UNIVERSITY OF SÃO PAULO ESCOLA POLITÉCNICA DEPARTMENT OF MECHANICAL ENGINEERING

Luís Eduardo Boni Cruz

Wind farm layout optimization based on numerical simulations

São Paulo 2019

Luís Eduardo Boni Cruz

Wind farm layout optimization based on numerical simulations

Original Version

Master Thesis presented to the Escola Politécnica da Universidade de São Paulo, to obtain the degree of Master of Science

Concentration Area: Energy and Fluids

Supervisor: Dr Bruno Souza Carmo

São Paulo 2019 Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catalogação-na-publicação

Cruz, Luis Eduardo Boni Wind farm layout optimization based on numerical simulations / L. E. B. Cruz -- São Paulo, 2019. 98 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica.

1.Layout optimization 2.Wind turbine 3.Actuator Disk 4.OpenFOAM 5.Dakota I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecânica II.t.

Acknowledgements

I would like to thank in first place my supervisor Professor Bruno Souza Carmo for being supportive, for understanding my reasons when I couldn't make progress in the work, for having patience, for motivating me during the development of this work and for guiding me through it.

I extend my thanks to my family and friends, specially my parents, my sister and my fiancee Rayra for understanding when I was not present because I needed to work on this project and for supporting me during this work.

I also want to thank my friend Eduardo Katsuno for all the help and discussions along this work, The OpenFOAM Foundation and Dakota's team at Sandia for providing free and high quality software and also the OpenFOAM community from the forum CFD Online, which directly or indirectly helped me finding the solution to the issues that appeared. In addition, I extend my thanks to the STI team at University of São Paulo for helping me with HPC issues and for allowing me to use the HPC structure.

Resumo

O presente trabalho consiste em desenvolver uma ferramenta para otimização de layouts de parques eólicos baseada em simulações de Dinâmica dos Fluidos Computacional (CFD), tanto do escoamento atmosférico quanto da interferência entre turbinas. Uma vez que não é viável simular todo um parque eólico usando a geometria completa de uma turbina eólica, surge a necessidade de modelos que representem os efeitos das turbinas no escoamento local, e o modelo mais comumente usado é o Modelo de Disco Atuador e suas variações. Definido o procedimento para a avaliação do comportamento de uma turbina eólica usando um modelo de CFD usando o software OpenFOAM, foi feito o acoplamento desse modelo com o software de otimização Dakota. Um Algoritmo Genético foi usado para a tarefa de otimização devido à sua robustez e as características do problema resolvido. Em posse dessa nova ferramenta, três diferentes casos de topografia foram testados considerando diversos números de turbinas em um domínio cilíndrico para atingir o melhor layout em termos de AEP que respeite as restrições físicas impostas. O processo de otimização foi bem-sucedido, haja visto a maximização de AEP. Além disso, o algoritmo foi capaz de evitar corretamente as esteiras geradas por máquinas a montante em cada caso, e ainda foi capaz de se beneficiar da interação das esteiras com o terreno durante o processo de otimização. Foi concluído que os resultados são promissores apesar dos significativos recursos computacionais necessários para o trabalho.

Palavras-chave: Otimização de layout. Turbina eólica. Disco Atuador. Dinâmica dos fluidos computacional. Algoritmo genético.

Abstract

This work consists in developing a tool for wind farm layout optimization based on Computational Fluid Dynamics (CFD) simulations of the atmospheric wind flow and inter-turbine interference. Since it is not feasible to simulate a whole wind farm using the complete geometry of the wind turbines, the need for models to represent their effects on the wind flow and the interference of one turbine on the others arises, and the most commonly used model is the Actuator Disk model and its variations. The procedure for wind turbine behavior evaluation using a CFD model was implemented in the OpenFOAM software, and this model was coupled with the Dakota optimization toolkit. A Genetic Algorithm was selected for the optimization task due to its robustness and the characteristics of the problem solved. With this new tool in hand, three different terrain cases were tested considering different numbers of turbines on a cylindrical domain in order to achieve the best wind farm layout in terms of AEP that respects the imposed physical restrictions. The optimization process was successful, leading to the maximization of the AEP. In addition, the algorithm correctly avoided the wakes generated by the upstream wind turbines for each case and was able to take advantage of the wake-terrain interaction during the optimization process. It was concluded that the results are promising despite the high computational resources required.

Keywords: Layout optimization. Wind turbine. Actuator Disk. OpenFOAM. Dakota.

List of Figures

Figure 1 –	Global Cumulative Installed Wind Capacity 2001-2017	1
Figure 2 –	Top 10 Cumulative Installed Capacity in December/2017	2
Figure 3 –	Stream tube representation	11
Figure 4 –	Actuator Disk and Variation of variables	11
Figure 5 –	Typical Folder Structure	16
Figure 6 –	Domain and Optimization grid representation	23
Figure 7 –	Optimization Workflow	24
Figure 8 –	Block structure of the domain (a) and domain mesh representation $\ . \ .$	28
Figure 9 –	Slice of the mesh of the Askervein Hill case	29
Figure 10 –	Example of Actuator disk cells (a) and surrounding mesh (b)	29
Figure 11 –	Objective function versus evaluations (a) and populations (b) for Case	
	1 with 12 turbines \ldots	30
Figure 12 –	Objective function versus evaluations (a) and populations (b) for Case	
	1 with 20 turbines \ldots	31
Figure 13 –	Wind velocity at $120\mathrm{m}$ from the ground of the best individual for Case	
	1 with (a) 12 turbines and (b) 20 turbines $\ldots \ldots \ldots \ldots \ldots \ldots \ldots$	31
Figure 14 –	Wind rose with sixteen sectors	32
Figure 15 –	Objective function versus evaluations (a) and populations (b) for Case	
	2 with 12 turbines \ldots	33
Figure 16 –	Objective function versus evaluations (a) and populations (b) for Case	
	2 with 20 turbines	33
Figure 17 –	Wind velocity at $120 \mathrm{m}$ from the ground of the best individual for Case	
	2 with 12 turbines and wind from (a) Sector 3 and (b) Sector 10	34
Figure 18 –	Wind velocity at 120 m from the ground of the best individual for Case	
	2 with 20 turbines and wind from (a) Sector 3 and (b) Sector 10	34
Figure 19 –	Specific AEP and Wake Losses for Cases 1 and 2	35
Figure 20 –	Altitude of the Synthetic hill	35
Figure 21 –	Objective function versus evaluations (a) and populations (b) for Case	
	3 with 12 turbines	36
Figure 22 –	Objective function versus evaluations (a) and populations (b) for Case	
	3 with 20 turbines	36
Figure 23 –	Wind velocity at 120 m from the ground of the best individual for Case	
	3 with (a) 12 turbines and (b) 20 turbines	37
Figure 24 –	Wind rose with four sectors	38
Figure 25 –	Objective function versus evaluations (a) and populations (b) for Case	
	4 with 12 turbines \ldots	38

Figure 26 –	Objective function versus evaluations (a) and populations (b) for Case	
	4 with 20 turbines $\ldots \ldots \ldots$	39
Figure 27 –	Wind velocity at $120\mathrm{m}$ from the ground of the best individual for Case	
	4 with (a) 12 turbines and (b) 20 turbines $\ldots \ldots \ldots \ldots \ldots \ldots \ldots$	39
Figure 28 –	Objective function versus evaluations (a) and populations (b) for Case	
	5 with 12 turbines \ldots	40
Figure 29 –	Objective function versus evaluations (a) and populations (b) for Case	
	5 with 20 turbines $\ldots \ldots \ldots$	40
Figure 30 –	Specific AEP and Wake Losses for Cases 3, 4 and 5	41
Figure 31 –	Wind velocity at $120\mathrm{m}$ from the ground of the best individual for Case	
	5 with 12 turbines and wind from (a) Sector 3 and (b) Sector 10	41
Figure 32 –	Wind velocity at $120\mathrm{m}$ from the ground of the best individual for Case	
	5 with 20 turbines and wind from (a) Sector 3 and (b) Sector 10	42
Figure 33 –	Askervein hill area colored by altitude values	42
Figure 34 –	Objective function versus evaluations (a) and populations (b) for Case 6	43
Figure 35 –	Wind velocity (a) and total pressure (b) at 120 m from the ground of	
	the best individual for Case $6 \ldots \ldots$	43
Figure 36 –	Original and normalized wind distribution	44
Figure 37 –	Objective function versus evaluations (a) and populations (b) for Case 7	44
Figure 38 –	Wind velocity and total pressure at $120\mathrm{m}$ from the ground of the best	
	individual for Case 7 and $9 \mathrm{m/s}$	45
Figure 39 –	Wind rose with sixteen sectors of the MERRA node	45
Figure 40 –	Plot of the Error function for $-3 \le x \le 3$	47
Figure 41 –	Smoothing region representation in grey	47
Figure 42 –	Linear variation of x with the distance from the center of the domain d .	48
Figure 43 –	Altitude values for the Askervein Hill region: original topography (a)	
	and smoothed topography (b) \ldots \ldots \ldots \ldots \ldots \ldots \ldots	48
Figure 44 –	Objective function versus evaluations (a) and populations (b) for Case	
	8 with 12 turbines	49
Figure 45 –	Objective function versus evaluations (a) and populations (b) for Case	
	8 with 20 turbines	49
Figure 46 –	Wind velocity and total pressure at $120\mathrm{m}$ from the ground of the best	
	individual for Case 8 with 12 turbines and wind from Sector 11 \ldots	50
Figure 47 –	Wind velocity and total pressure at $120\mathrm{m}$ from the ground of the best	
	individual for Case 8 with 20 turbines and wind from Sector 11 $$	50
Figure 48 –	Layout comparison - Synthetic Hill with 12 turbines (a) and 20 turbines	
	(b). The blue "x" represents the layout obtained in this work and the	
	red circle represents the layout obtained with Openwind \ldots	51

Figure 49 –	Layout comparison - Askervein Hill with 12 turbines (a) and 20 turbines	50
	(D)	52
Figure 50 –	Objective function × evaluations (a) and populations (b)-Case 1 - 14 turbines	61
Figure 51 –	Objective function × evaluations (a) and populations (b)-Case 1 - 16 turbines	61
Figure 52 –	Objective function × evaluations (a) and populations (b)-Case 1 - 18 turbines	62
Figure 53 –	Objective function × evaluations (a) and populations (b)-Case 2 - 14 turbines	62
Figure 54 –	Objective function × evaluations (a) and populations (b)-Case 2 - 16 turbines	63
Figure 55 –	Objective function \times evaluations (a) and populations (b)-Case 2 - 18 turbines	63
Figure 56 –	Objective function \times evaluations (a) and populations (b)-Case 3 - 14 turbines	64
Figure 57 –	Objective function \times evaluations (a) and populations (b)-Case 3 - 16 turbines	64
Figure 58 –	Objective function \times evaluations (a) and populations (b)-Case 3 - 18	04
Figure 59 –	turbines \dots	65 65
Figure 60 –	Objective function \times evaluations (a) and populations (b)-Case 4 - 16 turbines	66
Figure 61 –	Objective function \times evaluations (a) and populations (b)-Case 4 - 18 turbines	66
Figure 62 –	Objective function \times evaluations (a) and populations (b)-Case 5 - 14 turbines	67
Figure 63 –	Objective function \times evaluations (a) and populations (b)-Case 5 - 16 turbines	67
Figure 64 –	Objective function \times evaluations (a) and populations (b)-Case 5 - 18 turbines	68
Figure 65 –	Objective function × evaluations (a) and populations (b)-Case 8 - 14 turbines	68
Figure 66 –	Objective function \times evaluations (a) and populations (b)-Case 8 - 16 turbines	69
Figure 67 –	Objective function × evaluations (a) and populations (b)-Case 8 - 18 turbines	69

Figure 68 – Objective function versus evaluations (a) and populations (b) in the	
mesh sensitivity analysis with 16 turbines - no refinement	72
Figure 69 – Objective function versus evaluations (a) and populations (b) in the	
mesh sensitivity analysis with 16 turbines - $R1$	72
Figure 70 – Objective function versus evaluations (a) and populations (b) in the	
mesh sensitivity analysis with 16 turbines - $R2$	73
Figure 71 – Final layout for the case with no mesh refinement (a), R1 (b) and R2 (c)	73

List of Tables

Table 1 – Stability Classification According to the Potential Temperature Gradient	10
Table 2 – $\kappa - \varepsilon$ turbulence model constants	14
Table 3 – Common parameters in the optimization cases. \ldots \ldots \ldots \ldots	27
Table 4 – $\kappa - \varepsilon$ turbulence model constants for the optimization cases	27
Table 5 – Frequency distribution for sixteen sectors wind rose. \ldots \ldots \ldots	32
Table 6 – Frequency distribution for four sectors wind rose. .	37
Table 7 – Normalized wind distribution for Case 7. .	44
Table 8 $-$ Frequency distribution for sixteen sectors wind rose of the MERRA node.	46
Table 9 $-$ Element size and number of elements for mesh sensitivity analysis \ldots	71

