Wind Farm Layout Optimization in Complex Terrain based on CFD simulations

Luís Eduardo Boni Cruz[*], William Radünz[†], Bruno Souza Carmo[‡]

February 2019

Abstract

We present a tool for wind farm layout optimization based on Computational Fluid Dynamics (CFD) simulations of the atmospheric wind flow and inter-turbine interference using the open source code OpenFOAM and the Dakota optimization toolkit. The high computational power required to simulate a whole wind farm using the complete geometry of the wind turbines makes this option unfeasible and the need for models to represent their effects on the wind flow and the interference of one turbine into the others arises, being the Actuator Disk Model (ADM) the most commonly used. Once the procedure for wind turbine behavior evaluation using a CFD model is defined, the coupling between this model and Dakota optimization toolkit took place. Two cases were tested in order to achieve the best wind farm layout in terms of power output that respects the imposed physical restrictions for a given number of wind turbines and terrain.

Keywords: Turbine Siting. Complex Terrain. Actuator Disk. OpenFOAM. Dakota.

1 Introduction

The definition of a wind farm layout is one of the initial and most important stages during the wind farm development, representing a very sensitive topic in the business model. In the past decades the wind energy sector has been competing with more traditional energy sources, always trying to lower the cost of energy. One of the options to reduce this cost is through an optimization in the energy extraction, which in the wind energy industry can be achieved with an optimization of the layout of a wind farm On a given terrain.

Wind turbines disturb the flow and create a region with speed deficit and increased turbulence levels downstream known as wake, which leads to energy yield losses. The main goal of the wind farm layout optimization (WFLO) problem is to maximize the annual energy production (AEP), capacity factor (CF) or minimize the cost of energy

[*] Department of Mechanical Engineering, Escola Politécnica, University of São Paulo, Brazil

[†] Graduate Program in Mechanical Engineering, Federal University of Santa Catarina, Brazil

[‡] Department of Mechanical Engineering, Escola Politécnica, University of São Paulo, Brazil

(CoE) of a wind farm. It consists of an integer mixed-type problem containing both discrete and continuous variables and, thus, requires non-classic optimization approaches (GONZÁLEZ et al., 2014). Most applications were developed for offshore wind farms in which the incoming wind profile is assumed horizontally homogeneous which requires 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).

There are several tools in the market, either free or commercial, that are capable of performing a layout optimization, but there is a lack of CFD based optimization tools, which can give more accurate and reliable results, specially in complex terrains, in addition to being capable of dealing with complex flow scenarios such as wake interaction, wake recovery and the impact of the terrain in the wake of turbines. Here we present such a tool, and report results from two optimization scenarios: (i) a terrain with a synthetic hill, derived from a Gaussian function, for four wind directions; and (ii) a complex terrain, represented by the Askervein Hill, considering a unidirectional wind velocity with varying magnitude. The objective function of the optimization is the Annual Energy Production (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.

The remainder of this paper is organized as follows. In section 2 we present details of the numerical models used to calculate the flow over the terrain and through the wind turbines, and to obtain the wind turbine power and thrust. In section 3 an overview of the approach we adopted of the optimization problem is given. In section 4, we provide information about the setup of the cases and the meshes, and present the results obtained for the two optimization problems mentioned above. Finally, conclusions are drawn in section 5.

2 Wind flow and wind turbine numerical modeling

All the models described below were used or implemented on the open source software OpenFOAM (WELLER et al., 2002; OPENFOAM...,). It implements the finite volume method to solve conservation equations and has been vastly used and validated for many different computational fluid dynamics problems.

By neglecting the Coriolis and the atmospheric stability effects, surface layer winds may be modeled using the mass conservation and momentum budget equations, also known as the Reynolds-averaged Navier-Stokes (RANS) equations. The latter is given by the following expressions shown in index notation:

$$U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P'}{\partial x_i} + \frac{\partial}{\partial x_j}\left[2(\nu + \nu_t) S_{ij}\right], \quad i, j = 1, 2, 3 \tag{1}$$

$$P' = P + \frac{2}{3}\rho k \tag{2}$$

$$\nu_t = C_\mu \frac{k^2}{\epsilon}, \tag{3}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \quad i,j = 1,2,3 \tag{4}$$

where U_i are the mean velocity components, ρ is the density of the fluid, P' is a modified pressure term, ν is the kinematic viscosity of the fluid, ν_t is the kinematic eddy viscosity of the flow and S_{ij} is the mean strain rate tensor. RANS equations are solved employing turbulence models such as the $k-\epsilon$ model (section 2.1). Although winds are certainly an unsteady flow case, the long-term spatial distribution of the wind speed field within the wind farm area may be calculated via a steady-state simulation. Wind turbine wake effects are included by means of an actuator disk concept (section 2.3) instead of explicitly representing the wind turbine blades.

