia). Many facts have conditioned the course of the thesis and therefore, the content and arrangement that the reader can find in this dissertation, but I want to highlight some of the most important ones in this section.

July 2012 was the point in which I consider that I was of help for CTTC for the first time. In other words, when I stopped of purely studying for learning to move to the phase where I studied and learnt for production (either scientific or technical). During that time, the ongoing fields of research at CTTC were many, but I was proposed to join the Wind Energy team and I happily accepted. By then, some projects had been done with ALSTOM and Anortec to optimize the thermal behavior of a nacelle and design the blade of new wind turbine, respectively. Moreover, the CTTC was partner of the Offwindtech project, meant to stretch from 2011 till 2013, so we had certain deadlines to cover to that respect. Therefore, the agenda was partly set by the needs of the project which were related with a 3D CFD simulation of a 5 MW Offshore turbine of 126 m in diameter. Thus, the circumstances forced us to move into the details of wind energy on the field of CFD without letting us, or at least me, going through the analysis of simpler models on the first round. It is definitely not the best strategy to rush out in

that way, but nowadays, who (either individual or organization) can be doing science without this kind of pressure? Probably very few, if any. All in all, what I mean is that, the chronology of the decisions made and steps which configure the research line would have been different without those external agents. Definitely, and most importantly for the reader to know, the order in which the content of the thesis is arranged in this text does not match that course of events (chronological order).

Another issue that deserves some comments is the simulation platform and techniques which were available at CTTC by that time. Most scientist deal with research fields where team work and cumulative knowledge (know-how) are two indispensable components. Therefore, any new scientist would count on a certain set of resources when joining the group, and those resources can be seen at the same time as an opportunity and a limitation. In this sense, I must say that I have been lucky in my chance to join the CTTC group and take advantage of all the previous work that had been done that far, which was a lot. Just to mention a couple of facts that strongly influenced me for the better: 1) the existence of TermoFluids, an in-house 3D parallel aimed at HPC and C++ object-oriented code, capable of running leading-edge simulations involving hundreds if not thousands of CPUs; and 2) the size of the CTTC was about more than forty researchers between professors, engineers, doctors, PhD candidates and support staff.

The expertise of CTTC might be fairly considered quite high on the fields of HPC, Heat Transfer and certain aspects on aerodynamics. Those are the main fields of strength within the organization. However, in my opinion, in the field of Wind Energy we cannot say that we are a referent or a world leading institution. This leading role, must be attributed to NREL, DTU, TU Delft and some other academic and technical leaders. Thus, I believe that an extra degree of humbleness must be present during the whole writing.

Notwithstanding, I have no doubt that the whole work performed on this thesis is new and useful to CTTC. From the point of view of the scientific community, the contributions of this thesis might not represent a groundbreaking advance, but hopefully they will be of help for the upcoming research on the field of Wind Energy. From my point of view, if anything should be highlighted from the work I have developed with my CTTC co-workers, it must be the next aspects. First, the management of the information that configures a generic 3D airfoil object from the computational point of view. Second, the idea (and perhaps parts of the implementation) of the integration and automation of the blade CAD generation into mesh generation software. Third, the simple but effective AeroElectric coupling using the Blade Element Momentum Theory and the

curve Torque - Rpm of the generator. And fourth, the testing and application of the ideas of Adaptive Mesh Refinement, Immersed Boundary Method and Sliding Meshes to the CFD analysis of wind turbines as potentially successful simulation strategies for the future.

Once I am at the point of describing my work, I look backwards to those years at the beginning of this experience and now I think: *I shouldn't have done X, it would have been better to take the other path and do Y*. Probably, that thought would be common not only for Ph.D. candidates but for any person reflecting upon a past project. I guess that this is the hallmark of experience building and the learning-by-doing spirit necessary in any new adventure. I totally agree with the saying "If we knew what we were doing, we would not call it Research" and the consequence of this is: I acknowledge that many decisions made in the past years would have been better off in another slightly or radically different way. However it be, I stand with the best combination of pride and humbleness I am able to claim to defend the honest work I have performed in the last five-six years.

Summing up, I am sure that my contributions have been fruitful for CTTC, and I sincerely hope that the work presented here could be of help either directly or indirectly for the rest of the world. Anyhow, I must say that getting involved in this Ph.D., has lead me to understand the foundations of science, which has shaped me for the better. I do not hesitate even for a second to say that it has been one of the most rewarding experiences that I have ever undertaken both in the personal and professional dimension.

Abstract

This thesis focuses on the study and improvement of the techniques involved on a virtual platform for the simulation of the Aerodynamics of Horizontal Axis Wind Turbines, with the ultimate objective of making Wind Energy more competitive.

Navier-Stokes equations govern Aerodynamics, which is an unresolved and very active field of research due to the current inability to capture the relevant the scales both in time and space for nowadays industrial-size machines (with rotors over 100 m in diameter). Therefore, there is a need to aim at a combination of engineering and scientific models. The structure of this thesis is designed in accordance to the previous fact, so there are clearly two parts or approaches within the conducted research: zero dimensional models and CFD based analysis.

For those zero-dimensional and (computationally) cheaper approach, the efforts done were the next: visualization and side improvements of the BEM, an in-house code for pre-design purposes; designing an AeroElectric coupled algorithm to merge BEM with simple generator models that exclude details on electric circuits; designing a C++ code to study the dynamics of wind turbines coupling different component models; implementation of energy production based algorithms to optimize blades pre-designed with BEM, given the wind resources at the real location.

CFD-based analysis are meant to be the tools to design wind turbines in the future. Nowadays and for the years to come, they are and will be under ongoing research. The efforts done to this respect have been the next: exploration, implementation and analysis of Non-Inertial Reference Frame, Immersed Boundary Method, Sliding Meshes. Additionally, Adaptive Mesh Refinement and Wall Model LES methods have been explored too.

Both approaches (engineering models and CFD simulations) are very demanding in terms of geometry treatment. Dealing with geometry is a must. It is a crucial part. In fact, it is the first step that enables those studies and it is usually very

time consuming and also a tricky aspect. Therefore, the implementation of the right geometry-management environment was also developed and presented prior to the two aforementioned approaches. The main points of study in the geometry block were four. First, the management of the information that configures an abstract airfoil from the computational point of view. Second, the automation of the process to design the blade CAD for both meshing and manufacturing purposes. Third, the construction of a platform to manage the geometry of the different components of a wind turbine as required by the Immersed Boundary Method. And fourth, the development of a faster algorithm to perform the computation of the distances from a generic point to the blade STL file which was also required, once again, in the Immersed Boundary Method.

Summing up, the geometry aspects, the engineering models and the CFD analysis have been applied not only to scientific projects and experiments as the Offwindtech, NREL and MEXICO, but also to a commercial R&D case in collaboration with Termo Fluids S. L. and Enair S. L.

Contents

Dedication	i
Foreword	xi
Abstract	xv
1 Introduction	1
1.1 Context of Wind Energy	1
1.2 Wind Energy Technology	7
1.3 Wind Energy’s Market	10
1.4 State-of-the-art in Aerodynamics	13
1.5 The arrangement of this thesis	17
References	19
2 Geometry handlings	21
2.1 Airfoil digital conceptualization	22
2.2 Automatic blade generation	31
2.2.1 The AutoBlade process	32
2.3 STL management platform	38
2.3.1 Operations over STL objects	41
2.4 Distance function	49
2.4.1 Distance point-triangle algorithm	50
2.4.2 Combined effects of distance mesh-STL algorithm	51
2.4.3 Enhanced distance mesh-STL algorithm	54
2.5 Conclusions	59
References	60

3	Engineering models	61
3.1	Blade Element Momentum Code	65
3.2	AeroElectric code	68
3.2.1	Algorithm	69
3.3	Optimization	76
3.3.1	The optimization tool	77
3.3.2	The application to wind energy	79
3.4	Conclusion	87
	References	88
4	CFD models	91
4.1	Initial discussion	93
4.1.1	CFD software: TermoFluids	97
4.1.2	Meshing	99
4.2	Coriolis	103
4.2.1	Test 1: rotating empty cylinder	106
4.2.2	Test 2: crossed flow over simple geometry	107
4.2.3	Test 3: real WT rotor case	108
4.3	Immersed Boundary Method	114
4.3.1	Our Immersed Boundary Methods	115
4.3.2	From advection to direct computation	121
4.3.3	Results: Offwindtech, NREL and MEXICO	125
4.3.4	Conclusions	129
4.4	Sliding Meshes	130
4.5	Parked	137
4.6	Conclusion	142
	References	142
5	Conclusions and future research	149
5.1	Concluding remarks	149
5.2	Future research	151
A	Enair's project	153
A.1	Introduction	153
A.2	Original turbine: 3 kW	154
A.3	First turbine: 10 kW	157
A.4	Second turbine: 40 kW	164

Contents	xix
A.5 Conclusion	168

List of Figures

1.1	World Primary Energy Consumption in Mtoe (Million of tonne oil equivalent). Source: Schilling & Al. 1977 (modified).	3
1.2	Global cumulative installed wind capacity 2000-2015. Source: GWEC (modified).	4
1.3	2000-2015 evolution of the mix of installed power (in MW) classified by source in the European Union. Left, 2000. Right, 2015. Source: EWEA (modified).	5
1.4	Energy production by source in the European Union in 2015. Source: Eurostat (modified).	6
1.5	1980-2015 (mean) size evolution of land based HAWTs. Source: http://arstechnica.com/	10
1.6	Left: Top 10 countries in cumulative installed capacity by the end of 2015. Source: GWEC (modified). Right: Top 10 (turbine) manufacturers in the wind energy market. Source: JCR 2013 (modified) [10]	11
1.7	1998-2015 evolution of cumulative installed power in Spain in MW. Source: <i>Red Eléctrica Española</i> .	12
1.8	Breakdown of offshore CapEx costs based in German projects executed in 2012. Source: JCR 2013 [10] (modified).	13
1.9	Levelized Cost Of Energy of the main energy sources in the US. Source: IEA 2016 (modified).	14
1.10	Conceptual map of the contents of this thesis excluding introduction and conclusion.	18
2.1	STL file structure	23
2.2	STL cutting to obtain airfoil shapes across the blade	23

2.3	Problem description of the airfoil reordering. a) Group of (disordered) points as they are obtained from the cutting process. b) Result of the reordering proces.	24
2.4	Airfoil reordering algorithm. Step 1: extremis identification: a) computing distances to all points; b) couple TE-LE identified (but not who is who): c) computation of weighted angle indicator to determine whether A is LE or TE	26
2.5	Airfoil reordering algorithm. Step 2: a) chord line points discretization and search of their closest skin points; b) search of opposite board points; c) midpoints computation and drawing of the center line	28
2.6	Airfoil reordering algorithm. Step 3: a) identification of points; b) detail of components for the vectorial product that makes possible the discrimination: those outboard lead to vectors in different directions than those inboard; c) final reorganization of airfoils skin points	29
2.7	Airfoil reordering algorithm. Proof of concept: application to the real case of MEXICO experiment blade. Bottom: raw data. Top: effect of the algorithm	30
2.8	Final CAD model of the blade obtained with the "by-hand" approach employing Solid Works 2012	32
2.9	Technical details and content of the input and intermediate files used in the AutoBlade process	33
2.10	Airfoil modification steps within the AutoBlade process	34
2.11	Outcome of AutoBlade process: final cloud of points prior to ICEM curves and surface generation	36
2.12	Surface generation step and required elements in the AutoBlade process within ICEM scope	37
2.13	Final blade output of the AutoBlade process after ICEM performs all operations	38
2.14	Examples of wind turbine components in small (left) and large (right) wind turbines	39
2.15	Graphic scheme of the relevant virtual components for the aerodynamic study of wind turbines together with their main parameters- (*) yaw angle cannot be shown in this (vertical cut) perspective.	42
2.16	C++ object structure scheme of the relevant virtual components for the aerodynamic study	43
2.17	STL full reorientation process in a schematic way	45

2.18	The four steps of STL normal checking	46
2.19	Sequence of WT movement composition employing STL and the present geometry handling platform	48
2.20	Process of computing the shortest distance from a given 3D point to a concrete STL triangle.	50
2.21	Schematic 2D representation of distance computation of several point to STL triangle in a smooth convex object	53
2.22	Schematic 2D representation of conflictive regions for distance computation point to STL triangle in a sharp convex object	54
2.23	NEighbor Walking Algorithm (NEWA) proof of concept: first and second iterations over a real blade	57
2.24	NEighbor Walking Algorithm (NEWA) applied to three bodies: (left) cube; (middle) sphere; (right) real blade (with problematic and helping triangle heuristics)	58
2.25	The effect of NEWA together with the <i>opposite criterion</i> when applied over a rotating blade	59
3.1	Momentum theory and Betz Limit visualization.	63
3.2	Schematic arrangement of elements for BEM	64
3.3	BEM-user interface input-output file	66
3.4	Results of a BEM computation over a 40 kW blade	67
3.5	Schematic representation of the AeroElectric phases	71
3.6	Lift and Drag coefficients for the selected airfoil	72
3.7	Results of the AeroElectric code over a 40 kW blade	74
3.8	Wind characterization by means of Weibull Probabilistic Distribution with parameters 2 and 7	75
3.9	The twist and chord laws for the original 40 kW blade without the intervention of Optimus	80
3.10	Four images of the evolution of the genetic algorithm when employing TSR and C_l as design parameters for ideal laws of twist and chord	82
3.11	Evolution of the optimization algorithm in the chord optimization at the tip of the blade	83
3.12	Evolution of the optimization algorithm when dealing with the whole twist law at the same time	84

3.13 Map of individuals in generation 3 when computing the effect of limiting the chord and twist laws with the parameters MaxChord and MaxTwist, respectively.	86
4.1 The three levels of design of wind energy	92
4.2 Dynamic Stall (left) and Rotational Augmentation (right). The images are extracted from [3]	94
4.3 NACA 0012. Reynolds 50,000 at AoA 5° (bottom) and 8° (top), from [12]	95
4.4 Several examples of simulations run with TermoFluids: a) two-phase flow [24]; b) natural convection in high-temperature solar receivers [25]; c) compressors [26]; d) air curtains [14]; e) HVAC for automotive industry; f) combustion [27].	98
4.5 The 32 GB RAM of the 12-CPU machine employed in such meshes usually collapse during massive computing (left). The 64 GB resulted enough but close to the limit (right).	102
4.6 Scheme of the distances and nomenclature between an inertial and non-inertial reference frames.	104
4.7 Test 2: schematic representation of the flow in a Non-Inertial Reference Frame across an excentrically rotating cube (with respect the cylinder axis).	108
4.8 A quick glance at the Offwindtech Blade geometry.	109
4.9 Superficial mesh of the Offwindtech rotor observed from different angles and perspectives.	111
4.10 Volumetric mesh of the Coriolis-based simulation in the Offwindtech rotor. Left: octree step. Right: delaunay step. Middle: far view of the computational domain.	112
4.11 CFD results of the Offwindtech rotor employing Coriolis method after roughly 20° simulation. Left: zoom at the rotor zone, reasonable initial flow conditions. Right: far view from the whole domain, unacceptable flow conditions due to scattered random velocity gradients.	112
4.12 Schematic arrangement of the IBM Direct Forcing approach.	116
4.13 Heaviside smoothing of the distance function depending on the value of the ϵ parameter.	117
4.14 The set of simple figures that were employed in the IBM testing.	118
4.15 Details of the distance function validation in the case of the pyramid.	119

4.16	Details of the distance function validation in the case of a real blade (NREL): a) errors due to the random assignment when the vertex criterion was not implemented yet. b) mistakes on the distance calculation even when the NEighboring Walking Algorithm was running. c) final approach with all the heuristics developed: successful results in comparison to the STL blade.	120
4.17	Distance function interface at 0.5 filtered distance in the case of the MEXICO's full wind turbine. Top-left: still with problems. Bottom-right: final approach error-free with acceptable geometric accuracy.	121
4.18	Left: comparison of the STL blade shape (a, yellow), with the initial geometric loading (b, blue), and the after-30° advection (c, white). Right: The effect on power due to periodic geometric reloading.	122
4.19	Application of the distance function to the full wind turbine of the Offwindtech project employing the Neighbor Walking Algorithm for all the components (a), incorporating the boxes to limit the nodes to which the distance functions is computed (b), and the results of the process in (c) and (d).	124
4.20	CFD results of the Offwindtech rotor: coherent turbulent structures. . .	125
4.21	CFD simulation of the air flow around the MEXICO wind turbine employing IBM: a) pressure field on both the slice and the turbine surface, b) velocity field at the center slice, c) proof of concept of yaw STL capacity and d) instantaneous velocity field on the dynamic yawing case.	126
4.22	Power evolution of the MEXICO case at 24 m/s and -2.3° pitch angle. CFD predicted power: 21 kW. Experimental: 30 kW.	127
4.23	Pressure distributions in the 15 m/s MEXICO case at different span-wise stations. At the top, the 35 and 92 % sections of the blade, the ones fitting better the results. At the bottom, the 25, 60 and 82 % sections presenting large, small and medium discrepancies.	128
4.24	Instantaneous velocity field in the Hexascreen chemical mixer.	130
4.25	Meshing process in the Sliding Mesh approach: a) geometry, b) rotor (moving) mesh, c) background (static) mesh.	131
4.26	Left: Combination of the background and rotor mesh into a common one. Right: details of the prismatic boundary layer around an average section of the blade.	132

4.27	Essential details of the Sliding Mesh concept: a) scheme of the background and rotating meshes, b) and c) effect of the displacement of one mesh with respect the other and the corresponding mirror nodes.	133
4.28	CFD snapshots of the Sliding Mesh approach.	135
4.29	Power evolution of the turbines employing Sliding Meshes. Left: case 3 from Table 4.4. Right: case 4.	136
4.30	Evolution of the mesh using AMR: a) initial, b) final, employing only boxes to define control volume sizes, c) final, employing the distance function criterion.	138
4.31	Detail of the AMR meshes on the Parked conditions case. Top: mesh detail. Bottom: instantaneous velocity field.	140
4.32	Vortex test to discriminate the source of the errors experienced under the AMR approach.	141
A.1	Scanning process of the 3 kW blade. Top: blade itself. Left: blade scanning. Right: blade CAD model for the scanning software.	155
A.2	Reverse engineering of the 3 kW blade CAD model delivered by the scanning process. Top-left: blade in STL format. Bottom-left: geometrical cuts along the blade seen from the axis perspective. Right: extrated values representing the blade model.	156
A.3	Left: several airfoils that were analyzed before selecting the final one for the 10 kW alternative turbine. Right: comparison between the Enair10 and the new airfoil aerodynamic properties.	158
A.4	BEM comparisons between the ideal blade (dotted lines) and the final selected one (solid lines) for a combination of pitch and Tip Speed Ratio (λ) values.	159
A.5	Twist and chord laws of the final 10 kW turbine blade design together with an image of its the blade shape.	160
A.6	Flag position analysis. Torque computations for the turbine in standstill conditions as a function of the wind speed in [m/s] and the pitch angle. Left: low winds. Right: extreme winds.	160
A.7	Comparison of the initial nacelle (left) and the final improved design (middle and right).	161
A.8	Images of the 10 kW machine. Left: ultimate blade treatment. Bottom: erecting process. Right: final layout of the turbine ready for testing and operation.	162

A.9 Pre-design and BEM-based design comparison for the 40 kW turbine, together with the initial design of the 10 kW turbine, and some real instantaneous measures in the experimental campaign.	163
A.10 Schematic representation of the dynamics of wind turbines algorithm. . .	165
A.11 60-seconds-long dynamic simulation of the 40 kW turbine with unbalanced initial angular speed and fixed initial pitch angle for the first half of the simulation.	166
A.12 Sequence of the blade passage in the 40 kW blade employing Sliding Meshes approach. Top: horizontal plane. Bottom: vertical plane perpendicular to the rotor plane.	168
A.13 Sequence of the blade passage in the 40 kW blade employing Sliding Meshes approach at the rotating plane.	169

List of Tables

1.1 Qualitative comparison between a hypothetical ideal energy source and the most common energy sources available today.	5
3.1 Effects (on power) of the chord limitation keeping the maximum twist constant.	83
3.2 Effects (on power) of the twist limitation keeping the maximum chord constant.	85
3.3 Final set of cases and conclusion of all the steps of the optimization phase.	87
4.1 Different approaches tested in the 3D CFD applied to wind turbines. . .	101
4.2 Test 1: comparison of the two employed approaches, Inertial and Non-Inertial Reference Frames, for the same case.	107
4.3 The two different approaches tested in the case of a 3D CFD of wind turbines.	115
4.4 CFD cases for Enair's 10 and 40 kW turbines.	134
A.1 XFOIL results of the 20 airfoil sections extracted from the scanned 3 kW blade.	156
A.2 Comparison of the results between XFOIL and CFD for the midspan 3 kW blade airfoil (Enair10) for the Reynolds = 400,000.	157
A.3 Comparison of the six final blade designs for the 40 kW turbine.	164

Introduction

The introduction chapter is arranged as follows. The first section deals with the contextualization of wind energy, the global energy demand, the qualitative comparison to the rest of energies and the evolution of the wind energy installed power. After this, in the second section, the wind energy technology will be introduced, where the evolution of the technology will be commented together with the most common approach to harness the inherent power in wind: Horizontal Axis Wind Turbines – although some other alternative are briefly explored. The next topic to be covered, already in section 1.3, is the wind energy market. A review of the size of turbines is presented and the top 10 countries and manufacturers are also provided. Additionally, this section covers a brief revision of cost structure and a quantitative comparison to the rest of energy sources. The purpose of the fourth section is to introduce the reader to the state-of-the-art of aerodynamics, which is the first step of the wind energy conversion process and the core of this thesis. Finally, in the last section, the thesis objectives are exposed and a brief introduction of the rest of chapters is pointed out.

1.1 Context of Wind Energy

In the beginning of everything, according to our more refined scientific theories available today, there was a Big Bang (a big explosion) that was the origin of everything to come later. From that point, the universe matter reorganized itself and it has been proven to be continuously expanding – i.e. occupying more space. Accepting this event as a fact and trying to use our intuition combined with gravitational theory, our best common sense would claim that the universe must slow down due to the attraction among matter.

However, this seems not to be the case. The universe is expanding and the reasons remain obscure to us. We have coined a term to refer to the forces that shape the expansion phenomena: dark energy. In fact, the composition of the universe is not at all in accordance to what a non-expert human being would say: matter as we know it (water, air, iron, beds, computers, etc.) occupies 5% of space, dark matter stands for 27% while dark energy is believed to represent the remaining 68% [1]. I believe that this is a self-explanatory piece of data which represents how much do we know about the universe today, and more precisely about energy: nearly nothing.

Our understanding seems to be scarce in this fundamental part of the world, but we are continuously learning and widening our knowledge about it. Now, let us leave aside the far-fetched field of the universe and the dark energy to focus on a more comprehensive scope. We know that the sun is responsible, directly or indirectly, for most of our sources energy. In this sense, wind energy is not an exception: the average sun power striking the Earth is about 174,000 *TW* from which 86,000 *TW* actually reach the land, and 2 – 3 % of it converts to wind energy due to the unevenly heating of water, land and air masses. In 2000, the United Nations Development Program [2] carried out the World Energy Assessment in which they stated: "the rate of global commercial energy consumption is thousands of times smaller than the energy flows from the sun to the Earth." This means that, if our technology, driven by cutting edge scientific knowledge, was at the point to take advantage of 0.1% of our planet potential, we would have enough as to satisfy the global energy demand.

From 2000 to 2016, the worldwide energy consumption has grown around 30%. Another interesting piece of data according to the Global Energy Statistical Yearbook [3] is that from 2012 to 2015 the world energy consumption has remained in the range of 13300 Mtoe (Millions of Tone of Oil Equivalent energy), which is around 560 EJ (ExaJoules, 10^{18}). Similarly, the International Energy Agency quantified this magnitude in 530 EJ in year 2012 [4]. With all this figures in mind, we can infer that international organizations specialized in energy observation and studies coincide in the analysis of the current worldwide energy consumption.

Having said that, it would be interesting to have a look into the evolution of energy consumption of the whole humanity. Figure 1.1 leaves no room for doubt: the energy demand is on a clear growing pattern, especially if we take into account the trend of the last ten years. The discussion might be on the type of pattern but the growing trend is undeniable. In fact, OCDE countries experience a slight increment of energy demand but they are more or less stable, while, non-OCDE countries are growing faster: their consumption rate is twice as much as 20 years ago.

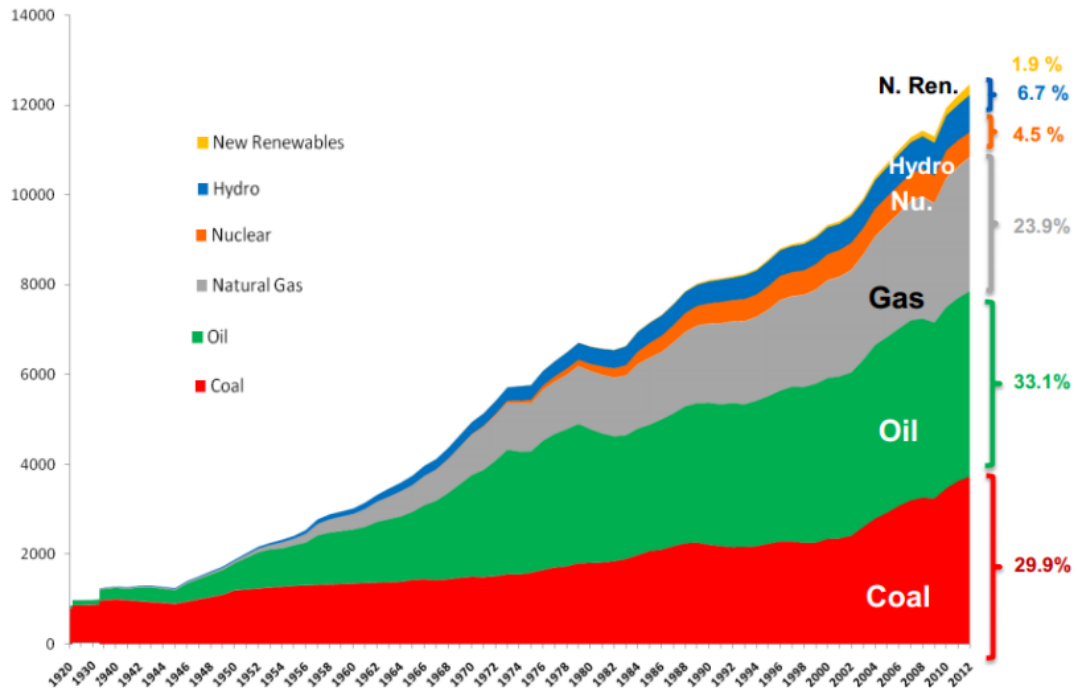


Figure 1.1: World Primary Energy Consumption in Mtoe (Million of tonne oil equivalent). Source: Schilling & Al. 1977 (modified).

The energy sector is complex and several thesis could be carried out to analyze it. The purpose of this chapter is not the detailed examination of the field but to provide enough information as to understand the context of wind energy. Furthermore, the main use of analysing the current and past situations is to be able to predict as accurate as possible the future in order to build up successful energy policies. In my mind, the most interesting data from the future predictions of the International Energy Agency for 2040 are the next three: i) the total energy demand grows 33%; ii) there will be a 70% lift in electricity demand; iii) and around 550 million people will still not have access to electricity. Those figures are just a prediction, but if the trend follows this path, it will lead us to a scenario in which we should have an effective plan to handle the situation.

Now, it is time to examine the Wind Energy panorama. The Global Wind Energy Council (GWEC) provides Figure 1.2, in which we can observe the trend of the total

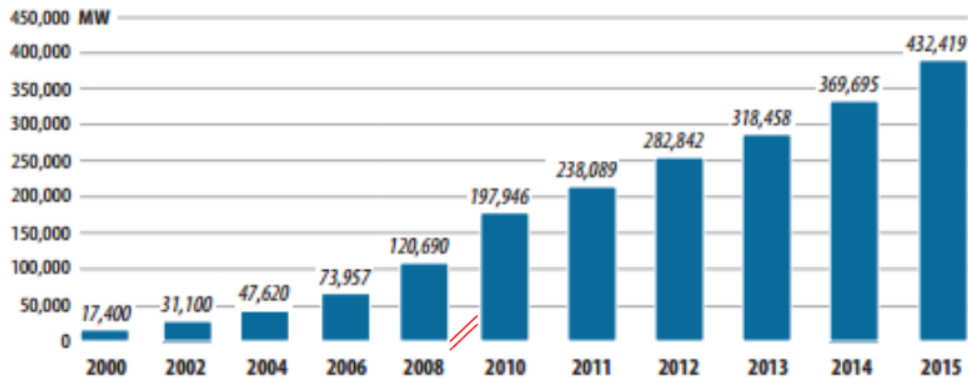


Figure 1.2: Global cumulative installed wind capacity 2000-2015. Source: GWEC (modified).

installed power in the world. As we can see, there is a clear growing trend. Focusing on the last 10 years, we observe an annual growth between 15 and 20%. At the end of 2015 the total power was 432.4 GW while by the end of 2014 was 369.7 GW, this means a 17% growth with the most recent data available.

In the specific case of the European Union (check [5] for details), the total installed wind power in contrast to the rest of energy sources has experienced an interesting growth: from 2.4% in 2000 to 15.6% in 2015, see Figure 1.3. When talking about the energy production, the contribution of wind energy in 2015 has been about the 10% of the total EU energy production, see Figure 1.4. Also, the energy production of wind energy in Spain was about 21.2% in 2013, and 20.4% in 2014 according to *Red Eléctrica Española*.

In the field of psychology, especially when dealing with emotional intelligence and/or leadership, it is common to do an exercise known as the Ideal Self. This is a self-knowledge exercise to achieve a clear picture of what we want to do with our lives, which was originally proposed by Richard Boyatzis in the Intentional Change Theory [6]. I believe that this kind of exercise can be adapted and replicated in many different fields, and in particular to the Ideal Energy source. What would it be the best imaginable energy source? If our dreams become true and anything can be asked for that energy source, how would it be? I have tried to do this exercise and at the same time to qualitatively compare this ideal source of energy to the most common sources available today. The outcome of this exercise can be consulted in Table 1.1.

Many parameters can be chosen to describe an ideal energy source, but I have just

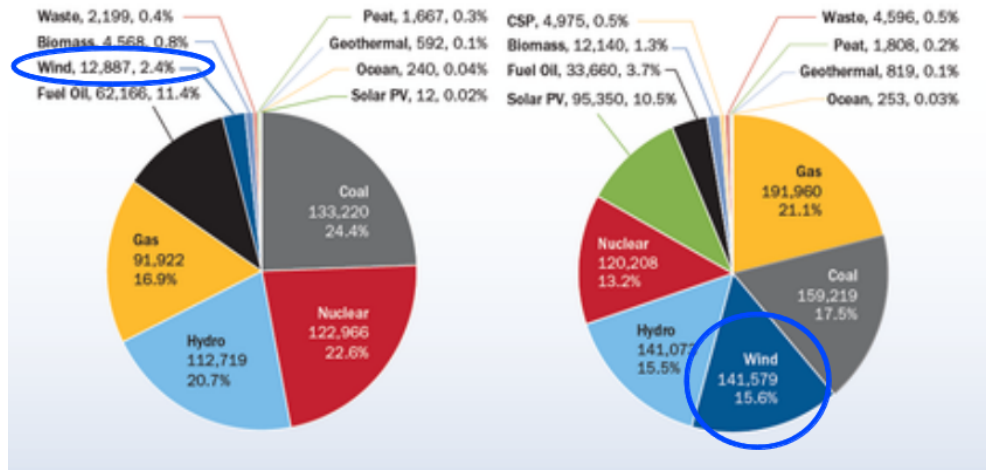


Figure 1.3: 2000-2015 evolution of the mix of intalled power (in MW) classified by source in the European Union. Left, 2000. Right, 2015. Source: EWEA (modified).

Category	Criteria	Ideal	Coal	Nuclear	Solar	Hydro	Wind
Space	Location	anywhere	flexible	security	sun	water	wind
Space	Size	zero	large	large	sizable	large	sizable
Energy	Production	infinite	high	very high	limited	high	limited
Energy	Input	none	coal	uranium	sun	water	wind
Energy	Output	none	CO ₂	radiation	none	none	none
Energy	Damage	zero	pollution	cooling	visual	visual	visual
Finance	LCOE	null	low	very low	medium	medium	medium
Finance	CapEx	none	medium	huge	medium	huge	high
Finance	OpEx	null	medium	high	low	low	low
Other	EROEI	infinite	high	low	quite low	quite high	medium
Other	Uncertainty	null	low	very low	high	medium	high
Other	Time	immediate	short	short	uncertain	very short	unknown

Table 1.1: Qualitative comparison between a hypothetical ideal energy source and the most common energy sources available today.

selected a few of them, the ones I think that are more relevant:

- **Location:** flexibility on the location of the power plant or the need of any special or specific requirements with respect its location.
- **Size:** this category mirrors the minimum size of a power plant that leads to

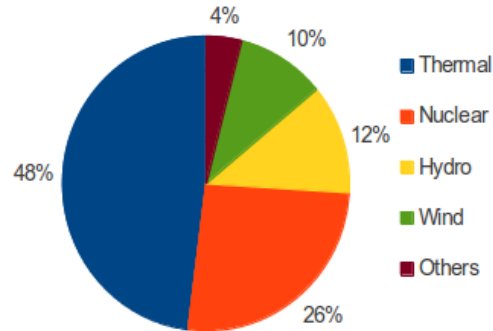


Figure 1.4: Energy production by source in the European Union in 2015. Source: Eurostat (modified).

competitive costs when facing massive energy production.

- **Production:** potential production if the source of energy would be scaled as much as reaching the limit that the Earth can handle, i.e. maximum space, maximum heat produced, etc.
- **Input:** the type and quantity of any material or immaterial requirements for the plant to be in operation.
- **Output:** added products or outcomes of the energy production process.
- **Collateral damage:** the undesired effects of the process that are inevitable due to the operating principle or nature of the source of energy.
- **Levelized Cost Of Energy:** ratio between the sum of costs over the whole lifetime of the power plant and the total amount of energy produced.
- **Initial Capital:** (CapEx) required investment before the power plant is to start operation.
- **EROEI:** the Energy Return On Energy Invested, or the ratio between the amount of energy produced and the energy used to run the energy production process [7].
- **Maintenance Cost:** (OpEx) monetary cost associated to repair and / or replace parts, pieces and systems due to flaws or deterioration during the lifetime of the plant.

- **Production uncertainty:** level of unpredictability in availability of the source of energy in the worst case scenario across the whole lifetime.
- **Production time:** the time lapse between the moment the decision to switch on the energy production and the moment that the actual production starts.

Table 1.1 can be interpreted in many ways, but I want to highlight two important aspects. First, comparing all the available sources of energy today, we observe that they all have different strengths and weaknesses, i.e. the high EROEI of hydro vs. its large visual impact; the huge potential of specific energy of nuclear vs. its dramatic long-lasting collateral damage of radiation; the renewability of solar and wind vs. their unpredictability or unreliability throughout time scales. Therefore, there is no energy source today with all pros and no cons. And the second aspect is, that although it might seem to us unrealistic, the concept of the Ideal Energy Source must inspire engineers, scientists and our society to keep working to make it possible. Today, the depicted Ideal Energy seems a far-fetched, magical and unattainable option, just as many technologies available today would have appeared to the humans living a thousands of years ago.

1.2 Wind Energy Technology

As its own name suggests, wind energy is a source of energy based on the existence of wind. Wind is the phenomena observed in the Earth's atmosphere consisting in the displacement of air masses (mostly) caused by the effect of uneven solar radiation across air, land and water bodies. The differential heating leads to changes in air density that imply the rising of air in a given location that forces the reorganization and migration of the air around those regions. Actually, the ultimate reason why wind exist is the pressure difference, and it can be also provoked by other reasons. For instance, the Earth rotation also causes the wind to be deflected due to the coriolis effect. Whatever the reason of the origin of wind, the operating principle of this source of energy is the transformation of the kinetic energy of the wind into electricity.

Many technologies have been proposed to harness the power of wind. The first machines to operate with wind are known from 500 DC. The well-known Netherlands windmill concept dates from 1850. With time, the wind energy market has adopted the (now typical) upwind 3-bladed Horizontal Axis Wind Turbine (HAWT), which is the technology that dominates the market today. However, there are other power production options such as the VAWT (Vertical Axis Wind Turbine) that take their small but interesting market share. The main benefit of VAWTs when compared to HAWTs is

that they are good at harnessing wind coming from all directions, they take advantage of any wind direction because they are omni-directional by nature while HAWTs must detect the wind direction and align their axis accordingly. VAWTs have many strengths: they are easier to manufacture; they can be grouped closely; they are lighter; and they perform good in high turbulence environments. On the other hand, they face three notable inconvenients: the dynamic stall (exposed in detail in Chapter 4, page 94); they operate closer to the ground where wind is weaker; and its operating principle is highly fatigue prone, especially for the blades. However, the crucial aspect that made HAWTs win the battle to VAWTs is their scalability, i.e. the ability to be scaled to produce greater amounts of energy per machine. Despite that, recently, some organizations such as Sandia National Laboratories or NREL are getting interested in studying VAWTs in depth to revalue them and take advantage of their strenghts.

Beyond the HAWT - VAWT polarization, there are many other alternatives, some of them innovative and some others even promising. For the sake of brevity, I will cover just three of them, which have something in common: I knew about them for the first time in the 2014 and 2016 editions of the congress *The Science of Making Torque of Wind*. The first one¹ consists in the combination of two ideas: downscaling and drones. Their objective is to make wind energy accessible to Sub-Saharan Africa using drones that, while flying, capture the wind energy in order to load a battery and deliver the energy to the people at ground. Their market also extends to remote areas, yachts and others. The second one is an original concept for offshore wind farms, where turbines do not make the conversion from mechanical energy to electricity in the nacelle. Instead, Antonio Laguna and his colleagues from TU Delft, propose a centralized production. All turbines in the wind farm contribute to pressurize the working fluid of a common hydraulic system that generates electricity in a single pelton-like turbine, which is common for all the wind turbines in the plant. The main advantage of this approach is the reduction of tones of materials in the turbine components, more concretely in the nacelle and tower. The third and last aspect that I want to highlight is the Makani concept². The idea behind that company is to use a kite instead of a usual HAWT in order to harness the wind energy. The blade-shaped kite flights in a vertical plane, describing a circle, and uses the flying speed as the wind input to power the eight smaller rotor-turbines that work as propellers, when landing and taking off.

Those three concepts are evidences of the diversity of ideas in the wind energy sector, but as it was said before, the great majority of the market is governed by HAWTs. Thus,

¹more information in <http://www.aenarete.eu/>

²more information in <https://x.company/makani/>

the aim of this thesis is to contribute to improve their aerodynamics. HAWTs have four main components: the blades, the hub, the nacelle and the tower. The blades are the knife-shaped objects that rotate in a plane (neglect pre-cone angle) perpendicular (neglect tilt angle) to the wind direction, in order to enable their airfoil-shaped cross-sections to operate at the best possible position to produce the greatest rotor torque possible. The hub is the part of the rotor where blades stem from, and also where the pitch system (used to rotate blades along their axis) also is nested. The nacelle is the space where the mechanical and electrical elements are located. Those elements enable the machine to do the conversion from mechanical torque of the shaft into electricity by means of the gear box and a electric generator coupling (usually a Permanent Magnet Generator). And finally, the tower is the structural component that holds the machine at the right height and transfers the loads with safety to the ground.

The equation that bounds the power produced by a wind turbine (Equation 1.3) is derived from the mass flow rate (Equation 1.2) and the specific kinetic energy (Equation 1.1) equations:

$$E_k = \frac{1}{2}v^2 \quad (1.1)$$

$$\dot{m} = \rho Sv \quad (1.2)$$

$$P = \frac{1}{2}C_p \rho \pi R^2 v^3 \quad (1.3)$$

where, E_k is the specific kinetic energy; \dot{m} is the mass flow rate of a fluid of density ρ , when flowing at v speed through a perpendicular surface S ; P is the power production of a wind turbine of rotor radius R with a power coefficient of C_p (meaning its efficiency).

As Equation 1.3 suggests, the most important parameters are the wind speed and the length of the blade because they follow square and cubic laws, respectively. In fact, this is the reason why the size of HAWTs has experienced a dramatic growth. Figure 1.5 shows the evolution of this growth through the last thirty years. As we can see, the average wind turbine rotor size in 2015 was 102 meters in diameter leading to a rated power of 2 MW. Up to date, the largest wind turbine blade is manufactured by the market leader LM Wind Power, and it measures more than 88 m (176 m in diameter), which is nearly as large as a soccer pitch or 7 typical shipping lorries.

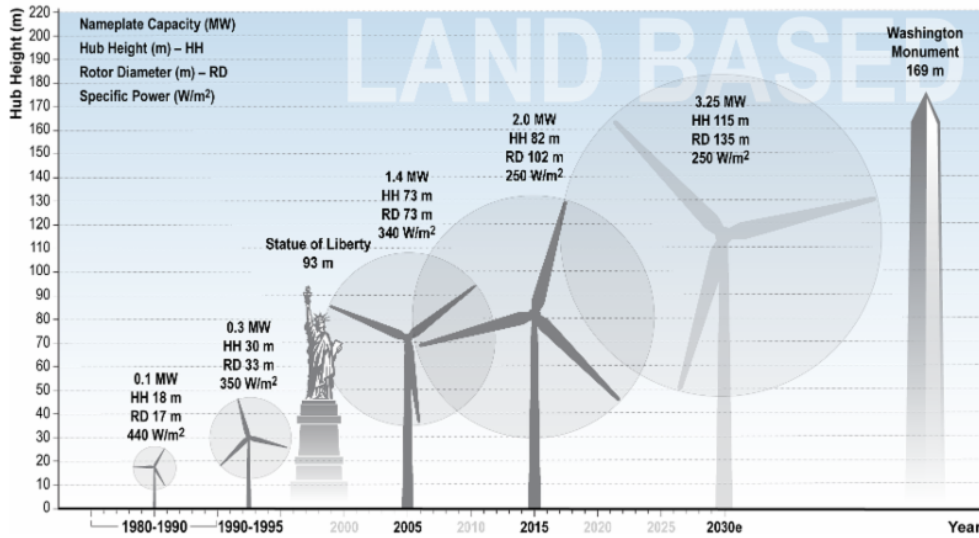


Figure 1.5: 1980-2015 (mean) size evolution of land based HAWTs. Source: <http://arstechnica.com/>

1.3 Wind Energy's Market

The World Wind Energy Association (WWEA) regularly provides a report [8] with an exhaustive analysis on the wind energy market. From the 2012 and 2014 report, we can have a closer look into the market. An interesting first indicator is the top 10 countries in installed power. In this respect, the GWEC [9] provides the most recent data. In Figure 1.6, we observe that China and the USA are the clear leaders having more than 50% of the market together. Then, the next three positions are for Germany, India and Spain, respectively.

When analyzing the market by regions, Europe and Asia historically account for the 75% of the market since 2010, while North America remains around 20%. In 2014, Asia overtook Europe's leadership and now, nearly 40 % of market is Asian while 35 % is European, which was in the reverse way in 2011. On the other hand, the offshore market is clearly dominated by Europe where UK, Germany, Denmark and Belgium are at the top of the rank, with the clear domination of the UK having 54 % of the offshore market alone. Another interesting piece of data is the specific installed power for both *per land* and *per capita*. Once again, Germany and Denmark perform great in both

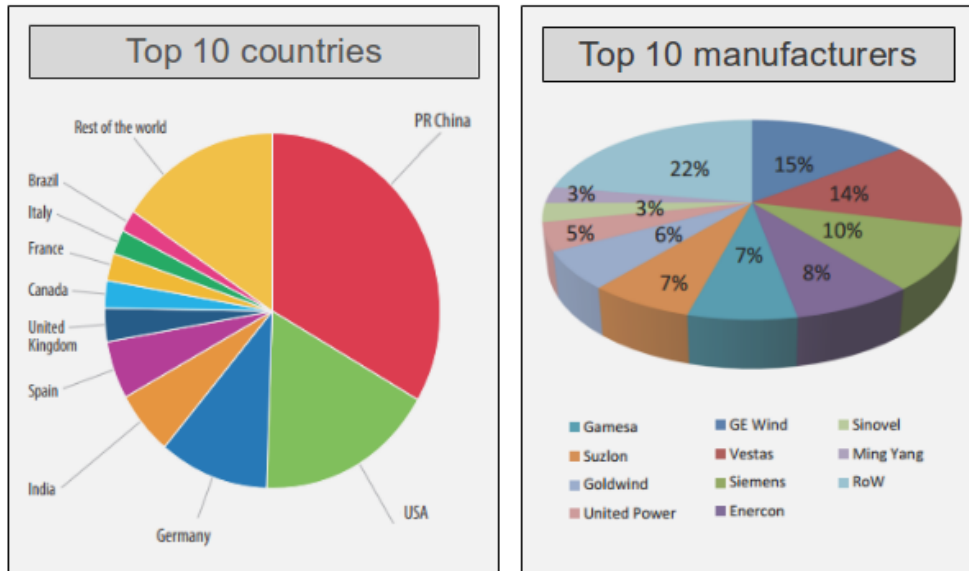


Figure 1.6: Left: Top 10 countries in cumulative installed capacity by the end of 2015. Source: GWEC (modified). Right: Top 10 (turbine) manufacturers in the wind energy market. Source: JCR 2013 (modified) [10]

of these indicators: Germany has installed $113 \text{ kW}/\text{km}^2$ and Denmark counts on $879 \text{ W}/\text{capita}$. This means that if the entire world had those density indicators, the whole energy demand could be supplied with wind energy.

The last aspect to highlight in this market-by-countries analysis before moving to costs of energy is the fact that politics and social aspects matter. Take the case of Spain: the country has proven to be a leader not only in wind energy but in renewable energies (in general) for the years between 2000 and 2010. However, the political decisions made during the last part of that decade ruined the evolution of the sector and provoked wind energy to reach a plateau in 2012 as it is observable in Figure 1.7. The situation is as dramatic as nearly stopping any new installation: just 27.5 MW were added in 2014 (0.1% increment). However, the 25th of March of 2014, a new energy production record was set by the Spanish wind energy sector: 46.9% was covered with this source. In addition, 20.4% of the annual demand in 2014 came from wind energy. Thus, in the Spanish case we observe the two sides of the same coin: on the one hand, the very good results of having worked well over time, and in the other hand, the block of the evolution

which will make Spain to stuck on the point it was in 2012.

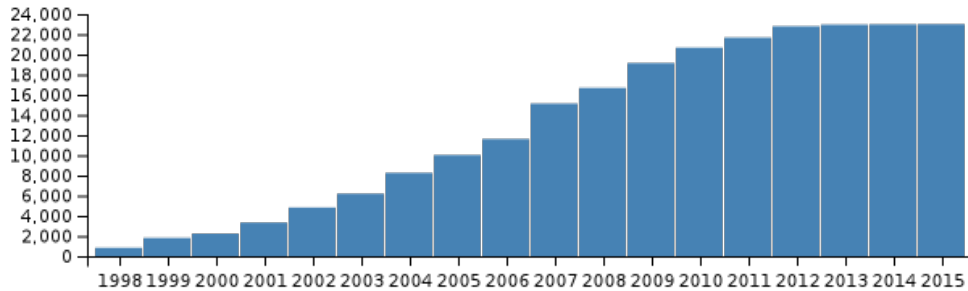


Figure 1.7: 1998-2015 evolution of cummulative installed power in Spain in MW. Source: *Red Eléctrica Española*.

Now, it is time to focus on analysing the costs of wind turbines. In this respect, references [10] and [11] provide an interesting point of view, but it must be said that it is quite hard to have access to real data due to the high secrecy and lack of transparency among projects and companies of the field.

Anyway, what it appears to be clear is the huge difference in the costs between projects, and specially between countries.

The trend of the growing size of HAWT seem to be uncoupled from specific costs because they remain constant around the 1,000 €/kW. What's more, this indicator has followed a slight but steady decrease along the recent years. One of the reasons for this uncoupling is the continuous improvement of the electricity production efficiency based on the fact that the amount of energy produced per unit of area of rotor has improved significantly. Wind energy is a capital intensive source of energy. As much as 75 % of total lifetime costs are within the first year of operation, and a huge part of them correspond to the wind turbine itself: around 75 % (in onshore projects; for offshore, the relative cost of turbine is lower, around 30 – 40 %). In 2003, the LCOE was about 6 – 8 c€/kWh in low average wind speed areas, while 4 – 5 in good regions, but we need to bear in mind that good areas are already occupied by current wind farms. Operation and Maintenance costs are confined in the range of 2.2 and 5.3 c€/kWh, strongly depending on the age of the machine: the older the machine, the costlier the operation and maintenance.

Offshore projects have evolved as much as to conquer broader waters as time goes on. For instance: the evolution in the wave resistance has been from 1.25 to 2.6m from

1996 to 2014 as well as to enable wind turbines to run at wind speeds of 10 to 15 m/s. Despite the good news of this fact, the offshore market is still waiting for a sufficient competition: Siemens has an overwhelming domination in the sector. Figure 1.8 shows the breakdown costs of an offshore wind project. Nowadays, the average wind farm is installed around 30 km from coast at waters of 20 m in depth, and it takes between 4 to 18 days per turbine.

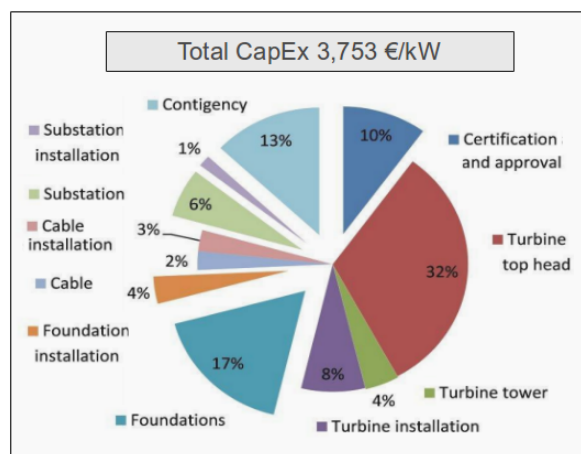


Figure 1.8: Breakdown of offshore CapEx costs based in German projects executed in 2012. Source: JCR 2013 [10] (modified).

Regardless of its inner competition, wind energy has to compete with the rest of energies in terms of LCOE. In this respect, the most recent piece of data that the author was able to find is shown in Figure 1.9. According to this data, the position of wind energy is quite good and has perspectives to improve further but the competition for the energy market is fierce since many of those sources cost between 5 and 10 c\$/kWh.

1.4 State-of-the-art in Aerodynamics

The goal of designing a wind turbine is to produce the maximum amount of energy at the lowest cost possible. The contribution of aerodynamics to this aim is to select the right combination of size, shape and materials when designing the machine. Special emphasis is put on the blade design without forgetting important aspects as location, environment, wind behaviour, tower and nacelle shapes, etc.