List of abbreviations and acronyms

- ABL Atmospheric Boundary Layer
- ADM Actuator Disk Model
- AEP Annual Energy Production
- ALM Actuator Line Model
- BEM Blade Element Momentum
- CFD Computational Fluid Dynamics
- DNS Direct Numerical Simulation
- GUI Graphical User Interface
- LES Large Eddy Simulation
- NREL National Renewable Energy Laboratory
- RANS Reynolds-averaged Navier-Stokes
- TKE Turbulent Kinetic Energy

List of Symbols

\dot{m}	Mass flow	[kg/s]
κ	Turbulent Kinetic Energy	$[\mathrm{m}^2/\mathrm{s}^2]$
μ	Dynamic Viscosity	$[\rm N\cdot s/m^2]$
μ_t	Turbulent Viscosity	$[\rm N\cdot s/m^2]$
ϕ	General Variable	[-]
ρ	Fluid Density	$[\mathrm{kg}/\mathrm{m}^3]$
heta	Potential Temperature	[K]
ε	Turbulent Dissipation Rate	$[\mathrm{m}^2/\mathrm{s}^3]$
a	Induction factor	[-]
A_d	Cross section area of the disk	$[m^2]$
A_w	Cross section area on the far wake	$[m^2]$
c_0, c_1, c_2	Thrust distribution constants	$[-, m^{-2}, m^{-4}]$
Ср	Power Coefficient	[-]
c_p	Specific heat at constant pressure	$[J/(kg\cdot K)]$
c_T	Thrust Coefficient	[-]
$C_{\mu}, \sigma_{\kappa}, \sigma_{\varepsilon}, C_{1\varepsilon}, C_{2\varepsilon}$	Coefficients of the k-e model	[-]
F_{obj}	Objective Function	[W]
K	Von Karman constant	[-]
p	Pressure	$[\mathrm{N/m^2}]$
p_0	Pressure reference state	$[N/m^2]$
P_{κ}	Turbulent production term	$[\rm N/(m^2\cdot s)]$
P_{kin}	Kinetic Power	[W]
P_{out}	Power output	[W]
R	Rotor radius	[m]

R	Universal gas constant	$[{\rm J}/({\rm kg}\cdot{\rm K})]$
r_i	Radial distance	[m]
S_M	Source term	$[N/m^3]$
T	Temperature; Thrust force	[K, N]
t	Time	$[\mathbf{s}]$
$T(r_i)$	Thrust distribution	[N]
T_{total}	Total thrust	[N]
U, u_i	Velocity field	[m/s]
U^*	Friction Velocity	[m/s]
U_d	Wind speed in the disk	[m/s]
U_w	Wind speed in the far wake	[m/s]
U_{∞}	Free wind speed	[m/s]
U_{ref}	Reference Wind Velocity	[m/s]
U_{wall}	Velocity parallel to the wall	[m/s]
V_i	Volume of cell i	$[m^3]$
V_{tot}	Total volume of the rotor region	$[m^3]$
x_i	Suffix notation for direction	[m]
z_0	Roughness length	[m]
z_g	Minimum z coordinate	[m]
z_{ref}	Reference height	[m]

Contents

	Acknowledgements	i
	Resumo	ii
	Abstract	v
	List of Figures	ii
	List of Tables	ci
	List of abbreviations and acronyms	ii
	List of Symbols	/i
1	INTRODUCTION	1
2	LITERATURE AND STATE OF THE ART REVIEW	5
2.1	Actuator Disk Concept	5
2.2	Optimization Methods for Wind farms	6
2.3	Optimization Software in the Market	7
3	THEORETICAL BACKGROUND	9
3 3.1	THEORETICAL BACKGROUND Atmospheric Boundary Layer	9 9
3 3.1 3.2	THEORETICAL BACKGROUND	9 9 0
3 3.1 3.2 3.3	THEORETICAL BACKGROUND	9 9 0 3
3 3.1 3.2 3.3 3.3.1	THEORETICAL BACKGROUND Atmospheric Boundary Layer Atmospheric Boundary Layer 1 Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1	9 9 0 3
 3.1 3.2 3.3 3.3.1 3.3.2 	THEORETICAL BACKGROUND Atmospheric Boundary Layer Atmospheric Boundary Layer 1 Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1	9 9 0 3 3
 3.1 3.2 3.3 3.3.1 3.3.2 3.4 	THEORETICAL BACKGROUND Atmospheric Boundary Layer Atmospheric Boundary Layer 1 Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1 OpenFOAM 1	9 9 0 3 3 5
 3.1 3.2 3.3 3.3.1 3.3.2 3.4 	THEORETICAL BACKGROUND Atmospheric Boundary Layer Atmospheric Boundary Layer 1 Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1 OpenFOAM 1 Basics 1	9 9 0 3 3 5 5
 3.1 3.2 3.3 3.3.1 3.3.2 3.4 3.4.1 3.4.2 	THEORETICAL BACKGROUND Atmospheric Boundary Layer 1 Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1 OpenFOAM 1 Basics 1 Case Structure 1	9 0 3 3 5 5 5
 3.1 3.2 3.3 3.3.1 3.3.2 3.4 3.4.1 3.4.2 3.5 	THEORETICAL BACKGROUND Atmospheric Boundary Layer Atmospheric Boundary Layer 1 Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1 OpenFOAM 1 Case Structure 1 Optimization Methods 1	9 0 3 3 5 5 7
 3.1 3.2 3.3 3.3.1 3.3.2 3.4 3.4.1 3.4.2 3.5.1 	THEORETICAL BACKGROUND Atmospheric Boundary Layer Atmospheric Boundary Layer 1 Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1 OpenFOAM 1 Basics 1 Case Structure 1 Dakota software 1	9 0 3 3 5 5 7 7
 3.1 3.2 3.3 3.3.1 3.3.2 3.4 3.4.1 3.4.2 3.5 3.5.1 3.5.2 	THEORETICAL BACKGROUND Atmospheric Boundary Layer Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1 OpenFOAM 1 Basics 1 Optimization Methods 1 Dakota software 1 Genetic Algorithm 1	9 9 0 3 3 5 5 7 7 7 7
 3.1 3.2 3.3 3.3.1 3.3.2 3.4 3.4.1 3.4.2 3.5 3.5.1 3.5.2 4 	THEORETICAL BACKGROUND Atmospheric Boundary Layer Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1 OpenFOAM 1 Basics 1 Optimization Methods 1 Dakota software 1 Genetic Algorithm 1	9 9 3 3 5 5 7 7 7 9
 3.1 3.2 3.3 3.3.1 3.3.2 3.4 3.4.1 3.4.2 3.5 3.5.1 3.5.2 4 4.1 	THEORETICAL BACKGROUND Atmospheric Boundary Layer Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1 OpenFOAM 1 Basics 1 Case Structure 1 Optimization Methods 1 Dakota software 1 MUMERICAL MODELING 1 Actuator Disk Model and Modifications 1	9 9 3 3 5 5 7 7 9 9
 3.1 3.2 3.3 3.3.1 3.3.2 3.4 3.4.1 3.4.2 3.5 3.5.1 3.5.2 4 4.1 4.2 	THEORETICAL BACKGROUND Atmospheric Boundary Layer Fundamentals of Wind Turbine Aerodynamics 1 Numerical Modeling 1 Incompressible Navier-Stokes Equations 1 Turbulence Model 1 OpenFOAM 1 Basics 1 Case Structure 1 Optimization Methods 1 Dakota software 1 Genetic Algorithm 1 NUMERICAL MODELING 1 Boundary Conditions 2	990333555777991

4.4	Optimization Workflow	24
5	WIND FARM LAYOUT OPTIMIZATION SCENARIOS	27
5.1	General Considerations	27
5.2	Mesh Characteristics	28
5.3	Results	29
5.3.1	Case 1: Flat terrain and unidirectional wind	29
5.3.2	Case 2: Flat terrain and sixteen wind directions	32
5.3.3	Case 3: Synthetic Hill and unidirectional wind	35
5.3.4	Case 4: Synthetic Hill and four wind directions	37
5.3.5	Case 5: Synthetic Hill and sixteen wind directions	38
5.3.6	Case 6: Askervein Hill and one direction	40
5.3.7	Case 7: Askervein Hill and one direction with Weibull distribution	42
5.3.8	Case 8: Askervein Hill and sixteen wind directions with varying wind speed .	45
5.4	Results obtained with Openwind	51
6	CONCLUSIONS	53
61	Suggestions for Future Work	54
0.1		54
	BIBLIOGRAPHY	55
	APPENDIX	59
	APPENDIX A – COMPLEMENTARY OPTIMIZATION RESULTS	61
A.1	Case 1 - Flat terrain and unidirectional wind \ldots \ldots \ldots \ldots \ldots	61
A.1.1	Optimization with 14 turbines	61
A.1.2	Optimization with 16 turbines	61
A.1.3	Optimization with 18 turbines	62
A.2	Case 2 - Flat terrain and sixteen wind directions	62
A.2.1	Optimization with 14 turbines	62
A.2.2	Optimization with 16 turbines	63
A.2.3	Optimization with 18 turbines	63
A.3	Case 3 - Synthetic Hill and unidirectional wind	64
A.3.1	Optimization with 14 turbines	64
A.3.2	Optimization with 16 turbines	64
A.3.3	Optimization with 18 turbines	65
A.4	Case 4 - Synthetic Hill and four wind directions	65

Optimization with 14 turbines

65

A.4.1

A.4.2

A.4.3	Optimization with 18 turbines	66
A.5	Case 5 - Synthetic Hill and sixteen wind directions	67
A.5.1	Optimization with 14 turbines	67
A.5.2	Optimization with 16 turbines	67
A.5.3	Optimization with 18 turbines	68
A.6	Case 8 - Askervein Hill and sixteen wind directions with varying	
	wind speed	68
A.6.1	Optimization with 14 turbines	68
A.6.2	Optimization with 16 turbines	69
A.6.3	Optimization with 18 turbines	69
	APPENDIX B - MESH SENSITIVITY ANALYSIS	71

1 Introduction

The wind energy market has shown an expressive growth in the past two decades, going from 24 GW of installed capacity in 2001 to almost 540 GW in 2016, an increase of about 2250% according to the Global Wind Energy Council (Figure 1). This growth is an evidence of the concern about environmental impacts related to other sources of energy combined with the investments on research and development to support the reduction of costs of this source, turning it into a competitive source of energy.

In terms of the energy matrix, the percentage corresponding to wind energy has increased in all countries investing on this source, which is strategic for energy planning. Rain and wind regimes are fairly complementary in Brazil. In seasons with low rain volumes the wind is ideal for wind power generation and vice-versa, and this allows saving water from the reservoirs during windy periods, so the water levels do not drop drastically.

The leading country in installed capacity in 2017 was China, with 35% of the installed capacity, followed by the United States of America, with 17% (Figure 2). Brazil is in the 8th place with 2%, which represents 12.8 GW. It is expected that until 2027 there will be approximately 23 GW of installed power from wind energy, representing about 12% of Brazil's energy matrix (Ministério de Minas e Energia. Empresa de Pesquisa Energética. Brasília: MME/EPE, 2018).

Wind farms in Brazil can have more than 100 wind turbines occupying an area of more than 100 km^2 and are usually divided in units of 30 MW. Due to the size and geometry of the wind turbines, a wind speed deficit and high turbulence zone is generated downstream the turbines. The generated wake causes energy yield losses commonly called wake losses and can propagate to regions located more than 3 km downstream the wind





Source: (Global Wind Energy Council, 2018)



Figure 2 – Top 10 Cumulative Installed Capacity in December/2017

Source: (Global Wind Energy Council, 2018)

farm. This means that it is important to evaluate the distribution of wind turbines within the available area not only to avoid wake interference on the turbines of that farm, but also to reduce the impact on other wind farms.

Being one of the first stages of the project, the layout of a wind farm is a critical factor for its characterization. With the improvement of knowledge and competitiveness of this market, this factor is being highly exploited and taken to the extreme in order to maximize the profitability of each project. In most cases, there are few wind resource measurements in the area where the wind farm is being planned to be installed, normally between 1 and 10 meteorological masts for a many kilometers wide wind farm.

It is known that the commonly used tools in the market are not able to predict the wind flow behavior in regions with high topographic and roughness complexity, because they use linear models with many simplifications and limitations in addition to analytical wake models, which were developed initially for offshore wind farms. For this reason, a computational fluid dynamics (CFD) tool was chosen to be used in this work, which despite having a higher complexity on its use, can achieve good results when correctly used (TAPIA, 2009). It was also noticed that there is a lack of CFD based optimization tools available in the wind market, so there is a need to have more accurate and reliable tools that can exploit as much as possible the available wind resource (BONANNI et al., 2012), justifying the development of this work.

The goal of this work is to develop a tool for wind farm layout optimization based on Computational Fluid Dynamics simulations of both atmospheric flow and inter turbine interference using the open source CFD software OpenFOAM combined with the open source optimization software Dakota, for a given topography and wind turbine model. The objective of the optimization is to obtain a wind farm layout that gives the maximum Annual Energy Production (AEP) respecting imposed constraints. The tool must be capable of considering real wind conditions during the optimization, such as multiple wind directions and wind velocities.