2.1 The $k - \epsilon$ model

The standard k-ϵ model proposed by (LAUNDER; SPALDING, 1974) is given by the following set of equations that describe turbulence kinetic energy and its dissipation rate budgets:

$$\frac{\partial}{\partial x_i}(U_i k) = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k - \epsilon, \quad i,j = 1,2,3, \tag{5}$$

$$\frac{\partial}{\partial x_i}(U_i \epsilon) = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] + C_{\epsilon 1} G_k \frac{\epsilon}{k} - C_{\epsilon 2}\frac{\epsilon^2}{k}, \quad i,j = 1,2,3, \tag{6}$$

$$G_k = \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j}, \quad i,j = 1,2,3, \tag{7}$$

where ν_t is the turbulence kinematic viscosity, k is the turbulence kinetic energy (TKE) and ϵ its dissipation rate, U is the wind speed, ν is the kinematic viscosity of the air and G_k is the TKE production term. Turbulence model constants used in this paper are shown in Table 1.

Table 1 – Turbulence model constants employed in this study

$C_{\epsilon 1}$	$C_{\epsilon 2}$	C_μ	σ_k	σ_ϵ
1.44	1.92	0.09	1.0	1.11

2.2 Boundary Conditions

Due to the length scale of a wind farm, the standard boundary conditions used for turbulent flow are not enough to describe this phenomena. It was observed (HARGREAVES; WRIGHT, 2007) that in a fetch domain, the inlet boundary layer is not maintained throughout the domain, making it necessary to modify the standard conditions. This classical set of modifications was proposed by Richards and Hoxey (RICHARDS; HOXEY, 1993) and is already implemented by default in OpenFOAM for the k-ϵ turbulence model.

The inlet is assumed to have a logarithmic profile as described in Equation 8

$$U(z) = \frac{U^*}{\kappa} \ln\left(\frac{z - z_g + z_0}{z_0}\right) \tag{8}$$

$$U^* = \kappa \frac{U_{ref}}{\ln\left(\frac{z_{ref}+z_0}{z_0}\right)} \tag{9}$$

where U^* is the friction velocity, κ is the Von Karman's constant, z_g is the minimum z coordinate, z_{ref} is the reference height at which the reference wind velocity U_{ref} must be specified and z_0 is the surface roughness.

Additionally, the turbulent kinetic energy and the dissipation should also be modified at the inlet as shown in Equation 10 and 11.

$$k = \frac{(U^*)^2}{\sqrt{C_\mu}} \tag{10}$$

$$\epsilon = \frac{(U^*)^3}{\kappa(z - z_g + z_0)} \tag{11}$$

At last, the terrain will be considered as fully rough and the velocity of the cells parallel to the ground U_w will be calculated with Equation 12.

$$U_w = \frac{U^*}{\kappa} \ln\left(\frac{z + z_0}{z_0}\right) \tag{12}$$

A pressure boundary condition was applied at the outlet.

2.3 Actuator Disk model

One of the most commonly used models to represent a wind turbine is the actuator disk model (ADM), which is based in the 1D momentum theory. This model requires a thrust distribution to be chosen, and this thrust will be applied to the flow during the numerical simulation. The advantages of using this model are the simplicity of the model itself and the reduced computational cost, once there are no moving parts or blade geometry to be modeled.

Some modifications were applied (Simisiroglou, et al., 2016) in OpenFOAM's native actuator disk model (*actuationDiskSource*). In the classic actuator disk theory, an upstream point is used to calculate to get the free wind velocity and calculate the thrust in the disk. The referred modifications aimed to remove the dependency of an upstream point, once the location of this point can be a subjective decision.