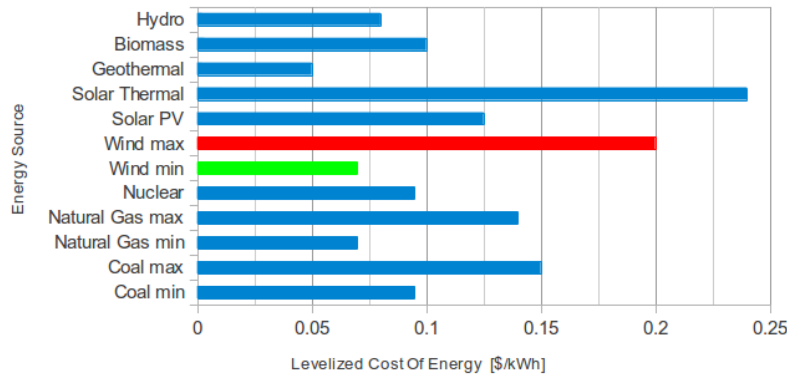


Figure 1.9: Levelized Cost Of Energy of the main energy sources in the US. Source: IEA 2016 (modified).

As exposed before, the origin of wind is the differential heating of the Sun across the Earth surface together with the coriolis effect due to the Earth rotation. Several other factors such as orography or sea-land breezes also affect the phenomena, that in either global or local scale make wind strongly unpredictable. Wind changes are so dynamic that wind turbines need a regulation system. Pitch control has become an essential part of the turbine, although there are some machines, especially with small wind turbines (say, below 40 kW), that still used stall control as a regulation mechanism. When winds are below 5 m/s there is not enough power as to run the production³. Likewise, when registering strong gusts, typically above 25 m/s, machines stop because it is not economically feasible to design them to work in such conditions (and withstand the strong loads associated). Between these cut-in and cut-out wind speeds, there is a physical limit to energy transformation: the Betz limit, which is 59.2%. It is a matter of logics that the whole kinetic energy cannot be harnessed, because it would imply that the air would stop right after the rotor.

Due to the changing nature of wind, the turbines must be able to work in extremely unsteady conditions. Therefore, each infinitesimal airfoil-shaped section of the blade along its radius must be working at a point in the $AoA-C_l$ ⁴ curve as to have a good aerodynamic performance but also having this characteristic unsteadiness in mind. It means that the actual working point is between the maximum lift and the maximum

³although it might be designed to start at 3 m/s

⁴Angle of Attack – Lift coefficient

lift-to-drag ratio. This is easier to say rather than to fulfill because the unsteadiness implies a continuous change of the Angle Of Attack. The AoA variation is caused by the change in direction of the apparent wind speed relative to the section might lead to the dynamic stall phenomenon [12], which is not well understood yet. The same is true for the rotational augmentation effect that the airfoils experiment, which is basically the observation of a delay of the stall mechanism with respect the behavior of the static airfoil (same Reynolds number, non-rotation). It is believed that the coriolis effect is responsible for the attachment of the boundary layer which ultimately implies the stall delay. Nowadays, this is an active field of research. Another interesting aspect of aerodynamics is the huge sensitivity (about 40% variation) of drag and lift coefficient when comparing the same airfoil in clean and dirty conditions. This fact, together with fatigue and deterioration of other components, might be the responsible for the continuous decrease of the overall wind farm productivity: 100% in the first year, 97% in the second, 96% in the third, etc. as presented in the [13] during the Torque 2016 congress.

As always in designing stages, we never have just one aspect in mind, instead we try to fulfill several criteria at the same time. This is the case of wind turbines too. Aerodynamics are crucial but cannot be taken as the only reference for design. In fact, the key is to couple the aerodynamic and structural efficiency to achieve a good compromise between the two antagonist aspects: a perfect aerodynamic design will lead to engineers to shape the airfoil sections as thin as possible to make them glide through the air flow. However, from the structural point of view, a blade behaves approximately as a cantilevered beam which would require sections to be as thick as possible to resist the loads. Here is the need to find the balance between this two opposite poles to resist the wide range of loads that the machine suffers: static, steady, cyclic, transient, braking, stochastic, impulsive and resonance induced ones [14]. Some of those loads are crucial for fatigue analysis, because they are exposed to roughly 10^9 cycles along the usual 20-year lifetime of the turbine (assuming a good location and design, and a 90% loading factor, i.e. 10% idle time). Usually, the most widely used material to withstand those loads is Fibre-Reinforced Plastic. The reason for this selection is the high specific properties (ratio to weight). In the end, the aerodynamic-structural trade-off implies devoting slender airfoils at the tip (tipically midspan outwards) where aerodynamics are more important than structural targets and a progressive increment of the thickness towards the root, where structural strength is indispensable. This approach leads to the design of blades with thicker airfoils at the root (above 15%), where the bending moment is greatest and the leverage is lower, but not so thick as to contribute to a massive

flow separation in this area. Additionally, root sections count on generous amounts of reinforcement to resist, mostly, the flapwise bending moments.

In this paragraph, some of the aspects of the aerodynamic design of a wind turbine will be stated under the rule-of-thumb approach. Roughly speaking, the ideal chord length of every station is given by a law that is roughly proportional to $1/R$ where R is the radius of the station, although there must be a maximum chord allowed for constructive purposes. The same type of law applies to the twist, but once again, for constructive reasons, the twist is limited to a 15 – 25 degrees range from tip to root. Moreover, there are two angles⁵ that become important to increase the clearance (distance from blade tip to tower) which are crucial when building upwind oriented rotors. The *tilt* is the angle between the horizontal plane and the wind turbine shaft axis, and the *cone* is the angle between a perpendicular plane to the shaft axis and the blade axis. Another interesting aspect is that blades are usually prebended not only for the sake of clearance increasing but also to allow for a controlled aeroelasticity behavior. Both angles, tilt and cone usually range from 2 to 6 degrees. Another interesting angle is known as *yaw*, which measures the misalignment between the rotor axis and the prevailing wind direction; the typical in-operation non-corrected yaw values for upwind rotors range from 2 to 10 degrees. The yaw reduction (and eventual elimination) is strongly sought since yaw misalignments imply a diminished energy production and a load imbalance.

All in all, the current state-of-the-art of aerodynamics is acceptably good as to design efficient wind turbines. Notwithstanding, the room for improvement is huge because this discipline is relatively young with respect to other scientific. High expectations are confided to CFD in the years to come, and that is the reason why the two biggest experiments performed in the field of wind energy are focusing on obtaining benchmark data for code validation. Those experiments are the NREL phase VI and the MEXICO.

The NREL experiment was completed in 2000, in the NASA Ames facilities. It is known as NREL Unsteady Aerodynamics wind turbine test (more info in [15]) where a two-bladed 10 m in diameter turbine was tested in different conditions inside the largest wind tunnel available at that time (cross-section of $24.4 \times 36.6 \text{ m}^2$). The second experiment is known as MEXICO (Model rotor EXperiments In COntrolled conditions), which started in 2008. It was divided in different phases, some have already ended and some others are still ongoing. The MEXICO experiment, performed on a three-bladed wind turbine of 4.5 m in diameter, provides reliable data with yawed and non-yawed flow conditions at different wind speeds (more information in [16]).

⁵for visual information check Figure 2.15 in page 42

1.5 The arrangement of this thesis

This thesis is arranged in five chapters and one appendix.

The present chapter is the first one, where we have covered the next content: *context of wind energy*, identifying the position of wind energy in the broader map of the energy production; the *wind energy's technology*, being HAWT the most common approach to massive energy production; the *wind energy market* where we have seen that, although being a *young*, wind energy is mature enough as to be trustable – keep in mind the 20 % growth rate of the last years; and finally, a short introduction about the *state-of-the-art* has been just provided in the previous section.

As the saying goes: *an image worths a thousand words*. Based on that, the author has considered useful to include a conceptual map with the key words and images that summarize at a first glance the content of this work. In Figure 1.10, all Chapters (in red color) stem from the [*Thesis: HAWTs*] title (in yellow color). Then, the scheme contains the main words of the sections's titles from Chapters 2, 3 and 4 as well as Appendix A, in green color. The most important images summarizing the work done are shown next to the corresponding labels. Notice that the objective here is not to provide each individual figure with a neat format, to do so, detailed comments and appropriate scales are used throughout the document. Instead, the pursued aim here is to provide a panoramic view to clarify what to expect from this thesis as a whole and also help to understand the arrangement.

Chapter 2 deals with the geometrical issues that represent the bottom line to develop the next two chapters. The geometry module is indispensable and, as shown later, it is the bottleneck during the Immersed Boundary simulations. The first aspect to cover in this Chapter is the development of an intelligent enough C++ object to handle the difficulties behind the automatic reconstruction of an airfoil shape. In the second section, it is explained how to build a code to automatically generate a blade CAD model already embedded in the 3D mesher ICEM. Section 2.3 exposes the methodology to generate a wind turbine computational model departing from their individual components modelled by an STL file. The final section of the chapter covers how to compute the distance function, in particular, from any mesh point to the wind turbine STL model, and in general, how to do so for the whole mesh with acceptable computing costs.

Chapter 3 is the first *computing* block of the thesis. It stands for the most common and useful engineering models when it comes to design wind turbines from the aerodynamic point of view. To begin with, the already existing Blade Element Momentum (BEM) code at CTTC was improved to be friendlier for the user and it was reshaped to

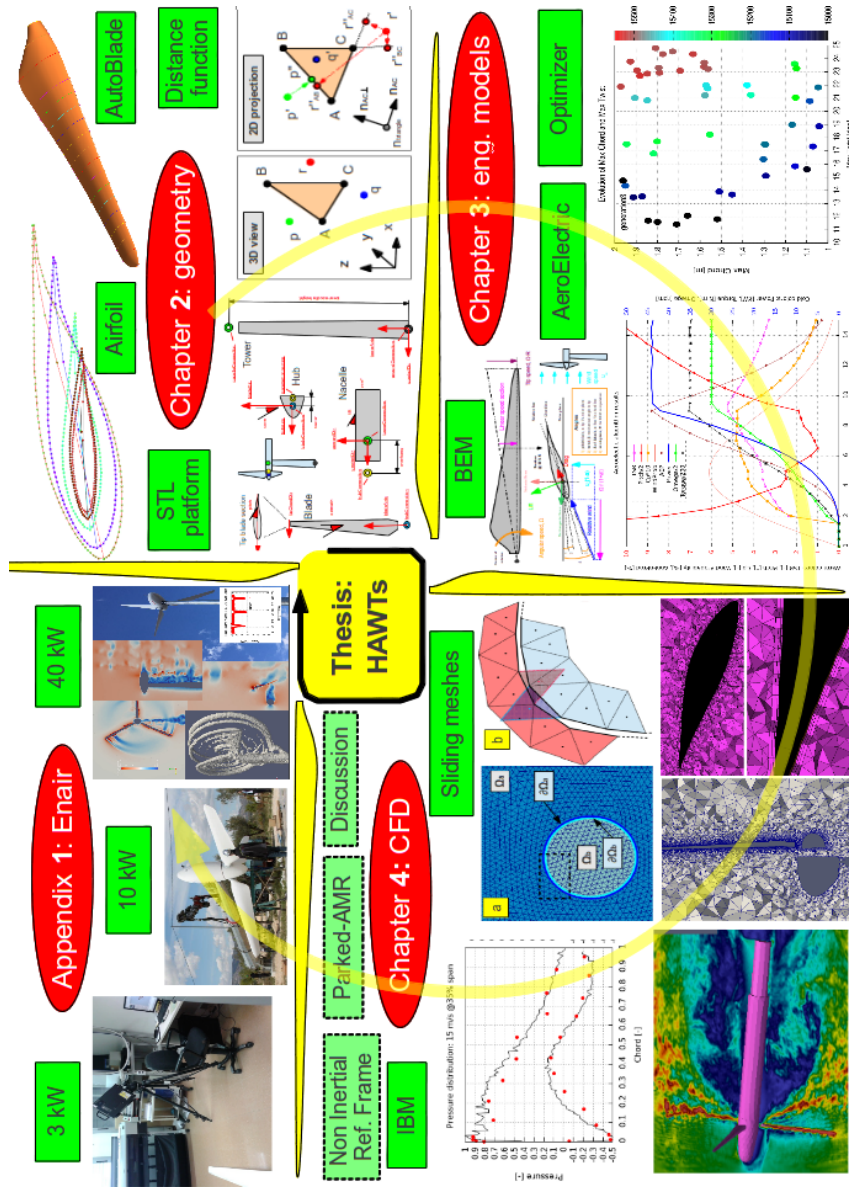


Figure 1.10: Conceptual map of the contents of this thesis excluding introduction and conclusion.

run massive computations. In the next section, the AeroElectric code is detailed. In here, the BEM method and code are enhanced by coupling in a simple but efficient way the behaviour of the electric generator. This allows the code to find the angular speed leading to the equilibrium of the machine. Additionally, a modifiable weibull distribution is included in order to enable the code to compute the Annual Energy Production of the turbine in a specific location. This enabled us to cover (in the final section of the chapter) an optimization tool in order to compare multiple blade designs and select the best one. The idea of the optimization tool is to find the right balance between the *ideal* aerodynamic design and the mechanical and structural constraints.

Chapter 4 is clearly the point at which larger efforts were dedicated. This chapter initially discusses the state-of-the-art of aerodynamics with special emphasis on the CFD studies around the wind turbine spatial scale. Next, three different approaches are implemented and tested with the help of the author's co-workers. The first approach uses the idea of Non-Inertial Reference Frames, which demanded the implementation of coriolis and centrifugal fictical forces. The second strategy shows how to employ Immerse Boundary Methods to the case of a 3D full wind turbine simulation. And the final and third method consists in the application of Sliding Meshes. Additionally, an extra section complements the whole picture by exposing the interest of the Parked Conditions case (static turbine) and the possibility to address it from the Adaptive Mesh Refinement and the Immersed Boundary Method.

The closure of this thesis is embedded in Chapter 5 in which the conclusions and future work are presented.

The last piece of unity of this thesis is presented in Appendix A, where a venture with a Spanish company of the wind energy sector is exhibited, based on the different phases that the project comprises. Three sections conceptualize the work done with Ingenia in collaboration with Enair S.L.: first, the reverse engineering of a 3 kW turbine is carried out; then, the design, optimization, installation and testing of a 10 kW turbine takes place; and finally, the design of a 40 kW turbine is done based on the experience of the smaller turbines together with the help of CFD simulations.

References

- [1] <https://science.nasa.gov/astrophysics/focus-areas/what-is-dark-energy>.
- [2] United Nations Development Program. World Energy Assessment. Statistical pocketbook 2015. .

- [3] <https://yearbook.enerdata.net/>.
- [4] International Energy Agency. World Energy Outlook 2012. .
- [5] European Commission. EU energy in figures. Statistical pocketbook 2015. http://ec.europa.eu/energy/sites/ener/files/documents/PocketBook_ENERGY_2015%20PDF%20final.pdf.
- [6] Richard Boyatzis et. al. An overview of intentional change from a complexity perspective. *Journal of Management Development*, 25:607–623, 2006.
- [7] Charles A.S. Hall et. al. EROI of different fuels and the implications for society. *Energy policy*, 64:141–152, 2014.
- [8] World Wind Energy Association. Annual report. <http://www.wwindea.org/world-wind-world-report-2012-launched/>, 2012.
- [9] Global Wind Energy Council. Global wind statistics. http://www.gwec.net/wp-content/uploads/vip/GWEC-PRstats-2015_LR.pdf, 2015.
- [10] Roberto Lacal Arántegui. 2013 JRC wind status report. Technology, market and economic aspects of wind energy in Europe. *JRC scientific and policy reports. European Commission*, 2014.
- [11] Paul Erik Morthorst. Wind Energy - The facts. Cost & Prices. 2, 2003.
- [12] Tonguag WEang. A brief review on wind turbine aerodynamics. *Theoretical & applied mechanics letters*, 2012.
- [13] Christian Back et. al. Full scale wind turbine test of vortex generators mounted on the entire blade. *Journal of Physics: conference series. Torque 2016.*, 753, 2016.
- [14] Kira Grogg. Harvesting the Wind: The physics of Wind Turbines. *Physics and Astronomy Comp papers*, 2005.
- [15] M.M. Hand et. al. Unsteady aerodynamics experiment Phase VI: wind tunnel test configuration and available data campaigns. *NREL*, 2001.
- [16] <http://www.mexnext.org/>.

Geometry handlings

Wind turbine simulations have to deal with plenty geometrical issues both when employing simplified or engineering models (as the ones to be presented in Chapter 3) or when running highly-detailed CFD-based simulations (as the ones that will be shown in Chapter 4). Therefore, a good geometry platform is a pre-requisite for any simulation modeling in order to cover this part successfully.

The ultimate interest of a wind energy project is the energy production (and its associated costs). From the aerodynamic point of view, the aim is to harness the maximum energy as well as ensuring the safety of the machine (and in particular the blades) in all loading scenarios. For this reason, the geometry (size and shape) of the blade is the crucial element and it is the center of all designing efforts. Therefore, final objective of engineers is to come up with a good blade design, which implies that the geometry definition is at the center of the process. The same is true for the CFD and high-level simulations. The final objective of running the simulation is to provide the performance of the machine in terms of torque, power and other indicators. The blade geometry is the final aspect to provide. Paradoxically, the geometry module for running CFD simulations is a strong requisite: the efforts to design and build a geometry platform from scratch are huge.

This chapter provides the explanation to build the platform that allows the wind turbine designer to deal with the geometry aspects of the process. First, the airfoil, which is not only the most elemental piece of the aerodynamics but also the center of the whole designing process, is digitalized with a robust tool that enables the code to flexibly read, design, print, cut and modify the essential parameters of the airfoil. Then, the *AutoBlade* code is presented. It requires the previously commented airfoil

conceptualization to design the shape of a blade and provide a CAD model valid for both manufacturing procedures and CFD simulations. The third block of this chapter deals with the developed platform to assemble the components of a HAWT using an STL file, including the blade geometry resulting from the previous section. Finally, the HAWT, modeled by STL objects, requires the coupling to the fluid dynamics part, and it is done by means of creating a distance field in the domain that provides the closest distance to the turbine.

Each of the four sections of this chapter aims at solving a specific problem required at some point in the computations of Engineering models (Chapter 3) and / or the CFD models (Chapter 4). Therefore, each section might be seen as trouble-shooter. For this reason the way to introduce each of them is problem-oriented from the beginning of each subsection.

2.1 Airfoil digital conceptualization

The problem definition is the next: there is a need to obtain an airfoil shape at a given section from the computational STL-based blade.

The situation presented above is key to analyze a CFD case based on the Immersed Boundary Method (IBM). For the blade performance analysis it is required to have the pressure distribution at some blade sections. For instance, in the MEXICO experiment, the blade stations at the 25, 35, 60, 82 and 92% of the radius were instrumentalized with pressure transducers to obtain the pressure distribution and monitor the performance of the blade. Thus, a tool to obtain pressure distributions must be incorporated to the CFD in order to validate the method.

As it will be shown in detail in Chapter 4, one of the approaches that were chosen for CFD analysis was the IBM. For the sake of following the content of the present chapter, the only information that needs to be revealed is the fact that an STL¹ file is employed to describe the geometry.

An STL file contains a finite number of 3D triangles. They are defined by three points and a vector representing the direction of the normal vector associated to the triangular surface defined by those three points. See Figure 2.1 to clarify this concept.

When running a CFD simulation in which a blade geometry is modeled following the STL approach, it is necessary to perform a series of cuts² over the 3D surface in order to obtain a set of 3D points to define the shape of the airfoil in that specific section

¹StereoLithography

²perpendicular to the blade axis

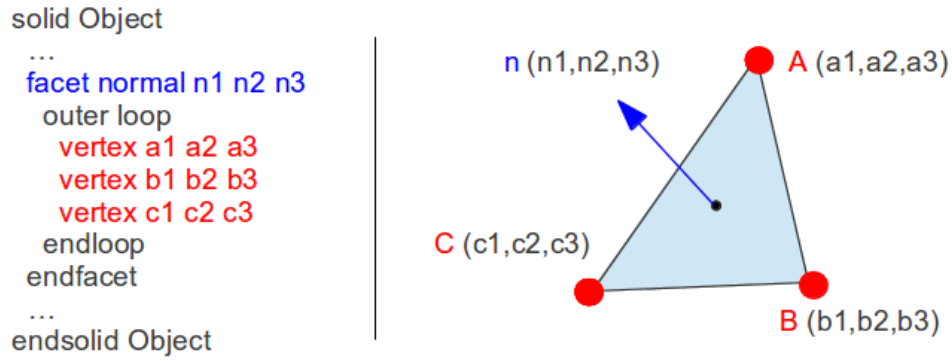


Figure 2.1: STL file structure

of the blade. See Figure 2.2 for graphic support to visualize one of these cuts (red plane).

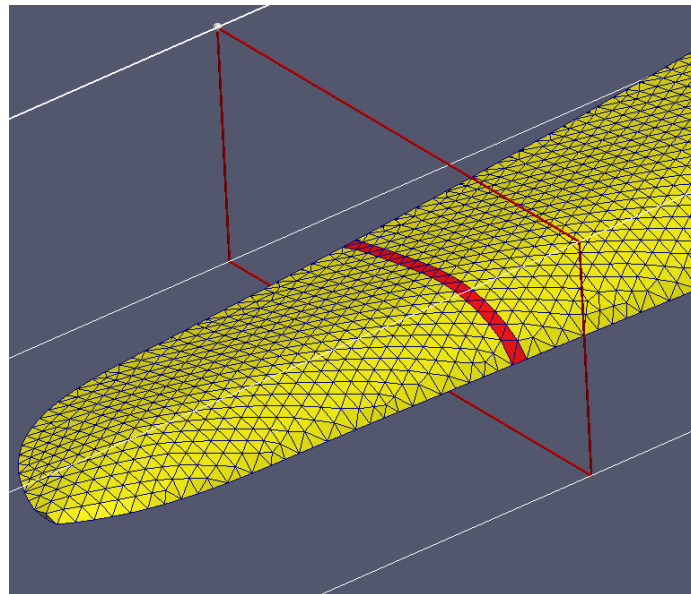


Figure 2.2: STL cutting to obtain airfoil shapes across the blade

The details of the intersection algorithm are not provided here, but the key concept

to run the algorithm is the plane intersection based on vector cross product and a series of *if* statements to check whether the crossing occurs or not. If the finite plane defined by the STL triangle and the infinite cutting plane actually cross and cut each other, then, the couple of points³ are added to an array of points that will end up shaping the airfoil in the way depicted in Figure 2.3.

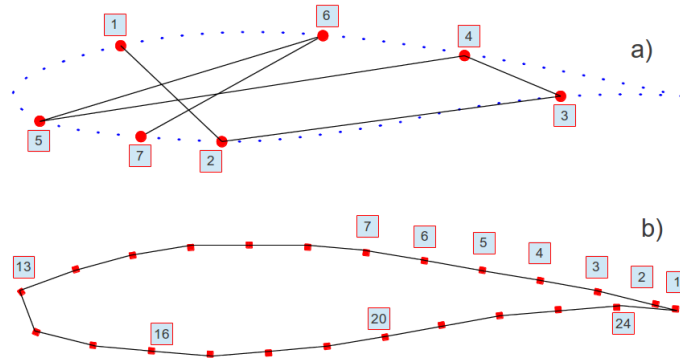


Figure 2.3: Problem description of the airfoil reordering. a) Group of (disordered) points as they are obtained from the cutting process. b) Result of the reordering process.

From the process described above, one might naively think that it is trivial to *connect the dots* just because human eye associates the shape defined by them automatically. However, the *problem* is far more complex than drawing points as children do in the cover of cereal boxes. Points are completely disorganized in the way that Figure 2.3a shows. Consequently, an algorithm is required in order to perform the point reordering to achieve the situation depicted in Figure 2.3b. This process is indispensable for the posterior plotting of pressure as a function of chordwise distance (see Figure 4.23 in page 128).

Before proceeding with the algorithm specification it would be great to state the other three applications in which this reordering procedure was required:

- **Automatic blade generator:** for the sake of building CAD models of blades in a systematic way, this algorithm was part of the engine to make that possible. More details will be provided in the next section of this chapter.
- **3D Scanning of a real blade:** a 2 m long blade was scanned within the context of the Enair's project. For more information refer to Appendix A. The output of

³those intersections of the edges of the triangles and the cutting plane

the scanning process was a 3D STL file representing the actual blade geometry. Then, this algorithm, in combination with the one from the previous item, was used to modelize a blade in a reverse engineering process.

- **CFD analysis performing Prismatic Boundary Layers (PBL):** for the sake of analyzing certain blade aerodynamic behavior, a PBL was required. This procedure implies that the Trailing Edge (TE) of a blade must be flat instead of sharp. For this reason, this algorithm is the bottom line to allow the transformation of blade CAD models with sharp TE into flat ones to run the PBL analysis.

The importance to deal with the airfoil conceptualization and reordering algorithm is justified by the abovementioned three situations and following the main reason: the virtual definition of the airfoil to enable the pressure distribution measuring during CFD simulation employing the Immersed Boundary Method. The algorithm is going to be described in the remaining part of this section. There are three major steps in the process:

1. **Extremes identification.** The first pattern that can be extracted from a disordered vector of points representing an airfoil is the couple Trailing Edge (TE) - (LE) Leading Edge. Clearly, they are the pair of points that are furthest from each other. Based on this fact, point-to-point distances are computed storing the maximum distance and the points that lead to it. Check Figure 2.4. In the upper part of the figure, sub-figure a), several point-to-point distances are computed from point 1. From them, the greatest distance (magenta color) is stored. The same process goes on for all points resulting that the chord length is the maximum distance registered, which is the one corresponding to TE and LE points, as shown in the middle part of the figure, sub-figure b).

At this stage, the pair of points representing TE and LE is identified, but now this question arises: Who is who? Up to now, there is no sign to state that point A is the TE or the LE. There are two possible options. The way to check who is who is by means of computing an indicator that roughly represents the weighted mean angle from each candidate (A and B) with respect all the rest of points defining the airfoil. Observe Figure 2.4c, in it we assume that LE is point A.

Equations 2.1 and 2.2 presents the formula to compute the weighted average angle from point A and B, respectively, to all the rest of points (P being the array, and P_i the point in the i -th position) with respect to the chord line.

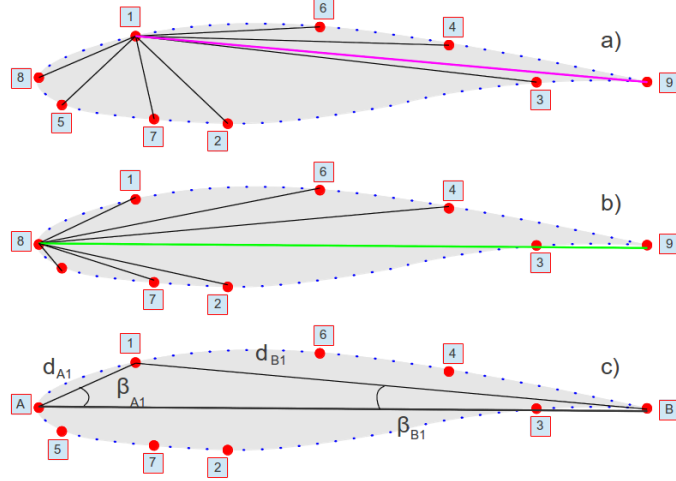


Figure 2.4: Airfoil reordering algorithm. Step 1: extremis identification: a) computing distances to all points; b) couple TE-LE identified (but not who is who); c) computation of weighted angle indicator to determine whether A is LE or TE

$$\alpha_A = \frac{1}{N} \cdot \sum_{i=1}^N \frac{|\mathbf{AB}| - |\mathbf{Ap}_i|}{|\mathbf{AB}|} \cdot \beta_{A,p_i} \quad (2.1)$$

$$\alpha_B = \frac{1}{N} \cdot \sum_{i=1}^N \frac{|\mathbf{AB}| - |\mathbf{Bp}_i|}{|\mathbf{AB}|} \cdot \beta_{B,p_i} \quad (2.2)$$

Equations 2.1 and 2.2, show how the angles from points closer to the candidate, A or B, weight more than those further. Like that, the in-built round nature of Leading Edge and the sharp nature of Trailing Edge should back up the validity of the indicator. Having both indicators, α_A and α_B , the LE would be the one having the greatest average. Therefore, the determination of LE and TE is successful at this stage of the algorithm.

2. **Center line construction.** Once the extremis (TE and LE) have been correctly identified, a center line is built. This center line is a curve departing from TE and finishing in the LE. Actually, this line is made by a number of connected segments forming a chain, in a way that all outboard points are located in one side of the mentioned line and all inboard points remain in the opposite one. But, the

question is how to do so without having more information than just the extremes of the airfoil.

Figure 2.5 contains the three sub-steps to be taken in order to draw the line. The first sub-step is to fragment the chord line in several segments, usually between two and five is enough if we break the chord in such a way that smaller segments are located near the TE to capture the high inboard curvature. In the example provided in Figure 2.5, the line was broken into four segments defined by three breaking-points (red color) roughly located in the positions (r_i) 50, 65 and 80 % of the chord line, where LE corresponds to 0% and TE to 100%. Then, the crucial part of this first sub-step is to compute distances from the breaking point, x_i , to all the skin points that define the airfoil. The closest point detected will be named as b_i , the green ones in Figure 2.5.

$$x_i = TE + \mathbf{chord} \cdot r_i \quad (2.3)$$

$$b_i = \min_{\forall p_i \in P} \|x_i - p_i\| \quad (2.4)$$

In the second substep, b_i is used to find a point located in the opposite board, o_i , that shares the closest x-coordinated in the chord direction to b_i . This is achieved by means of computing two rays in directions $\pm 10^\circ$ (green color) from the perpendicular chord line going through b_i as can be seen in Figure 2.5b. Usually, airfoil shapes contain more than 50 – 100 points, which is enough as to find a point in the opposite board (blue color) with $\pm 10^\circ$. But, in case it is not found, this situation can be solved by repeating this substep with a wider opening degree until a point is found.

$$o_i = \min_{\forall p_i \in P} \angle p_i, b_i, b_i' \quad (2.5)$$

The third and final sub-step consists of the actual creation of the center line by means of computing the average point, c_i in yellow color, being half way between b_i and o_i , just as Equation 2.6 shows.

$$c_i = \frac{b_i + o_i}{2} \quad (2.6)$$

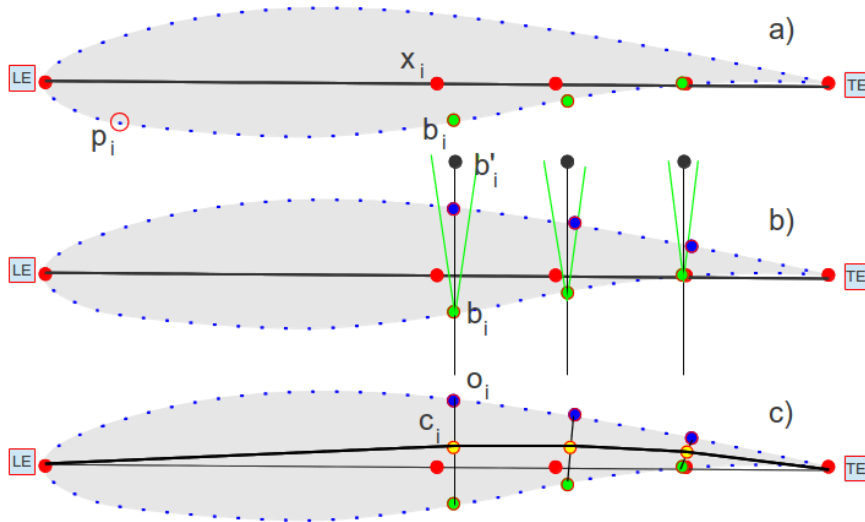


Figure 2.5: Airfoil reordering algorithm. Step 2: a) chord line points discretization and search of their closest skin points; b) search of opposite board points; c) midpoints computation and drawing of the center line

- Reordering by centerline.** From step 2 we reach a stage in which the centerline is created leaving all outboard points in one side of it, and all inboard points in the opposite side (see Figure 2.6a), although we do not have a list mapping which point is in each category. To make the distinction, a cross product is performed for every point employing two vectors: vector 1 from TE to point (in green color), vector 2 from TE to a point in the center line having similar x-coordinate (in blue color, as Figure 2.6b shows). If the resultant vector points out of the paper, the point is tagged in category one. Conversely, if it points towards the paper, the point is classified in the second category. Once we have ranked all the skin points, a calculation is done to discriminate which category is outboard and which is inboard. This calculation is the sum of the distances from skin point to the closest chord line point in each category. As it can be seen in Figure 2.5a, in average, the distances from points outboard are greater than the distances from points inboard. This is the criterion to be used to say which category is outboard and which is inboard.

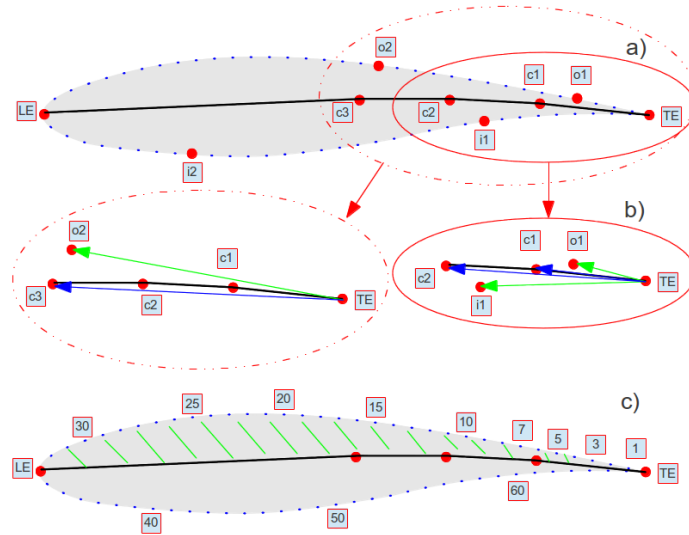


Figure 2.6: Airfoil reordering algorithm. Step 3: a) identification of points; b) detail of components for the vectorial product that makes possible the discrimination: those outboard lead to vectors in different directions than those inboard; c) final reorganization of airfoils skin points

At this stage, the reordering becomes simple to execute. We take as a reference the TE and compute distances from it to all the points in the outboard category; the one having the minimum distance will be the next one to be added to the final list of points where points stand in order. After the addition of the closest point into the list of ordered points, this last point is used as a new center for distance calculation, running a recursive method until all points are selected and reordered. When finishing with the outboard points, the LE is added and the process goes on with the inboard section. The outcome of this stage is the list of points representing the airfoil in an ordered way just as Figure 2.6c shows.

The previous three steps are mandatory in the reordering process and they are the minimum set of orders that fulfills the objective. However, an additional step was incorporated in order to check that the procedure finishes appropriately. The items to be checked are: any segments formed by two consecutive points must not cross the center line except for the TE and LE cases; the change of direction between two consecutive segments must not be greater than 90° except for flat TE airfoils.

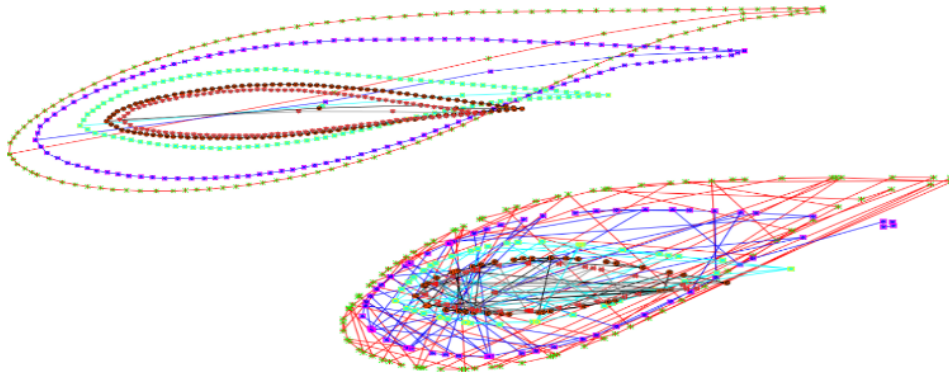


Figure 2.7: Airfoil reordering algorithm. Proof of concept: application to the real case of MEXICO experiment blade. Bottom: raw data. Top: effect of the algorithm

Furthermore, additional options were created to manage the number of points defining the airfoil: one for filling the gaps between consecutive points that are further away than the required tolerance introduced, and another to simplify the curve that represents the actual airfoil when too many points are employed.

After all, perhaps, it might appear to the reader that the presented algorithm is far more complicated than what would be necessary for a good performance. Perhaps, the reader might claim that using a reordering criteria based on the closest distance would be enough. In fact, that was the first algorithm tested by the author and, unfortunately, the outcome was not robust enough: many airfoils were not appropriately reordered. The problem was observed in the zones near to the trailing edge. In order to understand the failure mechanism, consider a point located in the outboard zone close to the trailing edge. In such case, some points of the inboard are closer to the selected point than the true next-point in the outboard region. This fact ruins the effectiveness of the algorithm purely based on the minimum distance criterion, because it is simply not enough to handle the complexity. The next idea to be tested was the reorganization purely based on the angle, but once again, other undesired points were erected as the ones to be selected when they should not. Afterwards, a combined criteria based on angle and distances was used. Again, the results were not satisfactory enough. In the end, the simplest and most robust algorithm to the best of the author's knowledge is the one presented above which has been tested across a large battery of cases.

A real example with the MEXICO blade is provided in Figure 2.7, where it is possible

to see whether the algorithm is being correctly applied or not over the 5 pressure stations of the real blade.

2.2 Automatic blade generation

The problem is the next: there is a need to obtain a CAD model of a blade reducing the human time devoted to the process.

The first time that the author faced the process of generation of a CAD model of a blade for CFD purposes was during a KIC InnoEnergy project named OffWindTech, in which the CTTC was a partner of the consortium during the period 2011-2013. It took us several days until we were able to generate a valid model employing SolidWorks 2012. There was a CAD model in an IGS format available for the consortium, but the problem was that the file was somehow corrupted or not well-built for the software we had employed to process it. We use the commercial software ANSYS ICEM to design the meshes that afterwards will be run with TermoFluids⁴, which covers the fluid dynamics module. Therefore, we had to create our own blade from scratch using Solid Works. This process involves finding the airfoil coordinates, opening them on Solid Works and connecting the dots of each airfoil to draw the curve defining the shape. Afterwards, the corresponding airfoil needs to be copied into the right plane. It requires repositioning, scaling and rotation until the whole skeleton of the blade is shaped. Then, the surface needs to be built from NURBS⁵ going through the ribs (airfoils), but the zone around the trailing edge was problematic (in all sections) because of tolerance issues together with the fact of having a high degree of torsion in the root airfoil-to-circle sections. This situation was overcome by means of breaking the airfoils in two parts: outboard and inboard. Like that, Solid Works can create the surface separately for those two regions and the final blade is finished. An example of the final output is shown in Figure 2.8. Then, when the blade model was imported into ICEM, those problems with the reading of the geometry disappeared, although other ones with superficial and volumetric meshing persisted.

In the end, we solved the problems and managed to come up with a computational model with an acceptable mesh. Unfortunately, the observed difficulties were not an isolated case: when dealing with other blades such as NREL Phase VI or the MEXICO experiment, we also had to deal with similar issues. Being that the case, we looked for help in external softwares to tackle this problem. We found NuMAD from NREL [1],

⁴the CTTC's in-house software. More details in Section 4.1.1 in page 97

⁵Non-Uniform Rational B-Spline, a mathematical concept employed in CAD design

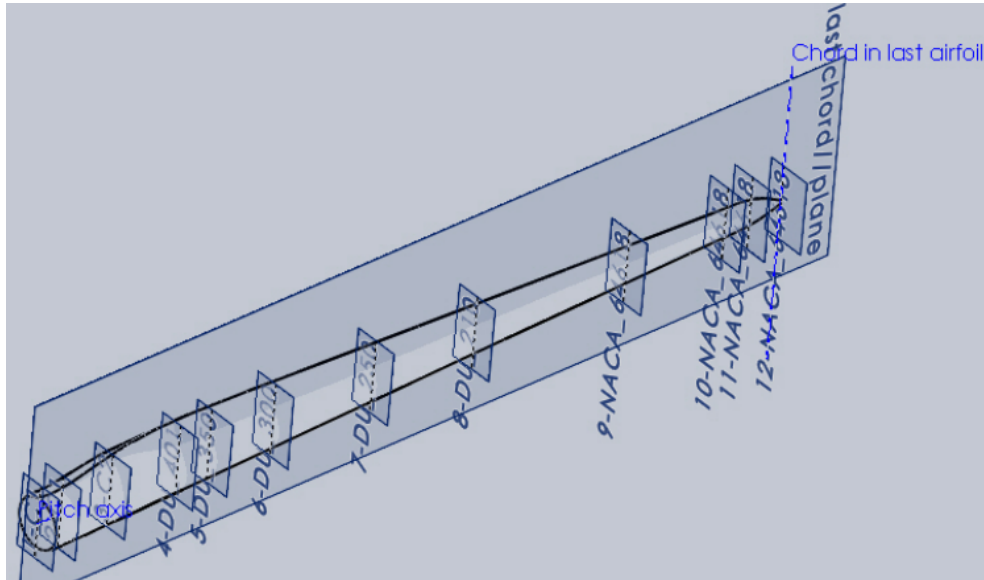


Figure 2.8: Final CAD model of the blade obtained with the "by-hand" approach employing Solid Works 2012

and it certainly was of help but we found two additional problems when using it: 1) the option for flat trailing edge airfoils was not available and 2) the geometry exporting file was not well-linked to ICEM ANSYS.

After all, NuMAD pointed us the direction to build our own code for automatic blade generation (*AutoBlade*), which would enable us to: build a blade such as the one in Figure 2.13 (in page 38) in a matter of a minute, without human intervention (slow and error prone) and already embedded in ICEM for meshing in a straightforward way. The aim of the next subsection is to detail the process and show how we created the tool to make it possible.

2.2.1 The AutoBlade process

The process of building a blade from scratch requires a couple of inputs: i) the 2D coordinates of the airfoils to be used in the blade design and ii) a table with all the parameters to bring the blade to reality. The system provides two files as an intermediate step of the process, one containing the 3D points of all airfoil sections in their final

location and an ICEM script to run the geometric computations to generate the blade departing from the cloud of points. And finally, the output is the blade geometry already generated inside ICEM, which makes the meshing easier. The whole process can be graphically seen in Figure 2.9.

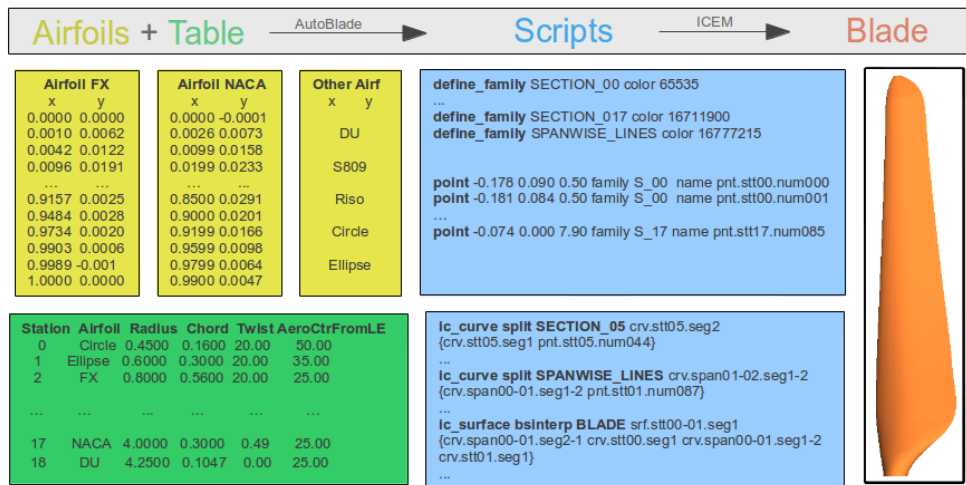


Figure 2.9: Technical details and content of the input and intermediate files used in the AutoBlade process

More specifically, the first input is an airfoil library that is just a folder with a collection of files containing the 2D coordinates of all the airfoils. The file name is just the airfoil name that the algorithm will search within the folder. Those airfoils typically contain between 100 and 250 points, usually with a higher density of points in the leading edge zone in order to capture the high curvature of the region. The second input is a table that contains all the required parameters to define the blade. Those parameters are the next: the *station* parameter is the section number; the *airfoil name* is the name of the airfoil and the file where the 2D data is registered; *radius*, which represents the distance between the rotation axis and the plane where the station (airfoil) is located; the *chord* is the distance from the trailing to the leading edge of the airfoil in the current station; the *twist*, which represents the angle that the airfoil at the section is rotated with respect to the tip station; and finally, the *% center airfoil from LE* indicates the edgewise-position of the airfoil, this position is going to determine the blade axis and the point from which the airfoil will be rotated and scaled.

With regard the outputs, there are two files that come out of the AutoBlade code. The first file contains geometry in ICEM format: it defines the points of each section already in their final location. This file also defines the families, or, in other words, the layers that ICEM uses to color and segment different groups of data. The second file is actually a script with the commands to process all the points and generate from them the final blade geometry. After all, the user only needs to open the first file in ICEM by means of going to the menu *File*→*Geometry*→*Open Geometry*. Once it is open, the geometry processing is done by opening the second file in this way: *File*→*Reply Scripts*→*Run from script file*. By doing this, in a matter of a minute, the final blade is finished.

Now, it is time to go to the heart of the issue and get closer into the way that those files are generated.

Script 1. Geometry defined by blade points. The first step is reading the table containing the blade information (see Figure 2.9, located in the bottom left corner). The information relative to the first airfoil is read and stored in *double* variables from C++, except for the airfoil name and the station number that are registered in *string* and *int*, respectively. Refer to Figure 2.10 for the graphic interpretation of the next steps. The second step, requires to search for the asked airfoil among the library files just to read the x-y coordinates (Equation 2.7) and build an airfoil in the way explained in last section. In doing so, points from the airfoil are listed in the right order and named systematically just to enable posterior ordered reference.

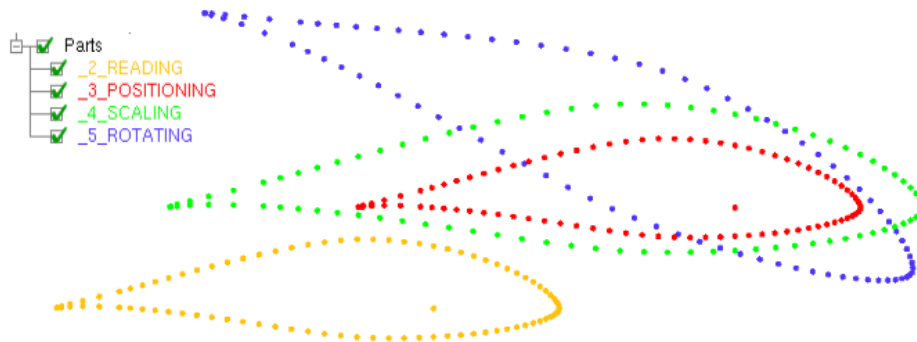


Figure 2.10: Airfoil modification steps within the AutoBlade process

The third step implies a pure translation of the airfoil as it is, just to make the aerodynamic center coincide with the blade axis as Equation 2.8 indicates. The fourth step proceeds with the scaling to convert the initial unitary chord length of the airfoil into the *chord* length size specified in the input table following Equation 2.9. The fifth step performs the rotation of the airfoil points according to the *twist* parameter to define the mechanical torsion angle of the station in the way that Equation 2.10 depicts. And the last step corresponds to the positioning in the final station according to the *radius* parameter just as Equation 2.11.

$$(x, y, z)_i = (x_{read}, y_{read}, 0) \quad (2.7)$$

$$(x, y, z)_i = (x, y, z)_i + (aeroCenter/100, 0, 0) \quad (2.8)$$

$$(x, y, z)_i = Chord \cdot (x, y, z)_i \quad (2.9)$$

$$(x, y, z)_i = (x \cdot \cos \alpha - y \cdot \sin \alpha, y \cdot \cos \alpha + x \cdot \sin \alpha, 0), \text{ where } \alpha = \text{Twist} \quad (2.10)$$

$$(x, y, z)_i = (x, y, z + Radius) \quad (2.11)$$

All these steps are done for just the first airfoil in the input table. If they are repeated over all the sections that define the blade, in the end, the process leads to a cloud of points such as the one in Figure 2.11.

Script 2. Geometry defined by surface generation. The first step is defining the curves that actually represent the shape of the airfoil section. To do so, the command *ic-curve point familyFormat curveName {listOfPoints}* is used so that: a) the *familyFormat* represents the layer⁶ in which the curve is going to be created; b) the *curveName* is the name that will be given to the curve; and c) the *listOfPoints* is the list of names that each of the points in the cloud received in the previous file. This is done for all the sections that the blade requires. Refer to 2.12 for graphic support to this explanation.

⁶a layer in terms of data management within the CAD software

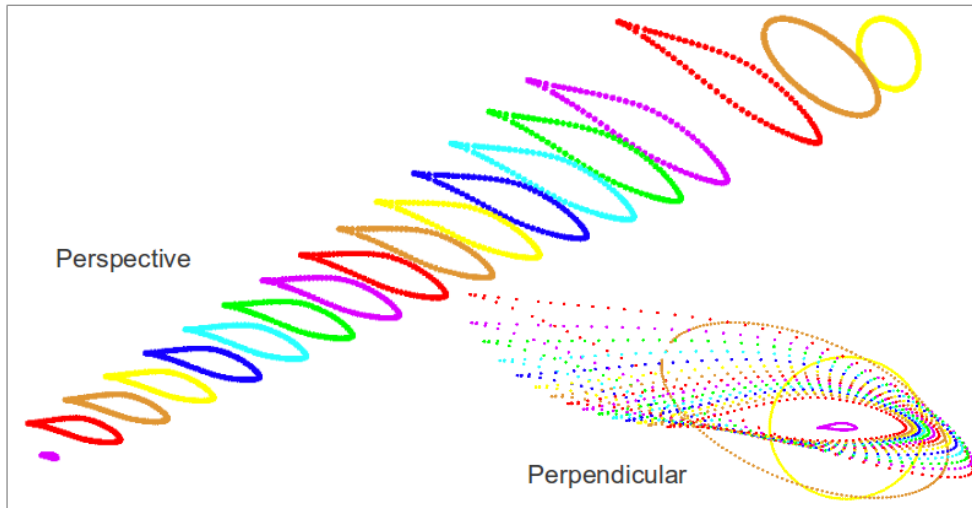


Figure 2.11: Outcome of AutoBlade process: final cloud of points prior to ICEM curves and surface generation

The next step is breaking the curves into segments by means of using the command `ic-curve split familyFormat nameCurveNew {nameCurveOld breakingPoint}` where a new name needs to be set for the second curve that will be generated when the original one is split exactly at the selected point, defined by the parameter `breakingPoint`. This step (the curve splitting) can be performed several times, but it is mandatory to be executed over the leading edge. The leading edge splitting is required in order to divide the airfoil in (a minimum of) two sections just to allow NURBS (see page 31) to adjust to the high curvature areas without problems. If this step is avoided, the blade is not well generated because the curvatures cannot be handled with just a single *envelope*. Therefore, a minimum of two sections are required (representing outboard and inboard) and the way to do so is by activating the trailing edge splitting automatically.

The third step is similar to the first one, but in this occasion the curves are created in the spanwise direction going through all the *breaking points*.

The fourth step is similar to the second but this time is applied over the spanwise lines employing the same *breaking points*. Now, the future blade surface is full of lines going both edgewise and spanwise somehow drawing a skeleton or a coarse

mesh.