The tool developed can receive as inputs the terrain, being it simple or complex, number and model of turbines to be considered, wind rose and sector-wise average wind speeds, with the assumption of a fixed roughness value all over the computational domain. It employs the Actuator Disk Model to represent the wind turbine rotor. The objective function of the optimization is the AEP and the constraint of the problem is the inter turbine spacing. The steady incompressible Navier-Stokes equations were solved and a genetic algorithm was used for the optimization task. It is expected that the tool can provide reliable results, specially in complex terrains, in terms of the capability of dealing with complex flow scenarios such as wake interaction, wake recovery and the impact of the terrain in the wake of turbines, which cannot be analyzed in most of the software available in the market today.

A description of this tool is presented in this work, and results for the following cases will be presented and discussed: (i) a flat terrain considering one and sixteen wind directions; (ii) a terrain with a synthetic hill, derived from a Gaussian function, for one, four and sixteen wind directions, and (iii) a complex terrain, represented by the Askervein Hill, considering one and sixteen wind directions, with a sector-wise wind velocity distribution.

In Chapter 2 we present the Literature Review of the main works related to methods for representing the turbine, wind farm wakes simulation and layout optimization, followed by a brief description of the most commonly used software in the wind industry. Chapter 3 contain the Theoretical Background related to the ABL and its classifications and also shows one way to derive the Actuator Disk main equations with Bernoulli's equation. In Chapter 4, the available ADM in OpenFOAM are presented and a proposed modification to one of its native models is shown. Additionally, the boundary conditions applied in the domain are briefly shown and the Optimization problem and Workflow are defined. Chapter 5 contains general considerations for the cases that were analyzed in terms of inputs and mesh and the results of each case, including a methodology developed for smoothing a region of the topographic map. It also discuss the mesh sensitivity and compare the results obtained in this work with results obtained using Openwind. At last, in Chapter 6 we present the conclusion of this work as well as suggestion for future works.

2 Literature and State of the Art Review

2.1 Actuator Disk Concept

A wind turbine is designed to be able to extract the kinetic energy of the incoming flow field by reducing the wind velocity downstream its rotor. Consequently, a thrust with a magnitude proportional to this energy extraction will be generated.

The actuator disk model consists on representing a wind turbine using a permeable and thin disc with the same rotor area of the wind turbine where the forces from the blades will be distributed. This concept was introduced by (FROUDE, 1889) and since then suffered several adaptations and improvements.

In (MIKKELSEN; SØRENSEN, 2003), both Actuator Disc and Actuator Line Models were tested for several configurations such as coned rotor, yawed rotor and tunnel blockage. In all tested cases there was agreement between the experimental data and theoretical results, although these methods were not recommended to be used as design methods due to the dependencies on grid and force-smearing resolution which affects the accuracy.

An improvement on the traditional actuator disk model was created by NREL (TOSSAS; LEONARDI, 2013). In that work, the actuator disk model (ADM) and actuator line model (ALM) were implemented for Large Eddy Simulations (LES) using the OpenFOAM toolbox as the flow solver. The wind turbine power and thrust coefficients were calculated and compared to a wind tunnel test showing good agreement in most of the operating velocities, but as the nacelle was not modelled there were still some disagreements between the calculated data and experimental results. Additionally, the Lillgrund offshore wind farm was simulated and the results obtained had good agreement with field measurements. It was concluded that the ADM and ALM show very similar results when simulated on a uniform flow, but only the ALM can capture the root and tip vortices. On the far wake the two presented models generated similar and acceptable results.

Another improvement was proposed by (SIMISIROGLOU et al., 2016), who modified the reference wind velocity used in the thrust coefficient curve provided by the manufacturer. This modification was done by replacing the undisturbed wind speed values with the calculated wind speed value at the disk using the standard ADM equations. By doing this, the thrust was calculated on each individual cell. This method showed a good agreement between the calculated and manufacturer's power and thrust curves for a single wind turbine immersed in a free stream. This approach will be used in this work and is described in detail later.

It is possible to find open source and commercial software that already have an ADM implemented, so there is no need to create a code for an existing application to use the ADM. OpenFOAM, ANSYS WindModeller, STAR-CCM+ and WindSim are four examples of software that could be used on a traditional analysis that required a simplified wind turbine modeling.

2.2 Optimization Methods for Wind farms

Wind farm layout optimization is a relatively new field of study that arose due to the development of a more mature market worldwide. Most of the optimization routines are based on a linear wake model first proposed by Jensen (JENSEN, 1983), which is only valid for flat terrains. The most commonly used optimization method for this class of problem in the literature is the Genetic Algorithm, which is fundamentally an attempt to imitate the behavior of natural evolution.

Most applications were developed for offshore wind farms in which the incoming wind profile can be assumed horizontally homogeneous, requiring only wake simulation (VASEL-BE-HAGH; ARCHER, 2017; PARADA et al., 2018; GUIRGUIS; ROMERO; AMON, 2016; HOU et al., 2017). These works essentially involved numerical wake simulation with the Jensen model and optimization based on genetic algorithm, greedy algorithm or particle swarm optimization (PSO). In complex terrain, however, the wind cannot be assumed horizontally homogeneous and therefore calculation with the more sophisticated computational fluid dynamics (CFD) is necessary (SONG et al., 2014; SONG et al., 2016; KUO et al., 2016).

A large number of different tests were made by (BONANNI et al., 2012) considering a flat terrain, the Jensen and CFD-based wake models and a Genetic Algorithm for optimization. Two cases, the first one with one wind direction and the second one with 4 wind directions, were optimized with both Jensen and CFD-based wake models, with the conclusion that there is an advantage of using the CFD-based wake model once it considers several physical issues, such as atmospheric wind profile, presence of the ground and turbulence.

A similar approach was used by (BONANNI et al., 2015), for a complex terrain, but in that work the undisturbed wind field was calculated using a CFD tool combined with a linear wake model. The obtained results showed good agreement with the intuitive solution, but the authors concluded that more investigation was required to fully validate the methodology.

Genetic Algorithms are not the only possible choice for this kind of optimization

problem, as shown by (SONG et al., 2014). On their work, the Greedy Algorithm was coupled with the Virtual Particle Model for the wake interaction. The idea behind the Greedy Algorithm is to find a local optimum on each stage of the solution aiming to find a global optimum. They concluded that the proposed method could avoid wake influence during the optimization process effectively and could be also used for flat terrain or offshore wind farms.

2.3 Optimization Software in the Market

As a common practice, most of the layout optimization software in the wind industry use a wind resource grid at hub height, which is typically generated in an external software. By using this 2D grid, important information such as the variation of the wind speed with height (also called wind shear) and the effect of the terrain in the wakes cannot be used in the optimization process, making it more simple in terms of the phenomena that are being captured.

The most known software for wind resource assessment in the market is WAsP (Wind Atlas Analysis and Application Program), developed by DTU (DTU Wind Energy, 2019). In this software, a Generalized Wind Climate is generated by removing the effects of the terrain (e.g. topography, roughness, surface heat flux, etc.) in the surrounding of the measurement points (typically, meteorological towers), creating a site-independent wind climate. This Generalized Wind Climate can be either extrapolated to the turbine locations or to a user-defined grid, which can then be used by the optimization software.

To perform this extrapolation, the inverse path is followed, by taking the terrain properties in the target point and applying to the Generalized Wind Climate. For horizontal and vertical extrapolation, WAsP uses a linear IBZ model, which according to the developers of this tool can adequately perform in flat and moderately complex terrain (DTU Wind Energy, 2019).

Acting as a modular software for planning and design, WindPRO can also perform resource grid generation and layout optimization (EMD International A/S, 2019). Although this functionality is available, the wind resource grid is either generated using WAsP in the background or read from an external file. The optimization module is simplistic, trying to place rows of turbines, changing their angles with respect to the wind flow and the turbine spacing. Another optimization option is the random layout optimization, where the optimizer tries to add turbines in the best available locations, without moving the already placed turbines. WindPRO can also be used to optimize noise operation modes.

With focus on the optimization, DNV GL's WindFarmer is a multi-variable tool, being able to consider the main costs of a wind farm (e.g cabling, roads, etc) in the optimization (DNV-GL, 2019). It has its own simplified flow model, but it is mainly used with an external wind resource file.

Another known software is WindSim, which is based on PHOENICS, the general purpose CFD code developed by Concentration, Heat and Momentum (CHAM) Ltd (CHAM, 2019). It solves the 3D Navier-Stokes equations using the Finite Volume Method considering a body-fitted mesh generated with control parameters defined by the user. In this software it is possible to calculate the turbulence intensity, velocity and pressure fields and also the inflow angle (angle between the wind velocity vector and the horizontal plane), and the user can also export a wind resource grid file to be used for layout optimization. There is also an optimization module called Park Optimizer, which can use the wind resource grid generated by WindSim combined with an analytical wake model.

Openwind, from UL Renewables, is one of the most known optimization software in the wind industry. It is user-friendly and versatile, allowing several variables to be considered in the same way as WindFarmer. It also has its own flow model, called Massconsistent model, which solves only the mass conservation equation to generate a 3D and divergence-free wind flow (UL Renewables, 2019). Another option is to use an external wind resource grid at the desired hub height from either WAsP, WindSim or other software that can export the data in the appropriate format. One advantage of Openwind is that it is easy to use and takes no longer that one or two hours to set up and run a case. It was not possible to find information related to the optimization method used in Openwind.

3 Theoretical Background

3.1 Atmospheric Boundary Layer

The application of CFD tools for wind farms has focused on the wind assessment over the terrain to obtain the undisturbed wind speed over a specific region. In this kind of application, it is necessary to take into account specific aspects related to the atmospheric behavior, once they have an important contribution to the accuracy of the solution.

Atmospheric Boundary Layer (ABL) is a concept that comes from meteorology and represents the lowest portion of the atmosphere, where most of the resources needed by human life are concentrated. In this layer it is possible to notice large gradients on physical quantities such as wind speed, temperature and humidity. The dynamics of this portion of the atmosphere is governed by the complexity of the topography, ground coverage (e.g. bushes, forests, lakes, deserts, cities) and region of the globe, which will have a significant impact on the temperature profile of the ABL.

According to Jacobson (JACOBSON, 2005), the ABL is defined as the region of the atmosphere between the Earth's surface and a 500 to 3000 m height that is influenced substantially by energy and moisture from the surface. Above the ABL there is a portion that could be called free atmosphere, where the most important governing phenomena are temperature and pressure variations on a planetary scale and the wind is called geostrophic wind. On this portion of the atmosphere, the geostrophic wind is approximately parallel to the isobar lines.

There are three basic classifications of the ABL regarding its thermal stability:

- Neutral: Turbulence is governed by mechanical phenomena and depends only on the vertical wind profile and surface friction due to the ground coverage. This is a rare case, but presents some important simplifications to solve the ABL problem. A logarithmic velocity profile is expected;
- Unstable: Due to the incidence of radiation from the sun, the air close to the surface receives heat and convection occurs. As the air is progressively heated during the day, buoyancy forces appear and stretch the eddies, mixing the air on the ABL and therefore a low wind shear is expected. This behavior normally occurs during the night;
- Stable: The surface-air interaction is governed by thermal and mechanical phenomena. In this case, convection is present due to buoyancy forces, but it is not

Potential Temperature Gradient	Classification
$\partial\theta/\partial z = 0$	Neutral
$\frac{\partial \theta}{\partial z < 0}$	Unstable
$\frac{\partial \theta}{\partial z > 0}$	Stable

Table 1 – Stability Classification According to the Potential Temperature Gradient

strong enough to stretch the eddies. For this reason, the atmosphere is stratified, and a high wind shear is expected. This behavior normally occurs during the day or in offshore cases.

One way to measure the atmospheric stability is by calculating the potential temperature θ , defined in Equation 3.1 as a function of the temperature T, a reference pressure p_0 , the local pressure p, the Universal gas constant R and the specific heat of air at constant pressure c_p .

$$\theta = T \left(\frac{p_0}{p}\right)^{\frac{R}{c_p}} \tag{3.1}$$

Each stability classification has a corresponding gradient of the potential temperature with respect to the height, as shown on Table 1.

3.2 Fundamentals of Wind Turbine Aerodynamics

Wind turbines are defined as devices that extract kinetic energy from the wind and convert it into electrical energy. Modern wind turbines are 3-bladed rotating machines with a horizontal axis, and the kinetic energy extraction is possible due to the aerodynamic profile of the blades. After the energy extraction, the wind slows down, creating a disturbed region downstream of the wind turbine.

A stream tube of circular cross section can be used to understand the flow disturbance created by a wind turbine. Assuming that the mass of air passing through the rotor do not mix with the outer air, the mass flow rate must remain constant inside the stream tube. The presence of the wind turbine makes the air slow down as it approaches the disk region, but since no work has been done there must be an increase of the static pressure to compensate for the velocity decrease, implying on an expansion of the stream tube.

When passing through the rotor, there is a static pressure drop which takes the pressure to a value below the atmospheric pressure. The region with reduced velocity and static pressure is called the wake of the wind turbine.

In order to obtain the pressure drop, velocity field and extracted power, it would be necessary to model the full 3D flow using a CFD code, including the rotation of the blades, nacelle, tower and other components. In fact, it could be done for a few wind turbines,

Figure 3 – Stream tube representation



Source: (JENKINS et al., 2001)

Figure 4 – Actuator Disk and Variation of variables



Source: (JENKINS et al., 2001)

but it would be unpractical to model an entire wind farm using this approach due to computational resource limitations. To overcome this complexity, simplified models can be used, such as Vortex Lattice, Panels Method, Blade Element Momentum (BEM), or even the ADM or ALM. Some of these methods, e.g. BEM, require the user to know detailed information about the blade geometry and profiles used, which is not easily obtained from wind turbine manufacturers. For this reason, the ADM was chosen for this work. Figure 4 illustrates the fundamentals of this models, which are explained in detail next.