From the actuator disk theory, the c_t is defined as $c_t = 4a(1-a)$, where a is the axial induction factor. By solving this equation for the axial induction factor, we obtain Equation 13.

$$a = \frac{1}{2}(1 - \sqrt{1 - c_t}) \tag{13}$$

Once the classic actuator disk theory is in 1D, some changes in the definitions are needed. The component generating thrust is the normal wind velocity at each point of the disk $\mathrm{U_n}$, which can be estimated by combining the axial induction factor definition from Equation 14 with Equation 13 and will be a function of the free wind velocity U_∞ and c_t.

$$a = \frac{U_\infty - U_n}{U_\infty} \tag{14}$$

$$U_n = U_\infty \left(1 - \frac{1}{2}(1 - \sqrt{1 - c_t}\right) \tag{15}$$

With Equation 15 it is possible to calculate the c_t on each cell based on the wind velocity on each cell.

The ADM theory is based on an infinitely thin disk, which on CFD could be represented by the face of the cells in the disk region. As this would represent a complex problem, OpenFOAM natively treats it in terms of volumes, calculating a volumetric force to be applied in the disk region. This is achieved by weighting the calculated thrust by the cell volume and the thrust on each cell can be calculated as shown in Equation 16.

$$T_i = c_{t,i} \frac{1}{2} \rho \left(\frac{U_{n,i}}{1 - a_i}\right)^2 A_{disk} \frac{V_i}{V_{disk}} \tag{16}$$

Here, V_i is the volume of each cell in the disk region, V_{disk} is the total volume of the disk and A_{disk} is the swept area of the wind turbine. By doing this, the actuator disk will be as thin as the mesh resolution allows.

The power extracted from the wind has to be calculated in order to calculate the AEP, this is done as the sum over each cell's power through its definition shown in Equation 17.

$$P_{out} = \sum_{i=1}^{k} P_i = \sum_{i=1}^{k} T_i U_{n,i} \tag{17}$$

3 Optimization Problem

3.1 Dakota toolkit

Dakota (ADAMS et al., 2014) is an open source software that was developed by Sandia National Laboratories (USA) to provide a set of optimization tools for engineering, solving structural analysis and design problems at Sandia in 1994. Nowadays it has expanded its capabilities, being able of handling parametric studies, optimization, uncertainty quantification, design of experiments and calibration. Dakota can be easily coupled with other software, specially those that can be run on the shell or command prompt.

In this work only the optimization class was used, being the genetic algorithm the chosen method for the optimization. This method was extensively used in the literature (BONANNI et al., ; SCHMIDT; STOEVESANDT, 2014) due to its robustness, independence of derivatives and ability to handle non-smooth problems.

3.2 Problem Formulation

Genetic algorithms are based in the evolutionary theory and try to emulate an evolution process through the generation of an initial population, reproduction, mutation and cross-over processes and selection of the most fitted individual. Each individual is characterized by a chromosome containing its information. In the case of a wind farm each chromosome contains the location of the wind turbines in the xy plane and the objective function, F_{obj}, of the optimization is the Annual Energy Production (AEP) of the whole wind farm, which shall be maximized and is defined by:

$$F_{obj} = 8766 \sum_{i=1}^{n} \sum_{j=1}^{m} P_j f_j \tag{18}$$

There are several constraints that a wind farm should comply with to be able to be built, the most common are environmental, archaeological, land ownership, generated noise, shadow flickering, neighbor wind turbines and inter turbine spacing. Due to the complexity of considering all the know restrictions of a wind farm, this work only considered a minimum inter turbine spacing in the xy plane as constraint, which has an important role when calculating the energy production and wake effect. This constraint is also needed to avoid overlapping of the actuator disks, which could lead to numerical issues.

We chose to solve a discrete problem over a grid of possible locations because it demands a lower computational time when compared to a continuous problem. The domain was defined as a 2 km radius cylinder, with a 10 x 10 grid located in the center of the domain and composed by 165 m x 165 m cells, being the vertices of this grid the possible wind turbine locations. A representation of the domain and discrete grid can be seen on Figure 1.

The wind turbine model chosen for this work is 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 hub height of 120 m.