The fifth step is the actual surface creation employing the command `ic-surface bsinterp familyFormat surfaceName {curve1 curve2 curve3 curve4}`" where *surfaceName* represents the name that the surface will receive and *curve1* to *curve4* are the names of the four curves that will be the edges of the finite surface.

And the final step is similar to the fifth one but applying it over the tip and the root surfaces just to close the geometry.

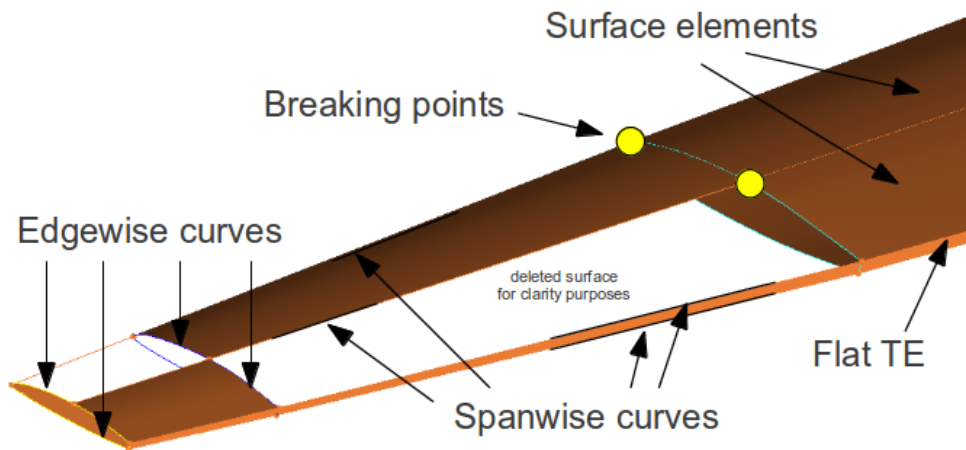


Figure 2.12: Surface generation step and required elements in the AutoBlade process within ICEM scope

As it was mentioned in the beginning of this section, with respect to Solid Works, the same behavior was observed in ICEM: when defining the blade in just one surface. Thus, the breaking point strategy defined just above, becomes indispensable to enable ICEM to properly create the blade surface without strange crossing areas in the trailing edge zones. Moreover, the peculiarity of handling a flat trailing edge is just a particular case with the current strategy. The only required aspect to deal with is to define two spanwise lines instead of one, as it was done for the case of a sharp trailing edge. The use of more than TE-LE breaking points was enabled for a matter of having the code created in a general and robust way. This aspect is adding value from the meshing perspective, but it is not required to increase the quality of the geometry of the surface.

Perhaps, a final comment is required with respect the `ic-surface bsinterp` command. There are several options available in ICEM in order to build up a piece of surface,

but the only one that was robust enough as to be applied for the present case was the *bsinterp*. The other strategies failed at one point or another, and they lead to protuberances and strange surface creations.

Anyway, by means of running the AutoBlade code following the steps detailed in this section, there has not been any case up to date in which the resulting blade becomes different than what was expected according to the input parameters. Thus, the conclusion is that the algorithm is robust enough as to produce blades such as the one presented in Figure 2.13 in a fast (about one minute), automatic (without human intervention) and safe way (no errors detected).

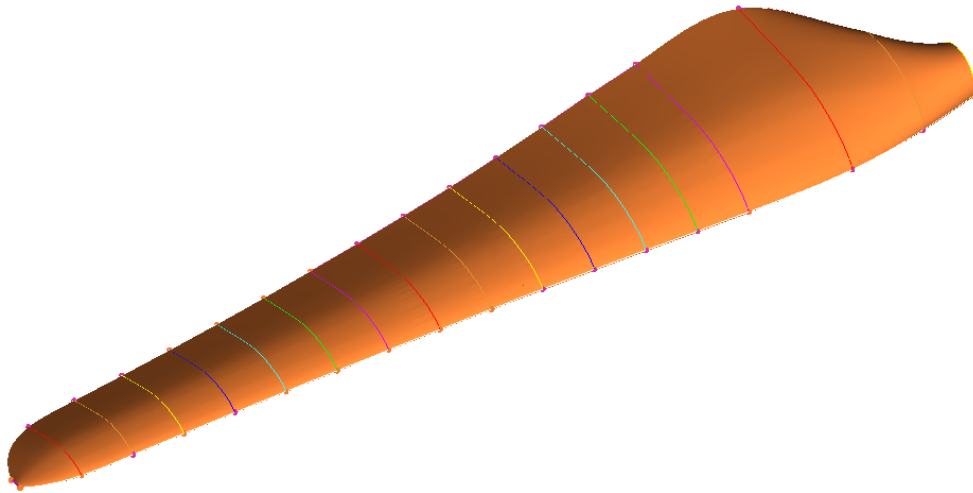


Figure 2.13: Final blade output of the AutoBlade process after ICEM performs all operations

2.3 STL management platform

There is a need of having a virtual platform to describe the geometry of a wind turbine by means of a set of STL files. When performing an aerodynamic simulation using CFD and Immersed Boundary, it is required to have a scalar field representing the distance from any point in the space to the wind turbine closest point. The mechanism to measure the distances to the machine is actually computing the distances to the STL objects.

From the mechanical and electrical point of view, there are thousands of components

inside a wind turbine (see two examples in Figure 2.14), each of them with different sizes and functions, but equally important for the machine functionality: screws, frames, shafts, bearings, wires, magnets, etc. From the aerodynamical point of view, the truly relevant components to be studied and improved are those which define the actual external shape, i.e. the ones in contact with the wind. In this sense, the main components are four: tower, nacelle, hub and blades, where the crucial one is the rotor (hub and blades). Therefore, special care must be put when designing those components.

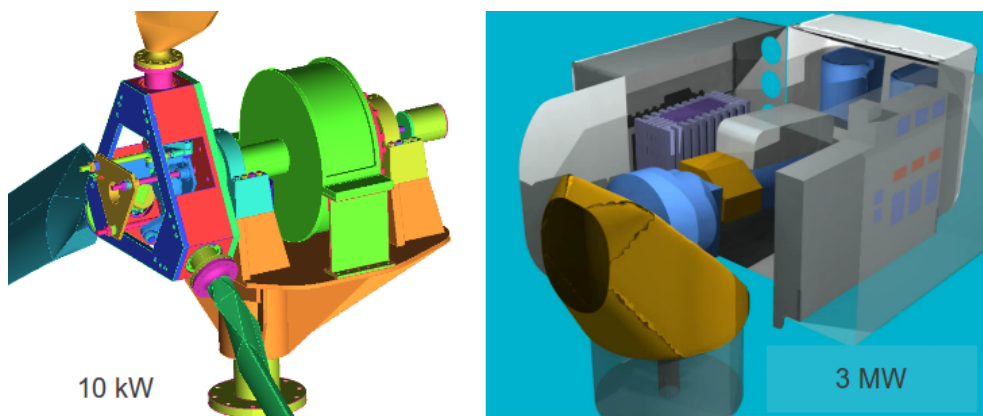


Figure 2.14: Examples of wind turbine components in small (left) and large (right) wind turbines

In such case, it make sense that the objects to be represented virtually on the simulation are exactly those four: tower, nacelle, hub and blade. The detailed geometry of each one of them and their relative positions are required for a sufficient degree of description, i.e. sizes, curvatures, details, etc. However, there are some parameters that are crucial for the physical definition of the turbine:

Tower height. The tower height is the distance from the ground to the position where the nacelle is mounted right on the top of the tower. This distance is especially important for two reasons: 1) it locates the hub in a position that ensures that the blades do not hit the ground; and 2) it has the objective of reaching high positions because the higher it is, the more wind energy can be harnessed.

Tilt. The tilt is the angle between the horizontal plane (ground) and the main shaft axis (hub) from which the mechanical energy is transferred. The tilt angle is typically between 2° and 6° and its aim is to increase the clearance between the tip of the blade and the tower during the blade passage.

Yaw. The yaw is the angle between the wind direction and the rotor axis if the tilt was undone. The desirable yaw angle is zero, however, due to the constant wind direction changes, this angle is usually non-zero. Actually, the average yaw misalignment is around 10° and it has an important negative impact on the power production. Upwind-oriented wind turbines, characterized by the fact that the wind encounters the rotor before than the tower instead of in the reverse way, suffer the mentioned misalignments because they require wind measuring and active alignment. Conversely, in downwind-oriented machines the yaw misalignment is usually zero since self-orientation towards the wind is an in-built behaviour that designers ensure. The downside of downwind turbines is the explicit blockage that the blade suffers when passing next to the tower, given that the blade heavily experiences the vortex shedding of the flow across the tower section. This is the problem that upwind-oriented machines are trying to avoid by means of active orientation. In those cases, the tower effect is also appreciable, but the effect is much lower.

Rotor overhang. The rotor overhang is the distance between the tower axis and the hub center. The greater it becomes the less aerodynamic disturbance the tower will have over the rotor. On the other hand, the further the rotor, the more material it needs to be employed and higher bending moments need to be resisted across different components: shaft, tower, etc.

Azimuth. The azimuth angle is the clock-wise measured angle between a vertical axis going through the hub center and the direction in which a reference blade axis is pointing to. It is the parameter that describes the position of the rotor when it is spinning. Taking the reference angle to be zero when blade one is aligned with tower, the blade passages will occur every $\frac{360^\circ}{N}$, where N is the number of blades (typically 3, sometimes 2).

Cone. The cone angle is the angle between the rotor plane (perpendicular to hub axis) and the blade axis. This angle is typically between 2° and 6° . The greater the value the bigger the blade tip clearance with the tower, however, as it grows, the cross-sectional area that the rotor exposes to the wind shrinks. Thus, it usually takes similar values as the tilt, in part, to compensate the perpendicularity of the blades when reaching the highest position of the cycle (synonym of greater winds and therefore greater potential harnessing powers).

Pitch. The pitch angle measures the rotation of the blade with respect its own londu-

dinal axis. More precisely, if the cone angle was zero, the pitch would be the angle between the rotor plane and the tip chord vector (from leading edge to trailing edge): zero pitch would imply that the tip chord vector lies on the rotor plane. As it will be explained in Chapter 3, the pitch is a crucial parameter for the efficiency of the machine and its control.

The parameters described above make reference to real operation parameters or geometry aspects of wind turbines. However, when it comes to defining a virtual platform for geometry handling for CFD purposes, some extra parameters need to be used to configure a reliable and flexible approach. Those additional variables are mainly about 3D positioning of objects. Concretely, they are points and vectors to enable the system to locate the STL objects with the right relative position.

In order to position a body in the three-dimensional space three variables are required: first, a point that will enable the reference for translations, and then, a couple of vectors that will unmistakably fix the orientation of the body in a particular position. That is the minimum number of variables needed to completely describe the position of a body. Thus, for all the STL bodies configuring the wind turbine, a point and two vectors are provided as it is shown in Figure 2.15. Moreover, as long as the tower, nacelle, hub and blades are geometrically connected, it becomes useful to define the concept of "connecting point" which would be the actual reference point to mount one object over the previous one. Clearly, the first object to be located must be the tower because it is connected to the soil; then, the nacelle is installed on the top of the tower; after this, the hub is added to the system; and finally, the blades are linked to the hub to configure the rotor and the whole wind turbine. This logic is mimicked in the virtual platform and the implications can be seen, graphically, in Figures 2.15 and 2.16. In the first one, the objects are drawn, and the corresponding variables are shown in the scheme. In the second one, the C++ Object creation is shown schematically by means of the main attributes that those objects have.

2.3.1 Operations over STL objects

In section 2.1, it was shown that an STL is an array of triangles in the 3D space. In the present section, so far, it has been shown the structure that represents the geometry of the turbine in the computational space. Now, it is turn to provide an explanation of how those objects are handled geometrically to arrange the platform for the posterior CFD simulation. Operations such loading and saving STL files are the basic ones. There is also the need to build checking functions, since CAD programs usually export STL

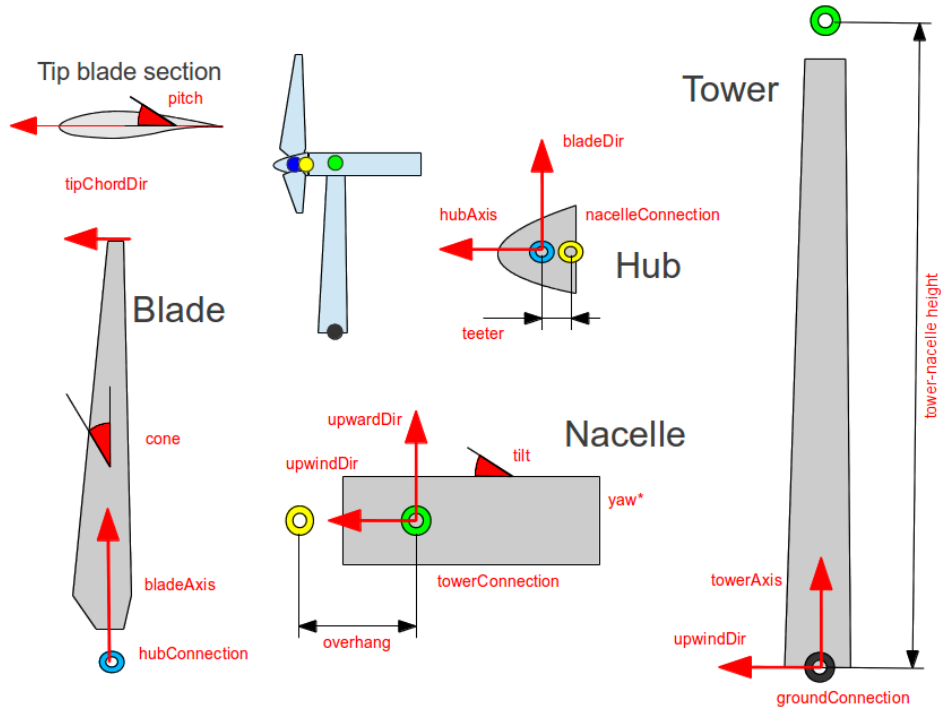


Figure 2.15: Graphic scheme of the relevant virtual components for the aerodynamic study of wind turbines together with their main parameters- (*) yaw angle cannot be shown in this (vertical cut) perspective.

files with holes or incorrect face normals. And finally, some transformation operations are required too, namely scaling, translating, rotating and the complete relocation and reorientation of an object.

Loading and saving. The loading of an STL file requires having a C++ class to build objects mimicking the same attributions. In fact, two classes are required: first, an object *FaceSurfaceBody* is created containing three 3D points for the vertices of the triangle and a 3D vector for the face normal. Then, another object is created, *StlSurfaceBody*, which contains an array of faces. With this data structure, the loading of the file is as simple as reading the data from the STL file and register the data following the patterns of the structure.

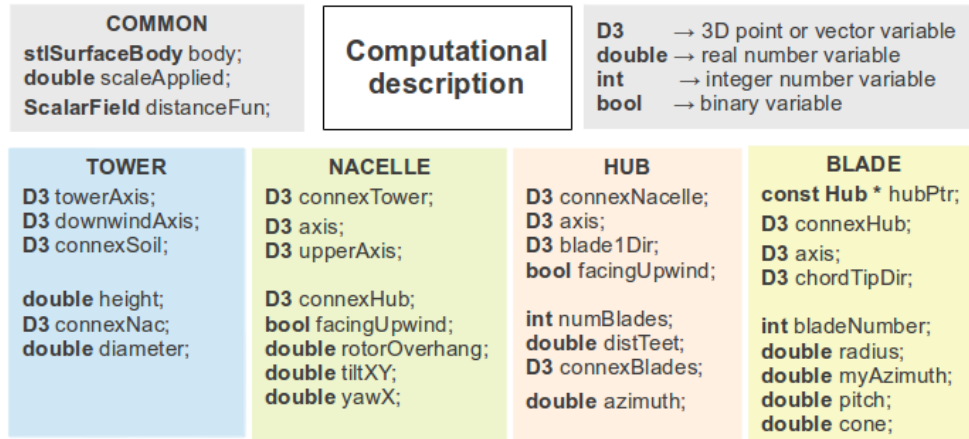


Figure 2.16: C++ object structure scheme of the relevant virtual components for the aerodynamic study

For the case of saving, two loops (a *for* nested within a *while*) need to be built: the big one (while) over the entire set of faces, and the nested one (for) over the points of the triangle. This, combined with the appropriate labels that the STL file structure is expecting, makes the saving function an easy implementation.

Translation. The objective of the translation function is to move the STL from point A to a point B, keeping the orientation constant. Therefore, the implementation is quite simple as the only required operation is adding the displacement to the corresponding coordinate of all the three-dimensional points (Equation 2.12), while the normals of the faces remain unchanged.

$$(x, y, z)_{final} = (x, y, z)_{initial} + (d_x, d_y, d_z) \tag{2.12}$$

Rotation. The objective of the rotation function is to fix an axis in the 3D space and perform the rotation of the object a rate determined by the given angle with respect to the specified axis. As it can be appreciated in Equation 2.13 this operation is far more demanding than the translation:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} u(ux + vy + wz)(1 - \cos\theta) + x \cos\theta + (-wy + vz) \sin\theta \\ v(ux + vy + wz)(1 - \cos\theta) + y \cos\theta + (wx - uz) \sin\theta \\ w(ux + vy + wz)(1 - \cos\theta) + z \cos\theta + (-vx + uy) \sin\theta \end{pmatrix} \quad (2.13)$$

In fact, the previous matrix only represents the rotation of an angle θ of a point with coordinates (x, y, z) over an axis determined by a unitary vector with components (u, v, w) that passes through the origin. Therefore, in order to make the rotation operation totally general, there is a need to perform a pre-translation over the point (as first step of Equation 2.14 shows – left) just to the rotation around an axis that goes through the point (p, q, r) . And after applying Equation 2.13, a post-translation needs (second step of Equation 2.14 – right) to be done to undo the first translation.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_i - p \\ y_i - q \\ z_i - r \end{pmatrix} \quad ; \quad \begin{pmatrix} x_f \\ y_f \\ z_f \end{pmatrix} = \begin{pmatrix} x' + p \\ y' + q \\ z' + r \end{pmatrix} \quad (2.14)$$

Scaling. The objective of the scaling function is to adjust the size of the object. This operation is usually performed after the loading of the STL file, but it can be applied whenever the user decides to. The scaling function requires two inputs: the scaling factor, S , and the scaling point (p, q, r) . The first one determines how the object is going to change the size. If the number is greater than one it will grow, and if it is lower it will shrink. The second parameter is the point to be used as a reference and from which the distances to all points will be increased or reduced according to the scaling factor. Equation 2.15 shows the detail of the way in which the triangle points are modified.

$$(x, y, z)_{final} = (x, y, z)_{ini} + S \cdot ((x, y, z)_{ini} - (p, q, r)) \quad (2.15)$$

Full reorientation. The objective of this function is to reposition an STL object in a different angle and location with respect to a given frame of reference. This tool is indispensable for the platform. It is especially necessary when reading the object for the first time due to the need to place the element in the desired location and orientation. The general case of the reorientation process requires to split it in three operations: a translation and two rotations. Figure 2.17 offers a visual explanation: with the first translation, the reference center of the object

matches the desired position. The second operation is the first rotation, which positions the object with one of their two reference axes aligned with the final desired position; and finally, with the last operation the object is totally reoriented from the initial position to the specified one. There must exist a more compact way to perform the same *full reorientation*, if not in just one step, with two of them could be enough. However, this specific movement has never been a bottleneck of the CFD simulation. Therefore, efforts have been done in other directions such as the distance calculation that will be explained later on in Section 2.4.

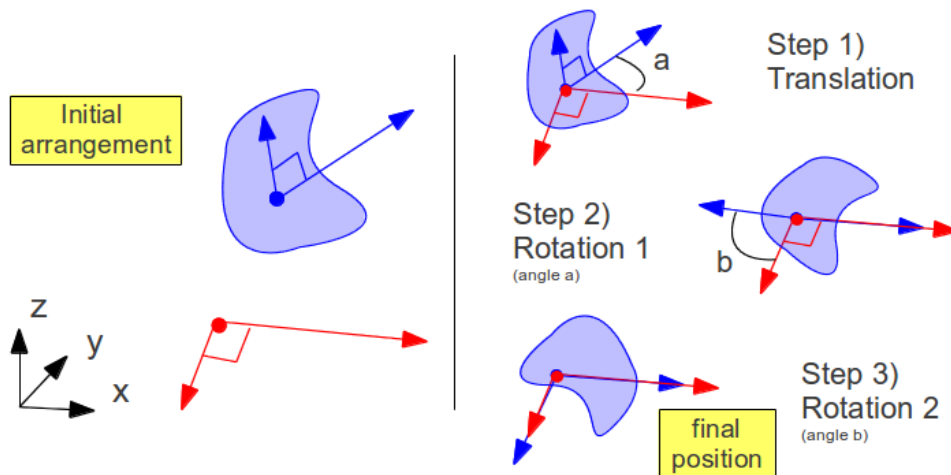


Figure 2.17: STL full reorientation process in a schematic way

Check STL. The objective of this function is to examine the loaded STL file in order to check whether it is totally correct or not before applying any other operation on it. For some reason, errors on STL files are common: they might lack some triangles to close the object as they should; they might suffer a problem with tolerances that also lead to non-continuous or opened objects; or they might have some face normal vectors that are not consistent with the criteria of pointing outwards the object.

a) Holes. The algorithm to check the absence of holes is based on the idea that any edge from any triangle, which is composed by two 3D points, must have exactly another triangle sharing the same edge. No less, no more than this: exactly one

match for any given edge. If there is no match it means that there is a hole. If there is more than one, it means that some triangles are repeated or that an incorrect connection is happening in the present edge. This approach not only allows the user to know if there are holes in the object, but also show, if there are repeated triangles. It can also report detailed information of all the registered problems in order to enable the user to visually check those regions to fix any anomaly.

b) Normals . There are four aspects to check with the normals: first, if it is actually perpendicular to the surface; second, if the normal vector modulus is equal to one; third, the consistency between faces; and finally, the vectors point outside the STL object. The first two refer to each individual face and the last two refer to the group of faces. All this substeps can be consulted in the scheme of Figure 2.18 for the sake of clarity.

The first aspect, *perpendicularity*, is checked by means of comparing the face normal with the computed vector resulting from the vectorial product between two vectors formed by the three points of the triangle: if the normal and the computed vector are parallel, perpendicularity is proved. If the vertices of the triangle are A, B and C, a possible configuration for the two face vectors might be \mathbf{AB} and \mathbf{AC} .

The second aspect, *unitary vector length*, is easily checked by means of computing the norm of the normal vector. Note that those two attributes not only can be checked in that way but also corrected if they are wrong.

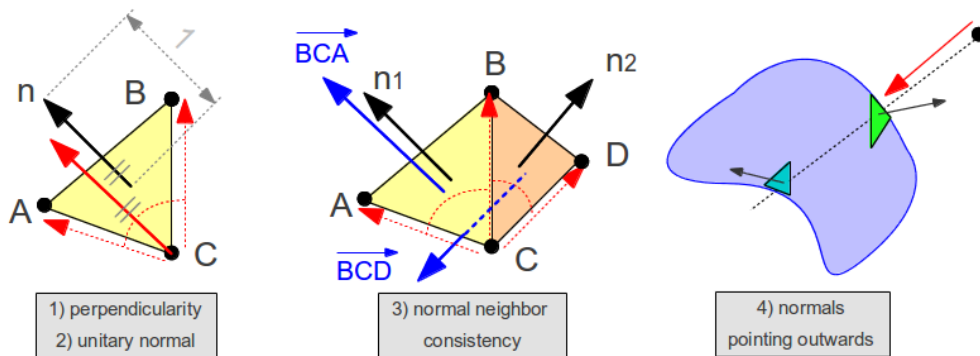


Figure 2.18: The four steps of STL normal checking

The third aspect refers to the *consistency of the normals*. They all must be pointing

in directions that make sense, i.e. outside the STL body. The way to ensure this is by checking that every pair of neighboring triangles are consistent with each other. A first triangle, selected randomly, is assumed to have the normal pointing in the right direction⁷. From there, following a domino effect, all the neighbors are checked and mended if necessary. A list of the checked and unchecked faces is built and dynamically adjusted as the algorithm goes on. In this way, the checking process avoids useless repetition. The way to check the appropriateness of a certain couple of triangles is schematically shown in the middle zone of Figure 2.18. Two normals already exist, \mathbf{n}_1 and \mathbf{n}_2 ; three vectors are computed: the common edge \mathbf{CB} ; the edge from triangle 1, \mathbf{CA} ; and finally, the edge from triangle 2, \mathbf{CD} . The vectorial product of \mathbf{CB} and \mathbf{CA} leads to the vector \mathbf{BCA} . The vectorial product of \mathbf{CB} and \mathbf{CD} leads to the vector \mathbf{BCD} . Both pairs of vectors are compared, on the one hand \mathbf{BCA} together with \mathbf{n}_1 , and in the other hand \mathbf{BCD} together with \mathbf{n}_2 . Both pairs (among the elements of the couple) must have equal directions (this was checked before), but one pair must agree on the sense and the other not. When this happens, the neighbor normals are consistent and the pair is built correctly. If this is not fulfilled, the triangle being tested gets its normal reverted.

So far, normals are proven to be consistent to one another, but in order to build a totally robust algorithm it is mandatory to execute another test. The objective of the fourth check is to ensure that the normals actually point outwards. Bear in mind that the third aspect checks normal consistency, but it was checked based on the assumption that the first selected triangle was correctly facing outwards. The way to check this aspect is shown in the right hand side of Figure 2.18. First, a random point far away from the object is chosen. Then, a random triangle is selected and a ray going from the external point and crossing the center of the triangle defines a line. This line is tested with all the rest of triangles of the body for the sake of counting and registering all the cutting points of the ray with the STL. The face normal vector of the triangle with the closest cut is selected for the final computation: a scalar product between its normal and the vector between the external point and its centroid. This scalar product must be negative, meaning that vectors point in opposite senses, which in turn implies that the closest triangle normal is actually pointing outside.

Reordering. This operation is used specifically in cases where the STL object is

⁷but it will be checked later on in the process (step 4).

as complex as to present problems in the distance computation explained in the next section. The order of the triangles in an STL is generally not important, but to perform certain operations (to be explained later on in Section 2.4), it is helpful to have an specific arrangement pattern. The objective of this function is to reorder the STL triangles according to the direction defined by a vector that the user delivers. All the centroids of the triangles are projected to a 3D line defined by the vector and a random point far away from the object (reference point). In this way, all triangles get naturally ordered by those projections, being the first point in the list the closest to the reference point. From there, the rest of points, in representation of their triangles, are reordered according to the distance to the reference point.

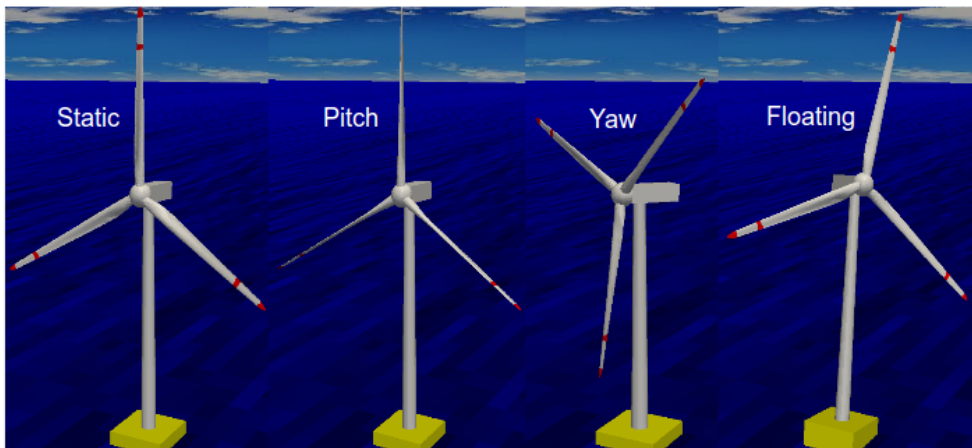


Figure 2.19: Sequence of WT movement composition employing STL and the present geometry handling platform

This section is closed, with the present description of STL platform for the geometric handling of wind turbine components. The correct behavior of all these functions created has been tested and verified extensively in multiple scenarios. Figure 2.19 shows a sequence of four images extracted from a full wind turbine simulation moving along all its degrees of freedom. In the first image (left), the turbine is seen in static conditions (reference). In the second image, the pitch capability is shown. The next figure shows the yaw angle and the final one incorporates the oscillating behaviour of the offshore platform. The azimuth variation can be seen throughout the process. Tilt and cone can

also be varied at wish but as long as they are constant for the whole turbine lifetime, they were not modified in the video from which the sequence is extracted.

2.4 Distance function

The problem definition is the next: there is a need to obtain a field representing the minimum distance from any point in the space to a group of STL objects.

The Immersed Boundary Method, used as a CFD strategy to study a full wind turbine, requires having a way to locate the interface that separates solid from fluid nodes, together with a mechanism to handle the imposition of the boundary conditions in that region of the domain. The solids are defined by the STL platform described in section 2.3, however, they are just the geometry specification. A way to connect the geometry with the fluid dynamics simulation needs to be found. The selected approach implies computing the distance from any point in the domain to the closest STL point (from a particular triangle). There is no room for doubt than when the distance is zero, the solid surface is located right there, on the node. The crucial consideration is that the distance must be signed (allowing positive and negative values) in order to be able to distinguish solid from fluid nodes. The criterion used for this is to assign the negative sign to the solid nodes (inside the STL), and the positive sign to the fluid nodes, those located outside the STL.

The operating principle of wind turbines implies that they must have rotating parts, which in turn means that simulating their behavior requires updating the distance field at any new iteration. In the cases that will be presented later on in this thesis (Chapter 4), a typical mesh would be in the range of tens of millions of control volumes (around 30 million) and a typical STL set of objects contains thousands of triangles (around 60 thousand). The approach all to all (control volumes and triangles) leads to enormous computing times. Therefore, it becomes impracticable to recompute these distances as frequently as once every iteration when the brute force approach is used.

In the beginning, we tried to use advection solvers to move (rotate) the distance field but it leads to imprecise geometry models. Thus, a combination of preprocess computations with the advection approach and a periodic geometry refreshing was done, but even in this case we faced some other problems. In the end, an alternative approach, the *NEighbor Walking Algorithm*, was built to solve this issue, allowing the recomputation of geometry every iteration.

2.4.1 Distance point-triangle algorithm

For now, let's set aside the computing frequency and focus on the unit of computing: any of those approaches ranging from those employing brute force to the most refine ones, rely on the same principle to measure distances. As it was mentioned before, the aim is finding the minimum distance from a given point to the STL surface. It means that loads of distances must be computed (order 10^9), which is a repetition of the same algorithm: the distance from the point to one triangle of the STL file.

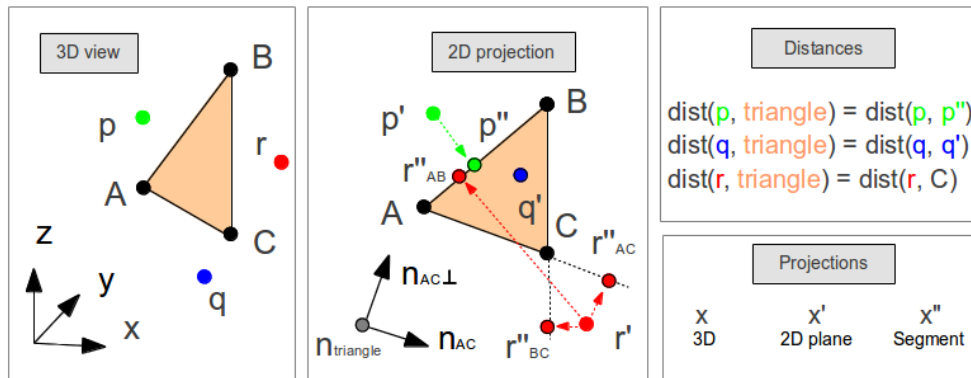


Figure 2.20: Process of computing the shortest distance from a given 3D point to a concrete STL triangle.

The algorithm to find the minimum distance from the point to the given triangle is explained in the following lines, graphic support can be found in Figure 2.20. Notice that the particular case of three different points (p , q , r) are presented along the three next steps:

- 1. Distance (absolute) evaluation.** The point q is projected from its original position to the infinite plane in which the finite 3D triangle is contained, which leads to the projected point q' . If such projection falls inside the triangle boundaries, the distance is measured as the norm of the vector from q to q' .

If the projection of the point over the plane falls outside the triangle, such as the case of points p' and r' , the distance is now computed with another function that measures the distance between a point and each of the three segments. To evaluate the distance between the point and one of the segments another projection step is required: the already projected point (such p' and r') is perpendicularly projected

once again to the infinite line defined by the finite edge of the triangle. If the projected point, p'' , falls between the two vertices, such the case p''_{AB} , the distance is computed by means of the norm of the vector between the original point p and the final projection p'' . Conversely, if the projected point, r'' falls outside the two vertices, such the cases r''_{BC} and r''_{AC} , the distance is computed between the original point and the closest vertex, in the case of Figure 2.20, the distance between r and C .

Thus, the minimum distance between the three projections is the one corresponding to the distance between the point and the triangle: 1) in the case of point r the closest point is C , 2) for point q it is the distance to q' , 3) but in the case of the point p , whose second projection (p'') falls between the vertices A and B , the minimum distance is the one between the original point and the second projection p'' .

2 Distance sign evaluation. The sign of the distance computed in the previous step is determined by the sign of the scalar product between the normal of the triangle and the vector built from the original point towards the centroid of the triangle. If such scalar product is positive, it means that the normal and the vector point in the same direction, i.e. it means that the distance should be negative because the point falls inside the solid. If the product is negative, the point is located outside and the attributed sign is the positive instead.

So far, the distance from a particular point, p , to an specific STL triangle has been detailed, but do not forget that the final objective is to find the minimum distance from point p to the whole STL, not just to one of its triangles. At this stage, it might seem that in order to complete the algorithm, the missing part is just the iteration over the STL triangles to find the minimum distance. It is true, but only partly true. The experience shows that dealing with geometry always throws exceptions and problems such us the ones detailed in the next subsection.

2.4.2 Combined effects of distance mesh-STL algorithm

It regularly occurs that a mesh point finds that the closest point in the STL is actually a vertex of one of its triangles, say vertex A from triangle T581. This vertex is common to more triangles than just the T581, perhaps the T18 and the T9262 share the same point. Undeniably, the absolute value of the distance is the same regardless of the triangle used for the distance computation, but the same is not necessarily true for the sign of

that distance. Then, two questions arise: 1) Which is the right triangle to be used for the sign computation? 2) How is the algorithm dealing with the situation? The first question requires a complex answer, to be provided in the remaining part of this section. In turn, the answer for the second question is "by chance". This means that as long as the STL triangles are not disposed in a specific order, the first triangle containing the vertex A will be the one whose normal will be employed in the sign evaluation.

In fact, those questions were not actively considered at the beginning of the research because the idea behind the method was based on the paradigm shown in Figure 2.21, where everything runs appropriately and all the zones compute the distances and signs without mistakes, both inside and outside the solid. This is like that because, for this case, all normals provide the same sign. Notwithstanding, the bodies represented by the STLs employed in this thesis might have sharp convex areas and that is the origin of the problem pointed out by the previous questions. For instance, the case of the blade in the trailing edge would imply that the body schematically presented in Figure 2.21 turns out to narrow down to something like the one in Figure 2.22.

Pay attention to Figure 2.22, as it can be seen, there are six different zones, five of them outside the object and another inside. From those outside, there are three of them where any mesh point located there computes the minimum distance to a vertex (vertex B), i.e. regions *Out d1*, *Out d2*, *Out III*. The first two regions might have problems with the sign of the distance. The sign for all the points falling in those two areas will be determined by one of the two normals of the triangles shown in the picture, and the result is dependent upon the one to be used: if the normal from the segment AB is used, the blue point drawn in region *Out d2* would be considered inside the body and the sign would be negative (incorrect). Conversely, if the normal from BC is used, the sign would be positive (correct) because the point would be considered to fall outside the body. For this specific case, the right normal to be used is the one from BC , but it is necessary to build a system robust enough as to deal with this kind of situations. As it was introduced before, the lack of specific algorithm commands in front of this case leads to random selection of the triangle, which is obviously error prone.

The proposed solution for sharp regions is to associate an enhanced normal for vertices. This special normal will be the one to be used when the closest point of the STL is a vertex. When this situation occurs, the normal will be determined by the sum of the normals of all the triangles sharing the vertex. For example, in the case of Figure 2.22, a new normal is computed and shown there in magenta color for vertex B . This normal is the normalized vector after adding the normals of AB and BC . Besides the enhanced normal, Figure 2.22 also shows an imaginary plane going through point B

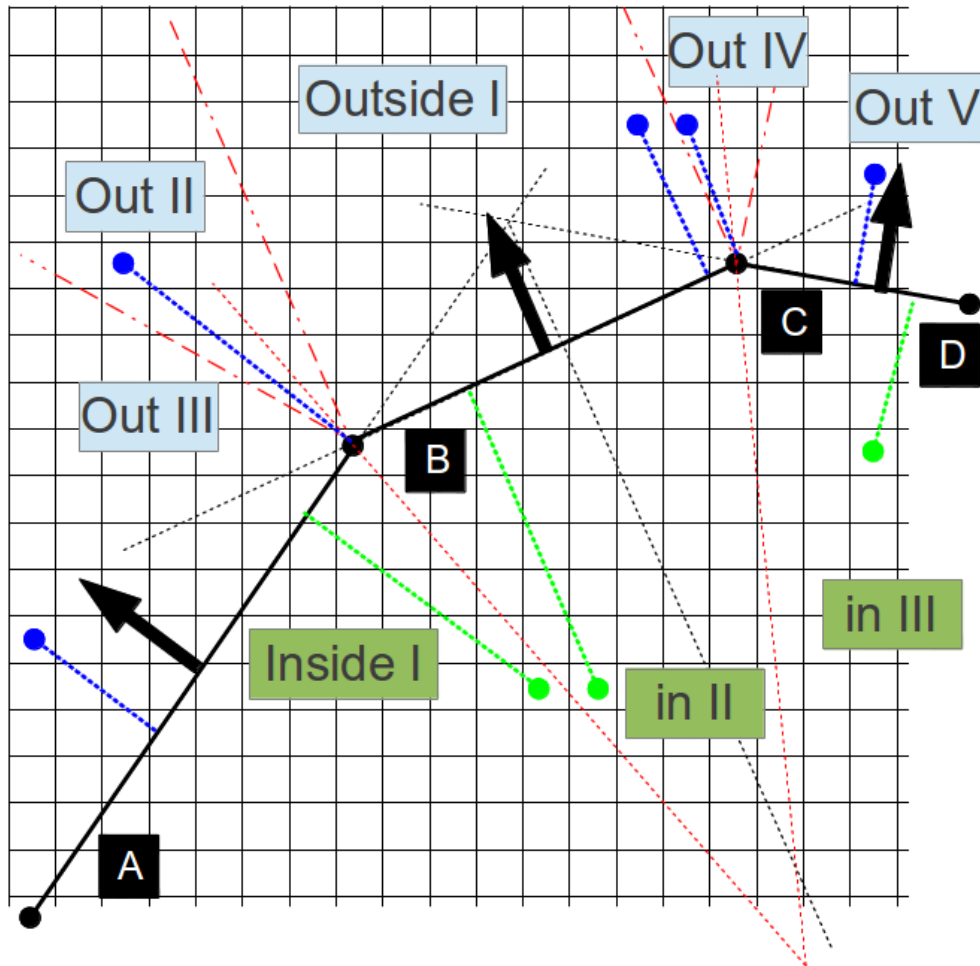


Figure 2.21: Schematic 2D representation of distance computation of several point to STL triangle in a smooth convex object

whose associate vector is the vertex normal, also with magenta color. Consequently, any point located in regions *Out III*, *Out d1* and *Out d2* is now safe from the presented problem because the new normal would consider them all to be outside the body.

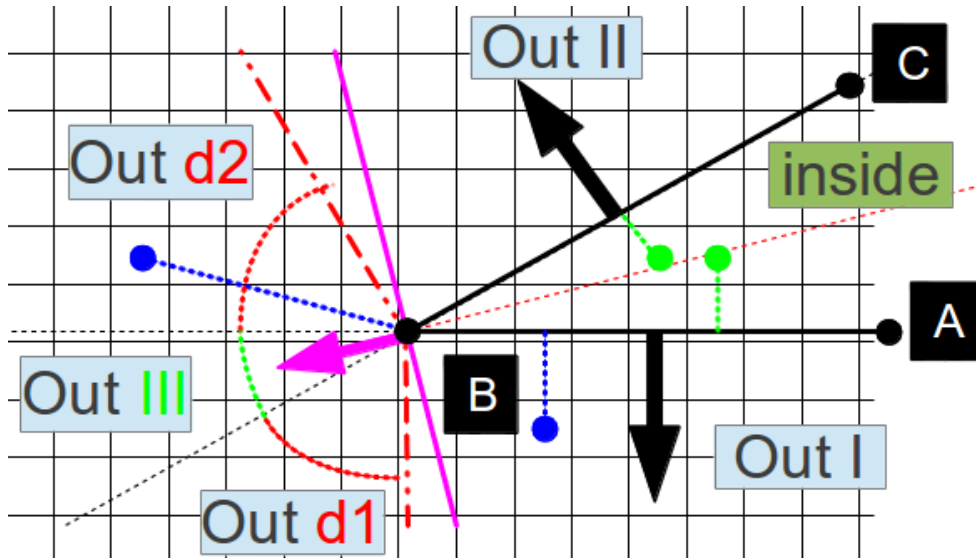


Figure 2.22: Schematic 2D representation of conflictive regions for distance computation point to STL triangle in a sharp convex object

2.4.3 Enhanced distance mesh-STL algorithm

The rotation of the blades and the need to constantly recompute the distance field leaves no room for doubt: the *brute force* approach is robust but at the same time, too expensive and slow (in terms of computational costs). Brute force is clearly not the most efficient way and several options exist to improve it such as limiting the number of operations or information reuse. As long as the rotation of the machine is below 1° per iteration⁸, the distances will not experience abrupt changes. This makes it interesting to study the feasibility of information recycling, and that is part of the content of this subsection.

Due to its simplicity, the first idea to be explored is the limitation of the all-to-all operations in a way that does not affect the quality of the results. There are three approaches that aim at this aspect:

1. **Static objects.** Usually, the cases to be studied keep constant pitch, yaw, tilt and cone angles as well as the rest of distances and parameteres described in the section 2.3, more precisely in Figure 2.15. The only one that is varying is the

⁸which in practice is even smaller than this value

azimuth, which is inherent to the working principle of wind energy. Thus, some objects are totally static and their distance functions do not need to be recomputed.

2. **Limiting boxes.** The meshes employed in the IBM for CFD are extremely dense in the rotor area, quite dense in the wake, tower and nacelle areas but have low densities in the rest of the domain. Thus, it might seem a good idea to limit a specific region of nodes that will need to compute the distance function. Do not forget that this distance is just a requisite for the ultimate goal: determining the interface of the turbine, for the sake of discerning fluid and solid nodes of the mesh. Therefore, and for instance, for the case of the tower, most of the domain is very far away from its interface. Thus, it makes sense to define a cylinder or a parallelepiped that encapsulates the tower alone. In this way, only the points inside the parallelepiped will go through the computing distance step. This means saving loads of operations by excluding the areas beyond the limiting box from the computation of the distance to the tower. The same strategy is followed for the cases of the nacelle and the rotor.
3. **Far distances computed with less precision.** Even when using static objects and limiting boxes. There are still some other situations in which time can be saved. Take the case of the rotor: the portion of the mesh that is concentrated in this region could be easily in the range of 60 – 80 %. Consequently, it is here where the crucial battle takes place. For those points that fall inside the limiting box but far away from the surface, they end up being with the label of "fluid". Why do we have to compute the distance to them with the maximum level of precision? There is no need. The distance from point to triangle is a much more expensive operation than just the distance from the point to the centroid or any of the vertices of the triangle. Thus, the way to compute the distances for those points inside the limiting box is just the norm of the vector from the mesh point to one point in the triangle. If that distance is greater than a reference distance, there is no need to compute it in a detailed way. If that distance is below the reference distance, then the maximum level of precision to compute the distance is utilized. The reference distance should be decided according to the relative size of the objects, there are many valid ways to fix it. One of them could be in relation to the size of the triangles, e. g. five times their characteristic size. Another one could be the 20 % of the tower radius, or the 10% of the nacelle length.

Clearly, the use of this strategy has two sides of the coin. The negative side is the double distance computation in the regions inside the reference area –

they compute the simplified distance first, and the detailed one afterwards. The positive side of the coin is experienced in those areas beyond the reference distance. For them, the detailed distance computation is never called because the simple distance suffices. It turns out that, the relative cost of the detailed vs. the simple distance computation is more than 20 times more expensive. The conclusion of this is that just by avoiding 5% of the detailed computations, this modification already makes sense.

In conclusion, this means that even when as little as 5% of the detailed distance computations are skipped (by only computing them in the simplified way), the algorithm already provides a time saving.

This package of options to optimize the computing time based on the ideas of limiting the use of distance calculation are truly helpful and they reduce the computing time considerably both for the static and dynamic objects. However, for the case of the rotor it is still unacceptable to refresh the distance every iteration by means of the tools exposed above. The crucial idea to cut down costs in the distance computation is the concept of the *neighbor walking* and related tools such as the *opposite criterion*, the *problematic triangles* and the *helping triangles*.

The idea behind the NEighbor Walking Algorithm (NEWA) is based on the information reuse and the self-directed distance computation. Figure 2.23 provides an example of the algorithm execution for the first and second iterations that will help to clarify these concepts. In the first iteration to find the closest distance from a point p to the STL, we are totally blind because there is no clue as to know which may be the closest point. The only possibility is trial-and-error. Then, the distance for a random⁹ triangle is computed. From there, the distance of its neighbors can also be computed and compared to the triangle. If any of the neighbor distances is closer to the point p , a next step is performed from the neighbors of it. This process goes on until the case that no neighbors are found to be closer.

The first aspect to notice here is the information reuse: if we store the closest triangle for all mesh points, the next iteration we already depart from a good guess, which would imply a huge time reduction. Bear in mind that because of CFD requires small time steps, the rotated angle would be far smaller than 1° . Therefore, big jumps in the STL triangle structure will never happen when it comes to find the closest triangle.

As the reader would imagine, this algorithm cannot be successfully applied to any case. If it is applied to a complex figure with holes, cuts, not continuous, with "valleys",

⁹for instance, the first in the STL list

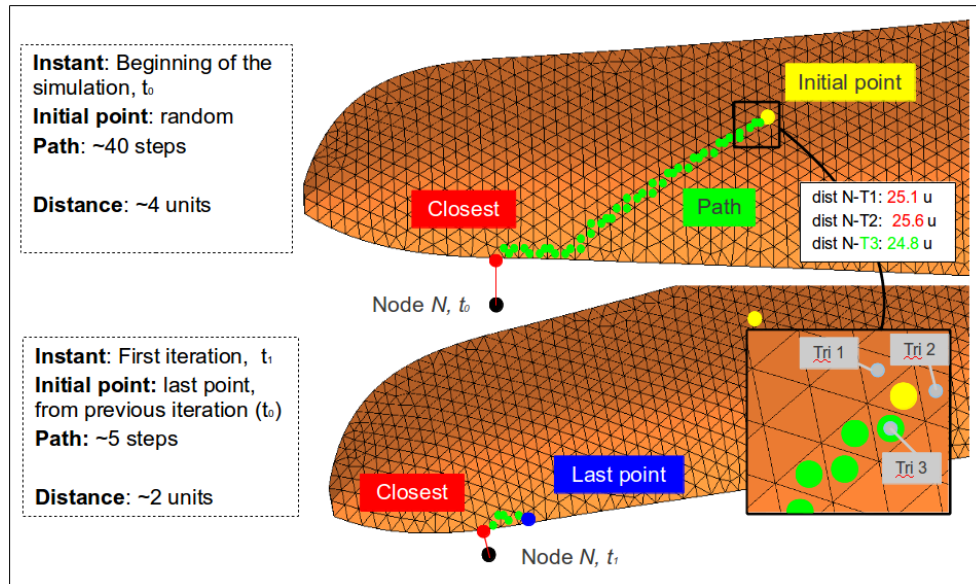


Figure 2.23: NEIGHBOR WALKING ALGORITHM (NEWA) proof of concept: first and second iterations over a real blade

etc, it will not succeed because it is based on heuristics. That is precisely the strength and, at the same time, a weakness. For the STL objects of this thesis (tower, nacelle, hub and blades) is highly useful and it cuts down costs dramatically, which would boost computational time. Notice, though, that this is definitely not a robust tool.

In fact, simple figures such cubes (parallepiped in general) can also present problems with the NEWA. When a point of the mesh finds the closest point of the cube to be one of its faces (not an edge or vertex), it can actually happen that the closest point is in the opposite face than it should! This undesired effect is consequence of the (lack of) curvature of the body and something must be done to equip the NEWA to sort out this problem. An extra degree of heuristics is incorporated to the algorithm: the *opposite criterion*. This is the direct solution to the problem that parallepipeds experience: when finding the provisional final triangle of the NEWA, an additional step is performed testing if a triangle from the opposite face is actually closer. Figure 2.24 shows the effect of the *opposite criterion* when applied to a cube, a sphere and the (modified) MEXICO

blade¹⁰.

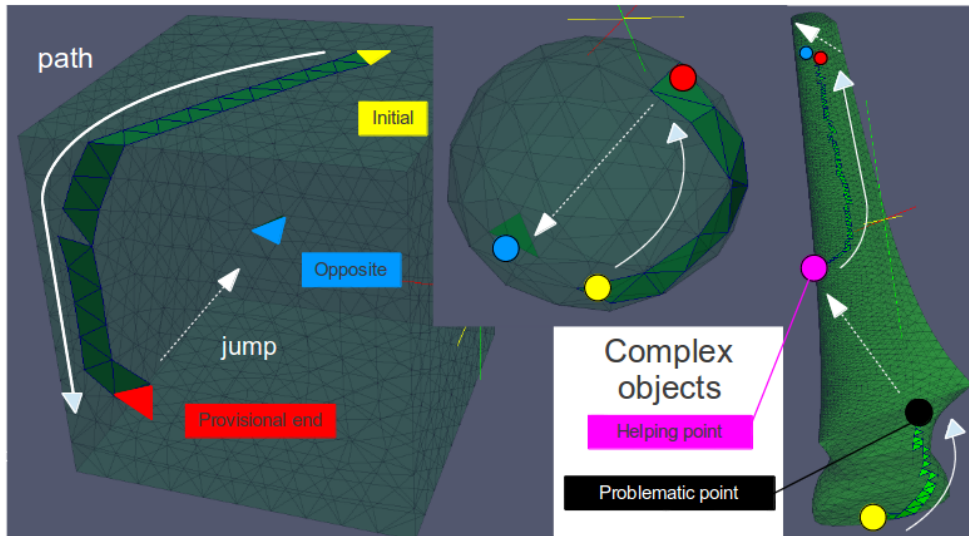


Figure 2.24: NEighbor Walking Algorithm (NEWA) applied to three bodies: (left) cube; (middle) sphere; (right) real blade (with problematic and helping triangle heuristics)

After all, it turns out that the NEWA, together with the opposite criterion work fine enough for certain objects, but it still fails with other ones. For instance, the blade in Figure 2.24 presented some problems for those points whose closest triangle was in the tip region because a "valley"-like zone is found in the highly curved root that prevented the NEWA to successfully *scape* from there. Because of that, one more layer of heuristics is added to those cases: the STL is reordered (in pre-process) and the triangles of the highly curve areas are considered to be *problematic triangles*. For those triangles, a few *helping points* are associated to them in order to restart the NEWA from a second point and check whether the minimum distance changes or remains constant. The effect of this measure can be seen in Figure 2.24. Those *helping points* are usually beyond the 30% of the blade span.