From the balance of momentum between the free flow upstream and far enough downstream considering a uniform distribution of thrust and no rotation, the thrust force T can be calculated from Newton's second law (Equation 3.2) and the kinetic power P_{kin} can also be calculated from the extracted kinetic energy (Equation 3.3).

$$T = \dot{m}(U_{\infty} - U_w) \tag{3.2}$$

$$P_{kin} = \frac{1}{2}\dot{m}(U_{\infty}^{2} - U_{w}^{2})$$
(3.3)

In these equations, \dot{m} is the mass flow rate, U_{∞} is the free stream wind speed and U_w is the far wake wind speed. By applying Bernoulli's equation in the region upstream of the actuator disk:

$$p_{\infty} + \frac{1}{2}\rho U_{\infty}^{\ 2} = p_{\ d}^{+} + \frac{1}{2}\rho U_{d}^{\ 2} \tag{3.4}$$

In Equation 3.4, ρ is the air density, U_d is the wind speed at the disk, p_{∞} is the free stream pressure and $p^+{}_d$ is the pressure immediately upstream the disk region. The pressure immediately downstream the disk, $p^-{}_d$, can be calculated in terms of the pressure jump Δp and $p^+{}_d$, as $p^-{}_d = p^+{}_d - \Delta p$. By applying again Bernoulli's equation, now downstream of the disk, Equation 3.5 is obtained.

$$(p_{d}^{+} - \Delta p) + \frac{1}{2}\rho U_{d}^{2} = p_{\infty} + \frac{1}{2}\rho U_{w}^{2}$$
(3.5)

Subtracting Equation 3.5 from Equation 3.4, we obtain an expression for the pressure jump, shown in Equation 3.6.

$$\Delta p = \frac{1}{2}\rho(U_{\infty}^{2} - U_{w}^{2}) \tag{3.6}$$

The thrust force can also be calculated in terms of the pressure jump and disk area, A_d , as in Equation 3.7,

$$T = \Delta p A_d = \dot{m} (U_\infty - U_w) \tag{3.7}$$

which combined with Equation 3.2 results in an expression which states that the wind velocity at the disk is the average between the free stream wind speed and the far wake wind speed (Equation 3.8)

$$U_d = \frac{1}{2}(U_{\infty} + U_w)$$
(3.8)

By introducing the axial induction factor $a = 1 - U_d/U_\infty$, it is possible to express U_∞ and U_w in terms of a as $U_d = (1 - a)U_\infty$ and $U_w = (1 - 2a)U_\infty$, and to calculate the thrust and extracted power using Equation 3.9 and Equation 3.10.

$$T = \rho A_d U_{\infty}^{2} 2a(1-a) \tag{3.9}$$

$$P = \rho A_d U_{\infty}^{3} 2a(1-a)^2 \tag{3.10}$$

At last, the non-dimensional power and thrust coefficients c_T and c_P are defined as shown in Equation 3.11 and Equation 3.12.

$$c_T = \frac{T}{\frac{1}{2}\rho A_d U_{\infty}^2} = \frac{\rho A_d U_{\infty}^2 2a(1-a)}{\frac{1}{2}\rho A_d U_{\infty}^2} = 4a(1-a)$$
(3.11)
$$c_P = \frac{P}{\frac{1}{2}\rho A_d U_{\infty}^{3}} = \frac{\rho A_d U_{\infty}^{3} 2a(1-a)^2}{\frac{1}{2}\rho A_d U_{\infty}^{3}} = 4a(1-a)^2$$
(3.12)

From the definitions above, the Betz limit can be obtained by doing $d(c_P)/da = 0$, which leads to a maximum $c_P = 16/27 = 0.593$ for an induction factor a = 1/3. The Betz limit represents the maximum power that could be extracted by a wind turbine with open rotor.

3.3 Numerical Modeling

3.3.1 Incompressible Navier-Stokes Equations

The governing equations of an incompressible fluid flow are called Navier-Stokes equations, and express conservation of mass and conservation of linear momentum.

The mass conservation equation is obtained from a mass balance: the rate of change in mass of a fluid element must be equal to the net rate of flow of mass into this element (VERSTEEG; MALASKEKERA, 1995). By applying this balance, the mass conservation equation for an incompressible flow is obtained in Equation 3.13.

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{3.13}$$

To obtain the equation for the conservation of linear momentum, a momentum balance must be done, and for a Newtonian fluid this equation is given by Equation 3.14, where S_M represents the source terms (eg body forces).

$$\rho \frac{\partial u_i}{\partial t} + \rho \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) + S_M$$
(3.14)

3.3.2 Turbulence Model

For most engineering applications the interest is on time-averaged values (e.g. mean velocities, mean pressure, etc), so averaging processes can be carried out, being the RANS (Reynolds-averaged Navier-Stokes) the most common one. Another reason to use the RANS model is that the computational power needed to solve a fully turbulent flow is not available to most engineers. Nowadays the only cases that can be solved are low to moderate Reynolds flows.

The averaging process consists on decomposing the variables into its mean and a fluctuating values. For a general variable ϕ , $\phi = \bar{\phi} + \phi'$, where $\bar{\phi}$ is the mean term and ϕ' is the fluctuating term. By applying this process to the Navier-Stokes equation, the following result is obtained:

$$\rho \frac{\partial \bar{u}_i}{\partial t} + \rho \frac{\partial (\bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \rho \overline{u'_j u'_i} \right) + S_M$$
(3.15)

Table $2 - \kappa - \varepsilon$ turbulence model constant	ts
--	----

C_{μ}	σ_{κ}	σ_{ε}	$C_{1\varepsilon}$	$C_{2\varepsilon}$
0.09	1.0	1.30	1.44	1.92

It can be noticed that six additional unknown terms, $-rho\overline{u'_ju'_i}$, known as Reynolds stresses, appear as a result of the time-averaging, leading to a system with more variables than equations. To close this system more equations are needed and are obtained using turbulence models.

One of the most used turbulence models in both wind engineering and general applications is the $\kappa - \varepsilon$ model due to its robustness and moderate computational cost. This model adds two new variables, the turbulent kinetic Energy (TKE or κ) and the urbulent dissipation rate (ε).

Developed by Launder and Spalding (LAUNDER; SPALDING, 1974), the standard $\kappa - \varepsilon$ model is a two-equation model, one for κ and other for ε :

$$\rho \frac{\partial \kappa}{\partial t} + \rho \frac{\partial (\bar{u}_i \kappa)}{\partial x_i} = \frac{\partial}{\partial x_j} \left(\frac{\mu_t}{\sigma_\kappa} \frac{\partial \kappa}{\partial x_j} \right) + P_\kappa - \rho \varepsilon$$
(3.16)

$$\rho \frac{\partial \varepsilon}{\partial t} + \rho \frac{\partial (\bar{u}_i \varepsilon)}{\partial x_i} = \frac{\partial}{\partial x_j} \left(\frac{\mu_t}{\sigma_{\varepsilon}} \frac{\partial \varepsilon}{\partial x_j} \right) + C_{1\varepsilon} \frac{\varepsilon}{\kappa} P_{\kappa} - C_{2\varepsilon} \frac{\varepsilon^2}{\kappa} \rho$$
(3.17)

with

$$P_{\kappa} = -\rho \overline{u'_i u'_j} \frac{\partial u_j}{\partial x_i} \tag{3.18}$$

These two equations can be translated to words (VERSTEEG; MALASKEKERA, 1995):

Rate of		Transport		Transport		Rate of		Rate of
change of	+	of κ or ε by	=	of κ or ε	+	production	-	destruction
κ or ε		convection		diffusion		of κ or ε		of κ or ε

The relation between κ and ε appears on the definiton of turbulent viscosity (Equation 3.19).

$$\mu_t = \rho C_\mu \frac{\kappa^2}{\varepsilon} \tag{3.19}$$

Five constants are present on this turbulence model $(C_{\mu}, \sigma_{\kappa}, \sigma_{\varepsilon}, C_{1\varepsilon} \text{ and } C_{2\varepsilon})$ and can be adjusted according to the problem being solved. The standard values of these constants (VERSTEEG; MALASKEKERA, 1995) are shown in Table 2.

3.4 OpenFOAM

3.4.1 Basics

OpenFOAM stands for "Open Field Operation And Manipulation" and is the leading free and open source software for CFD. It is completely written on C++ language, which allows the user to customize it as needed with no restrictions and is based on the Finite Volume Method and offers several discretization schemes (The OpenFOAM Foundation, 2019).

The software started to be developed by H.G. Weller and H. Jasak in 1992 at Imperial College London. Since then many changes took place and OpenFOAM is now owned by ESI Group and operates under the GNU license.

It is also equipped with pre-processing utilities for mesh generation, mesh manipulation and conversion, post-processing functions from the *postProcess* or by launching Paraview, and there is a large diversity of RANS, LES and DNS solvers for incompressible, compressible, combustion and multi-phase flows, among others. In addition, it can be run in parallel on personal computers, workstations or even a cluster with a large number of processors.

A common sense in the OpenFOAM community is that the learning curve of this software is steeper that those of other commercial software due to the lack of a Graphical User Interface (GUI) and detailed user guide. However, in counterpart, it has several tutorial cases that can be used as a starting point for a wide range of problems and also has a large community of users.

3.4.2 Case Structure

As there is no GUI, the user must navigate through several folders and files to set up a problem. A case is subdivided in some specific folders where the input files are placed. A typical folder structure used in this work is presented in Figure 5.

Three main folders are present in all cases run with OpenFOAM and must be highlighted:

- "0": this folder contains the boundary conditions' values for all involved variables. A subfolder is "include" and it contains information about the ABL profile and parameters for the initial conditions of all variables in the file "initialConditions";
- "constant": here the simulation type will be defined (RANS, LES, DNS) as well as the properties of the fluid. Additionally, the file "fvOptions" stores all the information about source terms. In this study the stored information is the rotor

Figure 5 – Typical Folder Structure



orientation, c_p , c_t , rotor swept area and c_t curve. All the mesh files are saved in the subfolder "polyMesh" after the mesh is generated;

• "system": it is responsible for all the controls of the simulation, such as time step, number of time steps, writing format, post processing, mesh generation parameters, convergence criteria, numerical schemes, parallelization. It also controls the coordinates, geometry and type of the source terms (file "topoSetDict").

If a simulation is run in parallel, a folder will be created for each processor with the name *processorn*, where *n* is the number of the processor, starting with zero. Each of these folders holds information about the part of the domain that will be solved by the processor, including the mesh, and boundary conditions to communicate with the other processors is also created for information exchange.

3.5 Optimization Methods

3.5.1 Dakota software

Creating an optimization software is out of the scope of this work, so we decided run the optimization calculations using an open source code. After researching for integration between open source optimization codes and OpenFOAM, Dakota (ADAMS et al., 2014) was considered the most suitable software.

Dakota is an open source software developed and released by Sandia National Laboratories (USA) and it was originally developed to provide a common set of optimization tools for a group of engineers solving structural analysis and design problems at Sandia in 1994. Today it has a variety of iterative algorithms. The most important available classes are Parameters Studies, Optimization, Uncertainty Quantification, Design of Experiments and Calibration. The software was also designed to be easily coupled with other simulation codes, especially ones that can be run on the Shell or Command Prompt.

In this work only the Optimization class was used and, among the several optimization methods available, the most suitable for this problem was a Genetic Algorithm. This method was used by Bonanni et al. (BONANNI et al., 2012) and Schmidt and Stoevesandt (SCHMIDT; STOEVESANDT, 2015) because it is a derivative-free method that can solve non-smooth problems. The optimization method that was used is called SOGA, which stands for Single-Objective Genetic Algorithm. It is a global optimization method that supports general constraints and a mixture of real and discrete variables. SOGA is part of the JEGA library available in Dakota.

3.5.2 Genetic Algorithm

The basics of every evolutionary algorithm is to emulate the natural process of evolution by population reproduction, selection of the most fitted individuals and application of some natural processes, such as crossover and mutation. Starting from an initial population, the value of the objective function for each individual will be computed, and the best fitted will be selected for reproduction in order to generate a new population and repeat the process. Each individual has its characteristics represented by a finite set of parameters (commonly referred as a chromosome), which will be exchanged (cross over) and eventually changed (mutation).

Cross over is the main mechanism acting on the production of new individuals because in this phase the child chromosome will be formed from a mix between each of the parent's chromosomes. The probability of cross over is generally high (about 0.8 to 0.95). Mutation is the process where random parts of the chromosomes are changed to try to avoid sterilizing a population and getting trapped on a local optimum. This mechanism has a much smaller probability of occurring when compared to cross over (about 0.1).

One of the advantages of this method is that a final set of optimal designs will be reached, which means that there may be more than one option for the designer to choose, giving an additional flexibility for the optimization process.