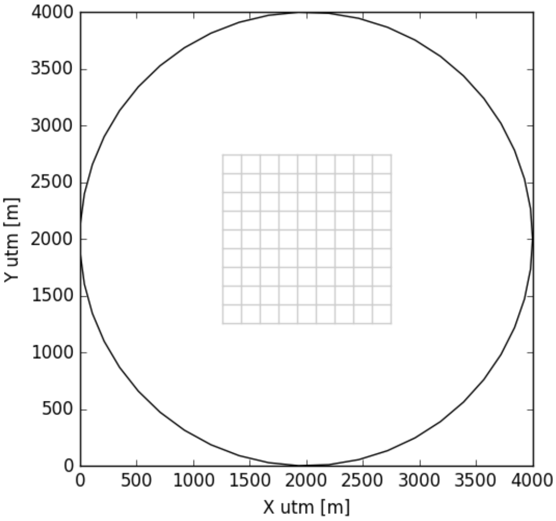


Figure 1 – Domain and Optimization grid representation

3.3 Optimization workflow

Neither OpenFOAM nor Dakota have a Graphical User Interface (GUI), so the coupling between them was done by coding using native scripts of both software and also Python scripts developed to generate inputs and manage files. Figure 2 shows the workflow of the optimization.

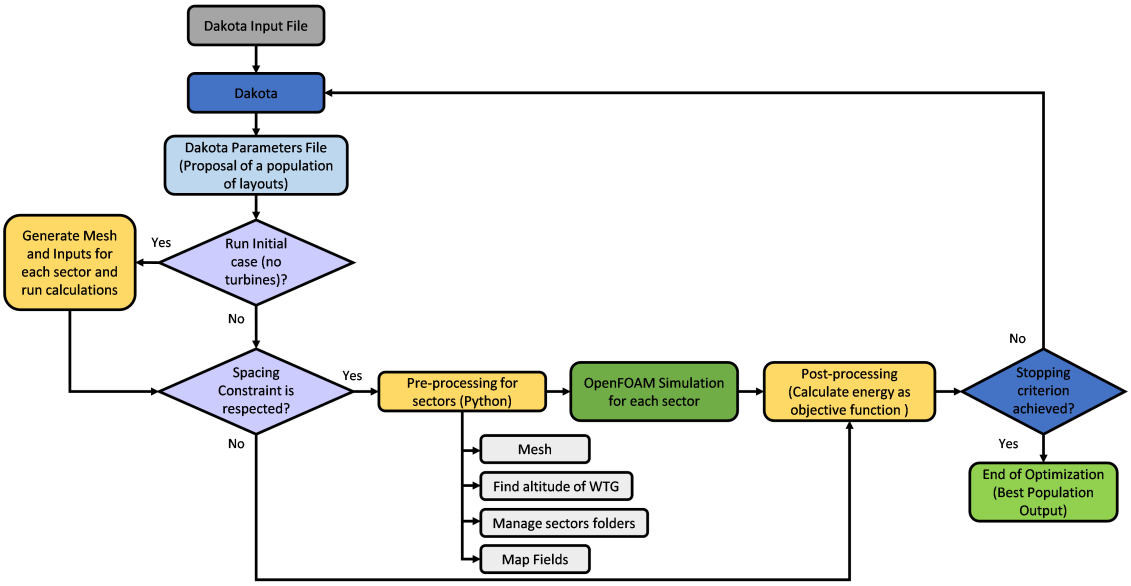

Dakota Input File
Dakota
Dakota Parameters File (Proposal of a population of layouts)
Run Initial case (no turbines)?
Yes
Generate Mesh and Inputs for each sector and run calculations
No
Spacing Constraint is respected?
Yes
No
Pre-processing for sectors (Python)
Mesh
Find altitude of WTG
Manage sectors folders
Map Fields
OpenFOAM Simulation for each sector
Post-processing (Calculate energy as objective function)
Stopping criterion achieved?
No
Yes
End of Optimization (Best Population Output)

Figure 2 – Optimization Workflow

An input file containing a randomly generated initial population is read by Dakota, and the individuals in this population should respect the spacing constraint. This initial population could be generated using Dakota, but it was observed that depending on the number of wind turbines in the domain, Dakota was not able to generate an initial population where all the individuals respect the constraint. For this reason, an external generator was developed in order to avoid this issue. As the spacing constraint is checked outside Dakota, in case any individual does not comply with it, its objective function will be set to zero. It is expected that Dakota do not suggest this kind of individual as the optimization evolves.