In the case of blades, the *opposite criterion* must be especially active because the problem is encountered frequently. Bear in mind that the blades are constantly rotating with the domain – the rotor plane – and it means that some mesh points "cross" the

¹⁰which was modified to amplify the curvatures at the root to make the test harder

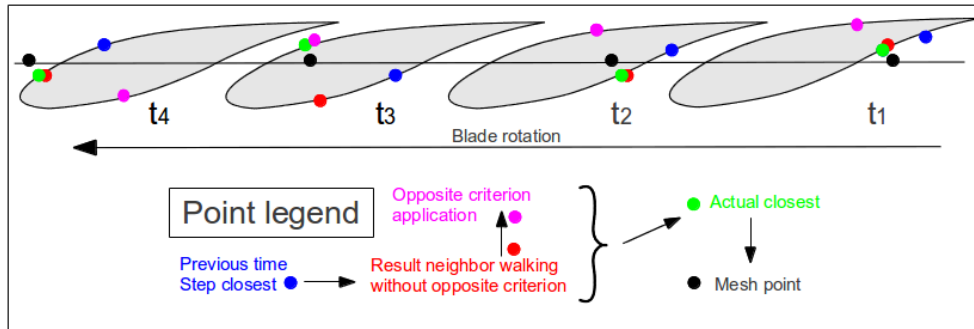


Figure 2.25: The effect of NEWA together with the *opposite criterion* when applied over a rotating blade

blade boundaries three times per spin. Figure 2.25 aims at this concept. The blade rotation is frozen at four steps from the right hand side of the figure to the left hand side. The black point represents the centroid of a particular mesh node in relation to the closest airfoil section, representing the blade in 2D. The blue point is the point from the closest triangle in the previous iteration. The red point is the closest point of the current iteration that the NEighbor Walking Algorithm was able to find and the green point is the actual closest point to the mesh point. The magenta point represents the test of the opposite point once the red one was found. As it is possible to see in the figure, instants t_1 , t_2 and t_4 do not require the opposite side criterion because the NEWA found the right closest point. However, the time step number 3 was not successful because it is exactly from the time steps $2 \rightarrow 3$ when the jump from the closest point takes place from the inboard section to the outboard section. It is precisely for those cases that the *opposite criterion* represents the crucial part of the NEWA in order to attain the true minimum distance point.

2.5 Conclusions

This chapter concludes here. In it, four sections have been covered. The first section has dealt with the problem of having an airfoil C++ object capable of solving the challenges associated to the basic blade design and, at the same time, provide the framework to analyze pressure distributions in a CFD-based simulation. In the second section, it has been exposed the construction of a code to automatically build blades from an airfoil library and a table with the blade parameters. The AutoBlade code has been proved

to be effective, fast, error-free and robust. Then, the third section has dealt with the platform that enables the virtual assembly of HAWT components based on the STL file approach. The framework can handle the natural degrees of freedom with which a turbine is designed and operates. Finally, an effective and fast alternative to the brute force approach has been developed for the mesh-to-turbine distance computation. More details of the distance evaluation will be provided in Chapter 4.

However, before doing that, it is time to develop an engineering tool in order to aim at the first level of approach to assess and design a wind turbine. This objective is covered in Chapter 3.

References

- [1] Jonathan C. Berg and Brian R. Resor. Numerical Manufacturing And Design Tool for Wind Turbine Blades: User's Guide. *NREL.*, 2012.

Engineering models

Engineering and science are always hand in hand: engineering is completely based on science, and science grows with the support of engineering. For the specific case of the aerodynamics of wind energy, science has pushed the boundary far away. The most refined and accurate scientific theories for predicting aerodynamics (in general) are defined by the Navier-Stokes equations (Equations 3.2). If we were able to solve those equations together with the mass conservation equation (3.1) with the maximum level of detail, the level of understanding about aerodynamics would be astonishing. However, this is not possible today because they cannot be solved analytically and we still lack the computational power to obtain a complete numerical resolution in many technological applications. This means that simpler methods need to be employed for the assessment and design of the machines that we are (technologically) able to build in the current time. In other words, the timings that engineers face in the designing process of wind turbines require tools for fast analysis and acceptable levels of accuracy.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (3.1)$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} \quad (3.2)$$

In this respect, Computational Fluid Dynamics is a relatively new approach which represents the cutting-edge tool for analysing aerodynamics with the maximum level of detail. However, the tandem accuracy of the results and time of computation is limiting the use of CFD in designing stages. The analysis and design of current wind turbines is mostly carried out by engineering models because the simulation is extremely fast and

the level of accuracy for most of designing scenarios is acceptable. The most common method to design wind turbines today is based in the Momentum Theory (MT). The combination of MT and the segmentation of the blade in different elements, represents the so-called Blade Element Momentum Theory (BEMT) or simply BEM.

The heart of this chapter is based on an in-house code called BEM, which was developed and implemented by my co-workers of CTTC¹. The BEM theory represents the foundation of the code and it is widely known and used. The purpose of this thesis is not explaining this theory in detail once again, because it can be consulted (for instance) in Chapter 2 of the Wind Energy Handbook. Please refer to [1] for further details. Notwithstanding, the main concepts of BEM theory are supplied in the remaining part of this introductory sections for the sake of continuity and completeness of the text.

Thus, the structure of this chapter is as follows. Right after this paragraph, an introduction to Momentum Theory will be exposed before entering in Section 3.1 which will cover the details of the BEM code available in our research group and the upgrades corresponding to the contributions of this thesis to it. The second section focuses on the AeroElectric code, an extension of the BEM code embedding the generator behavior into the aerodynamic part in order to account for the angular speed of the machine without external specification. And finally, in Section 3.3, an optimization for blade designs is exposed and validated in the context of a real case of a 40 kW turbine.

Introduction to Momentum Theory

The objective of a wind turbine is to harness as much kinetic energy from the wind as possible. The turbine can be modelled as a disc that offers a resistance (an axial force) in opposite direction to the wind (thrust). Assuming this simplification and applying the Momentum Theory to the flow around the disc, it is clear that the cross sectional area must increase after the rotor as shown in Figure 3.1 in the left side. The conservation of momentum equation in the axial direction, together with the mass flow rate and mass conservation principle will lead us to state that the power extracted from the flow is the one explained by Equation 3.3.

$$P = \frac{1}{2} \rho A_2 u_\infty^3 \cdot 4a(1-a)^2 \quad (3.3)$$

The first part of this equation is actually the total power inherent to the flow and the second part represents the power coefficient, c_p , expressed as a function of the induction factor, a , which is the velocity reduction at the disc section as v_2 shows in Figure 3.1.

¹with Aleix Báez Vidal being the main designer and developer

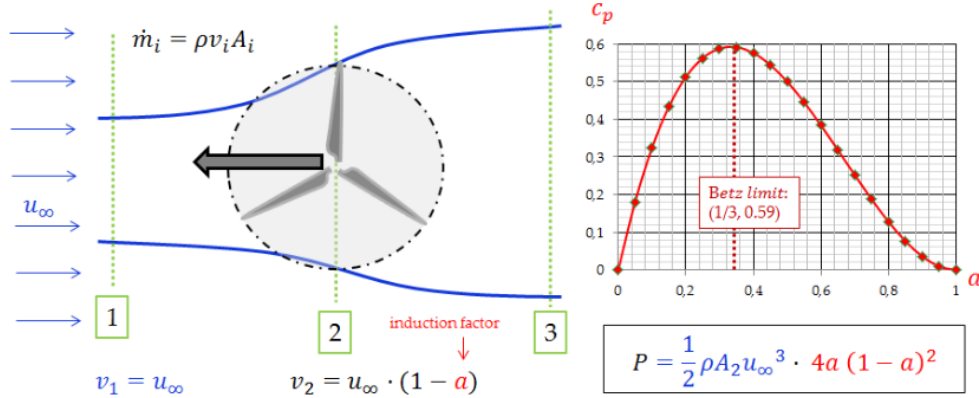


Figure 3.1: Momentum theory and Betz Limit visualization.

The optimization of this equation with respect to the induction factor yields to the value of 1/3 which in turn implies that the maximum c_p is 0.59, i.e. the Betz Limit.

The BEM consists in the multiple application of the Momentum Theory exposed above. The particularity of the method is the fragmentation of the rotor disc area into concentric annulus elements. Usually, twenty elements suffice for a good analysis. The key of this method is merging the lift and drag forces (at each blade element) with the conservation of momentum (both axial and angular). Figure 3.2 shows an schematic view of the velocities and forces in a given element of the blade.

Based on the speeds triangle shown in Figure 3.2, it is possible to link the speeds and the angles according to the trigonometric relationships available in Equations 3.4.

$$\sin \phi = \frac{u_{\infty}(1-a)}{W} \quad \text{and} \quad \cos \phi = \frac{\Omega r(1+a')}{W} \quad (3.4)$$

where: ϕ is the angle between the relative wind speed of the section (W) and the rotor plane; u_{∞} is the free stream velocity; a is the axial flow induction factor; a' is the tangential flow induction factor; Ω is the angular speed of the rotor; and r the radius of the section.

The aerodynamic forces that the flow induces over the section are governed by the definition of Lift and Drag, according to Equations 3.5.

$$\delta L = \frac{1}{2} \rho c W^2 C_l \delta r \quad \text{and} \quad \delta D = \frac{1}{2} \rho c W^2 C_d \delta r \quad (3.5)$$

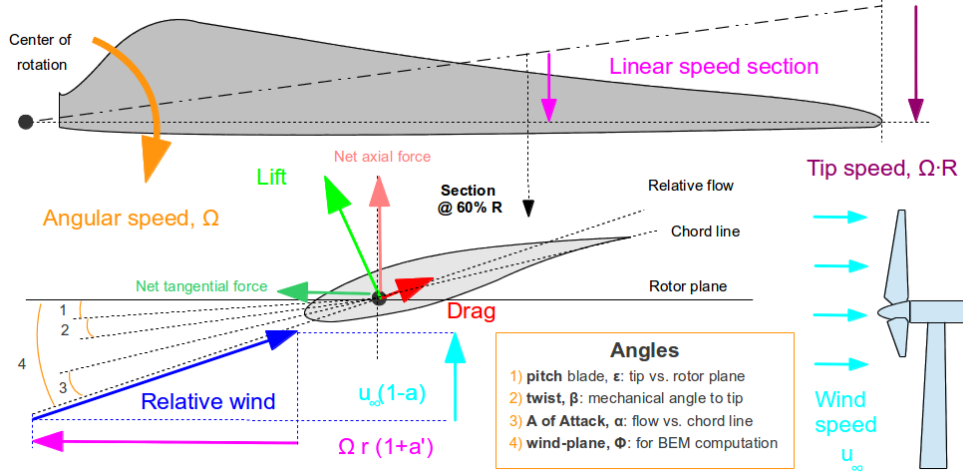


Figure 3.2: Schematic arrangement of elements for BEM

where: δL is the infinitesimal Lift force at the section as δD is to Drag; ρ is the density of the air; c is the chord length; and, C_L and C_D are, respectively, the Lift and Drag coefficients.

Additionally, the rate of change of axial momentum is given by Expression 3.6, where the mass flow rate across the rotor is \dot{m} , and u_3 is the velocity far away downwind.

$$\dot{m}(u_3 - u_\infty) = \rho u_\infty (1-a) 2\pi r \delta r (2a u_\infty) = 4\pi \rho u_\infty^2 a (1-a) r \delta r \quad (3.6)$$

The rate of change in momentum for the specific blade section, when related to the forces experienced on the airfoil, leads to an equation that, when rearranged, shows as Equation 3.7. The same is true for the tangential flow, leading to Equation 3.8. These two equations must be solved in order to find the induction factors, a and a' , that lead to the simultaneous fulfillment of the two conditions. The complexity relies on the fact that it is a non-linear system of equations: the different angles and the lift and drag coefficients are strongly coupled with the induction factors.

$$\frac{W^2}{u_\infty^2} \frac{Nc}{R} (C_L \cos \phi + C_D \sin \phi) = 8\pi (a(1-a) + (a' \lambda \mu)^2) \mu \quad (3.7)$$

$$\frac{W^2}{u_\infty^2} \frac{Nc}{R} (C_L \sin \phi - C_D \cos \phi) = 8\pi \lambda \mu^2 a' (1-a) \quad (3.8)$$

where: R is the Radius of the blade; $\mu = r/R$; N is the number of blades of the rotor; and λ is the Tip Speed Ratio (explained in detail on page 68).

All in all, the characteristics of the problem stated above force an iterative-based resolution method for all the sections in which the blade was divided. This is clearly much more convenient to be solved by means of a code, and that is the purpose of the next section: to provide an explanation with the particularities of our code as well as showing its capabilities. This would be the foundation of all the remaining sections of the present chapter.

3.1 Blade Element Momentum Code

The existing BEM code at CTTC was developed and validated by Aleix Báez years ago. As discussed in the introduction, the BEM equations for a specific blade element are simple but the convergence to a solution is relatively difficult to achieve due to the high sensitivity of the solution. The basic BEM code is equipped with the Prandtl tip and root losses to account for the special situations of those blade regions. Root sections are affected by the boundary layer of the hub and also by the transition from circular sections required for the hub connection to the first airfoil sections. Tip sections are exposed to the wake generation effect due to the abrupt change in circulation.

The code takes a series of inputs, mainly the geometry and operational parameters, and delivers a series of outputs. Both input and output parameters apply over the local and global scope. There is a file that represents the user-BEM interface for inputs and outputs, and its most important characteristics can be seen in Figure 3.3.

The file input global parameters consists of: a) basic geometry variables as the number of blades and its length (radius); b) operational parameters such the angular speed (ω) and the pitch angle; c) wind resources conditions such as the wind speed and the temperature of the air; and d) some numerical parameters for the sake of the root corrections performed in the algorithm. With respect to the output global parameters we can observe, for instance, the torque, the power and the power coefficient. Local input parameters are the corresponding to the chord and twist laws, i.e. the specific values at each radial location. Finally, local output parameters refer to the induction factors, angle of attack, contributions to torque and power, etc.

The contributions of this thesis to the existing code are mainly two: upgrading the interface and preparing the code for massive computations.

As one of the core chapters of this thesis (the present one) focuses in the engineering models, i.e. supply an added value to the BEM theory to be applicable in real wind

Global parameters	Element description
<pre> 1 ===== 2 # Overall Wind Turbine (Blade) Data 3 ===== 4 ----- INPUT ----- 5 NumBlades = 3 [units] 6 Radius = 7.9000 [m] 7 WindSpeed = 9.00 [m/s] 8 PitchBlade = -1.75 [deg] 9 xRoot = 0.040 [-] 10 Omega = 85.9 [rpm] 11 Temperature = 20.0 [°C] 12 ----- OUTPUT ----- 13 MaxChord = 1.8637 [m] 14 AirDensity = 1.205 [kg/m3] 15 TSR = 4.00 [-] 16 Power = 11050.60 [W] 17 Torque = 1227.84 [N·m] 18 Thrust = 1985.07 [N] 19 Cp = 0.1283 [-] 20 Cp/CpBetz = 0.2166 [-] </pre>	<pre> 22 ===== 23 # Blade data sections for BEMT 24 ===== 25 sectElem Chord r/R Twist AngWind a AoA 26 0 1.4200 0.1500 24.3 40.10 0.22 7.12 27 1 1.4500 0.2000 23.5 35.05 0.25 7.44 28 2 1.4200 0.2500 21.0 29.94 0.29 8.17 29 3 1.3700 0.3000 17.9 25.79 0.32 8.83 30 4 1.3000 0.3500 14.5 23.21 0.33 8.46 31 5 1.2300 0.4000 11.9 21.08 0.33 8.18 32 6 1.1600 0.4500 9.8 19.23 0.33 8.00 33 7 1.0900 0.5000 8.1 17.64 0.33 7.87 34 8 1.0300 0.5500 6.6 16.25 0.33 7.77 35 9 0.9702 0.6000 5.4 15.03 0.33 7.69 36 10 0.9252 0.6500 4.4 13.94 0.33 7.63 37 11 0.8801 0.7000 3.5 12.94 0.33 7.58 38 12 0.8100 0.7500 2.7 12.01 0.33 7.54 39 13 0.7644 0.8000 2.1 11.12 0.32 7.47 40 14 0.7234 0.8500 1.5 10.27 0.32 7.31 41 15 0.6864 0.9000 0.9 9.36 0.30 7.01 42 16 0.6529 0.9500 0.4 8.19 0.27 6.30 43 17 0.1000 0.9750 0.0 7.29 0.23 5.54 44 45 </pre>

Figure 3.3: BEM-user interface input-output file

turbine design projects, the author had to work on improving the existing code. A full restructuring of the interface file was done, introducing the conceptual separation of input and output parameters together with a human-readable formatting, renaming of tags and including their units. Additional aspects such as the determination of the air density by means of a state equation and the exclusion of user pre-process considerations such as the identification of the maximum chord were also worked out. Moreover, local parameters were reformatted with variables and units comprehensible at first glance for humans and the reordering, deleting and inclusion of the variables was also part of the actuation. The final outcome of this data restructuring can be seen in abovementioned Figure 3.3.

The second major change in the BEM code was the development of a platform to massively run a series of cases to analyze the performance of a given blade design in wide-enough working conditions sample. Additionally, a post-processing platform to visually present the results was also incorporated for the sake of enabling an easy comparison between different design approaches. Take the example of Figure 3.4, where a blade is tested along a range of 20 degrees of pitching angles, from -5 to 15 ,

and a range of 5 Tip Speed Ratios (TSR)², from 3 to 8. As a consequence of the analysis of one of the specific designs of the 40 kW blade designed for Enair S. L. (see Appendix A, it is possible to draw a couple of interesting conclusions: firstly, the range from 0 to 9 pitch angle and TSRs between 5 and 7 provide a power coefficient above 0.4 which is quite good; and secondly, the maximum values of c_p for TSRs between 5 and 7 occur for the same pitch angle, between 4 and 5 degrees with respect to the rotation plane, having the maximum performance of 0.47 for pitch 4 and TSR 6.

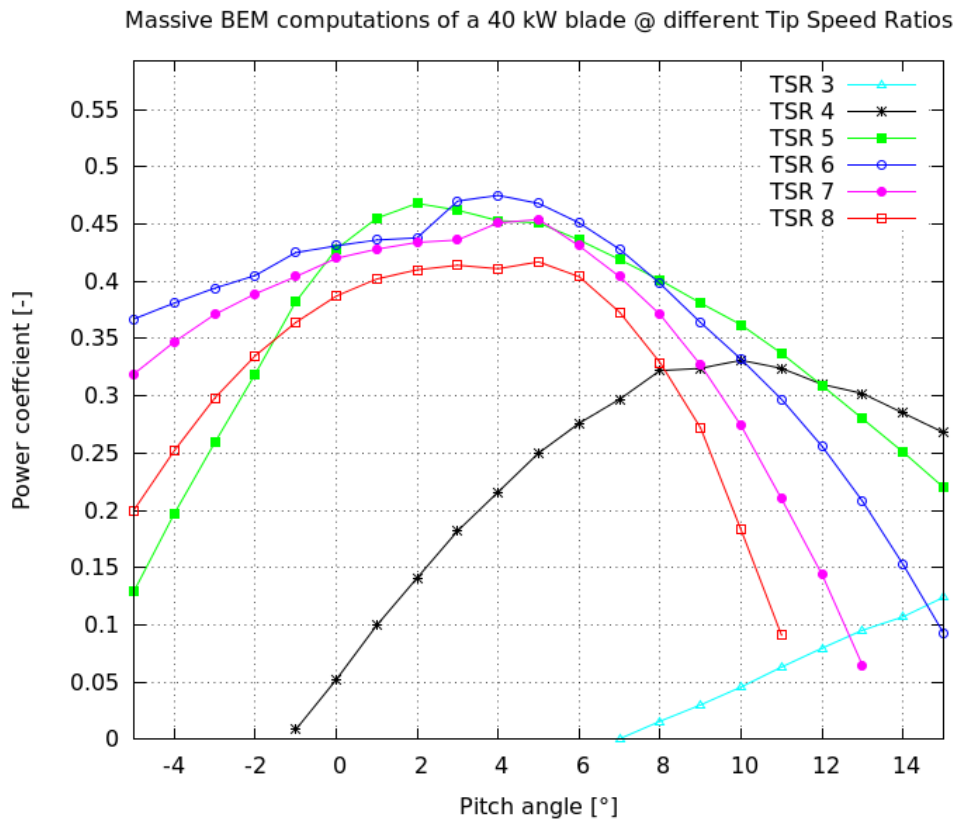


Figure 3.4: Results of a BEM computation over a 40 kW blade

The value of the improvements provided in this section can be summarized by

²to be commented in page 68

stating the next: a) BEM code is much clearer in terms of offering results to the user, and b) BEM has been upgraded to provide a platform for comparing wind turbine blade designs; both aspects contributing to the enhancement of the design process and the interpretation of the computations.

3.2 AeroElectric code

The BEM code is a zero dimensional model to approach to the wind turbine analysis with a good compromise between rigor and computational speed: it is a great option to analyze – in a matter of one second – the performance of a blade in a given tip speed ratio, wind speed and pitch of the blade. Among its weaknesses, it must be mentioned the inability to account for 3D effects due to the fact that any element is independent from all the rest; they are treated independently because the theory is not powerful enough as to consider the elements interaction. Nothing can be done to mend the previous weakness because it is intrinsic to the nature of the method, but there are some other aspects that make the BEM approach relatively uncomfortable and they can be circumvented.

Pay attention to the analysis provided in Figure 3.4. Once the geometry of the blade has been defined and the numerical parameters are chosen for convergence and precision purposes, only two input parameters can be used to model the operational degrees of freedom: the pitch and the TSR. Depending on the combination of these two parameters a better or worse blade performance is obtained – measured by the c_p . First of all, bear in mind that although the Tip Speed Ratio is a crucial and natural parameter for a wind turbine working point, the interpretation of the working point through TSR is not so direct for humans. The TSR is the product of the angular speed times the rotor radius divided by the wind speed, in other words, the ratio between the tip speed and the free stream velocity, see Equation 3.9. It naturally merges the external conditions of the wind, u_∞ , with the rotation of the machine, Ω , in a very useful way. This is characteristic of non-dimensional numbers such as the Reynolds or the Peclet.

$$\lambda = \frac{\Omega R}{u_\infty} \quad (3.9)$$

For the sake of performance clarification and display, it would be interesting to provide the power coefficient as a function of the wind speed, among others. Actually, one of the reasons to use the TSR is to avoid the need to specify the angular velocity.

It is not at all a trivial aspect to know the rotational speed of the machine because it depends on the wind speed and the pitch angle of the blade. With the BEM approach presented above, we either suffer the lack of a robust system to determine the angular speed or we force the user to supply this value. In both cases, we are not in an ideal scenario.

Fortunately, this problem can be sorted out with the integration of the overall electric behavior into the aerodynamic part. The idea is to include the behavior of the generator into the aerodynamic part to give birth to the AeroElectric algorithm which sorts out the angular speed problem. The generator model does not account for electric circuits, there is no need to implement this level of detail to obtain a good accuracy. Instead, the relationship between the electrical torque and the angular speed is used for the coupling. The only required piece of data is the chart with the angular speed and the electric torque variables. Usually, this relationship is exactly linear, so the only requirement is the nominal torque and the nominal angular speed to build the relationship by means of linear interpolation. Sometimes, the relationship is actually built by a slight curvature above the line connecting the nominal point and the origin of coordinates.

Then, the idea behind that simple procedure to approach this electrical part is to find a way to make the system consistent. When the turbine is working steadily at a given wind, pitch angle and angular speed, it means that the global aerodynamic performance matches the one from the generator, i.e. the power and torque from aerodynamic and electric parts are just the same.

From the computational point of view, the objective becomes to find the exact angular speed that leads to the same torque employing the two systems: the BEM code for the aerodynamic part and the Torque-rpm curve of the generator. This is true for a specific wind and pitch. It is expected that for any wind speed, it would exist a pitch value together with the angular speed that leads to the torques match, the challenge is to find those values. Actually, it can happen that there are several pitch values leading to the match. Ideally, the proposed algorithm would be able to supply which is the pitch and angular speed values that lead to the machine equilibrium³ that, from all the possible working points, yields the maximum power generation.

3.2.1 Algorithm

The algorithm is conceptually divided in three parts: data input, core of the process and postprocessing. The overview of the algorithm is available graphically in Figure 3.5, and

³where the electric and aerodynamic torque match

briefly introduced in this paragraph before getting into the details. The first part is the data input and consists of the three elements: A) the BEM file described in the previous section; B) the range of values for pitch and wind to be tested; C) the torque-angular speed curves of the generator. The second part is the core of the algorithm and it is arranged in a triple nested “for” loop that goes through the specified ranges of values from wind, pitch and omega. In this way, the BEM code runs all those cases to achieve the purpose exposed in the introduction of the present section. The third part deals with the output and the presentation and plotting of the relevant information. Now, it is time to explain each of these parts in detail.

Data input

The first part of the AeroElectric algorithm deals with the data input. The most basic information required for the algorithm is exactly the one presented in Section 3.1, see Figure 3.3. For the specific case of the AeroElectric code, not all the information presented in Figure 3.3 is necessary, since the wind speed and the angular speed will be overwritten for each of the cases to be run. The crucial information to be provided corresponds to the geometric category, i.e. the radius of the blade and the twist and chord laws together with the airfoil specifications (also C_l and C_d as a function of A_oA , see Figure 3.6).

The second element of the data input is a table with 12 parameters that define the ranges of the triple loop. The three first real numbers correspond to the wind specifications. The initial value to start the study is provided: it is typically 0.5 m/s. It makes no sense to start at 0 m/s because the wind turbine will not move, but values up to 5 m/s it may be acceptable to begin with. The second value sets the end of the wind analysis: it might be 15, 20 or 25 m/s according to the specific design of the turbine and its rated power. Typically, the higher the rated power, the higher this parameter (up to a limit). And the final wind parameter is the step that will be used for the analysis, 1 m/s is acceptable but 0.5 m/s is ideal. Thus, if the initial value is 0.5, the final value is 15 and the step is 0.5 m/s, a total of 30 wind points will be analysed: 0.5, 1.0, 1.5, 2.0, ..., 14.0, 14.5 and 15.0 m/s. The remaining 9 parameters of the table correspond to the repetition of the same principle described for the wind but for the case of pitch in three different scenarios (zones), three parameters for each zone: beginning, end and step of the range. The first wind zone corresponds to the start process of the machine and it usually goes from the specified beginning point of the wind range till 5 m/s, the second zone stretches until the pitching for power regulation (i.e. the wind speed in which rated power is achieved), and the third zone contains the remaining wind values. For

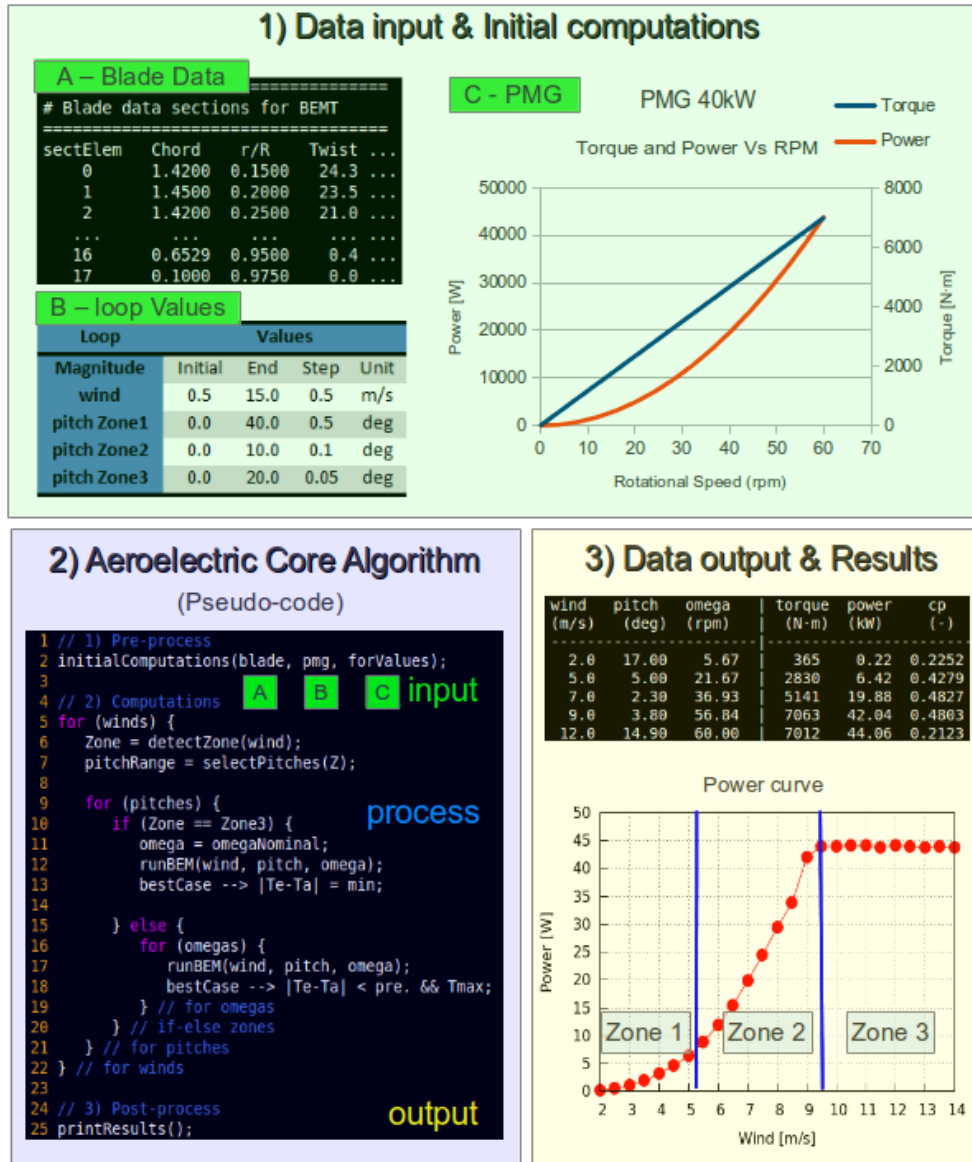


Figure 3.5: Schematic representation of the AeroElectric phases

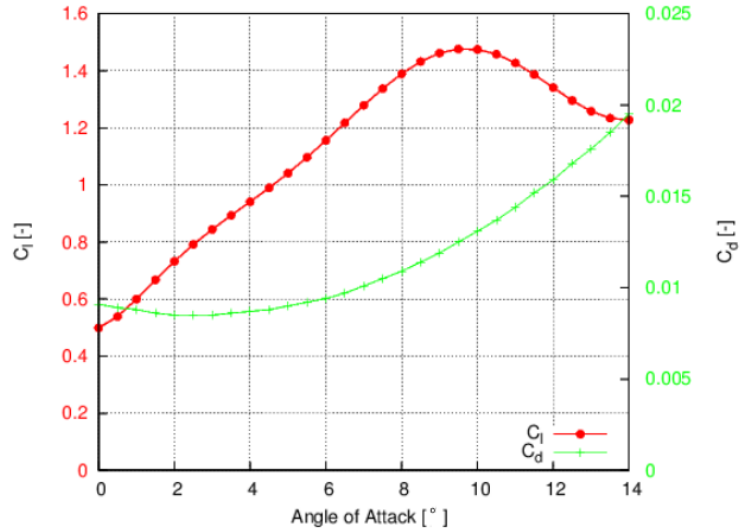


Figure 3.6: Lift and Drag coefficients for the selected airfoil

each of these zones, the sensitivity and expected pitch value for convergence is different, check Figure 3.5 (block 3, right-bottom corner). The first zone usually converges at pitch values greater than 10° and the computing precision is not as important as in the other two zones. For Zone 2, the expected value for convergence is below 10° and the precision should be lower than 0.5° . Finally, in the third zone, values for convergence usually go from 5° to 25° requiring finer precisions than those for Zone 2. In the end, these 12 parameters are used to exert control over the computational precision and duration of the simulation. Additional details will be provided later on when dealing with Figure 3.7.

And finally, the third element of the data input corresponds to the definition of the generator's behavior (T-rpm). In this case, both the linear or nearly linear behavior are approached in the same way: with a set of discrete 2D points from the origin to the nominal point. In order to handle this type of data, a C++ class has been created (Polyline) for the sake of performing point evaluation, interpolation, extrapolation and similar other demanded tasks, apart from the reading and printing functions.

Core of the algorithm

So far, the first part of the algorithm is covered: the data introduction. Now, it is

turn for the second part which is the core of the algorithm computation. Figure 3.5 (block 2) presents the pseudocode for the triple loop. The first loop is the wind one, the first operation to perform is the detection of the wind zone; from there, the pitch range is selected according to it. After doing that, the second loop comes in to deal with pitch angles, in here, a crucial differentiation needs to be done. In case of being in Zone 3, there is no need to go through the (third) omega loop because the omega will be constant: the rated one; consequently, the BEM is run and the torques from the aerodynamic and electric systems are compared: if they differ by less than 1% the results are stored as successful. Back to the pitch loop, if the current wind speed value belongs to Zone 1 or 2, then, the third loop is mandatory (omega). Now, departing from the analytical estimation of the omega to be converged⁴, the third loop is automatically built.

For all those cases with the specific combination of wind, pitch and omega, the electric and aerodynamic torques are computed and compared: when they are virtually the same it means that the machine works stably. For a given wind speed, several combinations of pitch and omega values might fulfill the torque match. From those points fulfilling the torques match, the one with the maximum torque is selected as the final point.

All in all, the triple loop checks the convergence of a combination of wind, pitch and omega for Zones 1 and 2, and a simplified version is used for Zone 3 due to the pitch regulation. The cases leading to a greater degree of convergence (similar torques) and greatest power and torque generation are selected as the best ones and they get ready for the data presentation.

Data output

The third and final part of Figure 3.5 to be commented is the results presentation. The algorithm prints automatically a file that contains all the relevant information from the computations. Obviously, the three key components are shown in first place: wind, pitch and omega. They are already the best pitching value (either in leading to maximum torque or the ones for power regulation – in case exceeding the nominal power) and the omega at which the machine will rotate for the specified wind speed. The parameters for performance evaluation such as the torque, the power, the power coefficient are also provided. Additionally, the TSR, the angle of attack of the tip and many other indicators can be shown in demand.

With the current approach of data input, algorithm processing and output of the results, a complete virtual testing of a blade design can be done in a matter of min-

⁴enabling a sufficient percentage variation above and below the value

utes (2 – 10, depending on the computing precision). Moreover, the plotting has been automatized as to produce a chart with all the information at a glance, see Figure 3.7.

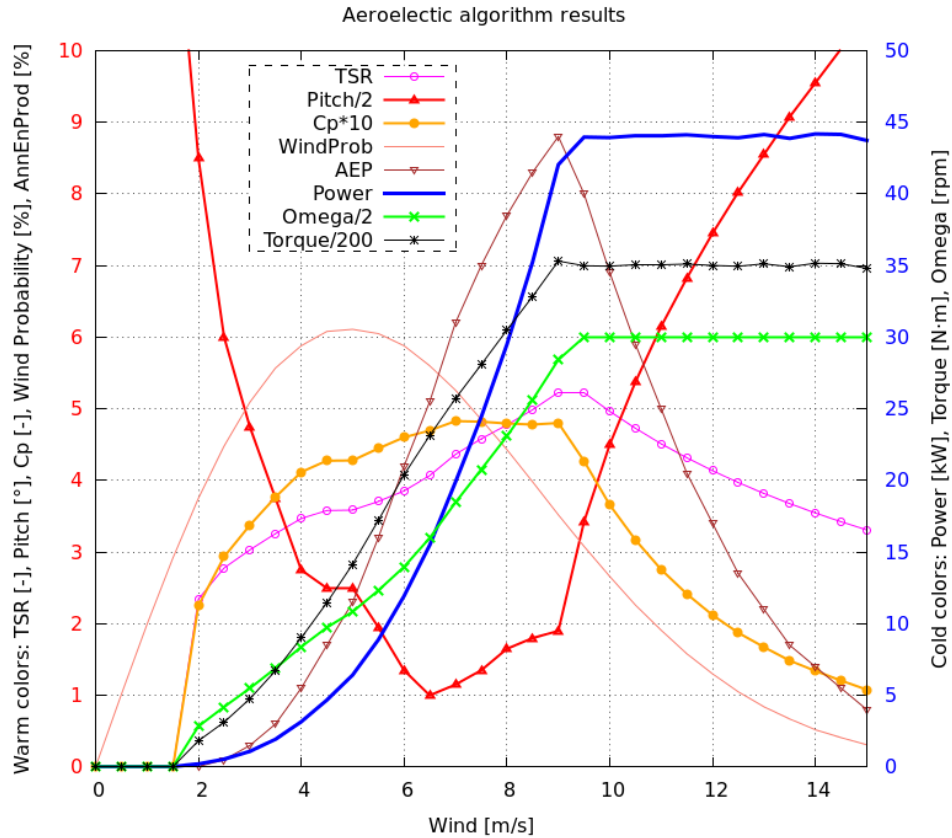


Figure 3.7: Results of the AeroElectric code over a 40 kW blade

Taking advantage of the platform already built and the software development program, the author considered of interest to include an additional idea to complement the AeroElectric algorithm. As it is, the algorithm provides a quite detailed level of information to assess the performance of a particular design, but it is perhaps too complex for a fast comparison between different models. It would be really interesting to compare different designs by means of using the least number of parameters possible,

ideally one. When this is the case the indicator would be the cost of energy as Botasso is presenting in [2] or, if not possible, rely on the Annual Energy Production (AEP).

The AEP is the total amount of energy produced in a year. In the international system, the Joules would be the unit of measure, but it is more common to offer MWh or kWh. If those numbers are worked out further, the same magnitude can also be shown as full equivalent production hours, which tells how many hours would have been required for the machine to work at full capacity to generate the same Annual Energy Production. Similarly, the AEP can be divided by the total number of seconds in a year to provide the average power production which is also a good strategy to understand which is the utilization rate of the turbine by comparing it with the rated power.

The first requirement to compute the AEP is to model the wind speed pobabilistic distribution. It is generally accepted that the wind speed follows a Weibull distribution whose parameters are adjusted to match the conditions of each location by modulating the amplitude and the length of the tail of the distribution. Thus, a Weibull-2-7 is a distribution with the frequency peak around 5 m/s with 90% of winds below 10.5 m/s and 99% of winds below 15 m/s, see Figure 3.8. The second aspect, the idle time, is provided as a percentage that refers to maintenance tasks and other reasons that force the machine to be stopped even though profitable wind speeds are registered.

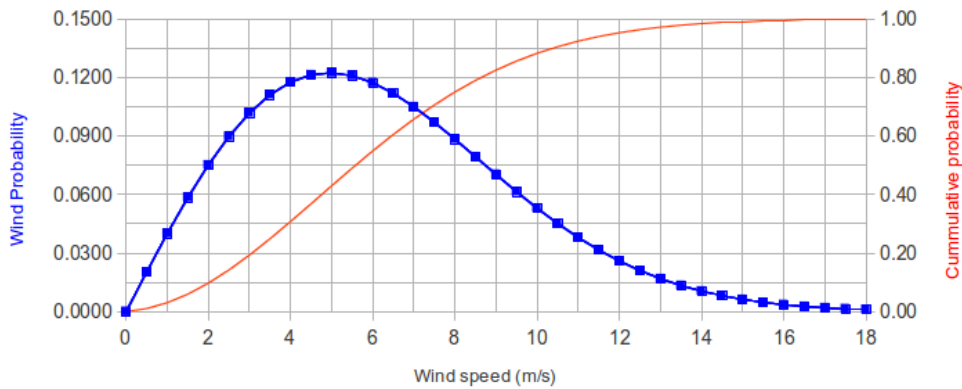


Figure 3.8: Wind characterization by means of Weibull Probabilistic Distribution with parameters 2 and 7

With the Weibull distribution and the idle time specified by the user, the AeroElectric code can compute the AEP departing from its output data based on the wind and power production vectors. A simple vector multiplication weightes the power with the wind

probability, which results in the final value of the AEP in form of averaged power.

At this point, the AeroElectric code combined with the introduction of the Weibull distribution and the AEP calculation becomes a closed box. It receives a series of pieces of data representing the blade geometry, the generator behavior, the wind resources at the desired location in order to assess the performance of the blade in those conditions by means of a single number: the average power of the turbine. This is a tool that definitely adds value in the design process of a wind turbine.

Despite the undeniable added value of the present tool, it can still be brought to a next level because it is susceptible to be combined with an optimization algorithm in order to refine and improve the input parameters based on the final indicator (any form of AEP). The next section deals with the merging of the AeroElectric code with the optimization engine.

3.3 Optimization

The ultimate objective when designing a wind turbine is to produce the maximum energy at the minimum cost possible. From the aerodynamic point of view, the energy production mainly depends on the geometric design (size and shape of the blade) and the cost depends on the amount of material and difficulty to manufacture. The maximization of the power performance can be derived from the application of the BEM theory. Analytical expressions (see Equations 3.10 and 3.11) can be obtained for both the twist and chord laws, and they would lead to the power maximization in the specific conditions that they were designed for.

$$\phi = \frac{2}{3\lambda\mu} \quad (3.10)$$

$$c = \frac{8\pi R \sin \phi}{3NC_{l,des}\lambda} \quad (3.11)$$

where: ϕ is the twist angle; λ is the Tip Speed Ratio; μ is the non-dimensional radius r/R ; R is the blade radius; N is the number of blades and $C_{l,des}$ is the lift coefficient of the airfoil at the design point (based on the desired Angle Of Attack).

Let us interpret the previous expressions. There are four parameters that influence the twist and chord: λ , C_l , N and R . The decision of the number of blades and their radius is something rather fixed for a given turbine while the Tip Speed Ratio and the Lift coefficient are parameters that allow us to play with. The TSR, or λ , appearing in the previous equations is a sensitive design parameter, meaning that the maximization

will happen when the machine works within a small range around the defined value. Obviously, once the value has been used for the blade design, and the blade is built in accordance to it, there is no room to change it⁵. This means that if the TSR varies at different working points, the correct value needs to be selected for the design. How much does the TSR change? Rescuing the plot offered in Figure 3.7, we can observe that most of the working conditions lay in the range 3.5 – 5.2. The range may be small in terms of absolute change, but it is actually huge in terms of relative variation and impact on the turbine’s performance. An acceptable idea would be to approximate the best value with the midpoint of the range: 4.4. However, the deduction of this value is based on an iterative process because, obviously, the chart from which it is deduced was obtained for a particular turbine whose original design-TSR might be different than 4.4. All in all, it would be much more appropriate to employ optimization methods to find out the TSR that leads to the best conditions possible.

The whole present section deals with the exposition of the optimization platform while being applied to a real case of the project that CTTC worked together with Enair S.L.⁶. The objective of the project is to design a three-bladed pitch-regulated 40 kW downwind turbine whose rated angular speed is 60 rpm. The blades were designed from scratch and all the design details were responsibility of CTTC and the author of this thesis.

A basic requirement before even thinking about an optimization algorithm would be the platform to define the blade geometry based on Equations 3.10 and 3.11, as well as flexibility on the geometry definition phase. The C++ Object Oriented Programming approach employed in the construction of the BEM code, [3], allows us to reuse some of the code. For the sake of the claimed flexibility, an additional object was created: the Multi-Lineal Design Tool (MLDT). The mathematical concept to do so is the polyline, i.e. the definition of a curve determined by an array of points. The object is equipped with interpolation, extrapolation and other useful functions. The MLDT, combined with file reading and modification, enables the automatic geometry variation for the sake of creating the multiple designs demanded by the optimization engine.

3.3.1 The optimization tool

Optimization is a crucial component not only in engineering but across many life disciplines. Therefore, it has been widely studied ranging from derivatives in basic calculus

⁵perhaps in the future, wind turbines will have flexible sections to modify the twist law depending on TSR

⁶for more details consult Appendix A

to the most advanced codes available today. The so-called evolutionary algorithms are robust and flexible but very time consuming. For concrete cases one might rely on local search algorithms but they can not deal with all levels of complexity. Several free-licence packages are available covering those topics to optimize in both single-objective and multiobjective cases. CTTC has made an effort in the last years to attain an optimization tool for the cases that we encounter in the field of Computational Fluid Dynamics and Heat Transfer. Since 2012, Imanol Aizpurua-Udabe has been working to integrate and adapt the available open-source software tools into *Optimus*, the in-house library that he has built during his doctoral studies [4].

Optimus is an optimization engine aiming at High Performance Computing that integrates the gradients methods with genetic algorithms, allowing the use of one or another on demand through specifiable switching criteria. It can deal with single or multiple optimization objectives, both in continuous and discrete problems. As mentioned before, the platform integrates (three) external tools. First, *Paradiseo*⁷, which is the result of an European joint work aiming at a general tool able to tackle all kind of optimization problems by means of using discrete local search methods that work in parallel, see [5] and [6] for details. Second, *Dakota* [7], a C++ library developed by Sandia National Laboratories that works with MPI (Message Passing Interface) whose specialization is systems analysis. And the third one, *Trilinos* [8], which is also developed by Sandia N.L., aiming at mathematical libraries specialized in parallel solver algorithms to give support to solve complex multi-physics engineering and scientific problems. Special attention was put in MOOCHO⁸, the local search algorithm, in order to be incorporated to Optimus.

Several validation tests have proven the effectiveness of the Optimus library, mainly in two categories: purely mathematical problems and real engineering problems that CTTC had to solve in the context of ongoing projects. In the mathematical side, four benchmark functions were selected to test the performance of single-objective capacities of the code: Rosenbrock, Rastrigin, Schwefel and Griewank. Additionally, two cases were also optimized with the present code: 1) a domestic refrigerator, where the thickness of the refrigerator and freezer wall had to be chosen for design; and 2) the exact section of a pipe employed in a specific flow pattern with incompressible flow.

The most important idea behind the optimizer employed in this thesis is the single-objective optimization by means of using a combined approach: mutation and parenting. These two ideas are used to obtain the individuals (cases with a specific parameter

⁷PARAllel and DIStributed Evolving Objects

⁸the Multi-functional Object-Oriented arCHitecture for Optimization from Sandia N.L.

combination) of a given generation (the group of individuals). Each generation contains a fixed amount of individuals across time. These individuals are the result of the *random* modifications –mutation– and the gene crossing –parenting– based on the traits of the best individuals of the previous generation.

Thus, Optimus, whose foundations are the ones presented above, represents a piece of appropriately validated software ready to be applied to the present case.

3.3.2 The application to wind energy

This subsection deals with the application of the optimization tool exposed in subsection 3.3.1, *Optimus*, to the particular case of wind energy, more concretely to the blade design. There are many ways to approach the optimization process, but the one proposed in this thesis consists of 5 steps that look at the blade design from different angles in order to achieve the best possible solution. Before getting into the details of those steps, they will be briefly exposed in the next paragraph.

The 5 steps of the proposed optimization approach are the following: 1) selection of a reference case which represents the starting point; 2) exploration of the best design employing ideal twist and chord laws based on two degrees of freedom: the C_l and the TSR; 3) departing from step 2: freeze of the best twist law and perform a systematic local variation of the chord law; 4) departing from step 3, invert the roles of twist and chord laws: firstly, individual modifications of twist, and finally a checking of all together; 5) departing from the best laws defined by steps 3 and 4, the maximum chord and maximum twist are limited (for constructive purposes) to assess the best trade-off between energy production and ease of construction.

Thus, in order to find a balance between a good definition of the blade and simplicity of calculations, the laws were modelled with a 5-point-polyline at sections 20, 40, 60, 80 and 99 %. Another relevant issue is the individual-collective testing performed in the steps 3 and 4. As long as BEM is a zero-dimensional model whose foundations assume that elements from different annulus do not interact, the blade can be assessed and optimized with the superposition principle. Anyway, the complete testing has been provided for the sake of completeness of the text.

Now, one by one, the application of the previous steps to the 40 kW case will be detailed in the next list:

1. **Reference.** The first desirable step of the process – although not indispensable – is to select a reference case that will be used to compare the battery of cases during the optimization procedure. In this way, the user will clearly have the

order of magnitude of the case to improve. Bear in mind that several previous work must be done by the engineer before the optimization is carried out. For instance, the decision of the machine radius must be made in early stages in function of the desired rated power. Moreover, the airfoil family should be also selected in accordance to the desired objectives and other circumstances such as the environmental consideration (ice accretion, dust, etc.). Previous experience might be used to define a blade that will set the scope of the process. In addition, a first AeroElectric simulation might also be useful.

In the example used to show the optimization process, the original blade design for the 40 kW machine was selected. It was carried out when the optimizer was not developed yet, so its design was purely based on human interpretation of several independent point-by-point AeroElectric computations. The airfoil was selected in order to have a smooth stall, a relatively high maximum lift and a low sensitivity to dirt. The airfoil Lift-Drag curves can be consulted in Figure 3.6 and the chord and twist laws are provided in Figure 3.9.

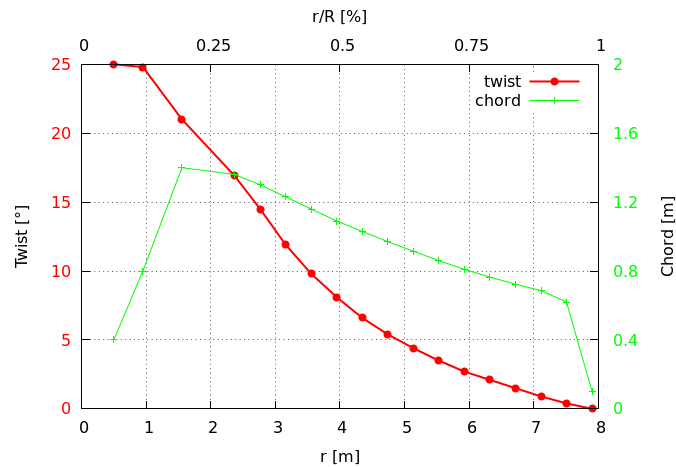


Figure 3.9: The twist and chord laws for the original 40 kW blade without the intervention of Optimus

- Ideal laws shaped by TSR and C_l .** As it was commented before, once the first layer of design parameters has been decided – such as blade radius and airfoil family – the key parameters for the blade design are TSR and C_l (see Equa-

tions 3.10 and 3.11). Therefore, a first simulation to be carried out is the one corresponding to an exploration of the best combination of parameters. Figure 3.10 reveals four images of the optimization sequence during the evolution of the genetic algorithm. As we can see, black-dark colors represent low values of the Annual Energy Production (Averaged power) predicted by the $TSR-C_l$ tandem, while cyan-red colors point to the best combinations. In this analysis, as it can be deduced from the right-bottom image, a good blade design might be achieved with the next parameters, $TSR = 4.0$ and $C_l = 1.8$, which would lead to an average power of 9,785 W.