Although it may seem a simple random search, genetic algorithms are randomized and exploit historical information of past populations in order to generate new search points. For detailed information about Genetic Algorithms, the reader is encouraged to read the book of Goldberg (GOLDBERG, 1989).

4 Numerical Modeling

4.1 Actuator Disk Model and Modifications

As shown before, the actuator disk has its origins in the 1D momentum theory. A thrust distribution must be chosen according to the problem to be solved or complexity of the model, and the three types of distribution available and commonly used in OpenFOAM (The OpenFOAM Foundation, 2019) are presented below:

• Uniform distribution: also referred as uniformly loaded disk, this distribution assumes that the total thrust force, calculated by

$$T_{total} = c_t \frac{1}{2} \rho U_\infty^2 A_d \tag{4.1}$$

is equally distributed in all points of the rotor. In OpenFOAM, the class *actua-tionDiskSource* is responsible for including this type of source term;

• Polynomial: this distribution is constructed by using a polynomial distribution of the total thrust force calculated by Equation 4.1, which is then distributed along the turbine radius according to

$$T(r_i) = \frac{(c_0 + c_1 r_i^2 + c_2 r_i^4)}{c_0 + c_1 \frac{R^2}{2} + c_2 \frac{R^4}{3}}$$
(4.2)

where c_0, c_1 and c_2 are coefficients chosen by the user, r_i is the distance from a point at the rotor plane to its center and R is the rotor radius. The class responsible for applying this type of source is called *radialActuationDiskSource*;

• Profile-based: this approach is based on the BEM theory, which uses information from the aerodynamic profile of the blades to calculate the forces acting on the rotor. By using this model, it is possible to vary the number of blades, use a tip effect correction and consider geometrical characteristics of the blades, e.g. twist angle and profile chord of each section of the blade. The main difficulty of using this model is that the required information about the blades is confidential and is hardly shared by the wind turbine manufacturers of the market. Due to this reason, this model was not employed in this work. The corresponding class in OpenFOAM is called *rotorDiskSource*.

Once the uniform distribution presents a simplified approach to the rotor modeling, it was chosen to be used in this work. A piece of information required by this model is a point upstream the turbine rotor where the wind velocity will be sampled to calculate the total thrust according to Equation 4.1. The idea is to select a point where the wind is not disturbed by any obstacles or other wind turbines so that the correct thrust can be calculated, and as the selection of this point does not have a mathematical representation there is a chance that this point could be located on a disturbed region. To avoid this issue, a set of modifications proposed by Simisiroglou et al. (SIMISIROGLOU et al., 2016) were used on the current work.

Thrust curves provided by wind turbine manufacturers are typically a table of $U_{\infty} \times c_T$, where U_{∞} is the undisturbed upstream wind speed. The first proposed modification was to adapt this information by using the wind speed at the rotor instead of the undisturbed wind speed. From the actuator disk theory, Equation 3.11 shows the definition of c_T . Solving this equation for the axial induction factor a results in

$$a = \frac{1}{2}(1 - \sqrt{1 - c_T}) \tag{4.3}$$

The wind velocity at each point of the disc U_n , which is the wind velocity normal to the disk and here it has the same meaning as U_d , can be estimated by combining the axial induction factor definition from Equation 4.4 with Equation 4.3 and it is based only on the free wind velocity U_{∞} and c_T curve.

$$a = \frac{U_{\infty} - U_n}{U_{\infty}} \tag{4.4}$$

$$U_n = U_\infty \left(1 - \frac{1}{2} (1 - \sqrt{1 - c_t}) \right)$$
(4.5)

Equation 4.5 shows the relation between the thrust coefficient and the free wind speed. This equation makes the second proposed modification possible, which is to calculate a thrust force on each individual cell by checking its velocity and finding the corresponding c_T .

The ADM theory assumes that the rotor region is represented by an infinitely thin disk, which could only be represented in a CFD mesh if the disk was located on the face of the selected cells to have the source term. Additionally, the area of this specific face would need to be known to calculate the thrust on this face. One way to overcome this issue is to distribute the total thrust considering the ratio, so the problem of accessing individual faces' area is avoided. OpenFOAM natively does it, so we decided to modify the already existing class *actuationDiskSource* so that no additional class needs to be created.

After that, the thrust force T_i acting on each cell can be calculated using Equation 4.6

$$T_i = c_{T,i} \frac{1}{2} \rho \left(\frac{U_{n,i}}{1-a_i}\right)^2 A_{disk} \frac{V_i}{V_{disk}}$$

$$\tag{4.6}$$

where $c_{T,i}$ is the thrust coefficient found from the modified thrust coefficient curve, $U_{n,i}$ is the wind velocity on the disk, A_{disk} is the swept area of the disk and V_{disk} is the volume of the actuator disk region (the subscript *i* denotes each cell of the actuator disk). The main interest is on the power extracted from the wind by the wind turbine, which will be used to calculate the AEP, and can be obtained from the sum of each individual cell's power according to its definition in Equation 4.7.

$$P_{out} = \sum_{i=1}^{k} P_i = \sum_{i=1}^{k} T_i U_{n,i}$$
(4.7)

4.2 Boundary Conditions

To model turbulent flows appropriately, in addition to using the equations shown in Section 3.3, adequate boundary conditions must be used to completely represent the relevant phenomena. As previously observed by (HARGREAVES; WRIGHT, 2007), the atmospheric boundary layer is not maintained on a fetch domain and some modifications on the boundary conditions are needed in order to maintain it. These modifications were initially proposed by Richards and Hoxey (RICHARDS; HOXEY, 1993) and are already implemented on OpenFOAM by default for the $\kappa - \varepsilon$ turbulence model.

On the inlet, it is assumed that the vertical wind profile follows a logarithmic law according to Equation 4.8.

$$U(z) = \frac{U^*}{\kappa} \ln\left(\frac{z - z_g + z_0}{z_0}\right) \tag{4.8}$$

$$U^* = \kappa \frac{U_{ref}}{\ln\left(\frac{z_{ref} + z_0}{z_0}\right)} \tag{4.9}$$

where U^* is the friction velocity calculated on Equation 4.9, κ is the Von Karman's constant, z_g is the minimum z coordinate, z_0 is the surface roughness height, U_{ref} is the reference wind velocity at height z_{ref} . In the same fashion, the turbulent kinetic energy κ and the dissipation ε need to be modified as shown on Equation 4.10 and Equation 4.11.

$$\kappa = \frac{(U^*)^2}{\sqrt{C_\mu}} \tag{4.10}$$

$$\varepsilon = \frac{(U^*)^3}{\kappa(z - z_g + z_0)}$$
(4.11)

Finally, the wall function for the terrain surface will consider the ground to be fully rough. This boundary condition is designed to be used in conjunction with the ABL inlet conditions and the velocity parallel to the ground cells U_w is calculated by Equation 4.12.

$$U_w = \frac{U^*}{\kappa} \ln\left(\frac{z+z_0}{z_0}\right) \tag{4.12}$$

It was stated by (RICHARDS; HOXEY, 1993) that a constant shear stress must be applied at the top of the domain so that the wind vertical profile does not suffer a decay along the streamwise direction. As in most of the applications this suggestion is ignored, a slip condition was applied in this work. A slip boundary condition was also applied on the sides of the domain. In the outlet, a pressure boundary condition was applied.

4.3 Optimization Problem

This work is focused on a layout optimization, so the chromosome of each individual contains a set of coordinates that represents the locations of the wind turbines on the xyplane and the objective function, F_{obj} , of the optimization is the Annual Energy Production (AEP) of the whole wind farm, which can be calculated using Equation 4.13 and summing over all n wind turbines and m direction sectors. In this equation, P_j is the kinetic power extracted from the wind on each turbine and f_j is the frequency distribution of the wind in each direction sector.

$$F_{obj} = 8766 \sum_{i=1}^{n} \sum_{j=1}^{m} P_j f_j$$
(4.13)

This is a maximization problem, so the optimization problem definition can be written as in Equation 4.14, where (x_i, y_i) is the coordinate of each turbine and d_i is the inter turbine distance, which should be equal or greater than the minimum values specified by the user, d_{min} , calculated in the xy plane.

$$\begin{array}{ll} \underset{x_i, y_i}{\text{maximize}} & F_{obj} = 8766 \sum_{i=1}^{n} \sum_{j=1}^{m} P_j(x_i, y_i) f_j \\ \text{subject to} & d_i \ge d_{min}, \ i = 1, \dots, n \end{array}$$

$$(4.14)$$

Wind farms are highly subjected to have wind turbine positioning constraints and care must be taken, otherwise the project may not be approved for the registration with the governmental agencies and construction. The most common constraints are environmental, archaeological, land ownership, generated noise, shadow flickering, neighbor wind turbines and spacing. Considering a whole set of restrictions would be very complex and time demanding, thus, to simplify the problem, the only constraint considered was a minimum inter turbine spacing. This also avoids having overlapping or very close actuator discs, both causing an ill treatment of the problem by the CFD calculations.

As a market good practice, this minimum inter turbine distance must be between 2 and 3 rotor diameters in order to avoid high turbulence zones produced by the wind turbine wake, so if any individuals had not respected this constraint it was immediately considered as unsuitable. It is also recommended to keep at least 8 to 10 rotor diameters between wind turbines on the downstream direction so that reduced turbulence levels can



Figure 6 – Domain and Optimization grid representation

be reached, but this is not a constraint of the problem because this result is expected to be obtained naturally from the optimization process.

We chose to solve a discrete problem due to the reduced search domain, which lead to a reduced computational cost when compared to a continuous problem. To do that, an area of approximately $1.5 \text{ km} \ge 1.5 \text{ km}$ on the xy plane inside the domain was selected as where the wind turbines could be placed and then this selected area was discretized on a 10 ≥ 10 grid. The prior selection of a restricted area inside the domain aimed to avoid interference of the mesh on the actuator disc region and the boundaries and also to keep a certain distance from the domain boundaries.

A representation of a typical domain, restricted area and discretized grid can be seen on Figure 6, being the vertices the possible locations of the wind turbines. This figure represents a horizontal slice of the domain and it is worth noting that each cell of the 10×10 cells grid has $165 \text{ m} \ge 165 \text{ m}$.

The wind turbine model chosen for this work was a SG 2.1-114 from Siemens Gamesa Renewable Energy, which has a 114 m rotor diameter and 2.1 MW of rated power, with a chosen hub height of 120 m. This hub height was fixed, while the quantity of wind turbines was specified for each case.

The optimization grid dimensions were the same for all the cases and according to its dimensions and the turbine rotor diameter, it was noticed that the turbines cannot be aligned in the x or y axis in neighbor grid cells, only "V" or diagonal shapes will respect the imposed inter turbine spacing constraint. This setup was intentional, because although there is a stronger restriction in the spacing, more complex turbine arrangements are



Figure 7 – Optimization Workflow

Source: Author

possible, as it will be seen in Chapter 5.

4.4 Optimization Workflow

Neither OpenFOAM nor Dakota have a Graphical User Interface (GUI) and the coupling between them must be done by coding using their native programming language. Additional Python scripts were written to manage input files and case folders and also to process the output from Dakota and OpenFOAM. A workflow of the optimization process may be helpful to understanding how the optimization process is and it can be seen in Figure 7.

Initially, an input file with all the optimization parameters is read by Dakota, initializing the process with a randomly generated population that respects the inter turbine spacing constraint. This population could be generated by Dakota via a seed number specification, but it was observed that as the number of wind turbines increased, this process became inefficient and could not find a suitable initial population. To solve this issue, a Python code was written so that a randomly generated initial population could be generated an inputted by the user.

In case any individual do no respect the imposed constraint, during the preprocessing the value of its objective function will be set to zero, and the optimizer will see it as a bad candidate. It is expected that, as the optimization goes on, this kind of layouts will not be suggested by Dakota anymore. The pre-processing is done using Python scripts, which involves preparation of input files for the CFD cases related to the mesh, boundary condition, location of the turbines in the mesh, turbine properties, orientation of the turbines and calculation of the altitude of each turbine location. After this, all the files and shell scripts are already generated and the CFD analysis is run for each case, considering a single direction, multiple directions or multiple wind velocities, depending on the case setup. In order to save time during the optimization, an initial case is run without considering the presence of wind turbines, and this solution is used as the initial condition in each case.

The convergence criteria can either be set up in Dakota based on the maximum number of iterations allowed or be based in the user's observation of the results (evolution of populations, result of each iteration, layout achieved, etc.). In case the maximum number of iterations is reached, the optimization stops and the best population available is checked, along with the best individual of this population.

5 Wind Farm Layout Optimization Scenarios

5.1 General Considerations

In order to test the optimization methodology, three different terrain cases were proposed, being them a flat terrain, a synthetic Gaussian hill and the Askervein Hill (TAYLOR; TEUNISSEN, 1987). The optimization grid with possible turbine locations has the same dimensions as shown in Figure 6, changing the domain size according to the topography under test.

For the following results, the quantities presented in Table 3 were not changed, unless otherwise specified, to try to keep each case's conditions as similar as possible. Each optimization case was examined maintaining the domain and increasing the number of wind turbines. The cases are typically constituted by 12, 14, 16, 18 and 20 wind turbines and for such cases, only the results for 12 and 20 turbines will be presented. The results for the other configurations can be found in Appendix A.