3. **Chord law.** Now, departing from the chord and twist laws that the $TSR-C_l$ parameters generate, the twist law is frozen as it is, and the chord law will be slowly modified. The twist and chord laws are approximated with 5 points distributed along the blade span: 20, 40, 60, 80 and 99 %. Consequently, 5 steps will be taken in order to approach the chord law optimization, freeing one of each degrees of freedom at a time. In this way, in the first step, the optimization will run multiple times the AeroElectric code with the frozen twist law and the section corresponding to the 20 % will have its chord freed, whose ranging r/R values go from 0 to 1. The optimization algorithm should find the r/R value leading to the maximum power alone. As expected, the evolution of the 20 % r/R should follow a converging pattern and the average power should grow as generations go on for all: minimum and maximum individual as well as the average value. The same routine is repeated to all the sections. Figure 3.11 depicts the evolution process for the 99 % section (tip). As it can be observed, the value converges to 0.068 (6.8% of the 7.9 m Radius, i.e. 54.4 cm) in roughly that 30 – 40 generations employing 20 individuals per generation.

Table 3.1 provides data in relation to the calculations of this step, but it will be commented in the end of the process.

4. **Twist law.** This step mimicks the previous one. The idea is to freeze now the chord law (optimized with the previous step) and free – one by one – all the twist sections until, in the end, the entire twist law is refined to achieve the maximum average power. This time, beyond the step-by-step approach of one section at a time (where the optimizer should find the best twist in the range 0 – 60), the simultaneous optimization of the 5 sections is performed.

Figure 3.12 shows how the convergence is far slower than the one-by-one approach shown in 3.11, where only 30 generations suffice to find out the optimum section

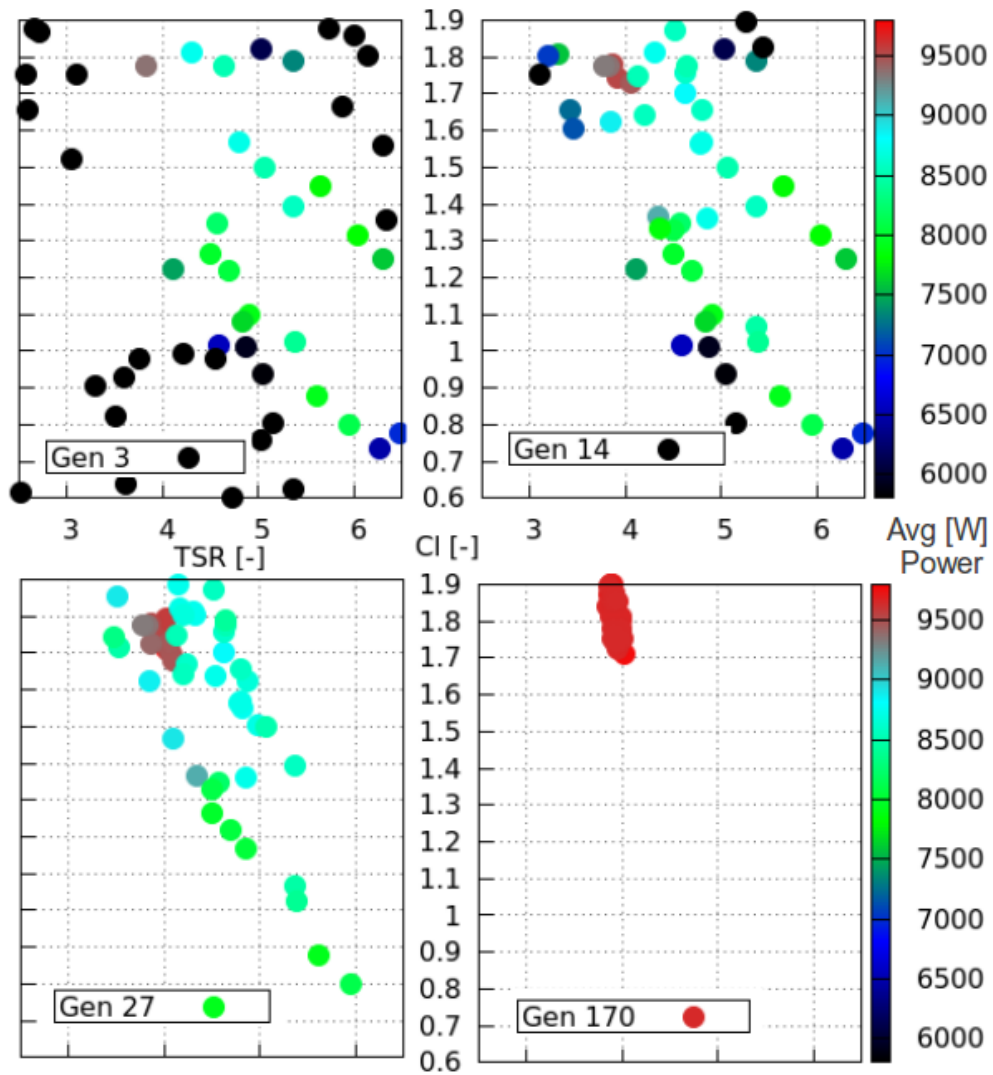


Figure 3.10: Four images of the evolution of the genetic algorithm when employing TSR and C_l as design parameters for ideal laws of twist and chord

with nearly no dispersion across the generation individuals, while in this case 600 generation is not enough to achieve the final value with less than 1 % error. De-

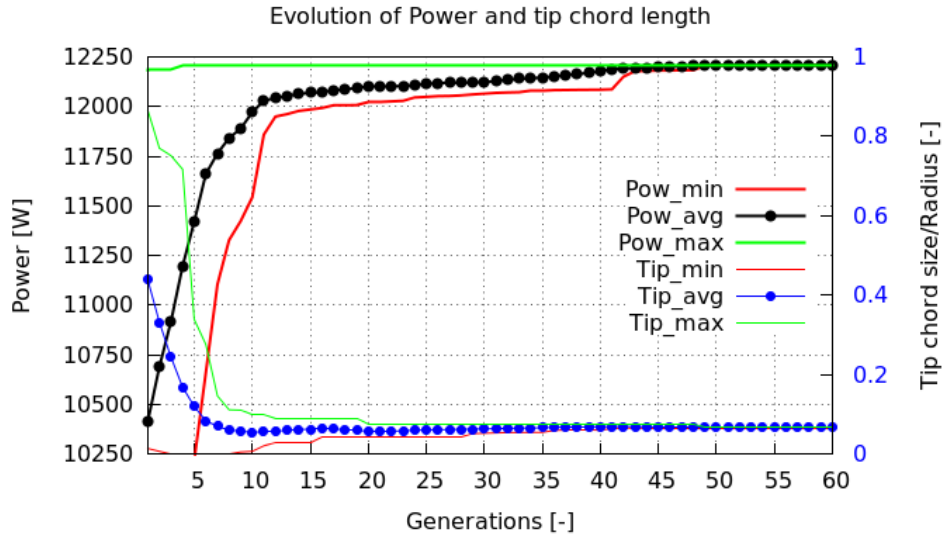


Figure 3.11: Evolution of the optimization algorithm in the chord optimization at the tip of the blade

MaxChord	MaxTwist	Power	Variation
m	°	W	%
2.00	25.0	15,696	reference
1.90	25.0	15,638	-0.37
1.80	25.0	15,630	-0.42
1.60	25.0	15,633	-0.40
1.40	25.0	15,559	-0.87
1.20	25.0	15,490	-1.31
1.00	25.0	15,306	-2.48

Table 3.1: Effects (on power) of the chord limitation keeping the maximum twist constant.

spite the uncertainty, the values seem to slowly converge and it actually coincides with the results of the one-by-one approach – not provided here for the sake of brevity. This test represents the truthful validation that, in terms of results, it

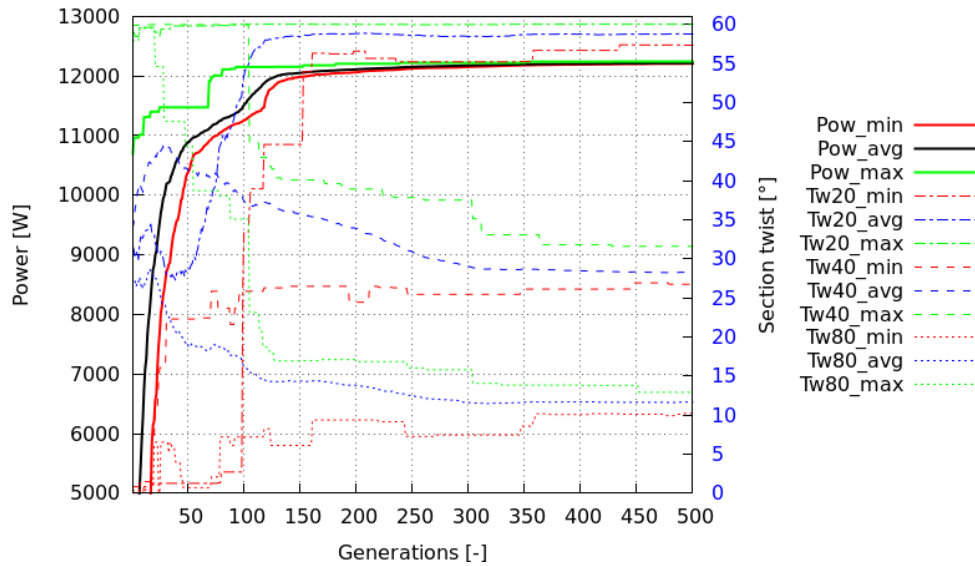


Figure 3.12: Evolution of the optimization algorithm when dealing with the whole twist law at the same time

is actually the same to perform the one-by-one approach or the all-together one. However, the computational cost of the former is much affordable than the latter: notice that just 150 generations will lead to the totally converged results while, in the second scenario, 3 to 4 times more generations are required to achieve similar confidence levels.

Table 3.2 provides data in relation to the calculations of this step, but it will be commented in the end of the process.

5. **Limiting chord and twist.** When integrating the results from step 3 and 4, the obtained chord and twist laws, as expected, tend to present the behavior of aerodynamically "ideal blades", i.e. having large twist and chord at root sections. Therefore, in the present case, the obtained values at the 20 % section are: a chord of more than 4 meters (52.7% of blade radius) and a twist angle of 58.4° . Clearly, those values are too high as to be considered reasonable for a real blade design, but reducing them would also lead to lower energy productions. It should be kept in mind that the objective is to produce the maximum energy at the lowest cost possible. Having such huge twist angles at the root makes the manufacturing

MaxChord	MaxTwist	Power	Variation
m	°	W	%
2.00	25.0	15,696	reference
2.00	24.0	15,650	-0.29
2.00	23.0	15,660	-0.23
2.00	22.0	15,646	-0.32
2.00	21.0	15,578	-0.75
2.00	20.0	15,591	-0.67
2.00	18.0	15,430	-1.69
2.00	16.0	15,400	-1.89
2.00	14.0	15,272	-2.70
2.00	12.0	15,101	-3.79
2.00	10.0	14,904	-5.05

Table 3.2: Effects (on power) of the twist limitation keeping the maximum chord constant.

more complex (and therefore costly). Furthermore, those big chords seem to be a waste of material and also a burden to transportation and aesthetics. The key question arises: what is the right balance between energy production criteria and manufacturing and material costs? To answer this question accurately enough, a reliable cost model should be built, in the line proposed by Botasso in [2]. Such models fall out of the scope of this thesis, so qualitative criterion is employed to make the decision, preferably in the presence of the maximum information possible in terms of the effect of marginal chord and twist reduction over the final average power.

Now, with the degree of confidence that the optimization paradigm provides, it is possible to perform an optimization and sensitivity analysis. Figure 3.13 offers a clear image of what is the effect of *cutting* the ideal laws at the *MaxChord* and *MaxTwist* pair of values. The point colors are clearly showing that the best combination resides in the upper-right corner, where the maximum allowed twist and chord are greatest within the allowed range of values, which is, naturally, what one would expect to find in such simulation. Once again, the internal consistency of the optimizer applied to the AeroElectric code is found to be high.

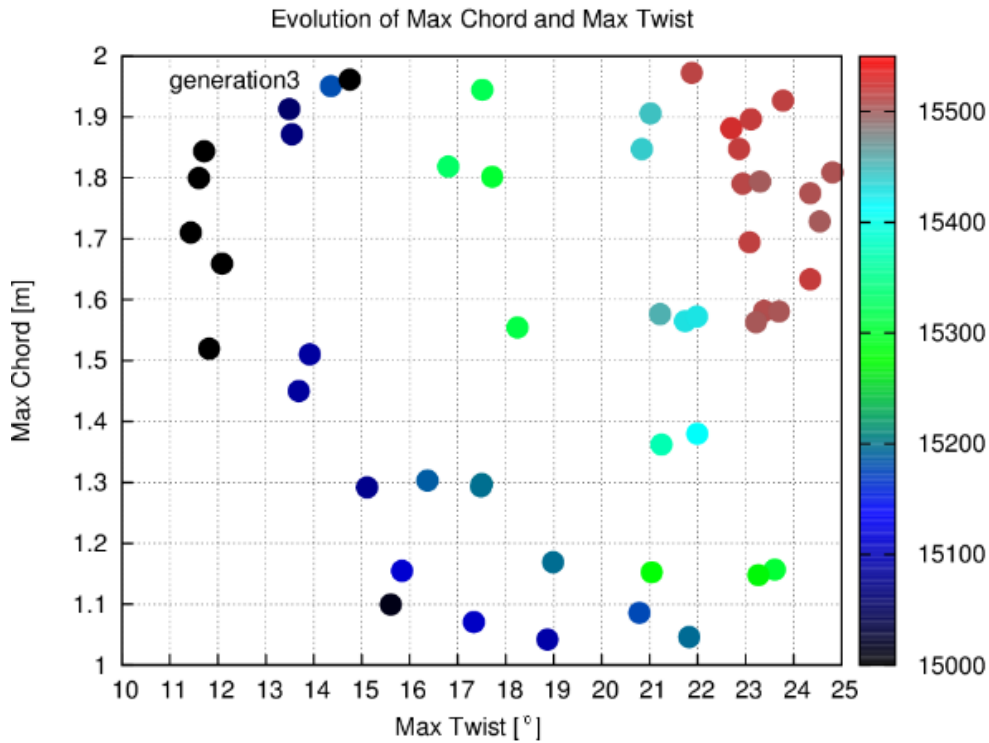


Figure 3.13: Map of individuals in generation 3 when computing the effect of limiting the chord and twist laws with the parameters *MaxChord* and *MaxTwist*, respectively.

The figures supporting the five steps provided above are truly helpful to understand the nature of the problem and the proposed solution. However, and specially when it comes to step number 5, tables are much clearer when data is required to give support in the decision making process. That's the reason why Tables 3.1, 3.2 and 3.3 are provided. The first table explores the effect of the *MaxChord* limitation when the *MaxTwist* is kept constant and the second table does the same but freezing chord and freeing twist. The effect on the power is shown in terms of relative impact with respect to the reference case – which is different from Step 1. The present one employs the laws from Steps 3 and 4 but limiting the *MaxChord* to 2.0 m and the *MaxTwist* to 25°.

In view of the results presented in Tables 3.1 and 3.2, some ideas can be obtained to combine the effects of *MaxChord* and *MaxTwist* together. A couple of new cases were

#	MaxCh	MaxTw	Power	Variation	Comments
-	m	°	W	%	about twist-chord laws
R	2.00	25.0	15,696	reference	ideal with optimizer procedure
A	1.45	25.0	15,283	-2.6	ideal-final real, no optimizer
B	1.20	15.0	15,165	-3.38	ideal with optimizer procedure
C	1.40	18.0	15,422	-1.75	ideal with optimizer procedure

Table 3.3: Final set of cases and conclusion of all the steps of the optimization phase.

run and their results are posted in Table 3.3. Case *R* represents the reference case. Case *A* is the original blade design when Optimus was still being developed. Cases *B* and *C* represent two alternative designs combining the acquired know-how.

Case *B* attains an average power of 15,165 W (representing a 3.4% reduction with respect the reference case) while having a thin blade (only 1.2 m of maximum chord) and a very low twist (of just 15°). Case *C* faces only a 1.8% reduction in average power and still supplies an interesting material saving (maximum chord is 1.4 m) and an acceptable 18° maximum twist.

At this point, engineers have all quantitative data to equip them with the precise information to make the qualitative decision of which is the best design, given the rest of conditions not considered in the model. In the end, the optimizer is a powerful tool that provides to the designer a series of data that enables a more accurate and effective decision making.

3.4 Conclusion

In this chapter we have dealt with zero-dimensional models that enable the engineer to design wind turbines with a good cost-accuracy relationship. At the beginning, the fundamentals of Momentum Theory have been covered slightly enough as to make understandable the necessary BEM introduction that leads to the BEM code explanation. The first contribution of this chapter is the improvement of the BEM-user interface, which allowed the generation of the platform for massive BEM computations as well as the post-processing required for graphic presentation of those results. Once the BEM platform has been brought to the next level, an additional step further is made: the inclusion of the generator's behavior based on the simple (but effective) T-rpm chart.

Such simple conceptualization enables the code to acquire a degree of independence from the angular speed of the machine because the algorithm is intelligent enough as to compute it automatically. For the sake of completeness of the module, the wind resource characterization by means of a Weibull distribution is implemented and used as the baseline for the computation of the Annual Energy Production of the machine when installed in a given location with the specified conditions. This structure, based on the AEP allows the coupling of the AeroElectric code with *Optimus* for optimization purposes. The tool represents an interesting advantage to systematically and automatically optimize a blade design with the least human intervention in the middle process. This technical advance enables the designer to delay the actuation until the point of the decision making, where human intelligence can weight qualitative, quantitative and other related factors in the blade design, (such as aesthetics) with the aim of achieving the best design possible.

References

- [1] Tony Burton et al. Wind Energy Handbook. Wiley. http://library.uniteddiversity.coop/Energy/Wind/Wind_Energy_Handbook.pdf.
- [2] Carlo L. Bottasso Pietro Bortolotti and Alessandro Croce. Combined preliminary-detailed design of wind turbines. *Wind Energy Science*, 2016.
- [3] Paul Deitel and Harvey Deitel. C++ How to program. Seventh edition. *Pearson Education, Inc.*, 2010.
- [4] Imanol Aizpurua-Udabe. *Parallel optimization algorithms for High Performance Computing. Application to thermal systems*. PhD thesis, Universitat Politècnica de Catalunya, Spain, 2017.
- [5] S. Cahon, N. Melab, and E. G. Talbi. Building with ParadisEO reusable parallel and distributed evolutionary algorithms. *Parallel Computing*, 30(5):677–697, 2004.
- [6] Paradiseo, a software framework for metaheuristics. <http://paradiseo.gforge.inria.fr>.
- [7] B. M. Adams, L. E. Bauman, W. J. Bohnhoff, K. R. Dalbey, M. S. Ebeida, J. P. Eddy, M. S. Eldred, P. D. Hough, K. T. Hu, J. D. Jakeman, J. A. Stephens, L. P. Swiler, D. M. Vigil, and T. M. Wildey. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and

sensitivity analysis: version 6.0 user's manual. Sandia Technical Report SAND2014-4633, July 2014. Updated November 2015 (Version 6.3), 2015.

- [8] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, and E. T. Phipps. An overview of the Trilinos project. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):397–423, 2005.

CFD models

Wind energy is the technology that humans employ to convert the kinetic energy of the wind into electricity to be used at homes and industry. The wind turbine is the minimum technological unit that enables this energy transformation, although the massive energy production for consumption takes place in the so-called *wind farms*. A wind farm is a collection of wind turbines installed in the same zone, typically within a few km. The wind turbines of the farm can share the wind resources and the same technological facilities (such as the transformation station), in order to take advantage of the economies of scale of such components.

In the same way that wind turbines are the main components of wind farms, airfoils are the elements that configure wind turbine (blades) as shown in Chapters 2 and 3 through the AutoBlade, BEM and AeroElectric codes. Thus, from the aerodynamic point of view, there are clearly three levels of design: the airfoil, the turbine (with special emphasis in the blade) and the wind farm.

Usually, the direction in which those levels are taken into account is the next: *Airfoil* \rightarrow *Turbine* \rightarrow *Farm*, meaning that the airfoil shapes the turbine and the turbine determines how the wind farm behaves. The reverse design direction, *Farm* \rightarrow *Turbine* \rightarrow *Airfoil*, is far less common. It would mean that the final wind farm layout would set the conditions for each turbine to perform an individualized wind turbine design in accordance to the level of turbulence, wind rose, wind speed probability and other wind turbine's wake impact. Additionally, the airfoils for each of those individual wind turbines would be selected according to the previously detected characteristics as well as other external factors such as dust, icing probability, etc. See Figure 4.1 for a visual description of this concept. In the end, a truly ideal design system should

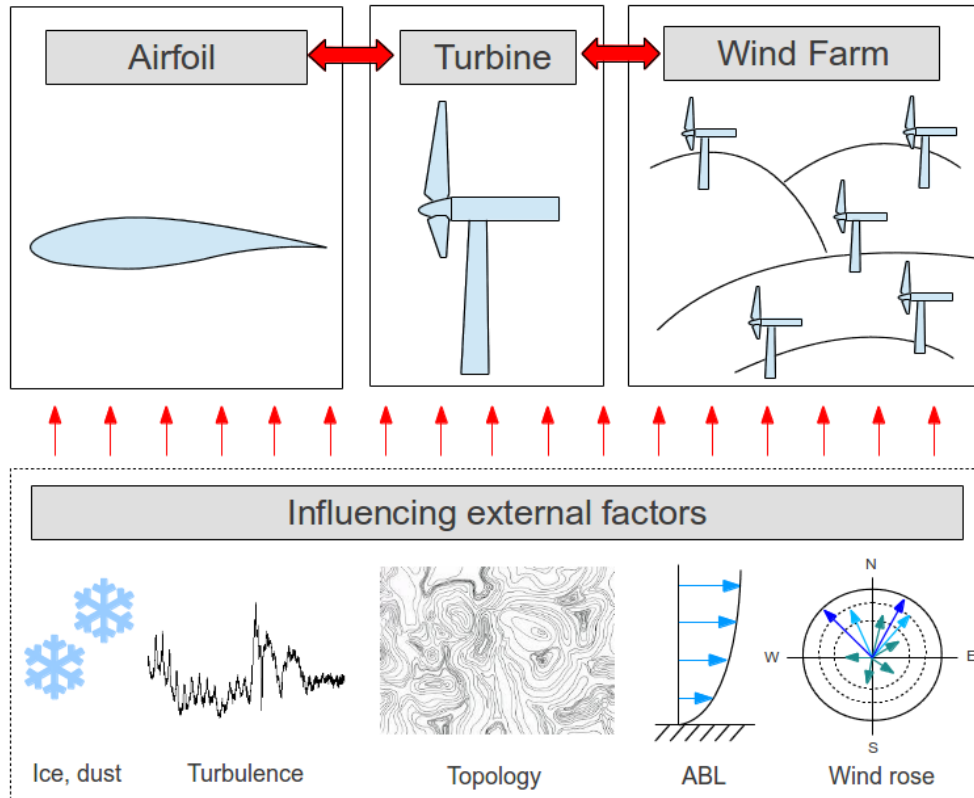


Figure 4.1: The three levels of design of wind energy

take into account all three levels simultaneously, with the maximum level of detail and accounting for the interaction between them.

This chapter is focused on the wind turbine level. The near wake generation and the power or torque production. The AeroElectric code based on the Blade Element Momentum Method (BEM) presented in Chapter 3, works great for a simplified view of the wind turbine operating states with an acceptable degree of accuracy: bear in mind that most of the current machines have been design with this method. However, BEM-based methods are limited, since they rely on 2D experimental (or numerical) results of airfoils and the underlying theory of the method cannot cover unsteady nor three-dimensional effects that occur in several critical working points. This is the main

reason that makes Computational Fluid Dynamics (CFD) an interesting option for the wind turbine analysis. CFD methods can potentially account for a wider range of details than BEM methods. For instance, and among others: atmospheric boundary layers can be included in the simulation by means of imposing such initial conditions; by nature, 3D effects are considered in the simulations; no experimental or previous results are needed to feed the computations.

The downside of CFD is that it is highly cost- and time-demanding. The accuracy of the cutting-edge methods for wind energy is currently being tested and validated in the scientific community. This chapter is the heart of the thesis and some contributions are expected to be done. Clearly, in this field, CFD will play an essential role in the years to come.

The structure of this chapter is as follows. First section provides an initial discussion or introduction about CFD and its specific state of the art, together with the explanation of the software employed (the in-house TermoFluids code) and the associated know-how to unstructured 3D meshing for these cases. The next three sections, 4.2, 4.3 and 4.4 describe, respectively, the three different approaches to perform CFD simulations over rotating wind turbines: Non-Inertial Reference Frame, Immersed Boundary Method and Sliding Meshes. Right before the conclusion, the last content-section deals with the application of the Adaptive Mesh Refinement methods to the Parked case of the MEXICO experiment.

4.1 Initial discussion

References [1] and [2], from 2003 and 2006 respectively, represent a good beginning to get into the details of wind turbine aerodynamics and aeroelasticity. The history of wind turbines is provided there, starting from the aerodynamics theory from Glauert, and also covering issues such as the wake generation, parametrization, evolution and interaction with other wakes. Moreover, vortex and BEM methods are sufficiently explained to understand the fundamentals of aerodynamics with the required specificity for rotating machines¹.

A couple of reviews are also interesting in order to depict the current state of aerodynamics of wind turbines. The first one, [4], deals with the micro-siting and wind farm layout optimization considering wind modelling and atmospheric boundary layer effects. Fatigue is also commented in the study as well as the Individual Pitch Control of blades which consists in setting different values of pitch for each blade in order to

¹some issues about helicopters are also commented there

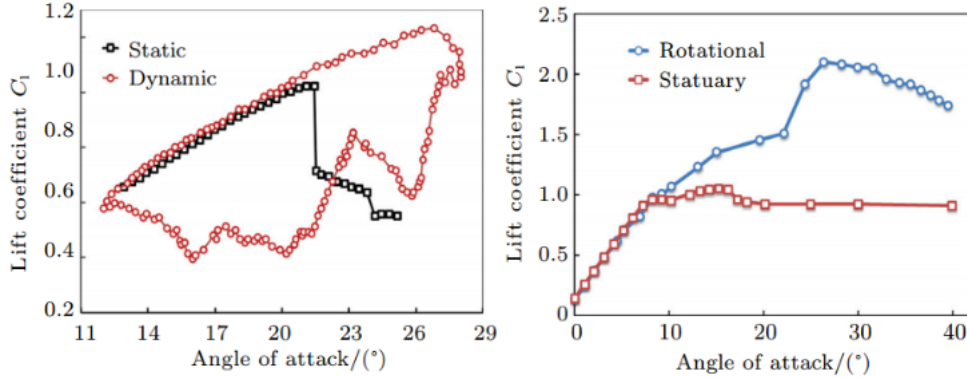


Figure 4.2: Dynamic Stall (left) and Rotational Augmentation (right). The images are extracted from [3]

increase the energy production or reduce the maximum loads. Additionally, issues like tower design and resonant frequencies are also explored. The second review, [3], unlike the first one, is centered purely on aerodynamical issues. At the beginning, it describes the BEM equations and a series of existing simulation software for different degrees of detail. Several interesting issues are covered in the paper, but only two of them will be mentioned here, check Figure 4.2 for graphic support.

1. **Rotational effects.** The paper covers the explanation of the rotational effects observed when comparing the airfoils on the blade and on static conditions. Stall delays as huge as 10° and C_l growths up to 100% are presented in the review. The doubling effect of the maximum lift coefficient is believed to be caused by the reattachment of the boundary layer due to the radial pumping effect of the Coriolis force. Additional insight in rotational augmentation can be pursued in the airfoil studies of the following references: [5], [6], [7], [8], [9] and [10].
2. **Dynamic stall.** The phenomena of the dynamic stall occurs when gradually increasing the Angle of Attack. After exceeding a critical AoA value representing the stall, posterior AoA reductions lead to an hysteresis instead of undoing the track followed in the increasing process (see left side of Figure 4.2). This is believed to be due to the difficulty to reattach the flow once it is detached after a critical point.

Both effects – dynamic stall and rotational augmentation – are of extreme inter-

est for wind turbine aerodynamics because they are constantly observed and their mechanisms remain unclear.

The aerodynamic interest in airfoils is common in the wind energy sector, see [11]. New airfoils are constantly investigated for the sake of obtaining specific conditions and characteristics in front of certain needs. For instance, a possible scenario is: a new airfoil wants to be designed for the root sections of a 100 m length blade to be installed in dusty areas. In such case, a low sensitivity to dirt should be one of the conditions. An additionally desirable feature is a smooth C_l decay in the stall region. Those might be enough reasons as to raise the interest on the search of a new airfoil.

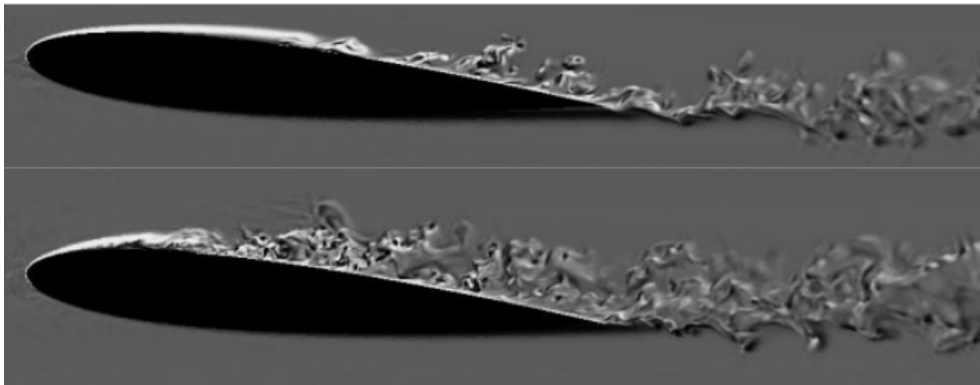


Figure 4.3: NACA 0012. Reynolds 50,000 at AoA 5° (bottom) and 8° (top), from [12]

The airfoil characterization usually aims at obtaining the pressure distribution in a concrete Reynolds number and Angle of Attack. With the pressure distribution, lift and drag coefficients – among other interesting magnitudes – can be computed. They are the main parameters of interest to wind turbine design purposes. Airfoil studies can be done from the experimental or numerical approaches, both having advantages and inconvenients. Experiments are expensive and need to be conducted carefully so that their results constitute a valid benchmark case. On the other hand, numerical studies require a validated software and hours of computing time but their solutions are far more accurate and enable much more detailed studies, see [12], [13] and Figure 4.3.

Figure 3.6, back in Chapter 3 (page 72), shows the airfoil drag and lift coefficients in function of the AoA. Clearly, what makes those airfoil simulations tricky is the stall mechanism, because it implies to correctly resolve the relevant scales of flow (see Figure 4.3). Usually, the most important scales are relatively large and that is good news.

Notwithstanding, the bad news is the fact that the small scales are actually strong mediators and catalyzers of bigger scales. Thus, being able to correctly resolve or model the small-medium scales is the heart of the problem that would lead to good results when simulating airfoils in the stall region. As predicted by Kolmogorov theory, the size of the scales depends on the Reynolds number according to the relationship in Expression 4.1, where η is the size of the smallest scale, L is the characteristic length (typically the size of the domain) and Re is the Reynolds number. As the Reynolds number increases, the size of the smallest scales shrinks, demanding finner meshes and shorter time scales to perform the simulation.

$$\frac{\eta}{L} \sim Re^{-\frac{3}{4}} \quad (4.1)$$

In this respect, Direct Numerical Simulations (DNS) manage to solve the smallest scales in the flow meaning that no modelling is necessary. This approach is the ideal one, as applied in [13] where a NACA 0012 at Reynolds 50,000 was solved (see Figure 4.3). However, this is not possible when the Reynolds number reaches the range of millions. For those cases, engineers and scientists have no choice but to recourse to models in order to predict the aerodynamic behavior of airfoils. Note that the corresponding Reynolds number to real wind energy cases ranges between 500,000 to 10,000,000.

In front of such a big Reynolds numbers, Large Eddy Simulation (LES) and Reynolds-Average Navier-Stokes (RANS) models are employed. Again, they have strengths and weaknesses. From the computational point of view, RANS are considerably cheaper but the solution is averaged on time, so the natural unsteadiness of wind turbines operation cannot be captured. Additionally, RANS are strongly dependent on tuning (as shown in [14]) unlike LES models that rely on fluid-flow inherent physics to modelize the effect of the smallest scales. LES can account for time varying solutions and represents a good alternative to DNS and RANS. However, since the LES computing time is still quite demanding, special care must be taken when selecting the cases to be run.

LES simulations of airfoils at relatively high-Reynolds are available in the literature. For instance, in [15], a DU (Delft University) airfoil at Reynolds 5 Million was simulated considering several Angles of Attack, including points in the stall region. Recently, LES Wall Models (such as the one exposed in [16]) can achieve similar results employing significantly less mesh. They manage to do so because of the superposition of a Prismatic Boundary Layer (PBL) stretching from the wall to the position of the first node adjacent to the wall. This strategy allows the LES mesh to be much coarser near the walls because of the *delegation* of this job to the PBL mesh, which is responsible to account for the stress tensor on the surface of the body.

In view of the evidence presented about simulations for airfoils, it might seem a far-fetched idea to even consider a whole wind turbine simulation. However, back in 2012, the CTTC directive board decided to get into the field of three-dimensional simulations of wind turbines as a matter of strategic positioning for future scenarios. Certainly, it can be considered a risky choice but it is also true that sooner or later, perhaps in 2040 or earlier, a full wind turbine could be simulated for the first time with LES.

In this context, two experiments are of extreme importance for the validation of CFD codes in wind energy aerodynamics, the NREL Phase VI [17] and the MEXICO [18] projects, both designed with the intention of providing benchmark data for posterior code validation. For instance, [19] is a marvellous paper because it compares the Unsteady-RANS (URANS) and Detached Eddy Simulation (DES, a variation of LES) in multiple scenarios of the NREL experiment: 5, 10, 15 and 25 m/s at 3° pitch, among many other conditions (such as wind constant at 15 m/s and varying pitch angle from -15° to 40° every 5°). The computational domain was discretized in 57 million control volumes employing the chimera technique (mesh overlapping, to be commented further in this section).

With regards to the MEXICO rotor there is a prominent number of studies, perhaps because it is more recent and the data provided are more complete. In the remaining part of this paragraph a few of them will be mentioned. Paper [20] presents three different approaches with the focus on the wake generation and vortex tracking: a periodic simulation using only 1 blade with both RANS and URANS and a third approach for the whole wind turbine with URANS. Another study, [21], focuses more on quantifying the possible influence of the MEXICO's wind tunnel over the measures taken in the turbine. Two additional papers deal with the analysis of the velocity deficit and the downwind field modification by means of comparing the PIV measuring of the experiment with the simulations: [22] employs Non-Inertial Reference Frames and [23] covers yawed cases.

Anyhow, whatever the simulation approach, two elements are essential: a powerful code to perform those high level computations and also a good meshing technique. Of course, both code and mesh strongly depend on the case, on the turbulence model and also on the numerical approach selected.

4.1.1 CFD software: TermoFluids

The Heat & Mass Transfer Technological Center (CTTC) originally started to work on heat transfer in the three forms: conduction, convection and radiation. Convection

requires mastering fluid dynamics and most of industrial and research problems imply sooner or later the contributions of numerical methods. With time, the application of knowledge to these areas and the collaboration with industry and academia, lead the CTTC researchers to build a powerful software and generate a spin-off specialized in consulting for cutting-edge equipment and systems in the field of Computational Fluid Dynamics and Heat Transfer (CFD&HT). The code was named TermoFluids and the company Termo Fluids S. L.².

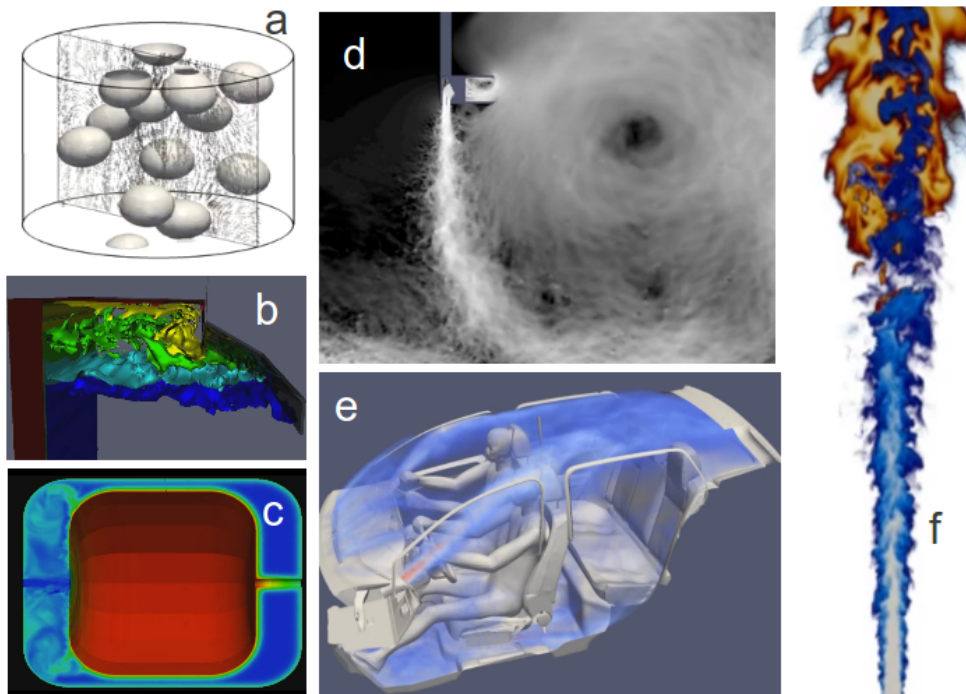


Figure 4.4: Several examples of simulations run with TermoFluids: a) two-phase flow [24]; b) natural convection in high-temperature solar receivers [25]; c) compressors [26]; d) air curtains [14]; e) HVAC for automotive industry; f) combustion [27].

TermoFluids is a 3D parallel unstructured code specialized for CFD&HT aiming at High-Performance Computing (HPC). The code is written in C++ using Object Oriented Programming (OOP) in order to be easy to grow and understand by the team of

²more information in <http://www.termofluids.com/>

developers³. The philosophy behind the code is to build it as general as to be applied in multiple cases and research fields: two-phase flow, combustion, radiation, fluid-structure interaction, concentrated solar power, heating, ventilation and air conditioning (HVAC), air curtains, refrigeration, compressors, etc. See Figure 4.4.

Navier-Stokes equations are solved by means of the Fractional Step Method [28] using symmetry-preserving numerical schemes. Several LES models have been implemented and validated [29]: QR, Variational Multi-Scale [30] or WALE [31]. Such models were successfully employed in the simulations presented in Figure 4.4.

One of the difficulties of the aerodynamics of wind turbines is its spatial and temporal multi-scale nature. Time-wise scales range from the frequency of the vortex shedding (small scale) of the tip airfoil section to the turbine spinning time (big scale). Spatial scales range from the boundary layer of the airfoil sections (tiny scale) to the size of the whole turbine (huge scale). This is clearly making the simulation more complex and costly. In this case, TermoFluids has proven to be accurate in similar natured problems e.g. in parabolic trough solar collectors in [32].

Finally, it is also worth to mention a couple of works done with TermoFluids in the field of wind energy. Firstly, the work carried out in the thermal optimization of the ALSTOM ECO 100 nacelle, [33], where different approaches were used to maintain the inner temperature below the target. And secondly, the relatively ample know-how that CTTC gathers in the field of aerodynamics of bluff bodies such as the flow around a sphere [34] or the previously mentioned DNS of the NACA 0012 airfoil [13].

4.1.2 Meshing

There are two clear pillars in a simulation: the code and the mesh. The validation of a code is a laborious task and it is the blueprint to be used in one way or another in function of the specificity of the case at hand. Conversely, the mesh is a very specific element that is absolutely particular of each case and a certain expertise is required in its construction and configuration. Even in the situation that the code and the computing infrastructure are exceptional, a mediocre mesh can ruin the results of the simulation.

The mesh constitutes the computational domain. It represents the geometrical model that will be used for computing. The real 3D physical space is represented by a finite number of volumes (elements, nodes) whose centroid will represent the properties of the cell for every parameter or variable of the simulation (velocity, pressure, etc.).

³now being around 20 engineers and scientists only in development tasks

The most common volume shapes are the tetrahedron – for unstructured meshes – and the hexahedron – for structured ones. Thus, a good mesh must aim at the same time at different properties/aspects: represent as precisely as possible the real geometry; maintain the quality (or aspect ratio) of all elements above (roughly) 0.3; ensure a smooth transition when growing control volumes; etc. Nowadays, it is still difficult to find a general, robust and automatic meshing tool achieving free-error high-quality meshes without hours of human intervention. Without doubt, a good mesh construction is strongly dependent on the know-how of the designer. Consequently, despite the technical aspects, it can be considered somehow as an art. Definitely, in the case of wind turbines this is also the case.

For the specific case of wind turbines, as it also happens with pumps, fans, propellers, and, in general, rotating machines, extra difficulties arise because there are rotating and static parts that force the simulation to come up with a method to adjust the mesh to the changing geometry over time. Of course, when simulating static domains, the mesh only needs to be created once; but for dynamics domains, a solution must be found. In the current state of the art, there are several options for this: i) simplifying the case and simulate just the rotating parts by means of using a *Non-Inertial Reference Frame*; ii) employing overlapping meshes (chimera), some static and some dynamic, proceeding with volumetric interpolations; iii) using what is called *Sliding Meshes*, where one mesh accounts for the background static domain and another for the rotating one in a way that both meshes slide one over the other one, meaning that no volume is overlapped because only a cylindrical shape does. There is an additional strategy named *Immersed Boundary Method* (IBM) that it is not so common to be applied to wind turbines⁴. There are some studies such [35] and [36] for airfoil flaps, but not for the full turbine.

In the context of this thesis, three different approaches were tested for a rotating wind turbine case: Coriolis (Non-Inertial Reference Frame), IBM (Immersed Boundary Method) and Sliding Meshes. Those techniques were tested in four different cases: Offwindtech project [37], NREL [17], MEXICO [38] and Enair's cases (see Appendix A). Two ways were employed to provide the meshes for the computations: the ANSYS ICEM mesher [39] and the AMR (Adaptive Mesh Refinement, [40]). In this way, the whole sequence of considered cases is presented in Table 4.1.

As it was explained in the Foreword section, there are some external reasons that forced us to follow the order presented in Table 4.1. Summing up, we had to go into the Offwindtech project for deadline reasons and after that, a validation-focused approach could come in to move to NREL and MEXICO cases. Notwithstanding, a couple of years

⁴to the best of the author's knowledge there is no precedent on this

#	Method	Case	HAWT	Mesh type	Software
1	Coriolis	Offwindtech	Rotor	Unstructured	ICEM
2	IBM	Offwindtech	Full	Unstructured	ICEM
3	IBM	NREL	Full	Unstructured	ICEM
4	IBM	NREL	Rotor	Structured	ICEM
5	IBM	NREL	Rotor	Structured	AMR
6	IBM	MEXICO	Full	Structured	ICEM
7	Sliding Meshes	NREL	Rotor	Unstructured	ICEM
8	Sliding Meshes	Enair	Full	Unstructured	ICEM
9	IBM	MEXICO Parked	Full	Unstructured	ICEM
10	IBM	MEXICO Parked	Full	Structured	AMR
11	Bluff body	MEXICO Parked	Full	Unstructured	AMR

Table 4.1: Different approaches tested in the 3D CFD applied to wind turbines.

later Enair’s project shifted the priorities again.

Clearly, the Immersed Boundary Method is the one where more efforts were done. Bear in mind the geometry handlings exposed in Chapter 2 to manage the STL components and provide a framework for airfoil reconstruction in order to obtain pressure distributions. Additionally, the meshes employed in this method (that will be explained in Section 4.3), posed many problems due to the huge size of them. The IBM method brought us, the code, the algorithm and meshing capabilities to the limit.

For instance, the computer employed for the meshing with ICEM was the most powerful machine at CTTC⁵: 32 GB (later on 64 GB) of RAM and 12 CPUs were not enough to create the required 3D CFD meshes: see in Figure 4.5 how the RAM was clearly limiting. In the end, the computer had the RAM increased to 64 GB and only like that we managed to compute the meshes we were aiming to.

Unstructured meshes, even when controlling for mesh size, were far more demanding (in terms of RAM) than structured meshes, probably due to the fact that ICEM data model for structured meshes allows to use a more compact working memory and also to build a lighter file than unstructured meshes. This fact was one of the reasons that lead us to try structured meshes, see case 4 in Table 4.1.

Details about meshes will be provided in each of the corresponding sections, since all

⁵besides the cluster, of course, where roughly 2,000 CPU are available with large RAM availability

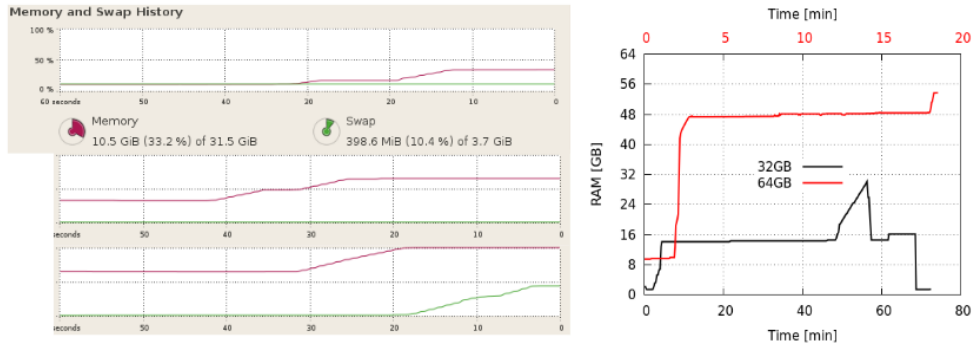


Figure 4.5: The 32 GB RAM of the 12-CPU machine employed in such meshes usually collapse during massive computing (left). The 64 GB resulted enough but close to the limit (right).

of them have particularities. However, some comments about Adaptive Mesh Refinement might be useful at this point. This approach is based on the recursive splitting of hexahedral volumes into eight smaller hexahedrons such as it is done in [40], based on the theory proposed in [41]. AMR is useful because it can automatically refine during running time where required. In the case of wind energy, it would be very costly, but the underlying idea of AMR can be used to both IBM of the pure rotor and also Parked conditions (developed later). For the IBM with structured mesh, a very coarse mesh can be initially generated with ICEM, that cannot be used in parallel in an efficient way, and then employing AMR to uniformly refine the initial mesh⁶ until obtaining the desired level of detail.

A final comment on this meshing introduction must be towards references [42] and [43]. They claim to be able to solve the flow around turbines employing meshes with roughly 1 million control volumes, achieving y^+ values in the order of 1 and having good results on pressure coefficient across the MEXICO blades employing LES methods. To the best of the author's knowledge (based on his own experience and the wide literature and reference delivered), a mesh with a million control volumes cannot be enough. It is not enough even to accurately solve the flow around an airfoil at Reynolds numbers (about one million) experienced in any of the turbine sections: in [15], 19 million cells were needed to solve just a single airfoil. How can it be possible to employ just 1 million for a whole turbine and obtain good results?

⁶employing parallel code and clusters

4.2 Coriolis

In science and engineering, when analyzing moving particles, objects or systems it is usual to define a Frame of Reference to which the position, velocity and acceleration are *referred* to. The mathematical solution of a situation or the description of a moving system does not varies with the frame of reference, they all represent the same *reality* but viewed from different angles. In most situations, a static Frame of Reference suffices. In such systems, the vector base remains constant with time and therefore they receive the name of Inertial Reference Frame. For certain type of situations, it is more convenient to use a specific frame of reference because it makes the analysis and its interpretation far easier. Conversely, a Non-Inertial Reference Frames is *dynamic*, i.e. either moving or rotating, in a way that implies acceleration – a change of the velocity vector, either by linear acceleration or just rotation. When the frame of reference is accelerating or rotating, the inclusion of Fictitious Forces becomes automatic as it will be derived in this section.

In the case of wind turbine simulation, there are rotating parts that are susceptible to be analyzed better from the point of view of a rotating (Non-Inertial) Frame of Reference. It has been repeated over and over again that the crucial components in wind energy aerodynamics are the blades, therefore, it makes sense to analyze the rotor alone without the tower and the nacelle. In this case, the *world* would be seen from the perspective of the rotor. If that is the case, a Non-Inertial Reference Frame would imply the use of a static mesh (non-rotating) but the fluid would appear to be rotating instead of flowing in a *straight* line stream. There is a clear benefit on approaching the problem in this way: the staticity of the mesh; but, this comes at a price: fictitious forces come into the game. Necessarily, the new forces need to be accounted somehow in the Navier-Stokes equations and the best way to do so is by adding them as a source term in the momentum equations. In the following paragraphs we will show how to include the corresponding additional terms.

Figure 4.6 represents the relationship between two different reference frames: an static one (S, inercial) and another one that can be either moving or rotating or accelerating (S', non-inertial). The mathematical representation of the position of a particle (or the centroid of a finite volume) is shown in Equation 4.2. The differentiation of the position leads to the velocity, Equation 4.3, and a second differentiation leads to the acceleration, Equation 4.4.

$$\mathbf{r}_{p/s} = \mathbf{r}_{p/s'} + \mathbf{r}_{s'/s} \quad (4.2)$$

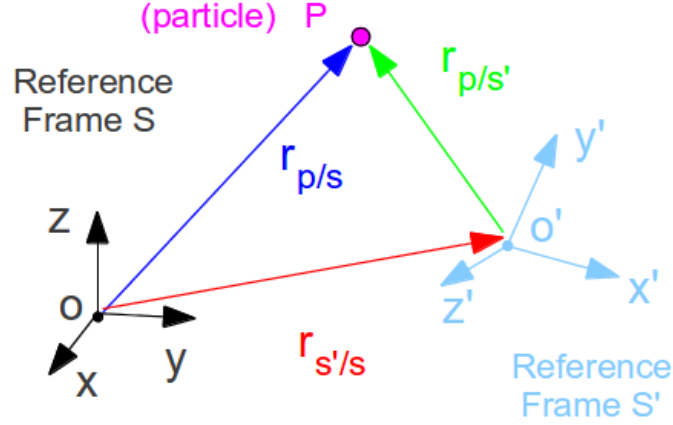


Figure 4.6: Scheme of the distances and nomenclature between an inertial and non-inertial reference frames.

$$\mathbf{v}_{p/s} = \mathbf{v}_{p/s'} + \boldsymbol{\omega}_{s'/s} \times \mathbf{r}_{p/s'} + \mathbf{v}_{s'/s} \quad (4.3)$$

$$\mathbf{a}_{p/s} = \mathbf{a}_{p/s'} + 2\boldsymbol{\omega}_{s'/s} \times \mathbf{v}_{p/s'} + \boldsymbol{\alpha}_{s'/s} \times \mathbf{r}_{p/s'} + \boldsymbol{\omega}_{s'/s} \times (\boldsymbol{\omega}_{s'/s} \times \mathbf{r}_{p/s'}) + \mathbf{a}_{s'/s} \quad (4.4)$$

The nomenclature in the previous equations is the next: \mathbf{r} is the vector position, \mathbf{v} is the vector velocity and \mathbf{a} is the vector acceleration; $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ are the angular velocity and acceleration of the Non-Inertial Reference Frame, and the subindexes are referred to p as the particle, point or centroid of the volume, s is the Inertial while s' is the Non-Inertial Reference Frame. Thus, for instance, $\mathbf{v}_{p/s'}$ is the velocity of the point p with respect the reference frame s' .