It should be noticed that the air density considered for both cases is different from the 1.225 kg/m^3 of the standard air density. The incompressible solver simpleFoam solves all the equations divided by the fluid density, so an air density of 1.0 kg/m^3 is convenient, because no further calculation from the physical parameters must be done.

Additionally, the $\kappa - \varepsilon$ turbulence model was used with its standard coefficients, unless otherwise specified, except for σ_{ε} , which was changed according to Richards and Hoxey (RICHARDS; HOXEY, 1993). The coefficients employed are shown in Table 4.

Rotor diameter []m]	114
Huh height [m]	120
Air density $[kg/m^3]$	1.0
Roughness [m]	0.03
Reference wind speed [m/s]	8.0
Domain radius [km]	2.0
Domain height [m]	600
Cross over probability	0.95
Mutation probability	0.10
Population size	20

Table 3 – Common parameters in the optimization cases.

Table 4 – $\kappa - \varepsilon$ turbulence model constants for the optimization cases

C_{μ}	σ_{κ}	σ_{ε}	$C_{1\varepsilon}$	$C_{2\varepsilon}$
0.09	1.0	1.11	1.44	1.92



Figure 8 – Block structure of the domain (a) and domain mesh representation

Regarding the Dakota setup, the optimization method used was the aforementioned SOGA using discrete variables, with a randomly generated initial population and without any imposed constraint within Dakota. The inter turbine distance constraint was verified by an external script as mentioned before.

5.2 Mesh Characteristics

OpenFOAM has two commonly used mesh generators: *blockMesh*, which generates block structured meshes, and *snappyHexMesh*, which generates meshes with hexahedral and split-hexahedral elements from triangulated surfaces. The structured block mesh is used as a background mesh and the triangulated surface will be used to snap the domain.

Another functionality of snappyHexMesh is to refine the mesh close to surfaces or inside defined regions. Such resources were used to apply an initial refinement of the mesh close to the terrain and in the region where the turbines would be located. This refinement is defined as a level, where the level number means that the elements in those regions will be divided 2^{level} times in all directions. In this study, a level 2 refinement was used, which means that the cell will be approximately 4 times smaller than the background mesh inside these regions.

A particular script was developed to generate the input file for *blockMesh* to produce the background block structured mesh in such a way that the most common number of wind directions (1, 4, 12 and 16) can be managed automatically, without any user input, by using 48 patches on the side of the domain that are combined into inlet and outlet conditions depending on the specific direction that is being evaluated. An example of the block structure is presented in Figure 8a and the mesh in the domain is shown in Figure 8b. Figure 9 shows a slice in a plane cutting one of the disks, illustrating the initial mesh refinement close to the surface and in the disk region for the Askervein Hill case.









Source: Author

The central square block is set up to have 2 km x 2 km and is where the turbines will be located. In all meshes, 20 elements were used in the z direction and the number of elements in the other directions are defined individually in each block and will be shown in each case. Each disk has around 12 to 16 cells along its diameter, which is considered an appropriate value (SIMISIROGLOU et al., 2016). No external mesh generator was used in this work. The selected cells and mesh on the vicinity of the actuator disk are exemplified in Figure 10.

A mesh refinement study was carried out and is presented in Appendix B. The conclusion was that the meshes we utilized in the calculations were enough to reach the goals of this work with satisfactory accuracy.

5.3 Results

5.3.1 Case 1: Flat terrain and unidirectional wind

The first test is composed by a flat terrain, chosen due to its simplicity and, at the same time, capability of testing some fluid dynamics and optimization combined results such as wake of upstream turbines avoidance and the ability of Dakota to deal and learn

Figure 11 – Objective function versus evaluations (a) and populations (b) for Case 1 with 12 turbines



Source: Author

from the external constraint imposed. The $\kappa - \varepsilon$ constants C_{μ} and σ_{ε} used in this case were 0.033 and 1.83, respectively. The incoming wind direction is negative y, equivalent to Sector 1 in a wind rose.

The background mesh of this case has 92160 elements and the final refined mesh has about 125000 elements (this number will slightly change according to the number of turbines in the domain).

The objective function value of each layout evaluation carried out by OpenFOAM is shown in Figure 11a and Figure 12a for 12 and 20 turbines, respectively. It is noticed that in the beginning of the optimization just a few individuals are shown as a consequence of the inter turbine spacing constraint, while in the end of the optimization there is a cloud of points. This means that although the spacing constraint was not explicit in Dakota, the algorithm was able to understand that this scenario is not favorable for the optimization and stopped suggesting this kind of configuration.

It is also noticed that between evaluation 1500 and 2400 there is no significant change in the objective function value for the scenario with 12 turbines, with a similar behavior for the scenario with 20 turbines. This can be understood as an indication of convergence. Next, the best fitted individual of each population is presented in Figure 11b and Figure 12b. A similar behavior can be seen in the populations, showing that the optimization can be considered as converged after approximately population 300 for 12 turbines and 600 for 20 turbines, since there is no change in the objective value after that. The total number of evaluations or populations does not necessarily means more simulation time to converge, once the individuals that do not fulfil the imposed constraint are included in the evaluations count and populations.

As a visual inspection in the CFD simulation, Figure 13 shows the wind velocity





Source: Author

Figure 13 – Wind velocity at 120 m from the ground of the best individual for Case 1 with (a) 12 turbines and (b) 20 turbines



values at 120 m from the ground on each point for the best individual of the optimization. It can be noticed that the optimization algorithm was able to avoid putting turbines in the wake of others and the turbines are placed as far as possible from each other in the wind direction. This secondary spacing is usually a constraint in other optimization software, but it was decided to not force it and let the tool decide the optimal spacing in this direction. Due to the spacing constraint and the discrete optimization grid dimensions, the "V" pattern is clearly seen.



Figure 14 – Wind rose with sixteen sectors

Source: Author

Table 5 – Frequency distribution for sixteen sectors wind rose.

Sector	Freq $(\%)$
S1	2
S2	4
S3	16
S4	10
S5	2
$\mathbf{S6}$	2
S7	2
S8	2
S9	8
S10	20
S11	14
S12	8
S13	4
S14	2
S15	2
S16	2

5.3.2 Case 2: Flat terrain and sixteen wind directions

For this case, a commonly used wind rose with sixteen direction sectors was considered for the same domain dimensions and mesh characteristics. This is a hypothetical wind rose with two prevailing directions, but with frequencies in all sectors, as shown in Figure 14 and in Table 5.

The objective here is to see how the optimization algorithm would deal with the wakes from prevailing and non-prevailing wind directions. The sectors are assumed to have the same inlet wind speed at hub height. The $\kappa - \varepsilon$ constants C_{μ} and σ_{ε} used in this case





Source: Author

Figure 16 – Objective function versus evaluations (a) and populations (b) for Case 2 with 20 turbines



Source: Author

are 0.033 and 1.83, respectively. As there is no change in the domain size and terrain, the mesh characteristics are the same as Case 1.

The orientation of the disk is fixed for each direction and the disk will be aligned with the direction sector instead of with the local flow as it happens with real turbines. This is considered a good approximation, once this level of accuracy is not needed for the optimization.

The same analysis of Case 1 is done here, and a similar evolution of the evaluations and populations is seen, as shown in Figure 15 and Figure 16. Figure 17 and Figure 18 shows the wind velocity at 120 m from the ground for the prevailing sector (S3 and S10) for both 12 and 20 turbines. From these results, more complex phenomena can be observed, such as wake deflection and wake merging.

Figure 17 – Wind velocity at 120 m from the ground of the best individual for Case 2 with 12 turbines and wind from (a) Sector 3 and (b) Sector 10



Source: Author

Figure 18 – Wind velocity at 120 m from the ground of the best individual for Case 2 with 20 turbines and wind from (a) Sector 3 and (b) Sector 10



In order to see how big the wake losses are, it is possible to use the specific AEP of each configuration and an insulated turbine in the domain as a reference value. This single turbine has an AEP of 12.2 GWh, so if we define the specific AEP as the total AEP divided by the number of turbines, the results shown in Figure 19 are obtained.

It was concluded that in Case 1 and Case 2, there is an almost linear increase of the wake losses and decrease of Specific AEP as turbines are added. One may think that by introducing more wind directions, the specific AEP should be smaller, because the wake losses would be higher, which is not always the case. This affirmation may be correct if we compare a wind rose with 1 and 16 sectors in a layout that can have a single string (or row) shape; in this scenario the wake losses from other directions different from the



Figure 19 – Specific AEP and Wake Losses for Cases 1 and 2





Source: Author

prevailing could be enormous. As the optimization algorithm was free to move the turbines in the desired shape, it was capable of finding a solution that does not have wakes all the time, reducing the overall losses.

5.3.3 Case 3: Synthetic Hill and unidirectional wind

In this case, we tested a terrain with a synthetic hill in the centre. The hill had the shape of a Gaussian function, which gave the height (z coordinate) as a result of a multiplication of two exponential functions, as shown in Equation 5.1.

$$z(x,y) = \begin{cases} e^{-1/(1-x^2)}e^{-1/(1-y^2)} & , -1 < x, y < 1\\ 0 & , |x,y| > 1 \end{cases}$$
(5.1)

The altitude of the generated terrain is shown in Figure 20.

Figure 21 – Objective function versus evaluations (a) and populations (b) for Case 3 with 12 turbines



Source: Author

Figure 22 – Objective function versus evaluations (a) and populations (b) for Case 3 with 20 turbines



Source: Author

The background mesh of this case has the same number of elements as before, but the final mesh now has about 290000 elements due to the presence of the terrain, which required some further refinement in this region.

As in the previous cases, Figure 21 and Figure 22 show the evaluations and populations evolution for the cases with 12 and 20 turbines. Again, a behavior similar to the previous cases was observed. The incoming wind direction is negative y, equivalent to Sector 1 in a wind rose. By visualizing the layout arrangement and the wind velocity field at 120 m from the ground from Figure 23, it is seen that there is a concentration of turbines in the center of the domain, where the wind velocity is higher, and it is also noticed that there is an interaction between the disturbed region upstream the turbine and the terrain.





Source: Author

Table 6 – Frequency distribution for four sectors wind rose.

Sector	Freq $(\%)$
S1	30
S2	60
S3	7
S4	3

5.3.4 Case 4: Synthetic Hill and four wind directions

Adding one level of complexity, a hypothetical wind rose with four wind directions was created to simulate this scenario, with the same inlet wind speed at hub height for all directions, domain size and mesh characteristics. This wind rose can be seen in Table 6 and in Figure 24. The mesh of this case was similar to Case 3, once the domain was the same and the terrain was not modified.

Again, the same behavior in the evolution of the evaluations and populations is observed and is shown in Figure 25 and Figure 26. In some cases, as in Case 4 with 12 turbines, the populations stopped evolving after population 250 and the improvement in the objective function was smaller after each step, so it was assumed that the optimization had converged and, in case there was still some gain, it was expected to be small, justifying finalizing the case.

The wind resource at 120 m height is shown in Figure 27 for the prevailing wind direction (Sector 2). Once more, it is noticed that there is an interaction between the terrain and the wakes, being the latest deflected in some specific locations. It is also noticed that the wake recovery is much faster for the turbines in higher wind velocity locations, located in the center of the domain.



Figure 24 – Wind rose with four sectors



Figure 25 – Objective function versus evaluations (a) and populations (b) for Case 4 with 12 turbines



Source: Author

5.3.5 Case 5: Synthetic Hill and sixteen wind directions

As a final test for the Synthetic Hill case series, the same domain and mesh characteristics of Cases 3 and 4 were tested with the sixteen sectors wind rose from Case 2. The evolution of the evaluations and the populations is shown in Figure 28 and Figure 29.

It is possible now to compare Cases 3, 4 and 5 in terms of the specific AEP and wake losses, also considering a single turbine in the center of the domain as reference, with an AEP of 15.2 GWh, and the comparison is presented in Figure 30. The case with the worst results in terms of specific AEP and wake losses was Case 4, although this was not an expected result in view of the previous comparison of Cases 1 and 2. This result can be explained by the non-uniform wind resource due to the presence of the terrain and to the different wind resource grid that is obtained for each sector (once the terrain is





Source: Author

Figure 27 – Wind velocity at 120 m from the ground of the best individual for Case 4 with (a) 12 turbines and (b) 20 turbines



not symmetric in all directions), which does not happen for a flat terrain. Under these circumstances, this result was considered a consequence of the wind rose frequencies and is understood as correct.

Another result from this comparison is that Case 5 is more efficient than Case 3 and 4 for a scenario with more turbines, but is less efficient than Case 3 for scenarios with less turbines. This leads to the conclusion that as the number of turbines increase, having more sectors in the wind rose could bring an advantage to the optimization.

In the sequence, the wind resource at 120 m from the ground is presented in Figure 31 and Figure 32 for the prevailing directions (Sectors 3 and 10).