Equation 4.4 represents the acceleration of a particle with respect a reference frame, S' , that might be rotating and accelerating with respect to a static reference frame, S , see Figure 4.6. There are five terms at the right hand side of the equal symbol, from the left to right they stand for: the relative acceleration of the particle to S' ; the Coriolis acceleration; the *drag* component, due to the angular acceleration of the S' reference frame; the centrifugal force; and the relative acceleration from reference frames S and S' .

In the present case, the Non-Inertial Reference Frame proposed is a rotating frame where the observer is located on the rotor. Therefore, some terms from Equation 4.4 are

identically zero such as the drag component and the relative acceleration between the reference frames S and S' , resulting in a slightly simpler equation:

$$\mathbf{a}_{p/s} = \mathbf{a}_{p/s'} + 2\boldsymbol{\omega}_{s'/s} \times \mathbf{v}_{p/s'} + \boldsymbol{\omega}_{s'/s} \times (\boldsymbol{\omega}_{s'/s} \times \mathbf{v}_{p/s'}) \quad (4.5)$$

Navier-Stokes equations for the wind turbine case (4.6 and 4.7) are the result of the conservation of mass and momentum, respectively.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (4.6)$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} \quad (4.7)$$

The momentum vectorial equation (4.7) is in fact the second Newton's law (Equation 4.8) introducing the Stoke's constitutive equation for Newtonian fluids. The right hand side of Equation 4.7 represents the *forces* and the left hand side the *acceleration* terms. The combination of Equation 4.5 into the second Newton's law (Equation 4.8) leads to Equation 4.9.

$$\mathbf{F} = m \mathbf{a}_{p/s} \quad (4.8)$$

$$\mathbf{F} = m (\mathbf{a}_{p/s'} + 2\boldsymbol{\omega}_{s'/s} \times \mathbf{v}_{p/s'} + \boldsymbol{\omega}_{s'/s} \times (\boldsymbol{\omega}_{s'/s} \times \mathbf{v}_{p/s'})) \quad (4.9)$$

Then, a final rearrangement and relabeling is required in order to include the centrifugal and Coriolis terms into the *Ficticial Forces* discussed before (and packed in Equation 4.10), which yields to the final expression: Equation 4.11.

$$\mathbf{F}_{ficticial} = m (2\boldsymbol{\omega}_{s'/s} \times \mathbf{v}_{p/s'} + \boldsymbol{\omega}_{s'/s} \times (\boldsymbol{\omega}_{s'/s} \times \mathbf{v}_{p/s'})) \quad (4.10)$$

$$\mathbf{F} - \mathbf{F}_{ficticial} = m \mathbf{a}_{p/s'} \quad (4.11)$$

Thus, the ficticial forces expressed in Equation 4.11 need to be included into the Navier-Stokes equations as a source term which is determined by Equation 4.12.

$$F_{ficticial} = \rho \cdot V_p \begin{pmatrix} 2(w\Omega_y - v\Omega_z) - x(\Omega_y^2 + \Omega_z^2) + \Omega_x(y\Omega_y + z\Omega_z) \\ 2(u\Omega_z - w\Omega_x) - y(\Omega_z^2 + \Omega_x^2) + \Omega_y(z\Omega_z + x\Omega_x) \\ 2(v\Omega_x - u\Omega_y) - z(\Omega_x^2 + \Omega_y^2) + \Omega_z(x\Omega_x + y\Omega_y) \end{pmatrix} \quad (4.12)$$

where Ω_i is the component of the angular velocity in the direction i ; x , y and z are the spatial coordinates; u , v and w are the velocity components; ρ is the density and V_p is the volume of the cell.

Besides the inclusion of the abovementioned source term into the Navier-Stokes equations, there was another critical issue to cover: the boundary conditions. It is well known that the outlet boundary conditions are quite problematic because they lead to numerical instabilities. Sometimes, they require mass corrections such as the ones presented in [44]. In our case, a new boundary condition had to be developed: at the beginning, when Neumann or mass corrective boundary conditions were employed, the cases diverged without exception. It was not until we implemented a boundary condition accounting for the rotation that the cases stabilized.

At this point, when the formulation of Navier-Stokes equations for Non-Inertial Reference Frames have been presented, the next step is to proceed with the verification.

4.2.1 Test 1: rotating empty cylinder

The first proposed situation for verification is the comparison of two different ways of approaching the same case. The case represents an infinitely large cylinder full of a static fluid when suddenly, the cylinder wall starts rotating with respect to its own axis. This situation is solved in two different ways:

- A. Static frame of reference.** In this case, the whole domain is static and the boundary conditions are the ones in charge to simulate the rotation of the cylinder. The fluid, initially static, starts acquiring momentum from the rotating walls and in the end of the simulation, the whole fluid rotates with the cylinder.
- B. Rotating frame of reference.** In this case, the situation is seen from the point of view of the rotating cylinder walls. The boundary conditions are static, but the whole fluid inside the cylinder is rotating from the beginning – bear in mind that from an external and static point of view, the fluid is actually static, but this is not the case of the rotating reference frame. Now, the terms corresponding to Coriolis and centrifugal fictitious forces are active. The observed behaviour of such simulations is that the fluid finally appears to be at rest from the point of view of the Non-Inertial Frame of Reference, which actually means that it is rotating at the same pace that the cylinder is rotating.

This easy verification test comparing the two different approaches of the same case (see Table 4.2) is just a first sign that the method is working. Additional tests were done

Case	Bondary Condition	Ref. Frame	Ini. fluid	End fluid
A	Rotating	Inertial	Static	Rotating
B	Static	Non-Inertial	Rotating	Static

Table 4.2: Test 1: comparison of the two employed approaches, Inertial and Non-Inertial Reference Frames, for the same case.

such as the application of the Method of Manufactured Solution (MMS) to check that the code implementation was correct.

Perhaps it is interesting to mention here that the pressure field across the cylinder, when employing ficticial forces, is found to follow a parabolic curve which is the consequence of the pumping effect of ficticial forces.

4.2.2 Test 2: crossed flow over simple geometry

The second test, see Figure 4.7, aims at a middle step between Test 1, where the fluid movement is not flow-like, and the final real conditions of a wind turbine. In this case, a relatively long cylinder enabled with inlet and outlet boundary conditions is the pipe where a fluid flows inside, encountering a body – a cube – which rotates excentrically with respect to the cylinder axis.

The geometry of the case is the next:

- the cylinder is 200 mm long and its radius is 50 mm
- the side of the cube measures 10 mm and its center of gravity is 15 mm away from the cylinder axis, rotating at 10 rpm
- the fluid flows at 0.1 m/s

This case represents a soft test (Reynolds number in the order of 10^3), designed to check if there is something wrong ongoing with the Non-Inertial platform (ficticial forces and boundary conditions) when the following two aspects coexist: encountering a body in cross-flow configuration and simulating the flow crossing the domain (specially for the outlet boundary conditions). This test was actually the responsible for showing that the *usual* boundary conditions (that we have always employed at CTTC, for instance, in airfoils simulations) were not appropriately working in this context.

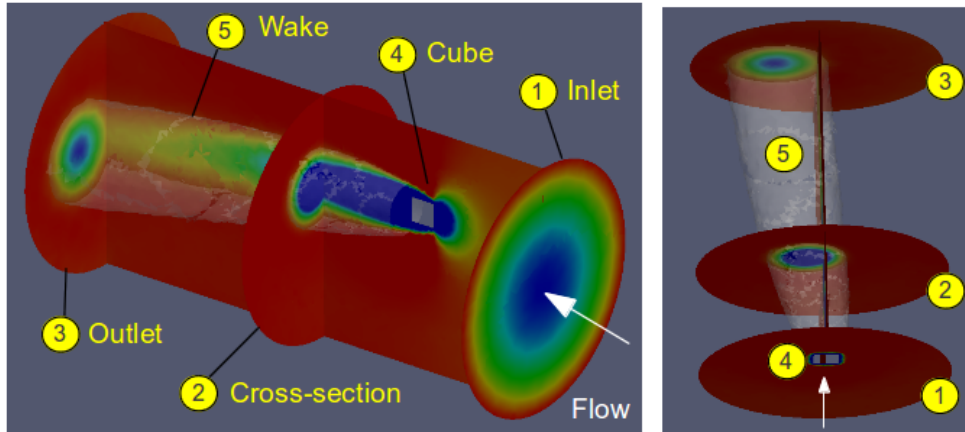


Figure 4.7: Test 2: schematic representation of the flow in a Non-Inertial Reference Frame across an eccentrically rotating cube (with respect the cylinder axis).

This simulation was run for a sufficiently long time as to observe no changes in the pressure and velocity fields i.e. the flow achieved a permanent solution. The wake was deflected in accordance to the flow rotation and outlet boundary conditions functioned well.

4.2.3 Test 3: real WT rotor case

After the successful second test, it is time to aim at a case with the real conditions and characteristics of wind turbines. The natural and reasonable step would have been to test this method in NREL Phase VI or MEXICO cases. However, as introduced in the *Foreword*, at this point in time, we aimed directly at the 5MW rotor originally developed by NREL [45] employed in the KIC InnoEnergy Offwindtech project [37]. A quick glance at the blade characteristics can be taken in Figure 4.8.

For the first time in the CTTC's history, we had to face the meshing of a such complex object in 3D. Up to 2012, we had no experience on this type of curvy and large objects. Therefore, we proceed to study in depth the ICEM User Guide [39] in order to get some insight on how to approach 3D unstructured meshing.

Roughly speaking, there are three big steps to generate a 3D mesh of any rotor: 1) obtain a reliable CAD model for the rotor geometry susceptible to be meshed; 2) generate a superficial mesh of the rotor model; 3) generate the three-dimensional mesh of the

#	Radius	Twist	Chord	Pitch axis aft LE	Thickness		Airfoil
(-)	(m)	(deg)	(m)	(*chord)	(%)	(m)	(-)
1	1.850	11.02	3.5420	0.500	100	3.5420	Cylinder 1
5	4.700	11.02	3.5420	0.500	100	3.5420	Cylinder 1
6	5.700	11.02	3.6034	0.487	96.28	3.4668	Cylinder 1
7	6.700	11.02	3.7731	0.465	88.27	3.3231	Cylinder 2
11	10.700	10.91	4.4877	0.392	55.70	2.4991	Cylinder 2
14	14.184	9.91	4.6114	0.375	40.10	1.8491	DU-00-W-401
17	16.710	8.87	4.6038	0.375	35.00	1.6114	DU-00-W-350
20	21.895	7.30	4.3564	0.375	30.00	1.3070	DU-97-W-300
24	29.996	4.88	3.8901	0.375	25.00	0.9726	DU-91-W2-250
28	38.209	2.54	3.3895	0.375	21.00	0.7118	DU-93-W-210
34	50.505	-0.34	2.6497	0.375	18.00	0.4769	NACA-64618
36	53.900	-0.98	2.4539	0.375	18.00	0.4417	NACA-64618
38	57.200	-1.61	2.2331	0.375	18.00	0.4020	NACA-64618
43	61.900	-2.22	1.2609	0.375	18.00	0.2270	NACA-64618
49	62.960	-2.29	0.2034	0.375	18.00	0.0366	NACA-64618



Figure 4.8: A quick glance at the Offwindtech Blade geometry.

domain.

The first step, the reliable CAD geometry might seem an easy one to attain but our experience tells that it is not so effortless. Pay attention to the fact that large and small dimensions coexist in the same object: the blade length is huge (63 m) while the thickness of the sectioned (cut) trailing edge is tiny (order of millimeters). This fact poses some tolerance problems that need to be sorted out. Additionally, some incompatibilities between IGS or STEP files⁷ produced by external softwares to ICEM were detected: the official Offwindtech blade model had some deficiencies in the trailing edge zones. The blade was not perfectly closed and we were forced to redesign it employing the data presented in Figure 4.8 and the SolidWorks software. As mentioned in Chapter 2, it is a long and laborious process, so, this motivated us to develop the AutoBlade program

⁷typical file formats for CAD models

later on. In the end, we managed to come up with a decent CAD model.

The second step is to generate a superficial mesh departing from the geometry. It turned out that the CAD model, even when being well defined, it was not appropriate for meshing. Probably, the fact that ICEM is not a 100% *deterministic* is not helping at all. As bizarre as it might sound, the same action performed at a certain point in time and executed later on, is not leading to the same results. The only possible explanation for this occurrence is that the RAM state and previous operations affect the course of events. As a matter of prevention, a conservative and step-by-step strategy was adopted, saving operations performed on the geometry to keep track of the process and be able to come back and forth to fight this *random-like* nature of ICEM. In the end, it was possible to find a methodology to achieve an acceptable superficial mesh (i.e. an STL-like mesh composed by an array of triangles, see Chapter 2). The right factor combination was the next: a) once the geometry is imported, run the process *Build Topology* with the local tolerance option active, and check that any curve of the model is associated exactly to two surfaces; b) set a reasonable element size according to the length of all curves defining the CAD model; c) compute the superficial mesh employing the technique named *Patch Dependent*. Those steps were proven to be sufficient for most of the cases. However, unexplainable problems arised in some CAD models and additional heuristics need to be employed. The outcome of the process can be observed in Figure 4.9. Perhaps it is the moment to highlight the fact that, one of the benefits of the AutoBlade code is that it guarantees 100% success when generating the superficial mesh departing from the CAD model that it delivers.

The third and final step to generate a mesh for the 3D case based on the Coriolis approach (Non-Inertial Reference Frame) consists of employing the mesh obtained in the last step (the second one) to grow from there the volumetrical one. This step is usually divided in three sub-steps: the *Octree* Meshing, the *Delaunay* Meshing and the *Smooth mesh globally*.

The Octree meshing, being run with the option "Existing mesh parts", generates a rough mesh that covers the whole domain while respecting the selected previous meshes already built (superficial, from 2nd step). It fulfills the need to mesh the domain, but the resulting mesh does not have the right quality to be run in TermoFluids. The Octree mesh is very block-like, see the left hand side image in Figure 4.10, where clear structured patterns are visible across the supposedly unstructured mesh. As a consequence of that, the Delaunay step becomes indispensable after the Octree meshing. The purpose of Delaunay is to remove those structured "patterns" across the mesh and truly attain a relatively smooth pattern in the control volumes connection. Sometimes,

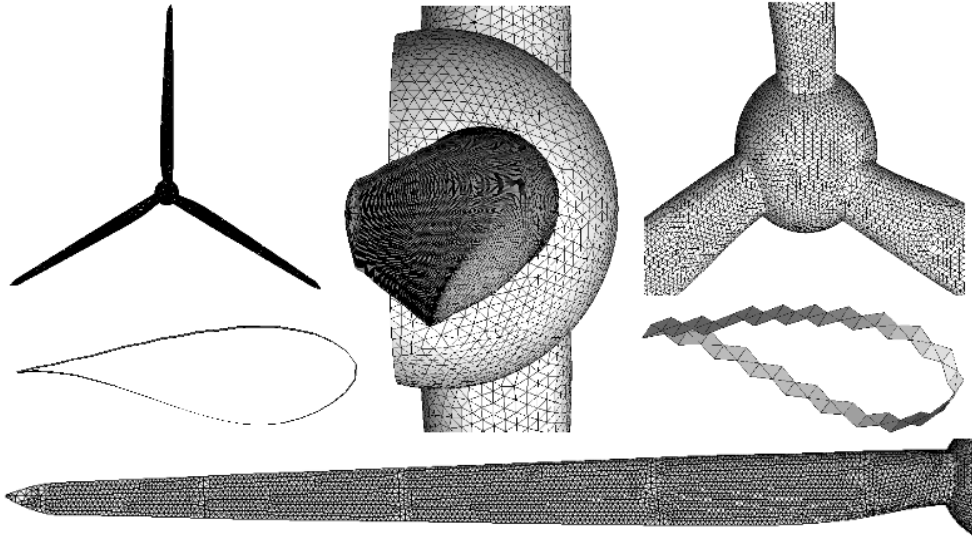


Figure 4.9: Superficial mesh of the Offwindtech rotor observed from different angles and perspectives.

Delaunay already provides a good-enough mesh in terms of the minimum quality of the elements and growing patterns. However, it has been detected that the extra smoothing is generally improving the results – see the right hand side image in Figure 4.10. It turns out that minimum quality values of 0.3 are easily attainable, sometimes it is possible to reach values of 0.5. This is crucial to ensure the *convergability* of the posterior simulation. In most cases, when simulations are run with meshes having minimum quality factors of 0.2 or below, they diverge⁸.

After creating the code particularities for the Non-Inertial Reference Frame simulations and a good mesh for the case, finally, the simulation can be performed. The most relevant case to be studied by the Offwindtech consortium was the one at the nominal power 5 MW that is expected to be achieved when the machine rotates at 11.7 rpm at a wind speed of 11.7 m/s. The first mesh had 1.9 million control volumes and was cylinder-shaped with a radius 5 times larger than the blade radius. A WALE LES model was selected to account for turbulence in this pure rotor simulation (no tower or nacelle) of the Offwindtech turbine (see case 1 from Table 4.1).

⁸unless the control volumes with such low qualities are located in non-critical regions of the fluid flow

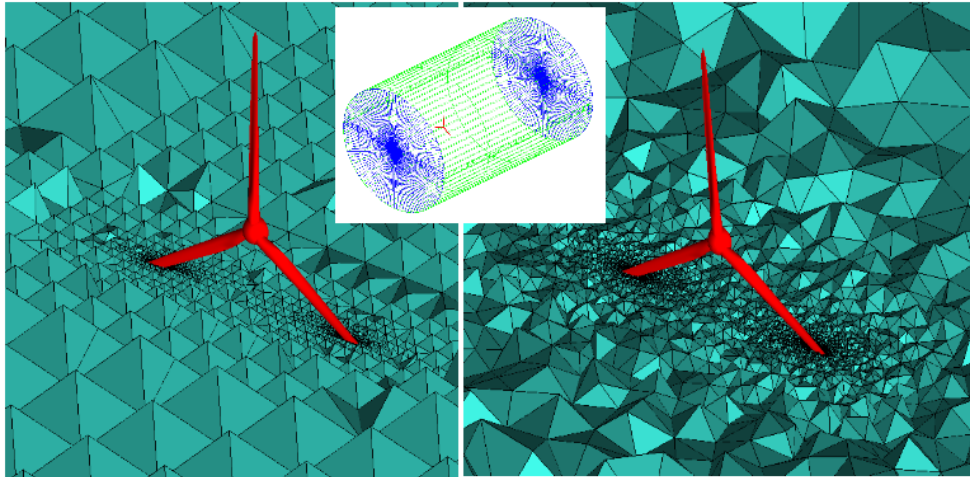


Figure 4.10: Volumetric mesh of the Coriolis-based simulation in the Offwindtech rotor. Left: octree step. Right: delaunay step. Middle: far view of the computational domain.

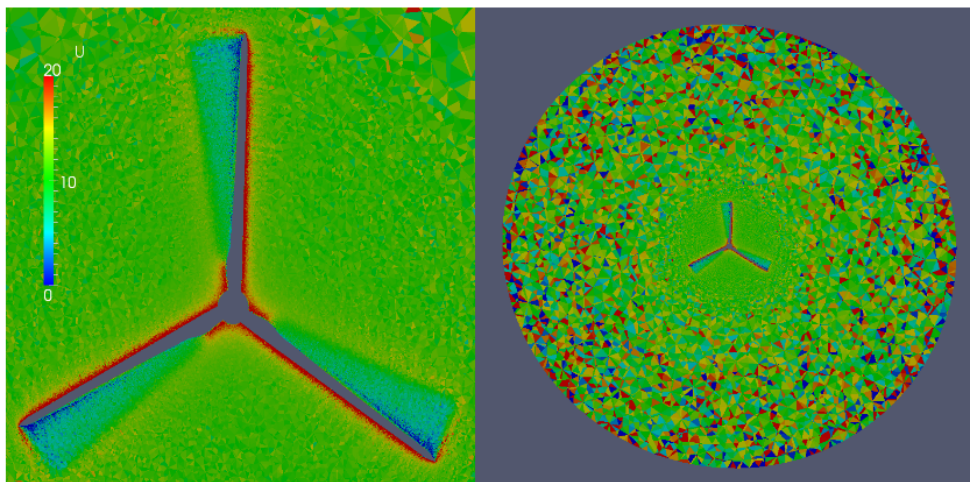


Figure 4.11: CFD results of the Offwindtech rotor employing Coriolis method after roughly 20° simulation. Left: zoom at the rotor zone, reasonable initial flow conditions. Right: far view from the whole domain, unacceptable flow conditions due to scattered random velocity gradients.

Figure 4.11 shows the instantaneous velocity around the rotor at a certain point of beginning of the simulation, when roughly 20° were covered. In the left hand side image, a reasonable initial velocity pattern is observed around and next to the rotor. However, it is just an optical and transient effect observed in this scale (zoom) at the beginning of the simulation. Now, pay attention to the image at the right hand side of Figure 4.11: strange patterns are clearly visible, specially in the zones far away from the rotation center. In view of these problems, we checked the code, rechecked the mesh, run the simulation with other models and virtually reviewed every possible human mistake that might be ruining the simulation. In our best attempts to solve the situation, we found no hint of what was the reason for these spurious velocity patterns. The best explanation we could find is provided next. Notice that the linear velocity of those nodes at the far regions, was as high as:

$$v = \omega \cdot r = \frac{11.7 \cdot \pi}{30} \cdot (5 \cdot 63) \approx 386m/s \quad (4.13)$$

The factor of high velocities alone could be enough as to introduce instabilities in TermoFluids' incompressible solver. The high velocities are the direct consequence of the large domain radius. Five times the blade radius might sound perhaps too much. However, relatively large domains are required in order to avoid the blockage effects of the boundaries over the turbine. But, even though the cylinder (domain) radius was reduced to 2 or 3 times the rotor radius, the same behavior was observed. Therefore, we suspect that the true reason to explain the divergences is the combination of several factors. A possible explanation is the coexistence of the high velocities themselves together with the location where they arise: the outer regions of the domain. Logically, those regions are not of interest for the wind turbine analysis, so relatively large control volumes are located there in order to concentrate the mesh in the relevant zone of study: near the turbine. This problem has a difficult solution because even in the case that we wanted to refine the far regions, it would quasi-impossible because the number of control volumes would grow exponentially.

All in all, to sum up this section, we conclude that a relatively big effort has been done to incorporate Non-Inertial Reference Frame into TermoFluids. The formulation was similar to the one successfully employed in [46], but the simulations invariably end up diverging because of – we suspect – the high velocities and their location where no mesh concentration is desired nor achieved. Further investigation, supported by an in-depth mathematical review of the formulation, would be required in this topic to make clear the reason of such divergence cases. According to [22], the simulation of a wind turbine rotor (MEXICO case) employing this methodology is possible, at least,

employing RANS. The authors of the cited paper harvested good results, so it would be expected to achieve the same milestone possible by means of the LES approach.

Anyway, this research strategy left us with a comfortable expertise in the meshing process and pointed to a set of problems that will be sorted out with the following approach to be studied: Immersed Boundary Method.

4.3 Immersed Boundary Method

Numerical simulations in the field of Computational Fluid Dynamics (CFD) usually conceive two type of boundaries: 1) those around solid elements and 2) those without a clear, so to speak, material delimitation, i.e. the boundary allows the fluids to flow across its surface. For instance, in a bluff-body⁹ simulation, there are boundaries in both far regions and around the solid to be studied. This conception clearly separates fluids from solids and does not consider the immersion of solids in the meshed areas, since they are meant to be occupied by fluids. However, many industrial situations have moving elements in the regions where fluid dynamics can add some value. For such cases, the traditional convention cannot be employed because the mesh is incapable to accomodate the geometry. Therefore, alternatives must be explored.

In this respect, Immersed Boundary Methods (IBM) is a family of methods that share the ability to approach CFD simulations with a *virtual* boundary of a solid instead of being *shaped* by the mesh boundaries themselves. Thus, the body is literally *immersed* on the mesh and that is the fact that gives the name to IBM.

IBM were originally introduced by Peskins, [47], back in 1972 in order to study heart valves. Since then, they have experienced a huge expansion and now they are recently being applied to many fields and situations as shown in [48]. A very recent review of IBM method is available in [49]. The application of IBM to complex three-dimensional flows has been proven to be feasible in studies like [50]. References [51] and [52] prove that these methods are recently beeing applied to the field of wind energy: the first one covers floating platforms, and the second one uses the Actuator Line method to study the wake interaction. However, to the best of the author's knowledge, there are no precedents of using IBM methods in the simulation of a wind turbine.

Immersed Boundary Methods can theoretically be applied to the case of a working wind turbine. The question is whether they are manageable in terms of mesh requirement and computational demand as well as the effectiveness of their prediction. The aim of this section is precisely to shed some light in the topic.

⁹airfoil, cylinder, sphere, aircraft, etc.

4.3.1 Our Immersed Boundary Methods

There is a whole family of Immersed Boundary Methods in function of how they treat the boundary, the steps employed to ensure its flow conditions and the order of those steps. In our CFD models we have employed two different methods which are developed in Table 4.3.

IBM Method	Order	Boudary	Characteristics
Smoothing	1	smooth	stable, no oscillations, better for moving objects
Direct Forcing	2	sharp	oscillations and numerical effects

Table 4.3: The two different approaches tested in the case of a 3D CFD of wind turbines.

The tested first method is similar to the Peskins approach, in terms of the boundary treatment: it is not a sharp one but a smooth one. We wanted to try this method to ensure the numerical stability of the code given the demanding conditions of rotating wind turbines, specially in the tip sections. Thus, the source term of the Navier-Stokes equations was modified in the regions next to the turbine solid boundary as well as inside the object while leaving the fluid zones unaltered. More information about the rate of change of the momentum source term will be provided later on in this subsection when introducing the Heaviside function.

After successfully run smooth boundary cases, the Direct Forcing approach was tested to analyze the effect of imposing the velocity directly. The idea of this method is to use a distance function to classify the nodes in the vicinity of the boundary into different categories. Some of them are used as points to impose the boundary condition (right on the boundary and inside the body), while those nodes immediately beyond the boundary (towards the fluid region) get their velocity interpolated between the known wall value and the outer fluid region. See Figure 4.12.

The details of how to impose the velocity at the Direct Forcing nodes are shown in the following lines. Let's depart from the time integration of the momentum equation:

$$\frac{\mathbf{u}_P - \mathbf{u}^n}{\Delta t} = \mathbf{RHS}^n + \mathbf{f} \quad (4.14)$$

Where \mathbf{u}_P is the predictor velocity, \mathbf{u}^n is the velocity at previous time step (n), Δt is the time step and \mathbf{RHS}^n includes the convective and diffusive terms. Imposition of

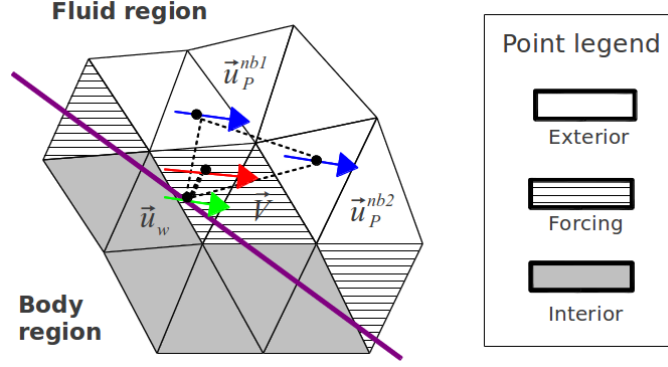


Figure 4.12: Schematic arrangement of the IBM Direct Forcing approach.

$\mathbf{f} = 0$ in Equation 4.14 takes place for the exterior points, while non-zero values are computed for the forcing points. The predicted velocity for the forcing points is imposed as dictated by Equation 4.15. See Figure 4.12. From there, the result is introduced into Equation 4.16 in order to compute the \mathbf{f} (a correcting term, for the sake of completeness in terms of registering the history of the *RHS* terms).

$$\mathbf{V} = a_w \mathbf{u}_w + a_{nb1} \mathbf{u}_p^{nb1} + a_{nb2} \mathbf{u}_p^{nb2} + a_{nb3} \mathbf{u}_p^{nb3} \quad (4.15)$$

$$\mathbf{f} = \frac{\mathbf{V} - \mathbf{u}^n}{\Delta t} - \mathbf{RHS}^n \quad (4.16)$$

where a_{nb_i} are the geometrical coefficients of the interpolation of each velocity \mathbf{u}^{nb_i} and the subindex w correspond to the same concepts applied at the wall point. With the current definition, the whole field of predicted velocities, \mathbf{u}_p , is computed. At this point, it is time for the Poisson equation to be solved and the final velocity field is corrected with the gradient of pressure as it is usually done in the Fractional Step Method.

Anyway, regardless the IBM approach, all methods do require an indicator to discern the location of the boundary (rotor surface) at any moment. We must recall the work presented in Section 2.3: the real boundary of the moving wind turbine is tracked at every moment (STL approach) and merged with the contributions of Section 2.4. The basis for the signed distance computation was presented for both general objects and real wind turbine components – specially blades. It is important to note that the distance evaluation is the foundation from which the IBM is erected.

As shortly introduced right above, the first Immersed Boundary Method to be tested stretches the effect of the boundary across a number of cells (smoothing) instead of limiting it to one of them (sharp behavior), as the second tested method does. Consequently, a mathematical filter must be applied to the distance function to ensure this gradualism. The smooth heaviside function is a good candidate for this task.

$$\phi = \begin{cases} d \leq -\epsilon & 0 \\ -\epsilon < d < \epsilon & 0.5 \left(1 + \frac{d}{\epsilon} + \frac{1}{\pi} \sin \left(\pi \frac{d}{\epsilon} \right) \right) \\ d \geq \epsilon & 1 \end{cases} \quad (4.17)$$

where: d is the distance value of the point delivered by the distance computation; 2ϵ is the distance where the boundary will be smoothed; and ϕ is the value of the smoothed distance.

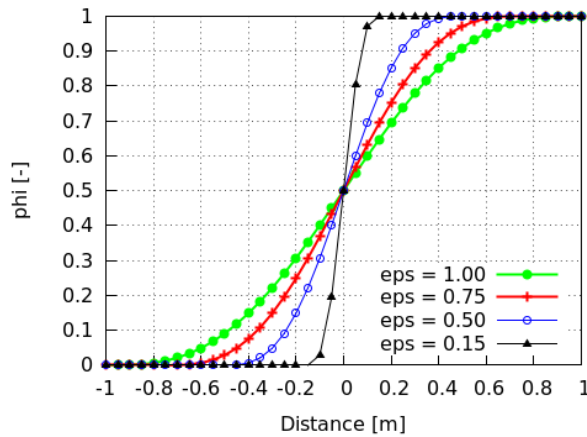


Figure 4.13: Heaviside smoothing of the distance function depending on the value of the ϵ parameter.

The result of filtering the distance function with the Smooth Heaviside function is a smoothed curve such as the ones appearing in Figure 4.13. As it can be observed, the effect of reducing ϵ is shrinking the space where the blurred boundary stretches. In the limit, when $\epsilon \rightarrow 0$, the boundary becomes sharp. Notice that the convention states that cell values of 1 are considered to be solid, fluids are tagged with 0 value and the blurred boundary adopts values between 0 and 1. Typically, the distance where the interface is desired to be smoothed, 2ϵ , is set in reference to the volume of the cells (V_{cell}) shaping

the mesh in the way shown by Equation 4.18, where N is the number of cells:

$$\epsilon \approx N \sqrt[3]{V_{cell}} \quad (4.18)$$

The distance function calculation and filtering was extensively validated by running the simulation in different conditions, combining several meshes with a wide range of objects such as the ones presented in Figure 4.14. From left-right and top-down, the STL figures tested are: a cube with the smallest number of triangles – two triangles per face; a cube with four triangles per face; a cube with multiple triangles per face; a sphere with roughly a hundred triangles; a tetrahedron; a squared-base pyramid; and a tetrahedron partially voided by a smaller tetrahedron (with the same base but having shorter height), generating a closed figure but with concave and convex regions.

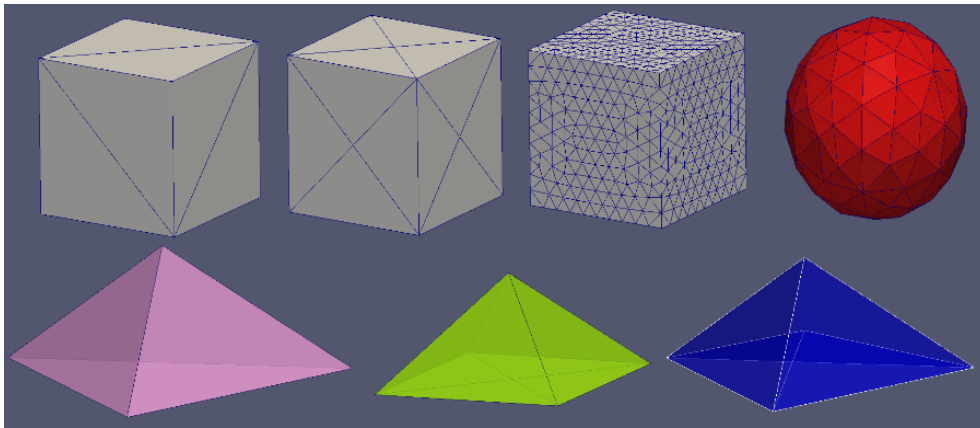


Figure 4.14: The set of simple figures that were employed in the IBM testing.

Those tests undeniably made a good service: many bugs and associated problems were found and solved. The multiple testing enabled the discovery of the *combined effect* of the distance field computation presented in Subsection 2.4.2. More concretely, Figure 2.22 showed with precision the associated problem to sharp edges. It was only detected when examining the results of simple STL objects because they enabled the author to follow up all the steps during the computation. In this way, a full understanding of the underlying problems was achieved and the solutions came along. Figure 4.15 shows the field of heaviside filtered distance of the pyramid case, employing an unstructured mesh. In the image, several elements coexist to display the whole idea. Firstly, the STL pyramid is split in two halves, one presents full opacity and the other is quite

transparent to enable us to observe the inner region of the figure. There is also a horizontal cut showing the filtered distance scalar field, which is the filtered distance function encompassing values ranging from 0 (blue) to 1 (red). The interface lies just where the 0.5 values are detected and a white iso-surface is built at this value, clearly mimicking the STL pyramid shape.

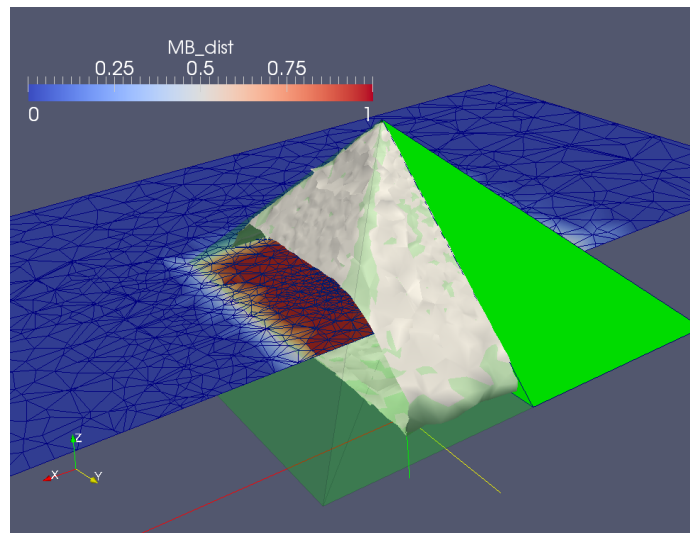


Figure 4.15: Details of the distance function validation in the case of the pyramid.

Having validated the distance function for simple objects, the focus was moved to the next level of complexity: real wind turbine components. As commented in Chapter 2, there are four crucial elements: the tower, the nacelle, the hub and the blade. Usually, tower and hub pose no problems because their geometries tend to be cylindrical and spherical, respectively. On the other hand, nacelle and blade are much more complex. The nacelle tends to be easy in most cases but in some others – such the MEXICO one – its geometry is tricky. And finally, the blade geometry is always complex because of three reasons: the twist law, the sharp region at the trailing edge zone and, specially, the transition to root sections.

Figure 4.16 contains three images to show the process of distance computation. Subfigure a) visualizes the effect of the inappropriate distance computation: the sign of certain space regions is incorrect (blue negative, while red positive). The solution to that problem was the one proposed in section 2.4.2: computing an averaged normal vector

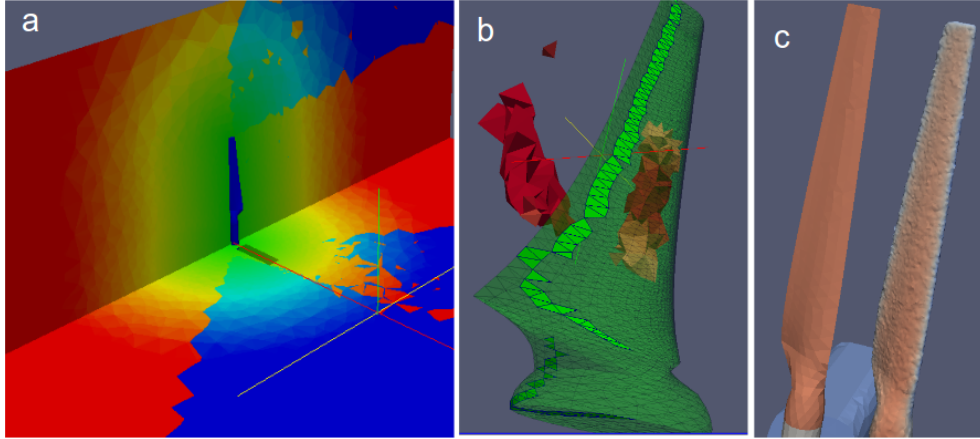


Figure 4.16: Details of the distance function validation in the case of a real blade (NREL): a) errors due to the random assignment when the vertex criterion was not implemented yet. b) mistakes on the distance calculation even when the NEighbor Walking Algorithm was running. c) final approach with all the heuristics developed: successful results in comparison to the STL blade.

for the cases where vertices of the STL were the closests to the cell centroid. Subfigure b) also shows some incorrect zones near the blade even when employing the *NEighbor Walking Algorithm* proposed in Chapter 2. Finally, Subfigure c) shows, for the case of the NREL blade, the final results of the combination of the heuristics commented in Section 2.4.3. As it is possible to see, all the artifacts and spurious boundaries are totally removed and the system is trustable. Same successful effects can be appreciated in the MEXICO turbine appearing in Figure 4.17.

The one-by-one (STL object) distance computation and smoothing have been proven to be succesful. The point now is how to approach the wind turbine as a whole. Bear in mind that the wind turbine geometry is modelled as a composition of STL objects and the present tool only resolves the objects one-by-one. The solution is as simple as computing the filtered distance function for all objects (independently of the rest) and then execute the overlapping by keeping the maximum value at every cell. Selecting the maximum value at every control volume as Equation 4.19 shows, ensures that solid parts are imposed over fluid ones, which contributes to shape the turbine.

$$\phi_{\text{wind turbine}} = \max\{\phi_{\text{tower}}, \phi_{\text{nacelle}}, \phi_{\text{hub}}, \phi_{\text{blades}}\} \quad (4.19)$$

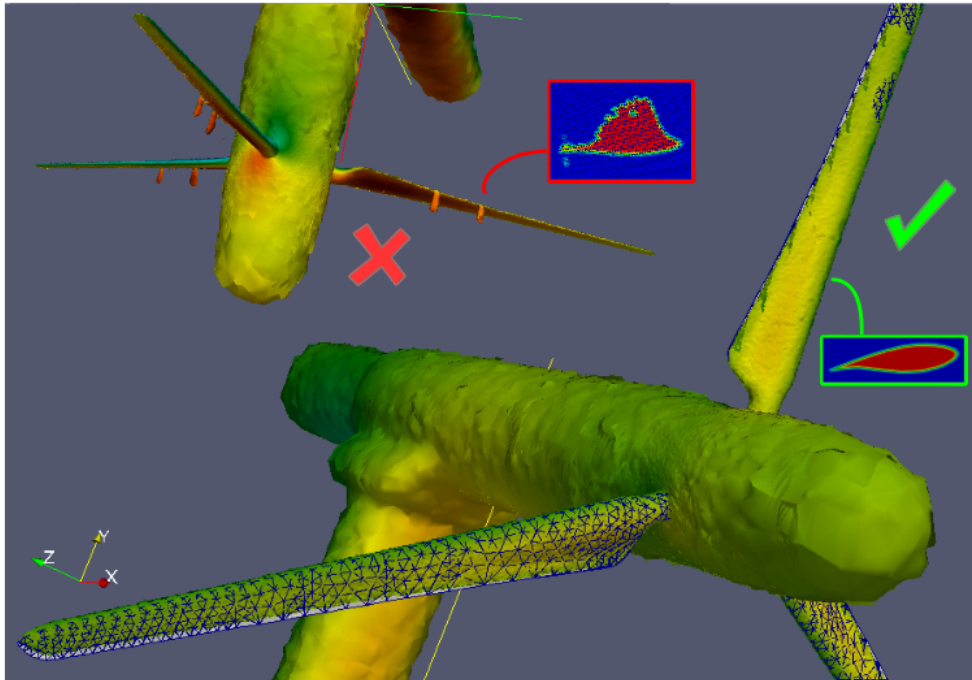


Figure 4.17: Distance function interface at 0.5 filtered distance in the case of the MEXICO's full wind turbine. Top-left: still with problems. Bottom-right: final approach error-free with acceptable geometric accuracy.

To conclude this subsection, it might be interesting to mention that with the so far presented strategy, the whole framework for IBM is totally defined in terms of geometrical description and mathematical formulation. However, nothing is said in terms of computational costs and affordability of the computations which is the next subsection's objective.

4.3.2 From advection to direct computation

The first attempt to solve the flow around a wind turbine in operating conditions was done employing advection solvers in order to *simulate* the effect of the solid-fluid interface, which is done by means of using Equation 4.20, valid for incompressible flows. The velocity of the advection field is given by the rotation pattern of the machine, and as

the reader can imagine, this advection comes along with the numerical diffusion which in practice implies the degradation of the surface quality.

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot (\mathbf{v}\Phi) = 0 \quad (4.20)$$

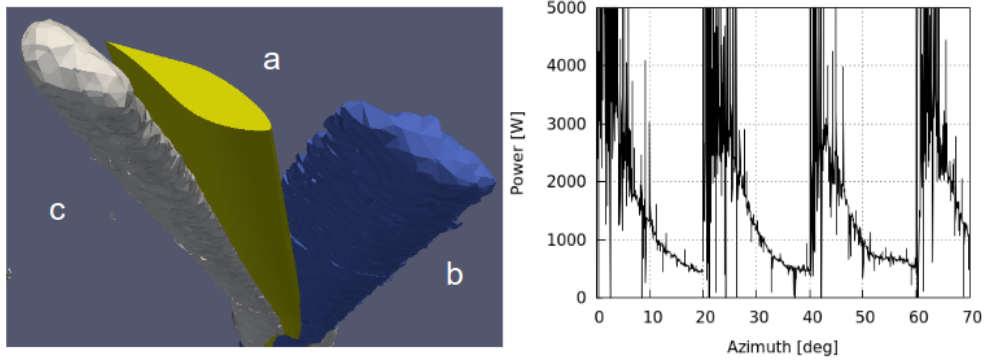


Figure 4.18: Left: comparison of the STL blade shape (a, yellow), with the initial geometric loading (b, blue), and the after-30° advection (c, white). Right: The effect on power due to periodic geometric reloading.

The image at the left hand side of Figure 4.18 compares the shape of the STL geometry (a, in yellow) with the initial geometry loaded (b, in blue) and also the advected boundary around 30° later (c, in white). The mesh in which the geometry was loaded and advected was around 10 Million of cells. As it can be seen, the rounding effect for the specific mesh configuration is dramatic. The area of the rotor can be computed by means of using Equation 4.21 and the normal of the surface at any cell is given by equation 4.22. In fact, the geometry degradation pattern is the responsible for the power decay observable at the right hand side of Figure 4.18.

$$A = \int_V \|\nabla\Phi\| dV \quad (4.21)$$

$$\mathbf{n} = \frac{\nabla\Phi}{\|\nabla\Phi\|} \quad (4.22)$$

Furthermore, beyond the geometry degradation problem, the initial geometry is clearly not representing the STL at the current mesh size. The conclusion from this initial idea of using advection solvers to account for the distance field rotation turned out to be unacceptable.

The next approach was the idea of combining the periodic geometric refreshment with a short advection. This means, for instance, that if the geometry is loaded at 0° , the advection can be performed for, say, 20° and then refresh the geometry when reaching that position. In such case, 18 reloading steps would be required to complete one spin.

After implementing the platform for general periodic reloadings, it was evident that not even a reloading frequency of 15° sufficed, and probably, higher reloading frequencies make no sense. As Figure 4.18 shows in the right hand side, the power rapidly falls when rotating just 10° . Therefore, this idea turned out to be impracticable only taking into account geometrical accuracy.

Ideally, advection needs to be removed and avoided, but the bottle neck of the whole distance computation is the astonishing computational cost. For the largest case run in this thesis, 60 million control volumes were employed in the mesh and 80 thousand STL triangles were used to defined the turbine geometry. This implies that $5 \cdot 10^{12}$ point-triangle distances were necessary to define the whole scalar field defining the immersed boundary. 512 CPUs demanded 4 hours of computing time to make that evaluation which is clearly prohibitive. The brute force approach is the limiting agent and a solution needs to be found to this.

At this point is when we came up with the *box restriction*, the *accurate-simplified distance filtering*, the *NEighbor Walking Algorithm*, the *Opposite criterion* and the rest of heuristics exposed in Chapter 2. The application of those principles led to a huge time saving that actually made the difference, since only 3-5 seconds – depending on the case – were required to reload the distance field with the maximum accuracy that the mesh enabled. Figure 4.19 agglomerates four different ideas: a) displays the path followed by the NEighbor Walking Algorithm for all the wind turbine components given a specific mesh point; b) represents the idea of the box limitation, where the different components only compute distances within their box; c) represents the Offwindtech's wind turbine geometry employing structured meshes – which clearly have a smoother look than the one offered by unstructured ones; d) from left to right, the comparison of the blade geometry in the STL, the isosurface representing the turbine boundary and the nodes selected by the same threshold.

With the current approach based on heuristics and intelligent algorithms, the change in computing time is radical with respect the brute force approach. Thus, we sorted out the days-long preprocess employing several GB of RAM and getting the geometry degraded. Alternatively, the new scenario offers a very fast preprocess, no extra memory usage and the maximum geometry precision. In other words, those improvements actually meant that this research line could go on instead of being closed.

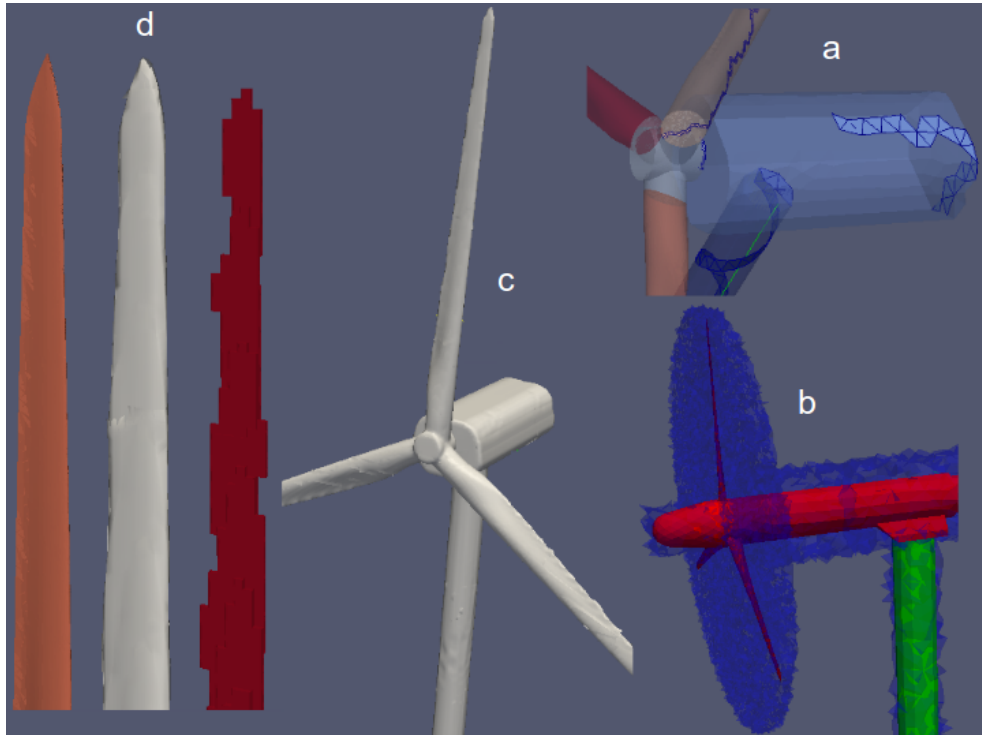


Figure 4.19: Application of the distance function to the full wind turbine of the Offwindtech project employing the Neighbor Walking Algorithm for all the components (a), incorporating the boxes to limit the nodes to which the distance functions is computed (b), and the results of the process in (c) and (d).

Full wind turbine 3D simulations of the Offwindtech project (see Figure 4.20) could be closed with the right level of detail according to the project expectations: the presented approach actually predicted the order of magnitude of the 5 MW offshore turbine. In spite of the reasonably good agreement of these global parameters, the reader should keep in mind that no validation was carried out. Caution must be present until the comparison to the scientific experiment is carried out. The aspect of the validation is considered in the next subsection.

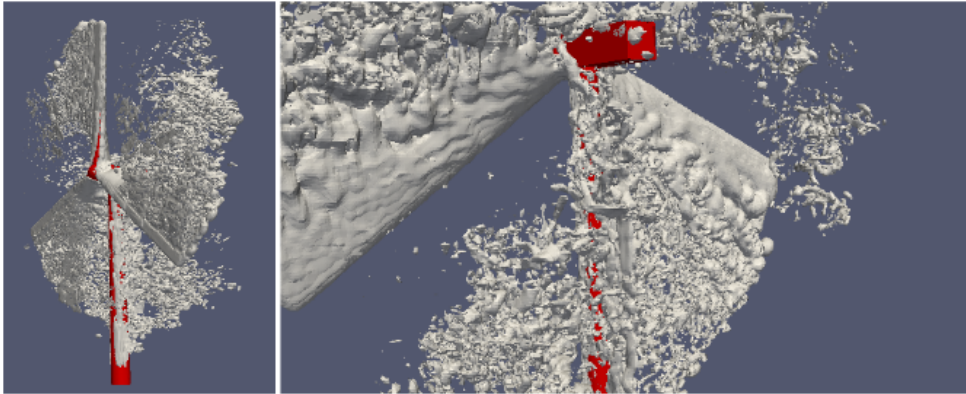


Figure 4.20: CFD results of the Offwindtech rotor: coherent turbulent structures.

4.3.3 Results: Offwindtech, NREL and MEXICO

The NEighbor Walking Algorithm and related heuristics have proven to have a point as a potential tool for wind turbine aerodynamics assessment. Now, it is time to actually compare the results of a sufficiently known case as the aforementioned NREL and MEXICO cases.

The degree of geometrical precision is theoretically only depending on the mesh density, so this is not the point of the discussion. What's more, the geometry definition for the MEXICO case – with a much smaller turbine than the Offwindtech 126 vs. 4.5 meters in diameter – is comfortably described in the cases presented in Figure 4.21 (leave the fluid-dynamic behavior aside for now). For the MEXICO case, thousands of hours were granted in the Barcelona Supercomputing Center (BSC, Mare Nostrum), which allowed us to run a wide range of situations, create videos and even testing situations in dynamic yaw. Meshes employing 40 to 60 million of cells were usual at that time.

The true essence of the validation consists in both the global and local indicators like torque and power (global), and the pressure distributions (local). The power is computed in two different ways. The first one is by employing only the pressure field and the heaviside distance function, and the other one uses the STL file and the pressure field. Both approaches match with discrepancies below 1 %. Equation 4.23 shows the general way to compute the torque and Equation 4.24 presents its conversion to the power.

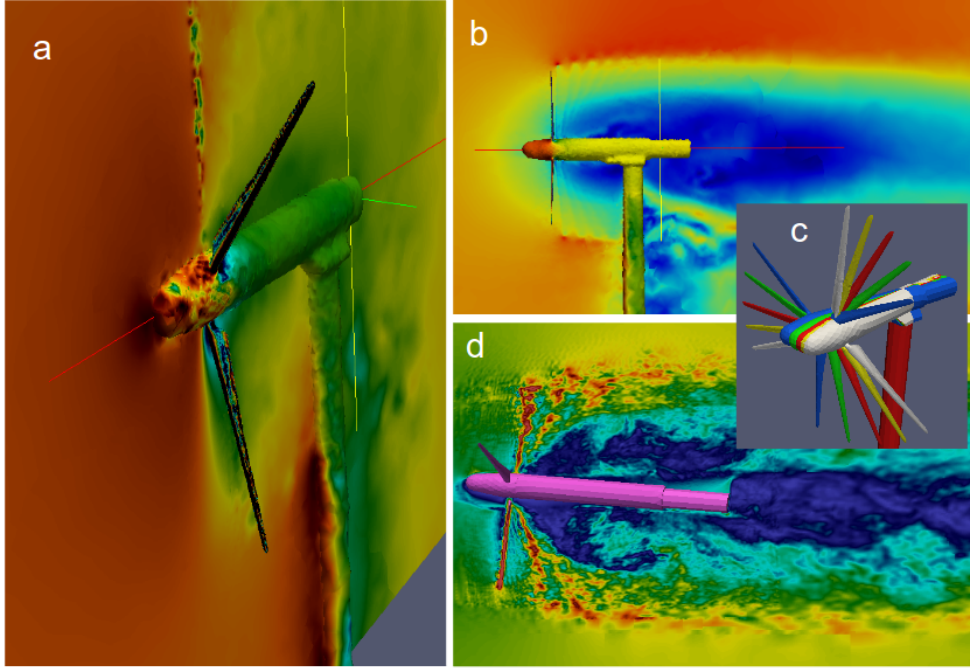


Figure 4.21: CFD simulation of the air flow around the MEXICO wind turbine employing IBM: a) pressure field on both the slice and the turbine surface, b) velocity field at the center slice, c) proof of concept of yaw STL capacity and d) instantaneous velocity field on the dynamic yawing case.

$$\mathbf{T} = - \int_S (\mathbf{r} \times p \mathbf{n}) ds \quad (4.23)$$

$$P = \omega T_{axis} \quad (4.24)$$

where: \mathbf{T} represents the torque; p is pressure of the element; \mathbf{n} is the unitary normal vector; ω is the angular velocity; and T_{axis} is the component of the torque applied to the rotating axis.

The power computation is the first parameter that one would check in order to know whether the simulation is potentially correct or not. If the power is in the range of the expected value, then a second step would be necessary: going into the local observation of pressure distributions. The MEXICO case with -2.3° pitch angle, and a free stream

velocity of 24 m/s is studied first. Figure 4.22 shows the power evolution over time. More concretely, and for the ease of interpretation, the independent variable represents the number of spins that the rotor has completed. As it is possible to see, there is a peak at the right at the beginning (27 kW), and around 3 spins later, the power gets stable at the approximate value of 21 kW which is quite different (roughly 30 %) from the 30.8 kW to be expected, according to the experiment.

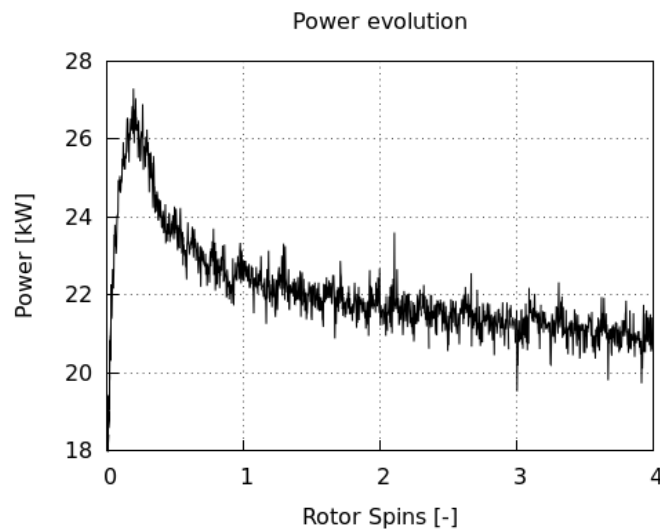


Figure 4.22: Power evolution of the MEXICO case at 24 m/s and -2.3° pitch angle. CFD predicted power: 21 kW. Experimental: 30 kW.

It is convenient to keep in mind that BEM-based methods, like the ones presented in Chapter 3, are able to predict quite accurately, and with much less effort than CFD-based methods, the power production in certain conditions. Thus, 30 % variations sounds like too much and definitely it is. However, bear in mind that the 24 m/s case is absolutely in stall. Perhaps BEM methods also become less accurate at these ranges.

When dealing with the 15 m/s case, the situation is different because the power prediction is the same range: the expected 12 kW that the experiment predicts. Now, the key is to check whether the method is behaving good at the local level or the power prediction is just a coincidence. The way to do so is by comparing the experimental and numerical results of the pressure stations of the blades, see Figure 4.23. The agreement is fairly good. Notice that the global trend is captured along the blade. Acceptable

results were obtained for sections at the 35, 60 and 92 % of the blade, while small discrepancies are present in the 82 % and considerable deviation showed up in the 25 %.

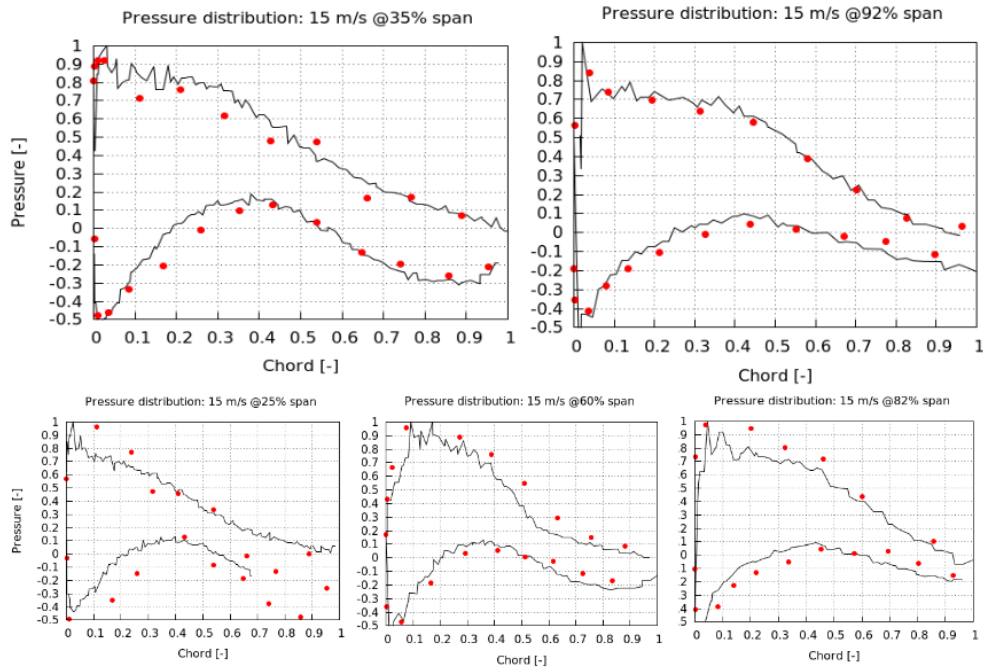


Figure 4.23: Pressure distributions in the 15 m/s MEXICO case at different span-wise stations. At the top, the 35 and 92 % sections of the blade, the ones fitting better the results. At the bottom, the 25, 60 and 82 % sections presenting large, small and medium discrepancies.

Generally speaking, the reasonable good results presented in Figure 4.23 might be promising for the feasibility of the method, but it should be mentioned that this is the best case available within the context of this thesis. Caution must be taken before considering this method absolutely valid. The MEXICO case is the most favorable case of all the ones tested here: the NREL is double size, the Offwindtech is more than ten times bigger and the purpose of CFD is to aim at industrial size machines. Therefore, mesh studies employing more than a hundred million control volumes should be the next movement in order to check whether there is a mesh convergence or not. However, such computationally expensive simulations are not within the scope of this thesis.

A good test to check the internal consistency of the method would be to simulate

2D airfoils employing solely IBM, with meshes built up with a 2D equivalent section to the 3D minimum slice capacity. In other words, the simulation would consist of an evaluation of a 2D airfoil of the rotating turbine with the same geometrical precision and Reynolds number as experienced in a section of the 3D case. Such simulations were carried out for DU (Delft University) family airfoils and the results were found to be consistent with the 3D results. Up to 30 % variations on the lift coefficient were present.

As we know, fluid vortices (scales) are downsized as Reynolds grows. In fact, the difficulty of the simulations are measured by this indicator. Definitely, the same idea applies to the IBM case. Some of my co-workers are dealing with industrial applications that allow them to use IBM in cases with heat transfer and fluid dynamics with complete success. For instance, in the cooling of electronic components with complex geometries and the convection inside a printer were modeled with the same tools, [53]. In the field of aerodynamics the same idea is replicated. [36] shows the simulation over a NACA airfoil with a moving flap (spoiler). When the flap is analyzed at Reynolds numbers in the order of 50,000 the results match the experiments while in cases with Reynolds up to the million become manifestly insufficient.

As a matter of anecdote it might be interesting to mention that the present work done for wind turbines shares its essence¹⁰ with a wide variety of other cases. Thus, the same work could be reused in fields beyond wind energy such as automotive industry and chemistry. First, my co-worker David Aljure slightly adapted the ideas presented here to the aerodynamics of cars, for the specific case of the flow around the wheels [54]. And second, a chemical mixer powered by a Rushton turbine was studied using the same technique. The mixer was designed to be used in the field of stem cells, see Figure 4.24.

4.3.4 Conclusions

In conclusion, the Immersed Boundary Methods seem to be a good methodology for the simulation of wind turbines in operating conditions. The point is that high Reynolds numbers are too demanding for the current state of the art of High Performance Computing and computing capacity. Perhaps, in a matter of years this method can play an important role in the field because of their high versatility to approach pitch and yaw dynamic changes as well as other aerodynamically interesting aspects. It can possibly be that in the near future the combination of them with other techniques such Adaptive Mesh Refinement can lead to a robust simulating platform within the paradigm of the

¹⁰the rotating nature and the complexity of geometry

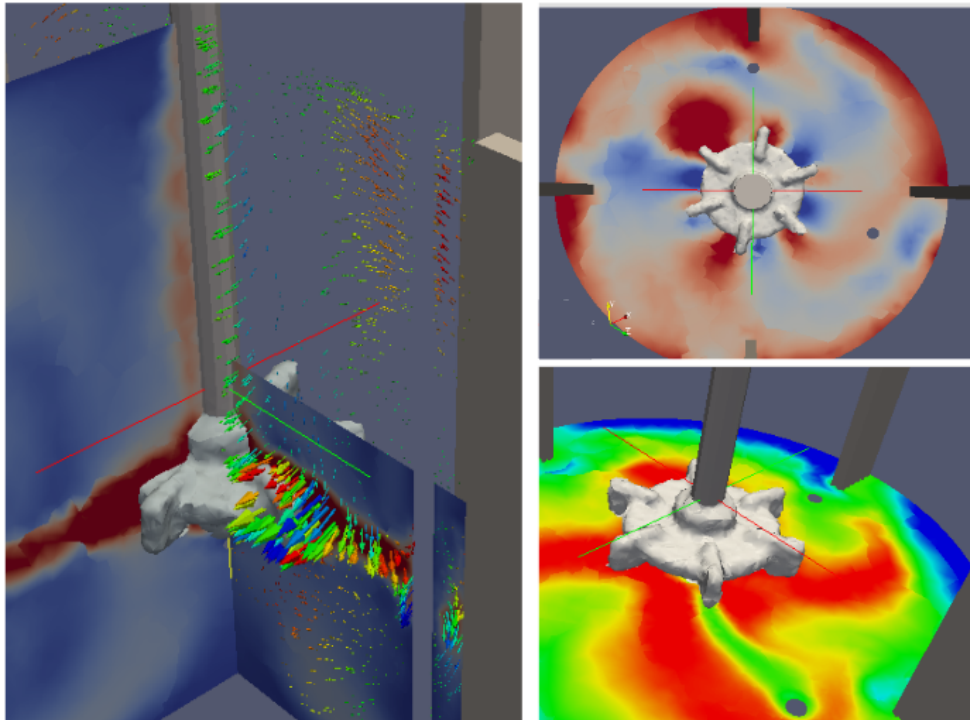


Figure 4.24: Instantaneous velocity field in the Hexascreen chemical mixer.

upcoming Exascaling.

4.4 Sliding Meshes

A third CFD-based approach to study the wind turbine's near wake aerodynamics has been implemented. It has been developed in the context of this thesis with the help of my co-workers, specially thanks to the strong contribution from Jordi Muela [55]. The method consists in the separation of two regions of interest encapsulating the rotating and the static components. Two different meshes are created, one for the rotating regions and another one acting as a background. The connection between the grids is actually the cylindrical shape of the rotating domain. When the cylindrical domain rotates over its own axis, its relative position with respect the background mesh changes

over time. The outer part of the rotating mesh actually "slides" over the inner part of the background mesh, that's why this method is called *Sliding Meshes*. Figure 4.25 shows an example of how the concept of Sliding Meshes actually applies to reality: a) shows the CAD model of the turbine, b) presents (an slice of) the rotor mesh, and in c) it is possible to see the background mesh.

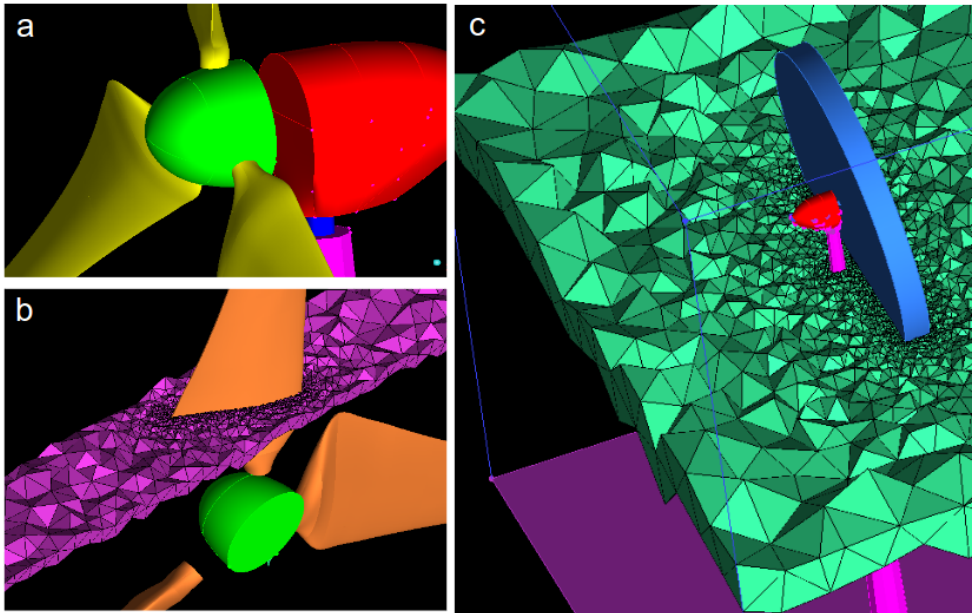


Figure 4.25: Meshing process in the Sliding Mesh approach: a) geometry, b) rotor (moving) mesh, c) background (static) mesh.

The incompressible Navier-Stokes equations, formulated in the so-called Arbitrary-Lagrangian Eulerian (ALE) are the next:

$$\nabla \cdot \mathbf{u} = 0 \quad (4.25)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot ((\mathbf{u} - \mathbf{v}_d) \mathbf{u}) = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \quad (4.26)$$

Where \mathbf{v}_d is the displacement velocity of the domain with respect the Eulerian Frame of Reference.

The background and rotating meshes are built in separate ways using ICEM ANSYS. Subsequently, they must be put together as a combined domain, but enabling TermoFluids to know who is who in order to perform the rotation of one while keeping the other static. Figure 4.26 shows a slice of the volumetric mesh when both meshes are already merged into a continuous domain. If it is watched in detail, the separation lines are actually visible (obliquous rectangular shape surrounding the rotor). Additionally, details of the mesh around an average airfoil of the blade are introduced at the right hand side of Figure 4.26. Notice that around 200 control volumes are used in the edge-wise direction and also a prismatic boundary layer (PBL) could be grown from the superficial surface of the blade because the flat trailing edge was introduced in the CAD model¹¹, check Chapter 2 for details on that.

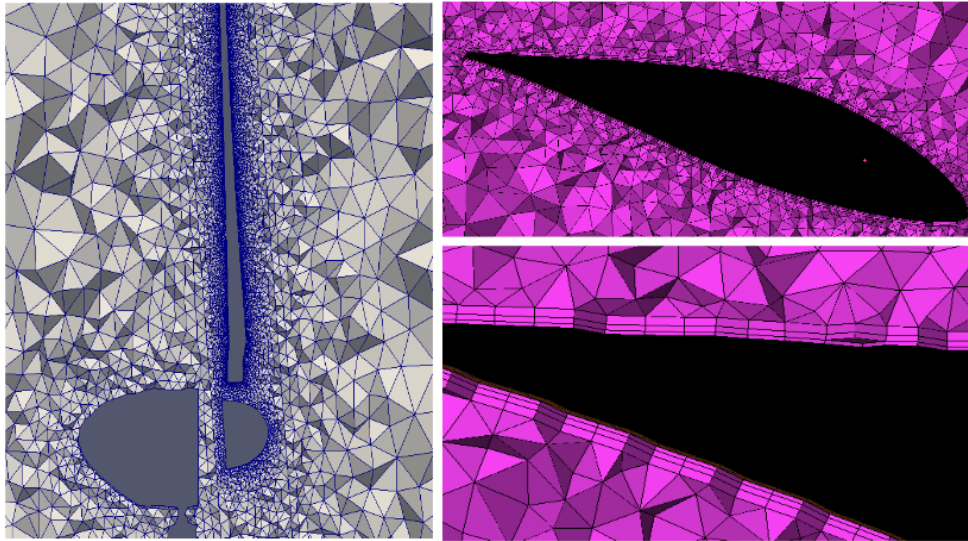


Figure 4.26: Left: Combination of the background and rotor mesh into a common one. Right: details of the prismatic boundary layer around an average section of the blade.

With the current strategy, some type of communication must be carried out between the two meshes in order to share information about pressure and velocities among others. This communication is actually done by means of employing a series of particles in both the static and dynamic meshes. The idea is to inject particles associated to

¹¹building a PBL from a sharp airfoil generates low quality pyramids that lead to divergence

the rotating mesh in the non-rotating region, immediately next to the sliding zone (see Figure 4.27b, blue color). Those particles will rotate in a geostationary orbit to the moving mesh. Likewise, the background non-rotating mesh injects some particles into the rotating region (red color). These particles represent the centroid of a virtual control volume that mirrors the geometry of the associated node. Thus, every boundary node in the sliding zone has a virtual element overlapping the opposite side mesh. The properties of this cross-boundary element are interpolated with the information of the surrounding nodes. In this way, this information is used to complement the fractional step method in the sliding zone neighbouring cells.

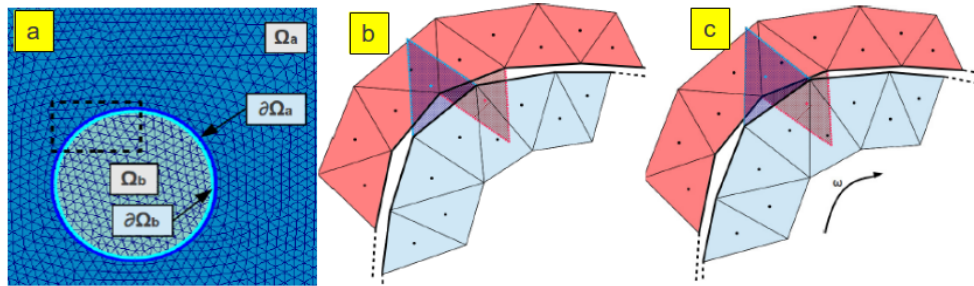


Figure 4.27: Essential details of the Sliding Mesh concept: a) scheme of the background and rotating meshes, b) and c) effect of the displacement of one mesh with respect the other and the corresponding mirror nodes.

Another interesting aspect when managing the Sliding Meshes approach is how the mesh rotation is actually performed. The answer to this question is what is called *Dynamic Meshes* which was mainly developed and implemented into TermoFluids by Olga Estruch, check [56] for further details. This technique actually allows the mesh control volumes to change in shape and size in order to adapt to moving boundaries or other specific criteria. The application of the Dynamic Meshes is easier in the case of Sliding Meshes because the elements do not change their size and shape, they only suffer roto-translations. From the point of view of the core algorithm, this situation represents an advantage, although the cell location and its face normals change over time, the volumes, surfaces and other element-related properties do not, which turns out a saving in terms of computational effort. Thus, the mesh movement is externally driven by the rotation regime of the machine's operating conditions. Additionally, notice that the tilt and cone angles can easily be implemented in the framework of this Sliding Meshes approach and also, the pitch behaviour could theoretically be studied with the

particularity that extra computing time would be required.

The Sliding Mesh approach was applied to the cases of Enair’s downwind 10 kW and 40 kW wind turbines. Their blades were completely designed at CTTC. Some geometrical details of them are displayed in Chapters 2 and 3, and additionally, Appendix A contains the relevant information about the process. The turbines were designed to achieve rated powers of 10 and 40 kW at the rotational speed of 120 and 60 rpm, respectively. The BEM-based characterization employing the AeroElectric code was used to predict the overall behaviour of the machine across the different working points. Those calculations are used as a guide for the decision making when it comes to select which cases might be potentially interesting for the 3D simulation. With the AeroElectric results in view, the best cases were those points near the design point, where the power production is in the range of maximum marginal contribution to the Annual Energy Production, see Table 4.4.

# [-]	Turbine Enair [kW rated]	Wind [m/s]	Ang speed [rpm]	Pitch [°]	BEM-Power [kW]	CFD-Power [kW]
1	10	6.5	78.5	2.5	4.7	4.0
2	10	8.5	97.7	2.5	8.8	12.0
3	10	10.0	120.0	2.5	12.5	13.0
4	40	7.5	41.0	2.0	25.0	17.5

Table 4.4: CFD cases for Enair’s 10 and 40 kW turbines.

Those simulations were carried out at the JFF (CTTC cluster) involving 512 CPUs for a couple of days each of them. This computing time is enough to observe the stable behaviour of the turbine. This state is reached with less than two rotor spins (in opposition to the 4-5 spins needed when using IBM). In this case, for the first time, we could see the effect of the tower interaction. In fact, one of the strongest points of performing 3D CFD simulations is the possibility to account for the power loss due to the shadow that the tower inflinges over the rotor. Rotor-tower interactions studies are scarce in the scientific literature. To the best of the author’s knowledge, only two have been published. This paper, [57], studies the interaction in an intelligent way: employing 2D interaction at a particular tower height. The only 3D study of such phenomena is available in [58], although in order to be fair it must be said that [19] also contains some short analysis about it. In conclusion, those studies show that tower-induced

perturbations are non-negligible: power can decay by as much as 25 % during 60° of the spin (a sixth of it!). Similar results are obtained in this thesis as it will be pointed out later.

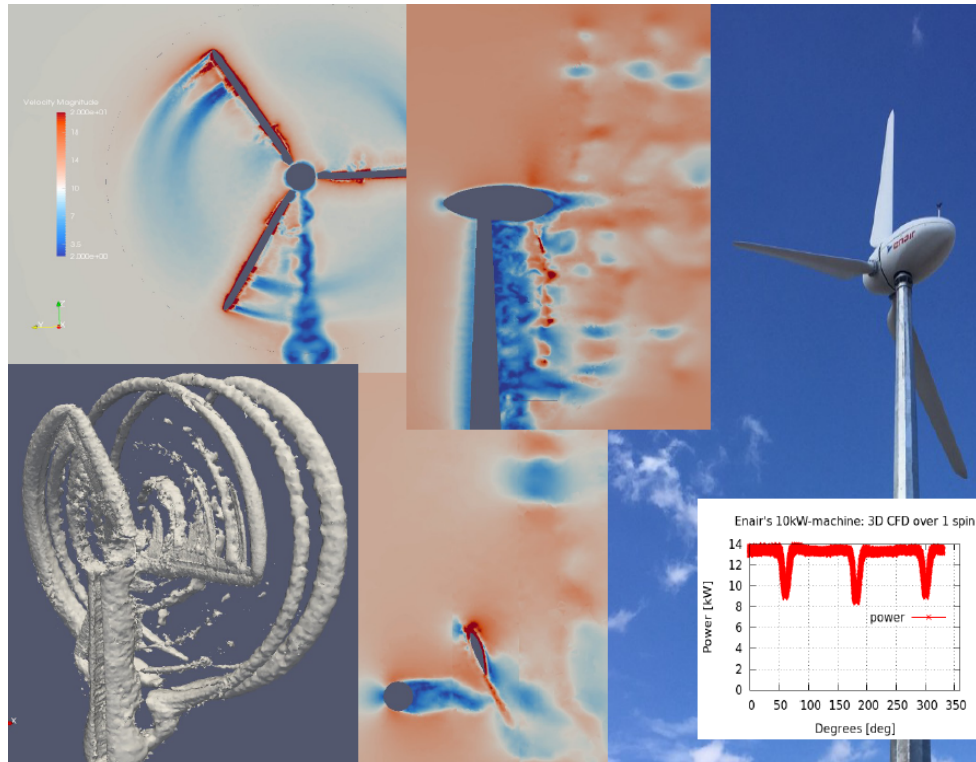


Figure 4.28: CFD snapshots of the Sliding Mesh approach.

At this point, an important fact must be commented. All simulations carried out with the Sliding Meshes method were found to have unphysical solutions. Undesired high wind speeds (as high as 10 times the expected value) showed up near the blade surfaces and the reasons for such spurious velocities were not found yet. We suspect that something related to the boundary conditions or the mass conservation of the Sliding Meshes method might be backfiring the normal course of the simulation. Perhaps the transition from the prismatic boundary layer (hexahedra) to the global mesh (tetrahedra) is also posing some problems. The author acknowledges that more investigation is

required in this respect. Anyhow, no divergence is experienced but the velocities compromise not only the reliability of the presented results but also the cost of the simulation because higher velocities imply smaller time steps and this increases the computing time. Therefore, in midst of this unwanted situation we thought of having an artificial velocity limiter to keep the velocities in the expected level. The criterion to do so was based on the fact that bluff bodies usually alter the flow field by a rate of 50 % the free stream velocity. Thus, the maximum velocity allowed in the domain was 1.5 times the tip relative wind speed.

With this provisional patch¹² we could carry on with the simulations and analyze the results. Figure 4.28 shows the qualitative results of the analysis, which were presented to Enair. The flow fields look reasonable and the detail of the power computation can be seen in Figure 4.29.

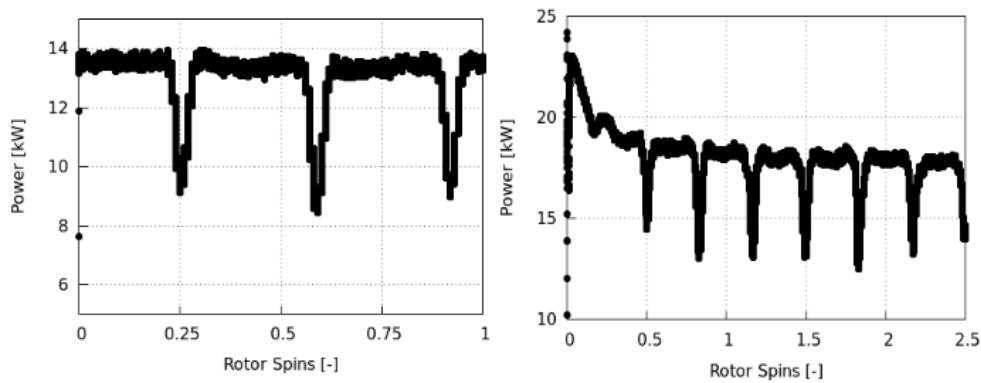


Figure 4.29: Power evolution of the turbines employing Sliding Meshes. Left: case 3 from Table 4.4. Right: case 4.

Figure 4.29 presents the power evolution of the 10 kW and 40 kW turbines when running in the conditions of cases 3 and 4 from Table 4.4, respectively. As the reader can see, the power reduction at the blade-passage behind the tower is quite relevant. For instance, in the 40 kW turbine case (in the right hand side of the figure), the net average power neglecting the tower interaction would be 18.5 kW, but due to the peak power reduction, which is in the order of 4.5 kW and stretches for about 7 % of the spin, the final averaged power turns out to be 17.55 kW, meaning a 9.5 % net reduction.

¹²it requires in depth investigation of the mathematical foundations of the model

A final idea must be commented in order to insist on the fact that those results are provisional (due to the maximum speed limitation exposed before). Clearly, additional work must be done to unravel the reasons leading to this unphysical results. The results presented are correct from the qualitative point of view, but in order to trust them from the quantitative perspective some work must be done to delete the *external* velocity limitation. Although the limit is set having a physical criterion in mind, this aspect must be worked out to avoid any possible arbitrariness.

4.5 Parked

Beyond the logically interesting cases where the wind turbine is rotating, there are some other situations involving the aerodynamics of wind turbines that demand a study to comprehend the underlying phenomena. One of these cases is the so-called *Parked* Conditions. This case is characterized by having a static rotor where the flow around the turbine is studied for two different reasons. First, because of security purposes when extremely high winds are registered (hurricane-like: 30–40 m/s). Nowadays, certifications of current wind turbines require them to withstand those high winds. The second reason is in the context of installing conditions (when erecting the turbine *in situ*). As the reader can imagine, the installation is done component by component in the natural order, so when it is the turn of a blade or the whole rotor¹³ to be assembled into the hub or nacelle, the precision required during that handling operation is incredibly high (order of milimeters). Ironically, in this situation, the wind is a big handicap, which of course is a desired condition for normal operation but a setback in this specific (installing) conditions because it causes a force over the blades or the rotor during the process carried out by huge cranes. Thus, the Parked conditions case is of extreme interest in the scientific and engineering world of wind energy for both the security in front of extreme wind conditions and the installation procedure.

It might seem that the non-rotating state of the standstill conditions is an advantage to solve the flow around the turbine with certain ease, but this only partly true. Indeed, the staticity of the geometry brings the benefit of keeping the mesh static. However, the downside of the case is that, from the point of view of physics, the resolution is much harder due to the nature of the blade design: the blade is twisted from root to tip (in most cases more than 15°). This means that either at the root or the tip, no matter the pitch angle of the blade, a region of the blade will always be in full stall. That is precisely the difficulty.

¹³it depends on the installation strategy

At this point, the Adaptive Mesh Refinement (AMR) plays an important role. AMR represents a flexible meshing paradigm that can theoretically adapt the mesh to the desired condition iteration by iteration. In other words, the mesh is not conceived as a time-wise static element. Instead, it is a dynamic entity. Despite the fact that AMR can apply to both structured and unstructured concepts, the application to structured meshes is far easier, and in the context of this thesis only structured meshes will be refined. Strictly speaking, AMR would require to refine the mesh during the simulation, but this is not the case of this thesis. The AMR method developed at CTTC and incorporated into TernoFluids (check the work done by my co-worker Oscar Antepará in [59] and [60]) was only employed in preprocess time in order to achieve the right level of refinement. This is adding an extra degree of control with respect the meshes provided by ICEM-ANSYS.

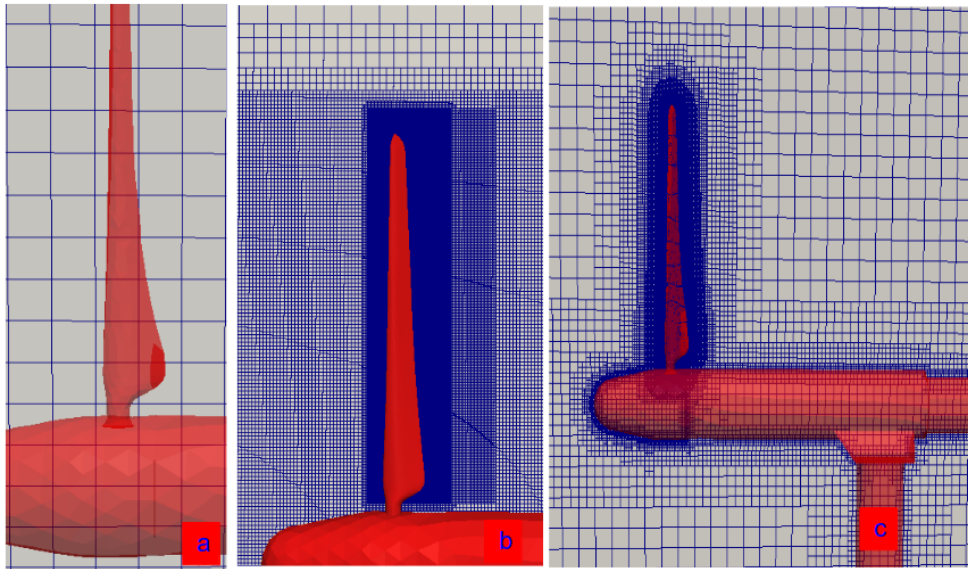


Figure 4.30: Evolution of the mesh using AMR: a) initial, b) final, employing only boxes to define control volume sizes, c) final, employing the distance function criterion.

The application of AMR in the context of this thesis is twofold. The first application is in the process of achieving a fine-controlled refined region in the IBM cases of the rotating turbines. This is of special interest when avoiding the RAM limitations presented before (recheck Figure 4.5) because ICEM is not designed to be run in clusters

and even with the most powerful computer at CTTC the memory was collapsed. The second application is connected to the present topic: the progressive refinement based on the distance from the cell to the wind turbine. A tool was specifically designed for the Parked Case scenario in order to take the benefits of employing a canonic structured mesh of any degree of refinement and automatically convert it into a glove-like mesh for the static turbine to be simulated. Figure 4.30a presents the simple initial mesh; b) shows the refinement by boxes fitting the different elements and the expected fluid-dynamic behavior of the region; and c) depicts the second level of precision where the mesh is refined as a function of the distance to the blade, which enables the saving of a numerous group of control volumes with respect to option b).

The highly-adapted mesh is built step-by-step from the uniform and structured original mesh. Since the AMR code is fully parallel, the mesh can be loaded by TermoFluids either in a cluster or on a single CPU. Once it is loaded, the distance function is computed and the mesh is refined according to it. A previous criterion based on the distance to the blade, tower, nacelle, etc. is specified in order to define the size of the control volume at each section. The combination of distance-based and box-based approaches is feasible and useful. In this way, a first level of refinement is achieved, splitting some control volumes in eight *child* control volumes. The second iteration is repeated in the same way: following the external meshing criterion to fulfill the right level of detail. Apart from this, additional logical criteria are taken within the AMR framework: the maximum gradient on refinement levels between two adjacent control volumes is one, i.e. the volume ratio from a cell to any neighbor is limited to eight. Otherwise, allowing more than one level, would lead to a volume ratio of sixty-four which is too much. This process goes on until all the requirements are fulfilled and then, it stops. The initial and final meshes are seen in Figure 4.30 a) and c); a cross section of the blade can be seen also in the upper part of Figure 4.31.

Figure 4.31 compares the section at a given part of the blade with both the mesh alone (above) and the CFD (below). In fact, as the reader can observe, the resolution is such, that the perceived airfoil shape of the first level of STL geometry refinement is actually chopped rather than smooth. Extra degree of STL refinement was introduced to avoid such deficient geometry, however, the problems that we faced were far more complex than those. Despite the quality of the simulation might seem good in Figure 4.31, the reader must be warned that this only shows part the flow field and other regions presented problems that could be clearly identified.

As exposed in the introduction of the current subsection, the interest of Parked Cases is precisely to study the situation in the presence of extreme winds. This fact is pushing

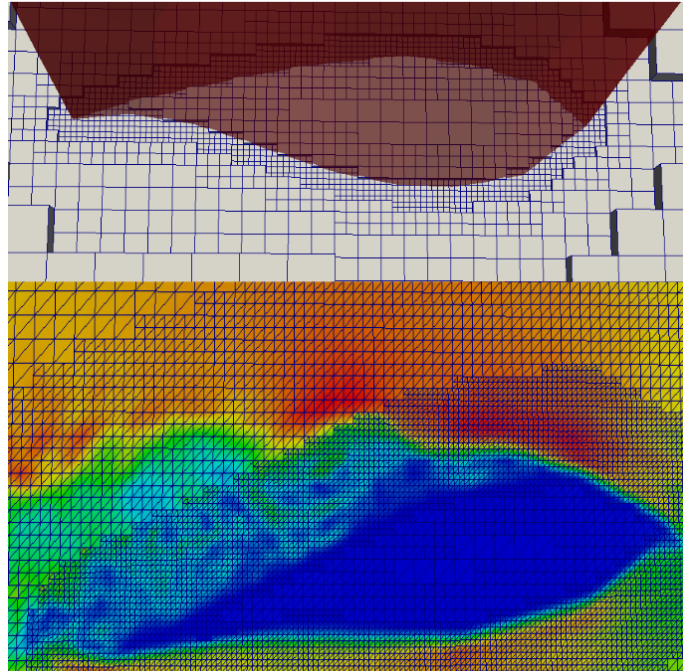


Figure 4.31: Detail of the AMR meshes on the Parked conditions case. Top: mesh detail. Bottom: instantaneous velocity field.

upwards the Reynolds number which becomes the main problem for AMR and IBM. The volume transition (ratio of 8) from a given cell to its neighbor when they have different levels of refinement seem to be too high for the arrangement of TermoFluids and the employed numerical schemes. For instance, the work of my co-worker Federico Favre on the field of AMR and IBM applied to the spoiler in a NACA airfoil, [36], concluded that when Reynolds numbers are limited to several tens of thousands (say 50,000), the applicability of such tandem is successful. However, similar cases with Reynolds up to the million were clearly proven to be unattainable by this strategy. Extra work must be done in this direction to upgrade the models and make them valid for typical wind turbine operating Reynolds.

Some more insights were obtained into this issue when an artificial case was set in accordance to [61]. This test was used to determine the reasons of why we do observe the spurious velocities with AMR. The test was, once again, carried out by Federico

Favre and it consisted in the pure convection of a big vortex, moving from the left hand side to the right hand side of Figure 4.32. In the middle of the domain there is an abrupt change in the size of the neighboring control volumes employing different meshing options: typical AMR, smoothed AMR and unstructured. For all the cases, spurious effects arise when the vortex reaches the transition: not only the vortex loses its shape but also a rebound effect is observed going backwards (instants t_3 and t_4). Such effects are inexistent when the vortex is advected in the two control meshes: when control volume size is kept uniform and also when the unstructured mesh gets a gradual growth of the control volumes. This study clarifies the source of the problems experienced in the Parked Conditions: it is not the interaction AMR–IBM or the high Reynolds (although it is relevant). The crucial aspect is the numerical schemes and the TermoFluids treatment in front of significant volume change between two neighbor cells.

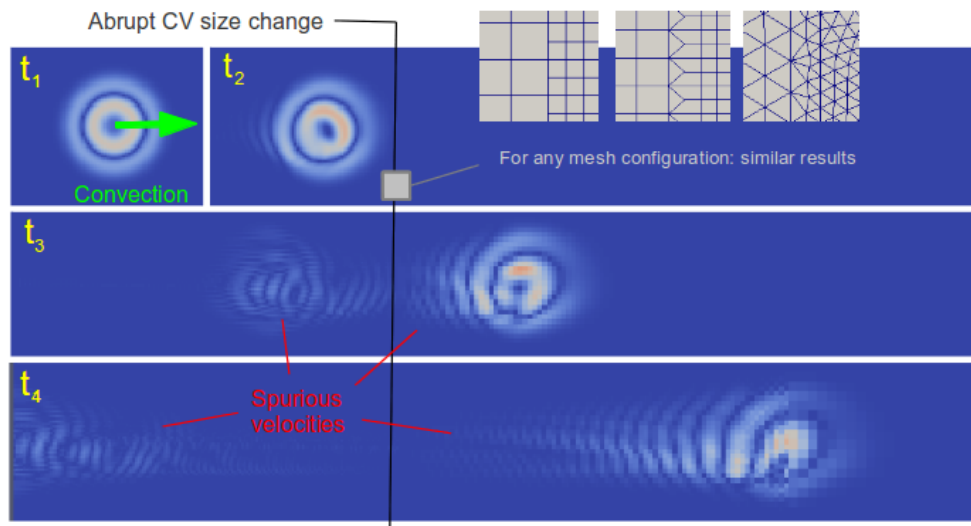


Figure 4.32: Vortex test to discriminate the source of the errors experienced under the AMR approach.

As mentioned above, the work from my co-worker Oscar Antepará on the field of AMR has been successfully applied to many cases, but the question is then: why it does not work in Parked cases and the vortex convection? A possible explanation relies on the fact that AMR is applied in the present cases with the *static* approach while in [59]

and [60], the refinement operates in a constant basis, following the physical behaviour. We believe that, leaving the high Reynolds issue aside and enabling truly dynamic AMR might help on the avoidance of the spurious numerical effects.

In the literature there are a couple of papers studying the standstill conditions of a wind turbine: [62] deals with the MEXICO case, and [63] was carried out during the task 3.1 of the KNOW-BLADE project. We have considered the possibility to perform similar simulations employing the technique mastered in the flow around airfoils but for the 3D case. First preliminar results were obtained and future work in this direction will be carried out. The incorporation of the Wall Model LES methods already being applied to the periodic 2D airfoil cases such as [16] might also be of help in this task.

4.6 Conclusion

In this Chapter we began with an ample discussion about the state of the art of CFD simulation, focusing on the wind turbine level and emphasizing on 3D studies. We have aimed at three different approaches for solving the flow around a wind turbine: Coriolis, IBM and Sliding Meshes. The implementation of Coriolis (Non-Inertial Reference Frame) led us to a situation where unexpected results were obtained: strong and unphysical gradients were detected in the far-field and they seem to be the cause of the simulations divergences. The second approach consisted in the employment of two different Immersed Boundary Methods in the MEXICO and NREL cases: the smoothed source term modification and the Direct Forcing approach. Reasonably good results were obtained for the MEXICO case at 15 m/s, but additional work must be done to contrast this information further. The third approach was based on the Sliding Mesh technique and qualitatively good results were obtained when externally limiting the maximum velocity of the field. Once again, an extra effort must be done in this direction as to clarify which type of boundary conditions might solve the problem. Now, besides the wind turbine rotating cases, an incursion was done in the Parked Conditions case, approaching it from the IBM perspective and employing AMR-based meshes for the automatization of the process.

References

- [1] M.O.L. Handsten et. al. State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in Aerospace Science*, pages 285–330, 2006.

- [2] L.J. Vermeer et. al. Wind turbine wake aerodynamics. *Progress in Aerospace Science*, 39:467–510, 2003.
- [3] Tongguang Wang et. al. A brief review on wind turbine aerodynamics. *Energy policy*, pages 12–34, 2012.
- [4] Aaron Miller et. al. Review of computer-aided numerical simulation in Wind Energy. *Renewable and Sustainable Energy Reviews*, 25:122–134, 2013.
- [5] Olof Grundestam et. al. Direct numerical simulations of rotating turbulent channel flow. *Journal of Fluid Mechanics*, 598:177–199, 2008.
- [6] Schreck S. et. al. Rotation Augmentation Disparitiens in the MEXICO and UAE Phase VI Experiments. 2010.
- [7] Srinivas Guntur and Niels N. Sørensen. A study on rotational augmentation using CFD analysis of flow in the inboard region of the MEXICO rotor blades. *Wind Energy*, 18:745–756, 2015.
- [8] Iván Herráez et. al. Insight into Rotational Effects on a wind turbine blade using Navier-Stokes Computations. *Energies*, 7:6798–6822, 2014.
- [9] Jeppe Johansen and Niels N. Sørensen. Aerofoil Characteristics from 3D CFD Rotor Computations. *Wind Energy*, 7:283–294, 2004.
- [10] Horia Dumitrescu et. al. Rotational Effects on the Boundary Layer Flow in WT. *AIAA Journal*, pages 408–411, 2013.
- [11] W.A. Timmer and R.P.J.O.M. van Rooij. Summary of the delft university wind trubine dedicated airfoils. *AIAA*, 352, 2003.
- [12] Baez A. et. al. Direct numerical simulation of the turbulent flow around a NACA 0012 airfoil at different angles of attack. *Proceedings of 24th Conference on Paralled CFD, Barcelona, Spain.*, 2011.
- [13] I. Rodríguez et. al. Direct numerical simulation of a NACA0012 in full stall. *International Journal of Heat and Fluid Flow*, 43:194–203, 2013.
- [14] Julián Ernesto Jaramillo Ibarra. *Suitability of Different RANS Models in the Description of Turbulent Forced Convection Flows. Application to Air Curtains.* PhD thesis, Universitat Politècnica de Catalunya, Spain, 2008.

- [15] Calafell J. et. al. On the Large-Eddy Simulation modelling of wind turbine dedicated airfoils at high Reynolds numbers. *Turbulence, Heat and Mass Transfer* 7, 2012.
- [16] Calafell J. et. al. A Dynamic Wall Model for Large Eddy Simulation on Unstructured Meshes. Application to Wind Turbine Dedicated Airfoils. *Turbulence, Heat and Mass Transfer* 8, 2015.
- [17] M.M. Hand et. al. Unsteady aerodynamics experiment Phase VI: wind tunnel test configuration and available data campaigns. *NREL*, 2001.
- [18] <http://www.mexnext.org/>.
- [19] Yuwei Li et. al. Dynamic overset CFD simulation of Wind Turbine aerodynamics. *Renewable Energy*, pages 285–298, 2012.
- [20] Th. Lutz et. al. Near wake studies of the MEXICO Rotor. 2010.
- [21] Réthoré P.-E. et. al. CFD model of the MEXICO wind tunnel. 2010.
- [22] A. Bechmann et. al. CFD simulations of the MEXICO rotor. *Wind Energy*, 14:644–689, 2011.
- [23] Niels N. Sørensen et. al. Near wake RANS predictions of the wake behind the MEXICO rotor in axial and yawed flow conditions. *Energy policy*, pages 12–34, 2013.
- [24] Balcázar N. et. al. A finite-volume/level-set method for simulating two-phase flows on unstructured grids. *International journal of multiphase flow*, 64:55–72, 2014.
- [25] FX Trias et. al. Direct numerical simulations of two-and three-dimensional turbulent natural convection flows in a differentially heated cavity of aspect ratio 4. *Journal of Fluid Mechanics*, 586:259–293, 2007.
- [26] Pont A. et. al. Direct numerical simulations of two-and three-dimensional turbulent natural convection flows in a differentially heated cavity of aspect ratio 4. *Tribology international*, 105:1–14, 2016.
- [27] Jordi Ventosa Molina. *Numerical simulation of turbulent diffusion flames using flamelet models on unstructured meshes*. PhD thesis, Universitat Politècnica de Catalunya, Spain, 2015.

- [28] CTTC. Introduction to Fractional Step Method. <http://projecte-hermes.upc.edu/Optatives/B%20-%20Analysis%20of%20thermal%20and%20fluid%20dynamic%20issues%20in%20industrial%20and%20aeronautical%20systems%20and%20equipment/FSM.pdf>.
- [29] Oriol Lehmkhul Barba. *Numerical resolution of turbulent flows on complex geometries*. PhD thesis, Universitat Politècnica de Catalunya, Spain, 2012.
- [30] L. Mazzei T.J.R. Hughes and K.E. Jansen. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, 3:47–59, 2000.
- [31] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turbulence and Combustion*, 62:183–200, 1999.
- [32] Hachicha A. et. al. Numerical simulation of wind flow around a parabolog through solar collector. *Applied Energy*, 107:426–437, 2013.
- [33] G. Colomer et. al. Proyecto de análisis del comportamiento térmico-fluídico en elementos de un aerogenerador: simulación numérica, validación experimental y diseño de nuevos aerogeneradores. 2011.
- [34] I. Rodríguez et al. Direct numerical simulation of the flow over a sphere at $Re = 3700$. *Journal of Fluid Mechanics*, 679:263–287, 2011.
- [35] WZ Shen WJ Zhu and JN Sørensen. Numerical Investigation of Flow Control Feasibility with a Trailing Edge Flap . *Journal of Physics: Conference Series. TORQUE 2014*, 524.
- [36] F. Favre et. al. On the fast transient spoiler deployment in a NACA0012 profile using LES techniques combined with AMR and IMB methods. *11th World Congress on Computational Mechanics, WCCM 2014, 5th European Conference on Computational Mechanics, ECCM 2014 and 6th European Conference on Computational Fluid Dynamics, ECFD 2014. Barcelona, Spain., 2014*.
- [37] Manuel Werner. Blade Geometry and CAD Model of the 5MW Reference Turbine. *IAG*, 2012.
- [38] K. Boorsma and J.G. Schepers. Rotor experiments in controlled conditions continued: New Mexico. *Journal of Physics: Conference Series*, 753, 2016.

- [39] ANSYS. ANSYS ICEM CFD User's Manual. <http://148.204.81.206/Ansys/150/ANSYS%20ICEM%20CFD%20Users%20Manual.pdf>, 2013.
- [40] Joao P. et. al. Adaptive mesh finite-volume calculation of 2D lid-cavity corner vortices. *Journal of computational physics*, 243:365–381, 2013.
- [41] Loïc Maréchal. Advances in octree-based all-hexahedral mesh generation: handling sharp features. 2010.
- [42] Borja Plaza et. al. Comparison of BEM and CFD results for MEXICO rotor aerodynamics.
- [43] Jang-Oh Mo et. al. Effects of wind speed changes on wake instability of a wind turbine in a virtual wind tunnel using LES.
- [44] A. Sohankar et. al. Low-Reynolds-Number flow around a square cylinder at incidence: study of blockage, onset of vortex shedding and outlet boundary conditions. *International journal for numerical methods in fluids*, 26:39–56, 1998.
- [45] J. Jokman et. al. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. . *NREL*, <http://www.nrel.gov/docs/fy09osti/38060.pdf>, 2009.
- [46] Shintaro Takeuchi et. al. Study of Solid-Fluid Interaction in Body-Fixed Non-Inertial Frame of Reference. *Journal of Fluid Science and Technology*, 1, 2005.
- [47] Charles S. Peskins. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 2:252–271, 1972.
- [48] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.
- [49] T. Gornak. A goal oriented survey on immersed boundary methods. *Fraunhofer ITWM*, 2013.
- [50] E. A. Fadlun et. al. Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations. *Journal of Computational Physics.*, 161:35–60, 2000.
- [51] Antoni Calderer et. al. Coupled fluid-structure interaction simulation of floating offshore wind turbines and waves: a large eddy simulation approach. *Journal of Physics: conference series. Torque 2014.*, 524, 2014.