Figure 28 – Objective function versus evaluations (a) and populations (b) for Case 5 with 12 turbines



Source: Author

Figure 29 – Objective function versus evaluations (a) and populations (b) for Case 5 with 20 turbines



Source: Author

5.3.6 Case 6: Askervein Hill and one direction

After testing two simple topographies, it is now time to test a real terrain, known as the Askervein Hill, which is located Northwest of Scotland and has an elliptical shape of 2 km in the major axis and 1 km in the minor axis, with approximate coordinates 57° 11' N, 7° 22' W (TAYLOR; TEUNISSEN, 1987). The simulation domain around the Askervein hill can be seen in Figure 33 and it has 2.8 km of radius and 800 m of height. This site is considered to be a complex terrain. The background mesh of this case has 130560 elements and the final refined mesh has about 370000 elements, which will slightly change according to the number of turbines in the domain.

Figure 34a shows the objective function of each evaluation, Figure 34b shows the best fitted individual of each population and finally Figure 35 shows the visual inspection



Figure 30 – Specific AEP and Wake Losses for Cases 3, 4 and 5

Source: Author

Figure 31 – Wind velocity at 120 m from the ground of the best individual for Case 5 with 12 turbines and wind from (a) Sector 3 and (b) Sector 10



Figure 32 – Wind velocity at 120 m from the ground of the best individual for Case 5 with 20 turbines and wind from (a) Sector 3 and (b) Sector 10



Source: Author

Figure 33 – Askervein hill area colored by altitude values



of the CFD simulation for this terrain. The incoming wind direction is 210° in a wind rose, equivalent to 240° in the trigonometric circle. This case was tested only with 12 turbines.

A convergence behavior similar to the previous results was noticed in this case, showing that the optimization reached a converged state after approximately evaluation 2050, corresponding to population 195. Figure 35 shows the wind velocity and total pressure neglecting the hydrostatic pressure term at 120 m from the ground for this case, for a better visualization of the waked region.

5.3.7 Case 7: Askervein Hill and one direction with Weibull distribution

A variation of Case 6 is now tested, considering a unidirectional wind for the same domain dimensions and mesh characteristics as Case 6, but with a set of different inlet wind velocities following a Weibull distribution. This distribution was generated based on the average wind velocity of 8 m/s and a k factor of 2.0, grouping the frequencies in the wind velocities 5, 7, 9, 11 and 13 m/s and normalizing the frequencies for those wind



Figure 34 – Objective function versus evaluations (a) and populations (b) for Case 6

Source: Author

Figure 35 – Wind velocity (a) and total pressure (b) at 120 m from the ground of the best individual for Case 6



speeds according to Figure 36. The numerical values are shown in Table 7 and the same incoming wind direction of Case 6 was considered.

Once more, it is possible to see in Figure 37 that the optimization algorithm performed as expected, although now the wind direction is fixed and the wind velocity is changing in magnitude, reaching a converged state after evaluation 2100 and population 160. The optimal distribution of turbines can be seen in Figure 38 for the 9 m/s wind speed, as well as the total pressure. This case was tested only with 12 turbines.

As observed before, there is a quick wake recovery in regions with high wind speed, in this case mainly on top of the hill, and the algorithm once more performed as expected, putting turbines in regions with higher wind speeds, but avoiding waked regions. It is interesting to notice that the turbine locations are clearly different from that obtained in



Figure 36 – Original and normalized wind distribution

Table 7 – Normalized wind distribution for Case 7.

V [m/s]	Freq $(\%)$
5	25
7	26
9	22
11	16
13	11

Figure 37 – Objective function versus evaluations (a) and populations (b) for Case 7



Source: Author

Figure 38 – Wind velocity and total pressure at 120 m from the ground of the best individual for Case 7 and $9 \,\mathrm{m/s}$



Figure 39 – Wind rose with sixteen sectors of the MERRA node.



Case 6, indicating that the distribution of wind speed is important, even if we consider only one wind direction.

5.3.8 Case 8: Askervein Hill and sixteen wind directions with varying wind speed

As a final test, the Askervein Hill was put closer to a real case, with sixteen wind directions and a specific wind speed for each sector, obtained from the Modern-Era Retrospective analysis for Research and Applications Version 2 (MERRA2) (GELARO et al., 2017) node 57.0 °N 7.5 °W at 50 m height in a hourly time series format. This series has almost 39 years of measurements for this point, but it was decided to use 20 years of measurements, from 1998 to 2018, for this test. Figure 39 shows the wind rose of this case and Table 8 shows the frequency and wind speed of each sector.

Sector	Freq $(\%)$	V [m/s]
S1	5.3	8.50
S2	4.7	8.10
S3	3.5	7.44
S4	2.8	7.27
S5	2.5	6.93
S6	3.6	8.17
S7	5.8	9.65
S8	7.4	10.31
S9	8.3	10.49
S10	10.1	10.90
S11	10.4	10.99
S12	10.1	11.02
S13	8.9	11.09
S14	6.6	10.10
S15	4.9	8.79
S16	5.0	8.52

Table 8 – Frequency distribution for sixteen sectors wind rose of the MERRA node.

In this case, and also for most of the real wind farms, there is a particular situation that had to be addressed, related to how OpenFOAM calculates and inserts the ABL in the inlet of the domain. The ABL inlet condition in OpenFOAM allows the user to provide one single value for the reference height and one single value for the ground altitude z_g , which is the same as having a terrain with constant altitude in the inlet.

For this reason, a procedure to smooth the surrounding of the Askervein Hill to a constant values was developed, making use of the shape of the Error function (also called the Gauss error function, or erf(x)) specially because of the two inflexion points present in this function, which leads to a smooth transition. This function is defined in Equation 5.2 (ANDREWS, 1998) and plotted in Figure 40.

$$erf(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt, -\infty < x < \infty$$
 (5.2)

It is necessary to define first a smoothing region, where the height values will be modified. If we define the inner circle with a radius R, where the terrain will not be modified, and an outer circle with radius R + r, we will have a circular ring region between these two circles where the terrain will be smoothed, as shown as a gray area in Figure 41. It was decided to calculate the new heights z_{smooth} by applying a scale and an offset factor to the erf function, as is Equation 5.3.

$$z_{smooth}(x) = a \operatorname{erf}(x) + b \tag{5.3}$$

Once the terrain boundaries can vary significantly depending on the wind farm, it is convenient to establish an adjustment factor, n_{erf} , to be chosen by the user and that

Figure 40 – Plot of the Error function for $-3 \le x \le 3$



Figure 41 – Smoothing region representation in grey.



Source: Author

will define the portion of the error function that will be used to adjust the terrain height. The argument x of erf(x) will be defined as having a linear variation as in Figure 42, so that $-n_{erf} \leq x \leq n_{erf}$, and it is possible to write x in terms of n_{erf} , R, r and d as in Equation 5.4.

$$x = 2 n_{erf} \frac{(d-R)}{r} - n_{erf}$$

$$\tag{5.4}$$

The altitude at the boundaries was chosen to be equal to the average altitude inside the domain defined by the radius R + r and it is desired that for d = R the terrain is not smoothed and for $d = R + r, z = z_{ave}$.

Applying the aforementioned conditions in Equation 5.3 and using the property erf(x) = -erf(-x), the final smoothing equation is presented in Equation 5.5, where z_{orig} is the original terrain altitude for the point being smoothed.

$$z_{smooth}(x) = \frac{(z_{ave} - z_{orig})}{2 \operatorname{erf}(n_{erf})} \operatorname{erf}(x) + \frac{(z_{orig} + z_{ave})}{2}$$
(5.5)



Figure 42 – Linear variation of x with the distance from the center of the domain d.



Figure 43 – Altitude values for the Askervein Hill region: original topography (a) and smoothed topography (b)



Source: Author

This smoothing procedure is direction independent, once the only used variable is the distance from the point being smoothed to the central point of the domain to calculate xusing Equation 5.4.

In this optimization case, a $n_{erf} = 1.5$ was used to smooth the terrain and the average altitude of the domain is $z_{ave} = 27.7 \,\mathrm{m}$. A comparison between the original and smoothed terrain can be seen in Figure 43. The domain and mesh characteristics remained the same as Cases 6 and 7.

In this last case, a behavior similar to all the previous 7 cases was observed in the evolution of the evaluations and population taking between 2000 and 3000 evaluations, or 200 and 400 populations, to achieve an acceptable convergence of the optimization. This




Source: Author

Figure 45 – Objective function versus evaluations (a) and populations (b) for Case 8 with 20 turbines



Source: Author

is presented in Figure 44 and Figure 45 for 12 and 20 turbines. A visual inspection of the wind resource grid and total pressure is presented for the prevailing wind direction (Sector 11) in Figure 46 and Figure 47.

A result already known by the author is that if the wind speed in high enough, the optimization algorithm will keep putting turbines in the wake of the others, as in Figure 47, where 4 turbines are placed in alignment with the incoming wind direction. This may make sense from the AEP point of view, because even with 4 turbines in the wake of the others, the wind speed is sufficiently high to allow a reasonable power extraction in those 4 turbines, but the turbulence levels will increase drastically downstream of each turbine, which is not a good operating point for the turbine and will generate high fatigue loads in the components, leading to a premature failure and economic losses.

Figure 46 – Wind velocity and total pressure at 120 m from the ground of the best individual for Case 8 with 12 turbines and wind from Sector 11



Source. Hutilor

Figure 47 – Wind velocity and total pressure at 120 m from the ground of the best individual for Case 8 with 20 turbines and wind from Sector 11



In view of that, this was considered a downside of the developed tools, which calculates the flow in steady state and do not consider the turbulent kinetic energy reaching the turbine in the objective function or constraints. However, this can also happen in the commonly used tools in the market, and to avoid this, the tools usually force a minimum spacing not only between neighbor turbines, but also in the prevailing wind direction. Although this was already known by the author, it was decided not to force this secondary spacing, to see how the tool would behave in this class of problems.

Another interesting result is that as the wind speed is very high, all the turbines will be operating close to the rated power, and because of that the improvement in the optimization process is not as expressive as in the other cases. Figure 48 – Layout comparison - Synthetic Hill with 12 turbines (a) and 20 turbines (b). The blue "x" represents the layout obtained in this work and the red circle represents the layout obtained with Openwind



Source: Author

5.4 Results obtained with Openwind

In order to see how different the results obtained with the presented tool and with tools available in the market are, the a similar optimization case was run using Openwind, considering Case 5 (Synthetic Hill, 16 directions) and Case 8 (Askervein Hill, 16 directions). To generate the wind resource grid file at hub height, WAsP was used.

As this software needs to have a wind speed distribution for each sector, a synthetic distribution was created for Case 5, considering a k factor of 2 and A factor of 9.5. These values were chosen so that the wind speed at the reference point was 8 m/s in WAsP, to keep the cases as similar as possible. In Case 8, the MERRA2 data was used, considering the same 20 years of measurement.

Figure 48 and Figure 49 show the layouts obtained for Cases 5 and 8 considering 12 and 20 turbines, where the blue "x" represents the layout obtained in this work and the red circle represents the layout obtained with Openwind. It can be seen that for a small number of turbines, both tools deliver a similar layout, but as the number of turbines gets higher, the differences between the outcomes become more clear. This is because the tool developed in this work considers more of the physics of the turbine wakes. Nonetheless, it is important to highlight that the inputs and restrictions are different, so it is not possible to compare the results quantitatively.



Figure 49 – Layout comparison - Askervein Hill with 12 turbines (a) and 20 turbines (b)

Source: Author

6 Conclusions

A tool for wind farm layout optimization was developed making use of CFD for better wake modeling and the Dakota toolkit for optimization. Although there is a reasonable computational cost when compared to semi empirical and approximate analytical models, CFD techniques for solving the flow field can give an expressive improvement of knowledge on the wake that has an important impact on the optimization process, justifying its use.

Eight optimization cases were tested using a Genetic Algorithm with the Annual Energy Production as the single objective function and their results were compared for the major division regarding the terrain complexity (Flat terrain, Synthetic Hill and Askervein Hill) considering different wind roses and wind speeds distribution for a given turbine model and hub height. As expected, the layout efficiency decreased as turbines were added to the domain, meaning that wake losses are introduced and that the number of turbines for the specified area is already over the optimum condition. Although this is not an ideal scenario for wind farms, it is appropriate to test the presented tool, once high wake losses are present and there is room for improvement. It was observed that there is an almost linear increase of the wake losses and decrease of Specific AEP as turbines are added, considering the cases with a Flat terrain and a Synthetic Hill. Despite the different conditions, the optimization algorithm showed a similar behavior in terms of the evolution of the objective function of each evaluation and the best fitted individual of each population, confirming that the applied method is suitable for a wind farm layout optimization.

There are clear evidences that more complex phenomena were captured during the optimization, such as wake deflection, different wake recoveries depending on the wind speed and interaction between the disturbances in the flow field caused by the terrain and by the turbines. These phenomena are not observed when using the tools available in the wind market and are seen as an improvement in the field of wind farm layout optimization, yielding more accurate results and should decrease the level of uncertainty in the design of a wind farm.

A wide range of scenarios can be handled by the tool, including a complete wind rose and wind distribution in each sector with any desired topography, but there is a high computational cost associated to this complete scenario. Instead, it is recommended that the average wind speed of each sector is used as a first analysis.

6.1 Suggestions for Future Work

There is still room for improvement in the developed tool, which could bring more information from real wind farms. A full validation of the power curve obtained with the ADM native from OpenFOAM could be done, accompanied by a calibration in the way the free wind speed in the c_T curve is transformed into the wind speed at the disk. This calibration would have the power curve as reference, and the actuator disk should be able to match the commercial power curve for the given turbine model. The wind profile downstream the turbine could also be verified using field measurements.

In terms of input conditions, usually a wind farm has real measurements inside the domain, so it is necessary to make use of this information to better estimate the wind resource. Again, a calibration of the inlet profile of each sector could be done, using as reference the average wind speed for each sector in the measurement point as done by Lovatto et al (LOVATTO; TRINDADE; DORADO, 2015). This would introduce more realistic wind conditions to the optimization. Still in the ABL modelling, a constant shear stress could be applied at the top of the domain as recommended by Richards and Hoxey (RICHARDS; HOXEY, 1993).

The optimization process could be improved by testing other strategies, such as increased population size, different optimization grid, different inter turbine spacing restrictions, etc. Additionally, different optimization techniques could be implemented, e.g. Machine Learning techniques based on operational data and Particle Swarm Optimization.

Bibliography

ADAMS, B. et al. A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual. 2014.

ANDREWS, L. C. Special functions of mathematics for engineers. 1st. ed. [S.1.]: Oxford University Press, 1998.

BONANNI, A. et al. Wind Farm Optimization Based on CFD Model of Single Wind Turbine Wake. *European Wind Energy Conference and Exhibition 2012*, 2012. ISSN 09536205.

BONANNI, A. et al. Wind Farm Optimization Based on CFD Simulation of Non-Flat Terrain. *Journal of Environmental Science*, 2015.

CHAM. CHAM Website. 2019. ">http://www.cham.co.uk/>. Accessed: 2019-04-21.

DNV-GL. DNV-GL Website. 2019. <https://www.dnvgl.com/energy/generation/software/windfarmer-analyst.html>. Accessed: 2019-04-21.

DTU Wind Energy. *WAsP Website*. 2019. <https://www.wasp.dk/wasp>. Accessed: 2019-04-21.

EMD International A/S. *WindPRO Website*. 2019. <https://www.emd.dk/windpro/>. Accessed: 2019-04-21.

FROUDE, R. E. On the part played in propulsion by differences of fluid pressure. *Transactions of the Institute of Naval Architects*, v. 30, p. 390–405, 1889.

GELARO, R. et al. The Modern-Era Retrospective Analysis for Research and Applications, Version 2 (MERRA-2). *Journal of Climate*, v. 30, n. 14, p. 5419–5454, 2017. Disponível em: https://doi.org/10.1175/JCLI-D-16-0758.1>.

Global Wind Energy Council. *Global Wind Report - 2017.* 2018. <http://files.gwec.net/files/GWR2017.pdf>. Accessed: 2019-04-21.

GOLDBERG, D. E. Genetic Algorithms in Search, Optimization and Machine Learning. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

GUIRGUIS, D.; ROMERO, D.; AMON, C. Toward efficient optimization of wind farm layouts: Utilizing exact gradient information. *Applied Energy*, v. 179, p. 110–123, 2016.

HARGREAVES, D. M.; WRIGHT, N. G. On the use of the $\kappa - \varepsilon$ model in commercial CFD software to model the neutral atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, v. 95, n. 5, p. 355–369, 2007. ISSN 01676105.

HOU, P. et al. Combined optimization for offshore wind turbine micro siting. *Applied Energy*, v. 189, p. 271–282, 2017.

JACOBSON, M. Z. Fundamentals of atmospheric modeling second edition. [S.l.: s.n.], 2005. ISBN 9781139165389.

JENKINS, N. et al. *Wind Energy Handbook*. United Kingdom: John Wiley & Sons Ltd, 2001. ISBN 0-4714-8997-2.

JENSEN, N. O. A note on wind generator interaction. *Riso National Laboratory Roskilde*, 1983. ISSN 01676105.

KUO, J. et al. Wind farm layout optimization on complex terrains - integrating a cfd wake model with mixed-integer programming. *Applied Energy*, v. 178, p. 404–414, 2016.

LAUNDER, B.; SPALDING, D. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, v. 3, p. 269–289, 1974.

LOVATTO, M.; TRINDADE, L.; DORADO, R. Predicting the inlet wind profile of the neutral atmospheric boundary layer for wind resource assessment over non-flat terrains using cfd. In: . [S.l.: s.n.], 2015.

MIKKELSEN, R. F.; SØRENSEN, J. N. Actuator Disc Methods Applied to Wind Turbines. 121 p. Tese (Doutorado) — DTU, 2003. Disponível em: http://orbit.dtu.dk/files/5452244/Robert.PDF>.

Ministério de Minas e Energia. Empresa de Pesquisa Energética. Brasília: MME/EPE. *Plano Decenal de Expansão de Energia 2027.* 2018. http://www.epe.gov.br/sites-pt/ publicacoes-dados-abertos/publicacoes/Documents/PDE%202027_aprovado_OFICIAL. pdf>. Accessed: 2019-04-21.

PARADA, L. et al. Assessing the energy benefit of using a wind turbine micro-siting model. *Renewable Energy*, v. 118, p. 591–601, 2018.

RICHARDS, P. J.; HOXEY, R. P. Appropriate boundary conditions for computational wind engineering models using the $\kappa - \varepsilon$ turbulence model. *Journal of Wind Engineering and Industrial Aerodynamics*, v. 46-47, n. C, p. 145–153, 1993. ISSN 01676105.

SCHMIDT, J.; STOEVESANDT, B. Wind farm layout optimisation in complex terrain with cfd wakes. In: *EWEA conference proceedings, Paris, France.* [S.l.: s.n.], 2015. p. 17–20.

SIMISIROGLOU, N. et al. The Actuator Disc Concept in Phoenics. *Energy Procedia*, The Author(s), v. 94, n. January, p. 269–277, 2016. ISSN 18766102. Disponível em: http://dx.doi.org/10.1016/j.egypro.2016.09.182.

SONG, M. et al. Optimization of wind farm micro-siting for complex terrain using greedy algorithm. *Energy*, v. 67, p. 454–459, 2014.

SONG, M. et al. Optimization of wind turbine micro-siting for reducing the sensitivity of power generation to wind direction. *Renewable Energy*, v. 85, p. 57–65, 2016.

TAPIA, X. P. Modelling of wind flow over complex terrain using OpenFoam. Dissertação (M.Sc. Thesis), 2009.

TAYLOR, P.; TEUNISSEN, H. The askervein hill project: overview and background data. *Boundary-Layer Meteorology*, v. 39, p. 15–39, 1987.

The OpenFOAM Foundation. *OpenFOAM Documentation*. 2019. <https://www.openfoam.com/documentation/>. Accessed: 2019-04-22.

TOSSAS, L. M.; LEONARDI, S. Wind turbine modeling for computational fluid dynamics. *National Renewable Energy Laboratory (NREL), Technical Report No.* NREL/SR-5000-55054, 2013.

UL Renewables. Openwind Website. 2019. https://aws-dewi.ul.com/software/openwind/ >. Accessed: 2019-04-21.

VASEL-BE-HAGH, A.; ARCHER, C. Wind farm hub height optimization. *Applied Energy*, v. 195, p. 905–921, 2017.

VERSTEEG, H. K.; MALASKEKERA, W. An Introduction to Computational Fluid Dynamics: The Finite Volume Method. [S.l.: s.n.], 1995. ISSN 02104806. ISBN 9780131274983.

Appendix

APPENDIX A – Complementary Optimization Results

In this Appendix, the results for 14, 16 and 18 turbines, for the cases where these quantities were tested, will be presented.

A.1 Case 1 - Flat terrain and unidirectional wind

A.1.1 Optimization with 14 turbines

Figure 50 – Objective function \times evaluations (a) and populations (b)-Case 1 - 14 turbines



Source: Author

A.1.2 Optimization with 16 turbines

Figure 51 – Objective function \times evaluations (a) and populations (b)-Case 1 - 16 turbines



Source: Author

A.1.3 Optimization with 18 turbines



Figure 52 – Objective function \times evaluations (a) and populations (b)-Case 1 - 18 turbines

A.2 Case 2 - Flat terrain and sixteen wind directions

A.2.1 Optimization with 14 turbines

Figure 53 – Objective function \times evaluations (a) and populations (b)-Case 2 - 14 turbines



Source: Author

A.2.2 Optimization with 16 turbines



Figure 54 – Objective function × evaluations (a) and populations (b)-Case 2 - 16 turbines

Source: Author

A.2.3 Optimization with 18 turbines

Figure 55 – Objective function \times evaluations (a) and populations (b)-Case 2 - 18 turbines



Source: Author

A.3 Case 3 - Synthetic Hill and unidirectional wind

A.3.1 Optimization with 14 turbines

Figure 56 – Objective function \times evaluations (a) and populations (b)-Case 3 - 14 turbines



Source: Author

A.3.2 Optimization with 16 turbines

Figure 57 – Objective function \times evaluations (a) and populations (b)-Case 3 - 16 turbines



Source: Author

A.3.3 Optimization with 18 turbines



Figure 58 – Objective function \times evaluations (a) and populations (b)-Case 3 - 18 turbines

A.4 Case 4 - Synthetic Hill and four wind directions

A.4.1 Optimization with 14 turbines





Source: Author

A.4.2 Optimization with 16 turbines



Figure 60 – Objective function \times evaluations (a) and populations (b)-Case 4 - 16 turbines

A.4.3 Optimization with 18 turbines

Figure 61 – Objective function \times evaluations (a) and populations (b)-Case 4 - 18 turbines



Source: Author

A.5 Case 5 - Synthetic Hill and sixteen wind directions

A.5.1 Optimization with 14 turbines



Figure 62 – Objective function \times evaluations (a) and populations (b)-Case 5 - 14 turbines

A.5.2 Optimization with 16 turbines

Figure 63 – Objective function \times evaluations (a) and populations (b)-Case 5 - 16 turbines



Source: Author

A.5.3 Optimization with 18 turbines



Figure 64 – Objective function \times evaluations (a) and populations (b)-Case 5 - 18 turbines

A.6 Case 8 - Askervein Hill and sixteen wind directions with varying wind speed

A.6.1 Optimization with 14 turbines

Figure 65 - Objective function \times evaluations (a) and populations (b)-Case 8 - 14 turbines



Source: Author

A.6.2 Optimization with 16 turbines

Figure 66 - Objective function \times evaluations (a) and populations (b)-Case 8 - 16 turbines



A.6.3 Optimization with 18 turbines

Figure 67 – Objective function \times evaluations (a) and populations (b)-Case 8 - 18 turbines



Source: Author

APPENDIX B – Mesh Sensitivity Analysis

A mesh sensitivity analysis was carried out to verify how the final layout would change according to the mesh resolution and to avoid misleading results and conclusion. As the result of the presented tool is the final layout, this was used for comparison between the sensitivity analysis.

For this evaluation, the case with a Synthetic Hill, 16 direction sectors and 16 turbines was chosen. The mesh was refined mainly in the central block of the domain, where the turbines will be placed and where the higher elevation zone is.

The number of elements in the mesh was increased in about 42% in each case, and the mesh description is presented in Table 9. The refined meshes are here called R1 and R2, and the element size of the background mesh, in the disk region and the number of elements are presented. The element size is also presented in term of the rotor diameter D, in this work D = 114 m. All the values presented in Table 9 were measured in the horizontal plane xy and the mesh was not refined in the vertical direction z, because the size of the elements is about the same size as the most refined case, being 30 m (0.26 D) in the background mesh and 7.5 m (0.07 D) in the disk region.

To evaluate how impacting the mesh refinement would be in the final layout, a new optimization was carried out, considering the best population of the Synthetic Hill case, 16 direction sectors and 16 turbines as the initial population, and the changes in the objective function and layout were monitored. In Figure 68, Figure 69 and Figure 70, the objective function in terms of the evaluations and populations is shown for each case. The strategy here was to run the optimization for about 25% of the total evaluations/populations needed to reach a converged state in the case with no refinement. As a result, it is noticed that the objective function has a different value in each case, but there was no expressive gain in AEP in each case. Figure 71 shows a comparison between the 3 layouts, where only one or two positions were slightly moved to other coordinates, but the ones in the center of the domain were not moved. These results indicate that the meshes employed in this study are sufficient to achieve the goals we set, and that further refinement of the meshes would not bring any major improvement, despite the increase in computational cost.

Case	Element size [m] -	Element size [m] -	Number of
	background mesh	Disk	elements
No refinement	$41.7 (0.37 \mathrm{D})$	$10.4 \ (0.09 \mathrm{D})$	286000
R1	$33.3 (0.29 \mathrm{D})$	$8.3 (0.07 \mathrm{D})$	406000
R2	$27.8 (0.24 \mathrm{D})$	$6.9 (0.06 \mathrm{D})$	579000

Table 9 – Element size and number of elements for mesh sensitivity analysis

Figure 68 – Objective function versus evaluations (a) and populations (b) in the mesh sensitivity analysis with 16 turbines - no refinement



Source: Author

Figure 69 – Objective function versus evaluations (a) and populations (b) in the mesh sensitivity analysis with 16 turbines - R1



Source: Author

Figure 70 – Objective function versus evaluations (a) and populations (b) in the mesh sensitivity analysis with 16 turbines - R2



Source: Author

Figure 71 – Final layout for the case with no mesh refinement (a), R1 (b) and R2 (c)



Source: Author