- [52] Xiaolei Yang et. al. Wind turbine wake interactions at field scale: An LES study of the SWiFT facility. *Journal of Physics: conference series. Torque 2014.*, 524, 2014.
- [53] F. Favre et. al. Numerical simulations of conjugate convection combined with surface thermal radiation using an Immersed-Boundary Method. *In proceedings of the 7th European Thermal-Sciences Conference (Eurotherm2016), Krakow, Poland.*, 2014.
- [54] F. Favre D. Martinez A. Oliva D. E. Aljure, O. Lehmkuhl. On the IBM approximation for the wheel aerodynamic simulation. *First International Conference in Numerical and Experimental Aerodynamics of Road Vehicles and Trains. France, Bordeaux, 2014*, 2014.
- [55] O. Lehmkuhl C. D. Pérez-Segarra A. Oliva Muela J., D. M. Valdivieso. Fluid-Structure Interaction of a Reed Type Valve Subjected to Piston Displacement. *27th International Conference on Parallel Computational Fluid Dynamics (PAR-CFD2015). UK, Scotland, Glasgow 2015*, 2015.
- [56] Olga Estruch et al. Fluid-Structure Interaction of a Reed Type Valve Subjected to Piston Displacement. *Compressor Engineering Refrigeration and Air Conditioning High Performance Buildings. US, Indiana, Purdue, 2014*, 2014.
- [57] Isam Janajreh et. al. Numerical simulation of tower rotor interaction for downwind wind turbine. 2010.
- [58] Frederik Zahle et. al. Wind Turbine Rotor-Tower Interaction Using an Incompressible Overset Grid method. *Wind Energy*, 12:594–619, 2009.
- [59] Antepara O. et. al. Parallel adaptive mesh refinement for large-eddy simulations of turbulent flows. *Computers and Fluids*, 110:48–61, 2015.
- [60] Antepara O. et. al. Parallel adaptive mesh refinement of turbulent flow around simplified car model using an immerse boundary method. *11th World Congress on Computational Mechanics, WCCM 2014, 5th European Conference on Computational Mechanics, ECCM 2014 and 6th European Conference on Computational Fluid Dynamics, ECFD 2014. Barcelona, Spain.*, pages 2603–2611, 2014.
- [61] Johan C. Kok. A symmetry and dispersion-relation preserving high-order scheme for aeroacoustics and aerodynamics. *European Conference on Computational Fluid Dynamics ECCOMAS CFD. TU Delft, The Netherlands, 2006*.

- [62] Ye Zhang et. al. Comparison of CFD simulations to non-rotating MEXICO blades experiment in the LTT wind tunnel of TUDelft. *The science of making torque from wind 2014*, 524, 2014.
- [63] Niels N. Sørensen et. al. CFD computations of wind turbine blade loads during standstill operation KNOW-BLADE TASK 3.1 Report. *Energy policy*, 2004.

Conclusions and future research

In this chapter, a short explanation will be offered for the most important conclusions of the present thesis. Additionally and as usual, the proposed future research lines will be exposed.

5.1 Concluding remarks

An ample introduction was provided in Chapter 1, covering aspects about the context of energy as a concept, the specifics of the wind energy technology, the current status of the wind energy market and the state of the art of the aerodynamic issues. We saw there that the rate at which the wind energy is growing annually is nearly the 20 % and the most common technology exploited is the one of Horizontal Axis Wind Turbines (HAWTs). We also commented that aerodynamics is a crucial aspect of wind turbines, which is not fully understood due to the high unsteady nature – in part because of turbulence.

Chapter 2 defines the geometrical basis for the remaining content of this thesis. The airfoil digital conceptualization is addressed first. This platform enabled us to both generate automatic blades and also to study the pressure distribution along the blade when using the Immersed Boundary Method (IBM) in the 3D CFD simulations. Section 2.2.1 detailed the process of the automatic blade generation (AutoBlade code) which allowed us to produce reliable CAD models with ease, without human errors and very quickly. The automation of the blade CAD generation facilitated us to produce STL objects that were employed in the definition and management of the wind turbine components for posterior CFD simulations. Finally, the bottom line for the IBM simulations

was presented: the ability to compute the distance function from all the mesh points to the turbine when defined by a group of STL objects.

The Blade Element Momentum (BEM) method was previously implemented in TermoFluids but it was improved in several ways as shown in Chapter 3. For instance, certain operations that the user had to do in pre-process were incorporated into the algorithm and preprocessing. Moreover, a friendlier user-interface was built for the sake of better understanding of the delivered results. An extra level of technical support was added to BEM, consisting in the inclusion of the electric generator's behavior. This meant to give birth to the AeroElectric code which is able to predict the angular rotation of the machine at the demanded working points. The Weibull probabilistic distribution was added to the AeroElectric code to enable one step further in the predictions: the Annual Energy Production (AEP). Finally, the engineering models exposed till that point were coupled with the CTTC optimization library in order to improve (in an automated way) the blade geometry by means of the AEP maximization.

Chapter 4 started with a discussion about the available CFD methods applied to wind energy, with special emphasis in the airfoil and wind turbine levels. The airfoil is the smallest unit with aerodynamic meaning *per se*, however, this thesis focuses more into the wind turbine scale. Three different approaches have been used to analyze the aerodynamic behaviour of a full wind turbine: the Non-Inertial Reference Frame, the Immersed Boundary Method and the Sliding Meshes technique. Coriolis and centrifugal fictitious forces were incorporated as a source term in order to study the rotor with the first approach. We could not find out a way to make this strategy work: we encountered divergences that we believe are caused by large velocity gradients in the far zones of the domain. The second approach, governed by the Immersed Boundary Method, required a huge effort in the geometrical computation, forcing us to come up with a series of heuristics to make the distance recomputation fast enough. The improvements were palpable and the simulations could be done. Although reasonably good agreement was achieved for certain working regimes in the MEXICO case, further investigation is required across different (wind) conditions. The last strategy to be tested was the Sliding Meshes. Qualitative good results were observed in the Enair's downwind 40 kW turbine, including the tower interaction. However, an extra effort must be done with regards the boundary conditions in the ALE formulation (Arbitrary Lagrangian Eulerian), as they are believed to be the cause of the high velocities experienced near the blades. Finally, a short incursion was done in the field of the Parked conditions case, testing the IBM with and without the Adaptive Mesh Refinement technique.

Within the context of this thesis, an industrial project was carried out in parallel,

allowing us to test in real conditions all the methods and techniques exposed along this text. The details of the project are available in Appendix A, which is divided in three parts. The reverse engineering of a 3 kW blade was performed in first place. The design of a 10 kW turbine was done next, which has been already built and tested successfully, being at the time of writing, in the certification process. And finally, a 40 kW turbine was designed based on the previous inputs.

Without any doubt, all the work performed in this thesis has been valuable for the organization supporting this work (CTTC). The contributions to the scientific community are believed to be the next: 1) the digital conceptualization of an airfoil and the robust identification procedure shown in Chapter 2; 2) the geometry treatment and implementation carried out in the AutoBlade code, where a blade is designed and embedded in a 3D meshing software even for flat trailing edge blades; 3) the AeroElectric simplicity and code capabilities, specially when merged with the genetic optimization tools as shown in Chapter 3; and finally, 4) the idea that it is feasible to apply IBM to the case of a 3D full wind turbine, specially as a future strategy.

Even in the first word of this thesis title (*towards*), the author already warned the reader about the difficulty of the task at hand. The results presented in Chapter 4 must be contextualized and always understood as a work in progress rather than a categorical answer, or a miraculous approach to solve the 3D flow around a full wind turbine employing LES. According to an estimation based in Komogorov scales and boundary layer theory, we believe that 6,000 million control volumes might be enough to perform a good LES simulation for the case of a current industrial size wind turbines. If Moore's Law keeps with the expected future projection, there will be a time in the near future where the explored CFD techniques (specially IBM) will suffice to analyze a wind turbine with detail and precision.

5.2 Future research

The ultimate aim of this thesis is to analyze wind turbines aerodynamics to understand the underlying physics in order to apply those concepts to improve the conditions and reduce the Levelized Cost of Energy. To that aim, many different research lines could be articulated and we believe that these three strategies might be specially relevant.

- 1 - **Airfoil.** The comprehension of the dynamic stall and the rotational augmentation effects is crucial for the design of wind turbines. We contend that developing a robust and automatic approach to study these effects might be of extreme

interest for the sector. Now, the simulations performed in this field, require tens of human hours only for meshing and post-processing stages. We would like to apply the AutoBlade know-how to make the meshing much more comfortable. Furthermore, an extra effort will be done to incorporate several other strategies currently ongoing at CTTC to build a virtual platform for airfoils. For instance, we find appealing the idea of the Wall Model LES combined with the automatic geometrical description and meshing. Ideally, the process would depart from the 2D coordinates of an airfoil, generate the geometry and the associated mesh, incorporate the LES Wall Models and deliver the chart Lift coefficient vs. Angle of Attack.

- 2 - **Wind turbine engineering models.** The idea is to develop the BEM further in order to achieve a more complete platform. A first block could be the idea of study the effect of tilt, yaw and cone angles. Additionally, a tool to compute the loads could be implemented and coupled with cost models accounting for blade manufacturing or for the tower expenses in function of the height and the materials. In this way, we could optimize the wind turbine design based on the Levelized Cost of Energy instead of the Annual Energy Production. In this way, the aerodynamic design and the electric system could be coupled not only with the structural part but also with the associated costs including those associated to human resources.
- 3 - **Wind turbine CFD methods.** Assuming that the computing power is going to keep growing, the idea of simulating a whole wind turbine using CFD would become feasible at a given point in time. Three strategies will be studied: 1) for the Non-Inertial Reference Frame approach, some additional tests could be carried out to explore the reasons of the divergences observed. For instance, radius-varying dumping could be introduced by means of setting a progressively higher density towards the zones far from the rotation center. 2) the IBM approach might become an interesting tool in the years to come in order to study not only the 3D effects but also the dynamic pitch and yaw. The proposal here could be focused on the mesh refinement and the inclusion of Adaptive Mesh Refinement (AMR). 3) the Sliding Meshes approach is believed to have the potential to be much more accurate than what we have shown so far. Our idea is to explore the boundary conditions issue in order to solve the unphysical velocities experienced and also incorporate the LES Wall Model simulations to account for the thin boundary layer around the blades.

Enair's project

The CTTC's spin-off, Termo Fluids S. L., is a company born to be the bridge between cutting-edge science and the market. Termo Fluids is proactively looking for partners and projects in the field of technical consulting. In the context of Barcelona's EWEA Annual Event in 2014, Termo Fluids hosted a stand and actively contacts in the field of Wind Energy. The author had the pleasure to participate in the fair and it was there when the connection TF-Enair was made.

A.1 Introduction

Termo Fluids S. L., in collaboration with *Ingenia* and in agreement with Enair S. L., started a joint venture to design a 40 kW downwind turbine. The project was titled as follows, *Cost efficient Wind Turbine of 40 kW of rated capacity* (ECIWIND), and was co-funded by the European Commission under the scheme: H2020-SMEINST-2-2014.

The implication of TermoFluids in the project began in April 2015 and finished in December 2016. Up to date, there are still phases of the project to be completed, being Enair fully in charge until the very end of the project.

At this point in the document, the author would like to thank Enair once again for agreeing to publish the work done to complement the content of this thesis. Enair agreed on the fact that all the information regarding the aerodynamic design can be disclosed except for the specific details of the airfoil shape employed in the design of the blades. Thus, in order to comply with the agreement, the airfoil information will be omitted.

Termo Fluids' duties with respect to the project can be synthesized in three steps: 1)

analyze the already existing 3 kW upwind turbine in order to understand its behavior and use the insight to develop the next turbines; 2) design a 10 kW downwind turbine as a first step towards the final machine; 3) design the 40 kW downwind turbine. These steps are the skeleton of the appendix's structure. Three sections are provided together with a short conclusion at the end.

A.2 Original turbine: 3 kW

Enair already had a 3 kW upwind turbine working for some years. The idea was to depart from that working machine to design the 10 and 40 kW machine taking advantage from the already successful turbine.

Therefore, the first step consisted in analyzing the already existing turbine. Enair send us one of the blades to be studied and we performed a reverse engineering procedure. The 3D scanning of the blade was subcontracted. The *Centre de Visió per Computador*¹ at the *Universitat Autònoma de Barcelona* was the chosen organism to perform the scanning. They employed a *Konica Minolta Vivid 9i*² in order to obtain a computational model of the blade. See Figure A.1 for visual information about the process.

The precision of the scanning service is far more than acceptable for the task at hand: it can capture structures in the order of microns. In the present case, obtaining a geometry with precision of millimeters is enough. Therefore, we demanded a CAD model (STL based) with a lesser degree of detail than what it was actually possible. In

¹in English: Center for Computing Vision. More information at: <http://www.cvc.uab.es/>

²more info in <http://sct.uab.cat/sti/es/content/escaneo-3d>

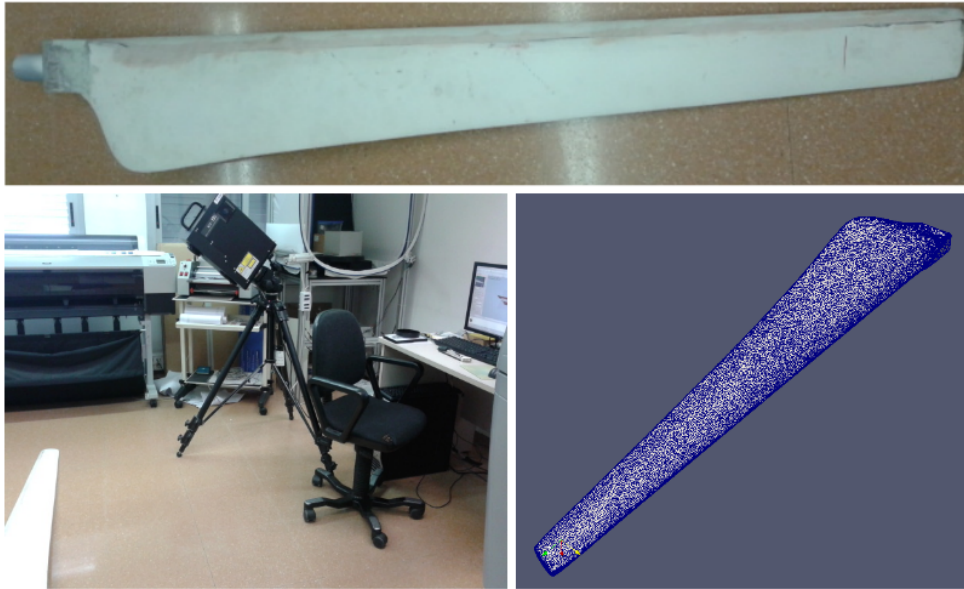


Figure A.1: Scanning process of the 3 kW blade. Top: blade itself. Left: blade scanning. Right: blade CAD model for the scanning software.

this way, the size of the file could be downscaled and kept within acceptable limits³. With this computational model in STL format, we run the cutting tools of Chapter 2 to obtain a number of sections that were analyzed in detail afterwards. In this way, it was possible to obtain the required data to represent the blade and rebuild it with the AutoBlade software developed. See Figure A.2.

At this point, the complete geometry of the 3 kW blade was available, including the details of the airfoils composing the stations along the blade axis. Once the geometry is defined, the next step is to investigate the aerodynamic behaviour of the extracted airfoils. As a first approximation, massive XFOIL⁴ computations were run for each of the (20) sections employed in the model. Results for the simulations of the corresponding Reynolds numbers (roughly 400,000) at a good range of Angles of Attack are shown in Table A.1.

As commented in the introduction of Chapter 4, the simulation of a 2D airfoil is

³notice that the original STL file contained more than 2 million triangles

⁴a software to study inviscid flows around airfoils in subsonic conditions

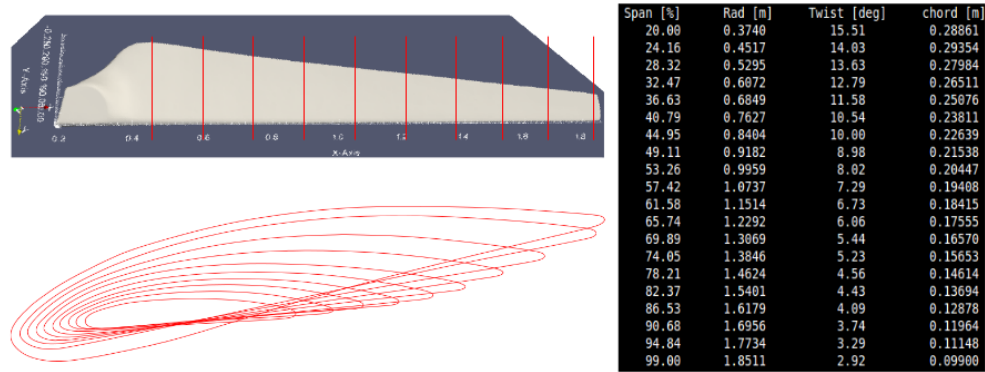


Figure A.2: Reverse engineering of the 3 kW blade CAD model delivered by the scanning process. Top-left: blade in STL format. Bottom-left: geometrical cuts along the blade seen from the axis perspective. Right: extrated values representing the blade model.

	Enair 0		Enair 1		Enair 2		Enair 3		Enair 4		Enair 5		Enair 6		Enair 7		Enair 8		Enair 9		
AoA	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	
-5	0.17	0.015	0.24	0.018	0.25	0.018	0.27	0.017	0.26	0.016	0.19	0.016	-0.055	0.023	0.23	0.016	0.29	0.015	0.21	0.014	
-2.5	0.44	0.013	0.50	0.014	0.51	0.014	0.54	0.014	0.51	0.013	0.45	0.013	0.33	0.017	0.48	0.013	0.51	0.013	0.47	0.012	
0	0.69	0.012	0.76	0.013	0.77	0.012	0.79	0.012	0.76	0.012	0.69	0.012	0.73	0.012	0.68	0.011	0.74	0.012	0.72	0.012	
2.5	0.89	0.012	0.94	0.011	1.08	0.011	1.10	0.011	1.11	0.011	1.05	0.011	1.047	0.010	0.72	0.010	1.03	0.011	1.01	0.011	
5	1.04	0.014	1.19	0.012	1.22	0.012	1.27	0.012	1.25	0.012	1.17	0.012	1.19	0.013	0.69	0.014	1.21	0.013	1.21	0.013	
7.5	1.21	0.020	1.34	0.017	1.37	0.016	1.45	0.015	1.39	0.015	1.31	0.016	1.26	0.019	0.67	0.021	1.38	0.016	1.41	0.016	
10	1.21	0.038	1.43	0.026	1.48	0.023	1.50	0.024	1.45	0.026	1.43	0.024	1.20	0.028	0.64	0.032	1.48	0.024	1.55	0.022	
12.5	0.89	0.079	1.52	0.040	1.55	0.037	1.38	0.050	1.38	0.049	1.45	0.042	1.14	0.037	0.61	0.042	1.55	0.040	1.64	0.035	
15	0.56	0.119	1.43	0.072	1.51	0.065	1.27	0.076	1.31	0.073	1.44	0.063	1.082	0.046	0.58	0.051	1.49	0.071	1.65	0.060	
17.5	-	-	-	-	-	-	-	-	-	-	1.43	0.084	-	-	-	-	-	-	1.38	0.12	
20	-	-	-	-	-	-	-	-	-	-	1.42	0.10	-	-	-	-	-	-	1.10	0.18	
22.5	-	-	-	-	-	-	-	-	-	-	1.41	0.12	-	-	-	-	-	-	-	-	
25	-	-	-	-	-	-	-	-	-	-	1.40	0.14	-	-	-	-	-	-	-	-	
	Enair 10		Enair 11		Enair 12		Enair 13		Enair 14		Enair 15		Enair 16		Enair 17		Enair 18		Enair 19		
AoA	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	Cl	Cd	
-5	0.24	0.014	0.21	0.014	0.27	0.014	0.18	0.014	0.21	0.015	0.17	0.015	0.097	0.016	0.10	0.019	-0.1	0.021	-0.1	0.022	
-2.5	0.46	0.012	0.48	0.012	0.54	0.011	0.45	0.012	0.47	0.012	0.46	0.013	0.40	0.013	0.47	0.014	0.2	0.019	0.24	0.016	
0	0.61	0.011	0.69	0.0090	0.77	0.009	0.63	0.009	0.72	0.010	0.72	0.010	0.68	0.011	0.74	0.011	0.51	0.015	0.57	0.011	
2.5	0.91	0.012	0.98	0.011	1.42	0.011	1.03	0.010	1.07	0.010	1.05	0.009	0.98	0.010	1.05	0.010	0.97	0.012	0.91	0.013	
5	1.12	0.014	1.15	0.013	2.05	0.013	1.18	0.012	1.21	0.012	1.18	0.012	1.09	0.012	1.15	0.014	1.11	0.016	1.01	0.019	
7.5	1.32	0.018	1.33	0.017	2.67	0.015	1.33	0.016	1.35	0.016	1.23	0.021	1.17	0.018	1.27	0.022	1.21	0.024	1.20	0.024	
10	1.48	0.023	1.49	0.024	3.30	0.017	1.42	0.025	1.43	0.027	1.09	0.039	1.31	0.026	1.42	0.030	1.37	0.032	1.30	0.037	
12.5	1.57	0.035	1.55	0.039	3.93	0.019	1.45	0.042	1.38	0.052	0.95	0.057	1.27	0.049	1.39	0.054	1.34	0.053	1.29	0.073	
15	1.10	0.10	1.36	0.083	4.55	0.021	1.40	0.072	1.28	0.081	0.81	0.075	1.16	0.076	1.35	0.077	1.24	0.080	1.19	0.11	
17.5	-	-	-	-	5.18	0.023	-	-	-	-	-	-	-	-	-	-	-	1.15	0.11	1.11	0.14
20	-	-	-	-	5.81	0.025	-	-	-	-	-	-	-	-	-	-	-	1.05	0.13	1.01	0.18
22.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.95	0.16	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.86	0.18	-	-

Table A.1: XFOIL results of the 20 arfoil sections extracted from the scanned 3 kW blade.

a complex issue. XFOIL simulations, as long as they are inviscid, are not meant to provide good results in stall conditions. RANS simulations can account for an extra degree of precision than XFOIL, but even in this case, the results must be taken with

caution near the stall region. Therefore, the XFOIL results were only trusted in their linear region and, for the sake of comparison, a series of LES simulations were run at several Angles of Attack in the case of the Enair10 airfoil (the one corresponding to the 50 % span of the 3 kW blade). The comparison of the LES and XFOIL simulations can be consulted in Table A.2.

AoA [°]	Cl -XFOIL [-]	Cl -CFD [-]	Absolute Error [-]	Relative Error [%]
0.0	0.619	0.576	0.0429	7.4
5.0	1.126	1.051	0.0742	7.1
10.0	1.482	1.387	0.0945	6.8
12.5	1.575	1.491	0.0849	5.7
15.0	1.101	1.577	0.4761	30.2

Table A.2: Comparison of the results between XFOIL and CFD for the midspan 3 kW blade airfoil (Enair10) for the Reynolds = 400,000.

As shown in Table A.2, XFOIL results approximately agree with CFD up to 12.5° where the stall is predicted by the CFD simulations. As expected, XFOIL results are only trustable in the linear region.

A.3 First turbine: 10 kW

As a midstep towards the design of the 40 kW turbine, a 10 kW prototype had to be designed, built and tested. The first designing option was the brute scaling of the 3 kW machine in order to adopt the corresponding size of 10 kW machine.

$$P = \frac{1}{2} C_p \rho \pi R^2 v^3 \quad (\text{A.1})$$

Equation A.1⁵ shows the concept behind the scaling factor. With the corresponding values, the approximate size of the blade is 4 m (8 m in diameter). An alternative to the (3 kW) scaling strategy wanted to be presented. This means that a new family of airfoils had to be selected to configure the blade section shape. Moreover, new chord length and twist laws will have to be determined although the size of the blade would be constant.

A vast number of airfoil shapes were considered. The left side of Figure A.3 shows some of them. The selection of the airfoil took place with the next criteria: a) maximum

⁵already exposed in Chapter 1 back in page 9

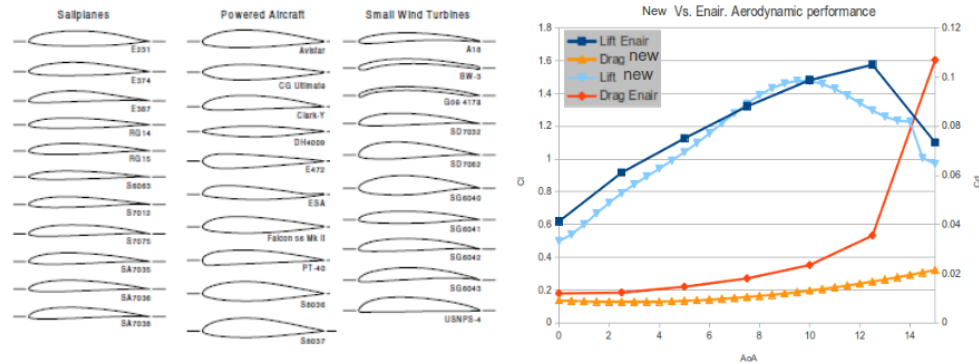


Figure A.3: Left: several airfoils that were analyzed before selecting the final one for the 10 kW alternative turbine. Right: comparison between the Enair10 and the new airfoil aerodynamic properties.

lift as high as possible; b) a smooth stall region; c) low sensitivity to dirt. Having these requirements in mind, an airfoil was chosen and its lift and drag curves across different Angles of Attack (AoA) are depicted in the right hand side of Figure A.3. Additionally, the properties of the Enair10 airfoil were overlapped in the chart to enable the comparison. As it is possible to see, when contrasting the selected airfoil with respect Enair10 airfoil we obtain the next: a) the maximum lift value is at the same level; b) it has a smoother stall region; c) is less sensitive to dirt accretion effect. Additionally, the maximum lift-to-drag ratio of the new airfoil is higher and the trailing edge is sharper, which reduces the component of noise associated to the shear layer⁶.

Several blade designs were compared, including blades with different maximum twist and chord lengths, and also the 3kW-just-scaled blade. The performance of the just-scaled blade in front of any of the multiple specific designs of the new approach is quite poor. Figure A.4 shows the performance comparison between the ideal blade – no limitation in maximum chord and twist – and the final design – a manufacturable option. As it is possible to see in Figure A.5, the final blade design fulfills the aesthetics criterion, keeps the maximum chord and twist within manufacturable limits and shows an acceptable performance (no big reduction with respect the ideal one).

Beyond the analysis executed to evaluate the performance of the turbine at different pitch angles and Tip Speed Ratios, some other studies were required to ensure the right

⁶because of the better accommodation of outboard and inboard boundary layers

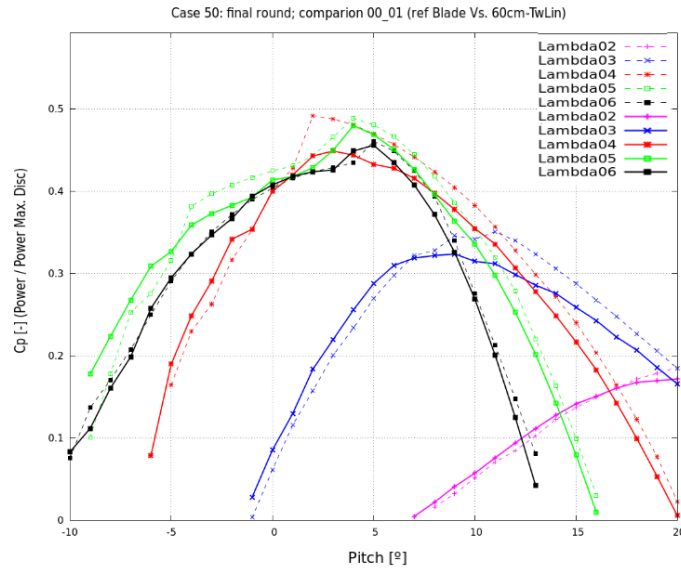


Figure A.4: BEM comparisons between the ideal blade (dotted lines) and the final selected one (solid lines) for a combination of pitch and Tip Speed Ratio (lambda) values.

functioning of the turbine. One of them is with respect the concept of *flag position*, which means finding the right pitch angle of the blade in order to get the minimum torque possible (ideally null) when the turbine remains in standstill conditions. The flag position angle is the one that places the blade in a position that *half* of it contributes to the torque towards one direction and the other *half* offers opposition⁷. Consult the study in Figure A.6. In the left side, it is shown the axial Torque as a function of the pitch angle⁸ and the wind speed. The best pitch angle to start up the machine is 75° and the flag position is 89°.

Due to the already fixed mechanical design of the pitching system, no more than 75° were available without redesigning the pitch system (and the corresponding inner pieces inside the hub). For operation purposes, the minimum pitch angle required was 4° while the required pitch for null torque in Parked conditions is 89°. Therefore, from the aerodynamical point of view, the desired pitch range should be 85°, but only 75° were available. A compromise was sought and a brake was installed in order to resist

⁷it is not physically the half of the blade, but the half of the torque.

⁸being 0° when the tip section chord is coplanar with the rotation plane

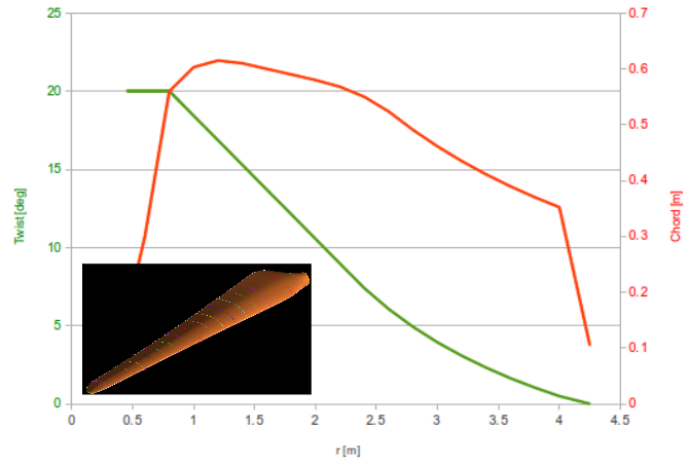


Figure A.5: Twist and chord laws of the final 10 kW turbine blade design together with an image of its the blade shape.

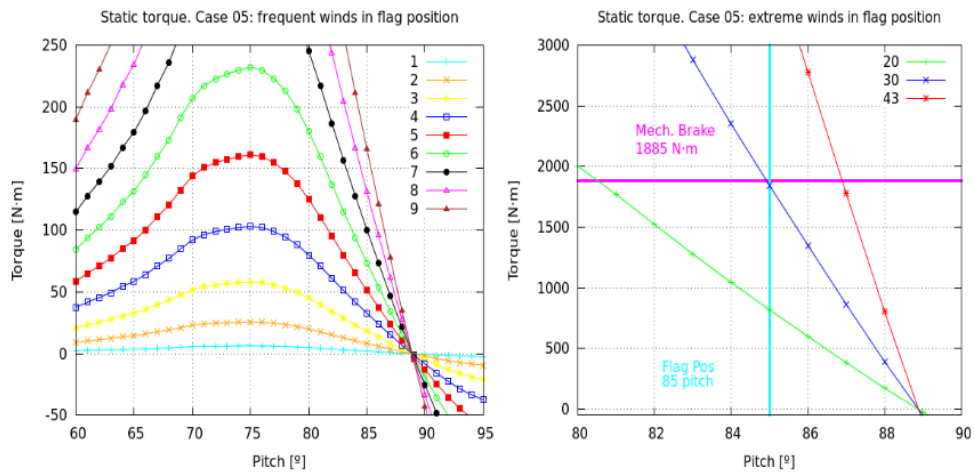


Figure A.6: Flag position analysis. Torque computations for the turbine in standstill conditions as a function of the wind speed in [m/s] and the pitch angle. Left: low winds. Right: extreme winds.

the torque in static conditions, given the fact that the null torque could not be reached. Figure A.6, in the right hand side, shows the ability of the braking system to withstand strong winds.

Another issue to cover was the external shape of the nacelle. The initial prototype was studied with CFD and tested in both cross-flow and operating conditions. The fact that it is a downwind oriented wind turbine makes the nacelle design a crucial aspect for fast and effective self-orientation. The criteria to design a new nacelle was the next: a) low drag in operating conditions and b) high self-reorientation yaw moment when having cross-flow. No CFD images have been included about the nacelle study for brevity reasons, however, the initial and final design are available in Figure A.7.

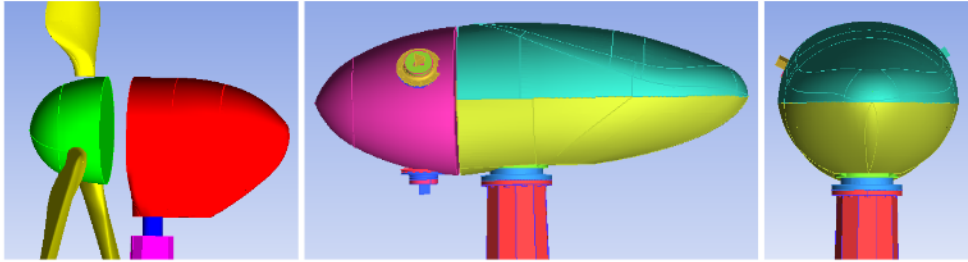


Figure A.7: Comparison of the initial nacelle (left) and the final improved design (middle and right).

With all the aerodynamic study completed, the contributions of the Finite Element Analysis for the design of the mechanical parts together with the rest of engineering aspects that Enair covered, the prototype was brought to reality. Figure A.8 shows the last constructive details of the blade in the top-left corner, down below the erection process of the turbine in the testing location and its final disposition in the right hand side.

The turbine's performance evaluation has gone through different stages. At the beginning, some wiring and electrical problems arised and the project had to stop to find the flaws and replace the damaged components. An experimental measuring campaign was carried out by Enair in the testing site and the results are available in Figure A.9. The machine is not certified yet, therefore, the power curve must be considered as provisional until the official testing is carried out, which will be done in the following months.

Figure A.9 contains information about the 10 kW turbine and also the 40 kW one.



Figure A.8: Images of the 10 kW machine. Left: ultimate blade treatment. Bottom: erecting process. Right: final layout of the turbine ready for testing and operation.

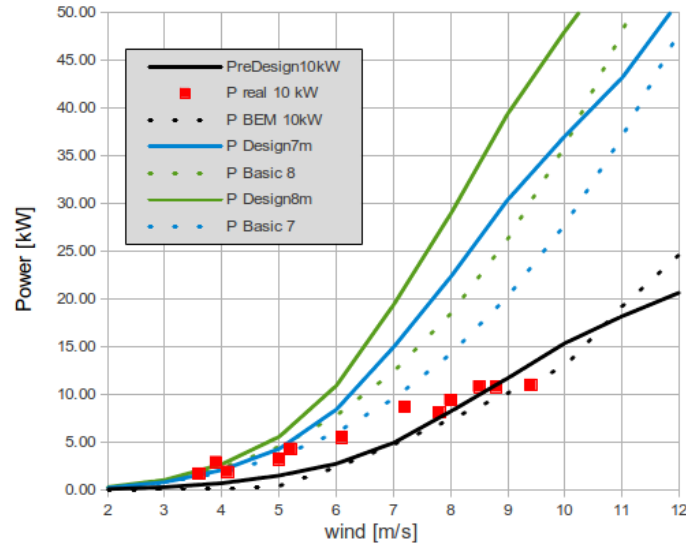


Figure A.9: Pre-design and BEM-based design comparison for the 40 kW turbine, together with the initial design of the 10 kW turbine, and some real instantaneous measures in the experimental campaign.

The figure shows the initial pre-design of the 10 kW turbine in a black solid line⁹ and the BEM-based predicted behavior in a black dotted line. They show very good agreement, meaning that BEM results are reasonable. Additionally, the results of the experimental campaign for the 10 kW turbine are shown in red color square dots. Those results below 7 m/s seem not to be correct because they reveal a power performance which violates the Betz limit. Those results above 7 m/s are much more trustable and adjusted to the BEM predictions. Altogether, the fact that some of the values are – at least – dubious, forces us to interpret them with caution. Once again, we should not forget that all the measured data from the real power and wind are provisional. From the available measures, the power is much more reliable because of the electric measuring systems. Conversely, the measured wind speed might be altered from reality. Although the anemometer is installed 40 cm above the nacelle’s upper surface, its position does not guarantee to capture the true free stream velocity.

The rest of the content of Figure A.9 will be discussed within the next section.

⁹based on Equation A.1 and a rough estimation of the power coefficient (c_p)

A.4 Second turbine: 40 kW

The first step in the design process of the 40 kW downwind turbine is to determine the size of the blade. A clear limitation for this case was the target of keeping the turbine within the category of *small wind energy*¹⁰. The regulation accepts turbines with a rotor swept area below 200 m^2 , i.e. the maximum rotor radius to fulfill this condition is 7979 mm. Back to Figure A.9. Two different blades were selected in the pre-design of the 40 kW turbine: a 7 and 8 m long blades (blue and green, respectively). Two different approaches were followed for each: the dotted line corresponds to a constant power coefficient of 0.3 and the solid line represents a more refined power curve, employing c_p values based on the experience of the 10 kW design.

The target is to achieve the rated power at the lowest wind possible. Since Enair did not state a limitation, even a larger rotor (say with a 10 m radius) would be acceptable from the mechanical and weight-related point of view. Finally, the rotor radius was set to 7.9 m, meaning that the tip of the blade would be at this position and the root will have in consideration the hub radius, to allow a good physical coupling.

#	Design	Blade Law		Maximum allowed		at CpMax			AEP
	TSR	Twist [°]	Chord [m]	Twist	Chord	CpMax	TSR	Pitch [°]	[-]
1	4.5	ideal	ideal	36	1.86	0.489	4	2	Reference
2	4.5	linear	ideal	18	1.86	0.477	5	5	-
3	4.5	ideal	linear	36	1.37	0.483	4	2	-
4	4.5	linear	linear	18	1.37	0.473	5	4	-5.63%
5	4.5	manual	manual	20	1.45	0.479	5	4	-
6	4.5	manual2	manual	25	1.45	0.475	5	4	-2.04%

Table A.3: Comparison of the six final blade designs for the 40 kW turbine.

The next aspect to cover in the 40 kW blade design is the specific geometry of the blade. The airfoil employed to shape the blade was the same than in the 10 kW turbine, but the twist and chord were reconfigured. At this point in time, the Weibull distribution approach was ready to be used and the Annual Energy Production (AEP) was susceptible to be computed. Therefore, the criterion to select the best of all the design possibilities contemplated was based on the AEP computation. Table A.3 shows the tabulated values of the final round comparison hosting six blade possibilities. Blade 1 was the reference, without limits in the maximum chord and twist values, having ideal laws. Blade 6 was

¹⁰the Spanish term in the regulation is the so-called *minieólica*

found to attain a good compromise between a good AEP (only 2 % reduction with respect the reference blade) and a reasonable maximum twist and chord values along the blade (25° and 1.45 m).

At that point in time, the blade optimizer was being developed but its application to Enair's case did not arrive on time. Therefore, the optimization process was manually done as exposed in the previous paragraph.

The BEM-based algorithm is a useful technique to predict the behaviour of the machine in *equilibrium*, taking into account the aerodynamic and electric torques. In fact, this state is an idealization of the reality. The truth is that the machine is continuously unbalanced, so it experiences a dynamic readjustment in its angular speed in function of the resulting net torque.

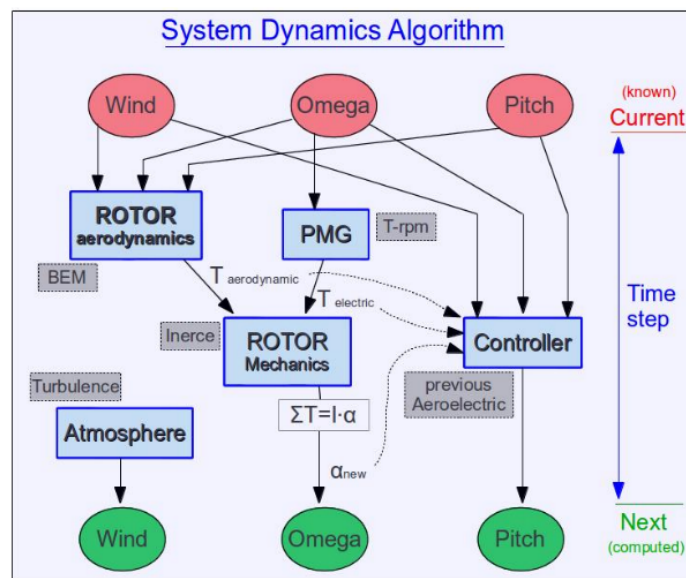


Figure A.10: Schematic representation of the dynamics of wind turbines algorithm.

For the case of the 40 kW turbine it was desirable to have a platform to simulate the dynamic behaviour of the turbine in front of wind speed changes and other possible perturbances. To that aim, a Simulink-like code was generated in order to study the responses of the turbine. The code was built in C++ and it was designed according to the paradigm presented in Figure A.10.

The idea of the dynamic algorithm is to run an unsteady simulation coupling the

behaviour of the different embedded systems separately, in order to integrate them all at the right moment. The basic parameters of the simulation are the wind speed, the angular speed and the pitch angle. Those parameters are governed by the atmospheric conditions, the torque balance and the controller, respectively. With them the next subsystems are computed:

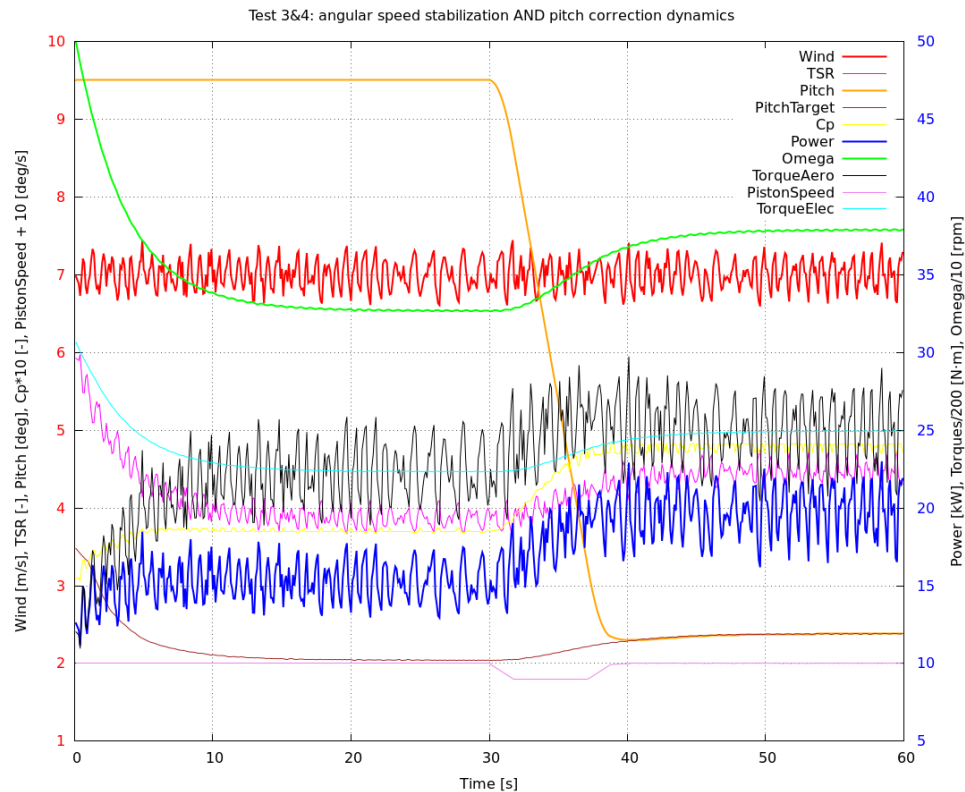


Figure A.11: 60-seconds-long dynamic simulation of the 40 kW turbine with unbalanced initial angular speed and fixed initial pitch angle for the first half of the simulation.

Rotor aerodynamics. The core of the aerodynamics is the largely discussed BEM method. It receives the inputs of wind, angular speed and pitch angle and delivers the aerodynamic Torque.

Generator (PMG). The electric generator¹¹ is modeled with the curve Torque - rpm. Therefore, it receives the angular speed and delivers the electric Torque.

Rotor mechanics. The rotor mechanics receives both the aerodynamic and electric Torques, and delivers the angular acceleration taking into consideration the inertia of the whole rotor.

Controller. The controller receives all the inputs of the system. Based on the tabulated information fed by a previous AeroElectric simulation, it decides which should be the next target pitch that the piston should pursue.

Piston. The piston receives the state of the pitching system and the target from the controller in order to execute the movement to accommodate the best pitch for the next time step.

Atmosphere. The atmosphere is the conceptualized entity that manages and fixes the wind speed at any time during the simulation.

Several simulations were carried out to check the stability and response of the machine. Here, for the sake of brevity, only one of those test will be exposed. Figure A.11 shows 60 seconds of a simulation that starts with a non-converged situation where the angular speed is greater than it should (initial value 50 rpm). In 20 seconds the system stabilizes and the angular speed becomes 32 rpm. Additionally, the pitch angle was simulated to be fixed at 9.5° instead of being free to be determined by the controller-piston group. At time 30 seconds, the system is freed again, and 20 seconds later a new stationary value is achieved where the optimal pitch for the given conditions is 2.4° . Notice the effect of disposing the best pitch angle. When being incorrect (9.5° , at time 25 seconds) the power was 15 kW. Conversely, when being the optimal (2.4° , at time 55 seconds) the power was 20 kW. This means a 33 % increase.

The final aspect to cover here refers to the CFD simulations already presented in the Section 4.4. The additional contribution to the CFD simulations presented in Chapter 4 is the sequence of the blade passage seen from different perspectives, check Figures A.12 and A.13. In both figures, the shadow of the tower is clear and the interaction blade-tower can be seen through the velocity fields appearing in the sequence.

¹¹in this case a Permanent Magnet Generator (PMG)

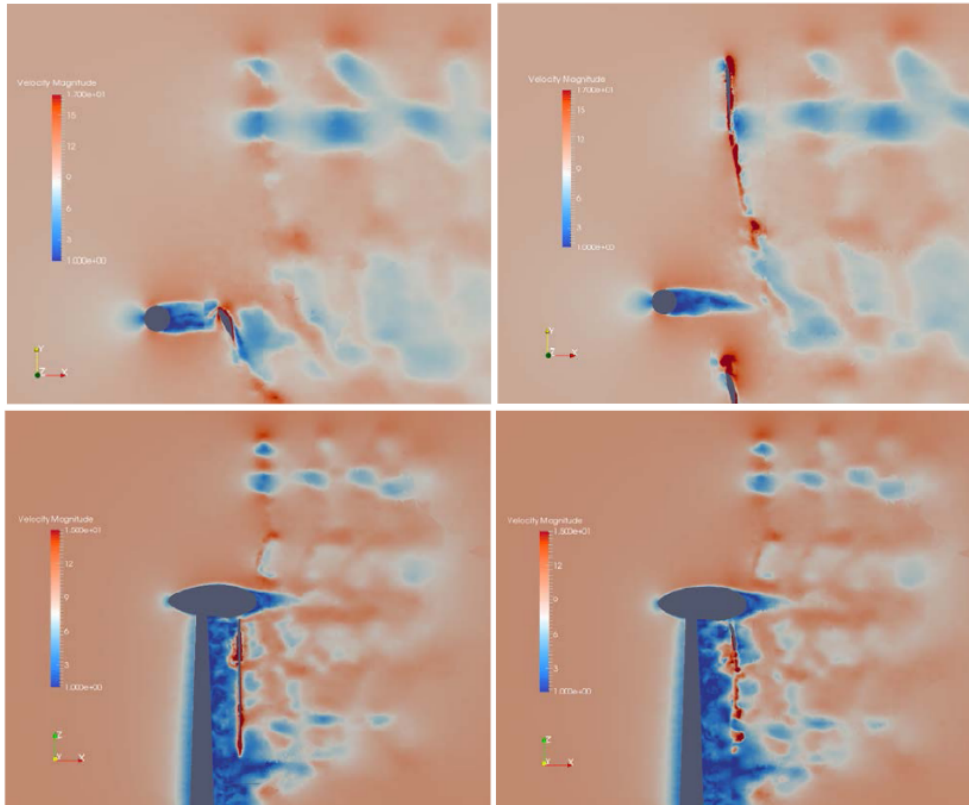


Figure A.12: Sequence of the blade passage in the 40 kW blade employing Sliding Meshes approach. Top: horizontal plane. Bottom: vertical plane perpendicular to the rotor plane.

A.5 Conclusion

The objective of this appendix is to show how the capabilities described throughout the thesis have been successfully applied to a commercial case. The 10 kW machine is already a working prototype and it is about to be commercialized with massive production. The 40 kW machine is still in the first part of the manufacturing process yet.

Thus, two downwind turbines have been designed from scratch. The process followed to do so was the scanning of 3 kW already-existing turbine in order to perform reverse engineering and upscale the model to 10 kW. Several 2D CFD simulations were carried

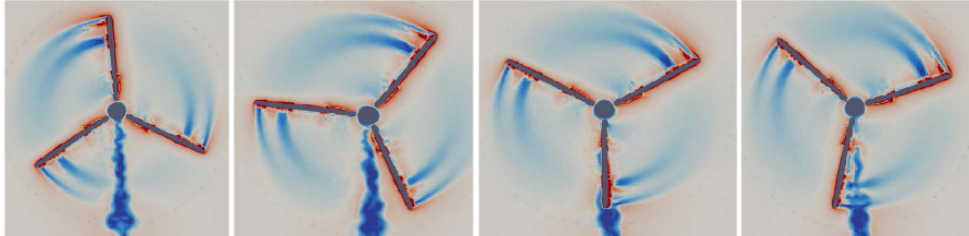


Figure A.13: Sequence of the blade passage in the 40 kW blade employing Sliding Meshes approach at the rotating plane.

out to analyze the performance of the profile of the previous machine and the new proposed airfoil. The comparison of both airfoils turned out to be much more favorable to the new, so it was the selected one to the final blade design.

Optimization of the 10 and 40 kW machines was done by means of comparing the performance of several designs. In the 10 kW case, the AeroElectric code and TSR-pitch comparisons pointed out the best approach. In the 40 kW case, the AeroElectric code was combined with the Weibull distribution which allowed the computation of the Annual Energy Production. The AEP was used as the main criterion to define the best possible blade design. In the end, the automatic optimization presented in Chapter 3 was validated and employed in the current case, but the tight deadlines did not allow us to use this technique to improve the already defined 40 kW turbine.

All in all, based on the preliminary experimental campaign over the 10 kW turbine, we can say that the aerodynamic design has been successful. Further and final testing will have to be done with the power curve certification during the following months.