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, even considering a single direction, multiple directions or multiple wind velocities. 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 or be based in the user's observation of the results (evolution of populations, result of each iteration, layout achieved, etc.)

4 Wind farm Layout Optimization

4.1 General Considerations

In all the test cases, the grid with possible locations of the turbines was kept in the center of the domain, once more than one direction can be used in the optimization. Besides, the quantities in Table 2 were kept constant.

In the incompressible and steady state solver *simpleFoam*, all the equations are divided by the fluid density, so the air density value of 1 kg/m^3 was chosen for convenience, this way no further calculations are needed to post-process the results.

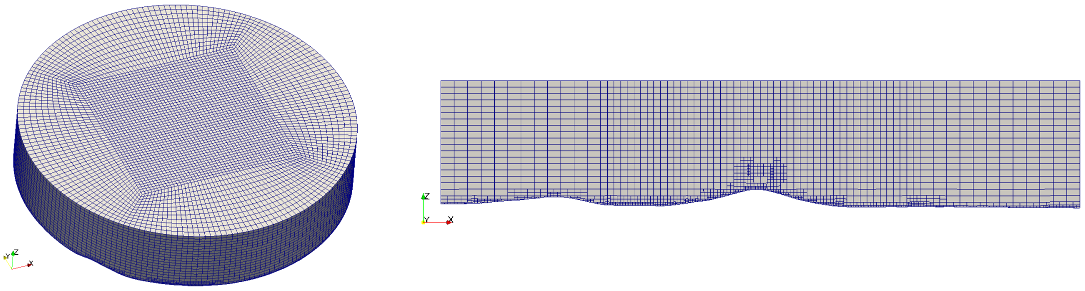
Table 2 – Common parameters in the optimization cases.

Rotor diameter (m)	114
Huh height (m)	120
Air density (kg/m3)	1.0
Cross over probability	0.95
Mutation probability	0.10

4.2 Mesh Characteristics

OpenFOAM has two commonly used mesh generators: *blockMesh*, that 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.

Figure 3a shows a general view of the mesh in the domain while Figure 3b shows a slice in the direction of the incoming wind, illustrating the initial mesh refinement close to the surface and in the disk region for the Askervein Hill case. The background mesh is formed by 92160 cells, while the mesh after the standard refinement has about 280000 cells for all the cases to be studied. Each disk, in the diameter direction, has 12 cells, which is considered an appropriate value (SIMISIROGLOU et al., 2016).



(a)

(b)

Figure 3 – Mesh representation for the Askervein Hill case.

4.3 Case 1: Synthetic hill and four wind directions

4.3.1 Topography description and Considerations

The first test case, named Bump terrain, was derived from a Gaussian function, which calculates the z value of the terrain based on a multiplication of two exponential functions, as shown in Equation 19.

$$z(x,y) = \begin{cases} e^{-1/(1-x^2)}e^{-1/(1-y^2)} & ,-1 < x,y < 1 \\ 0 & ,|x,y| > 1 \end{cases} \tag{19}$$

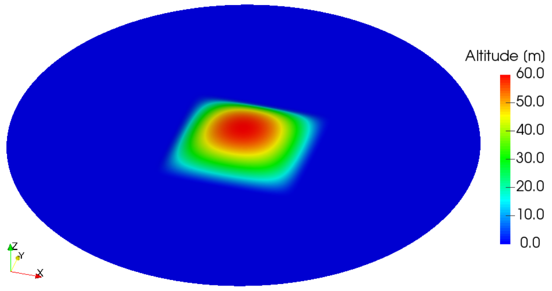


Figure 4 – Altitude of the Bump terrain

Figure 4 shows the altitude of the generated terrain. Additionally, a hypothetical wind rose with four wind sectors was created to simulate this scenario, considering a the same wind speed for all directions. This wind rose can be seen in Figure 5.

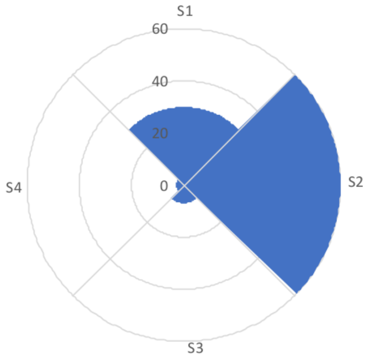


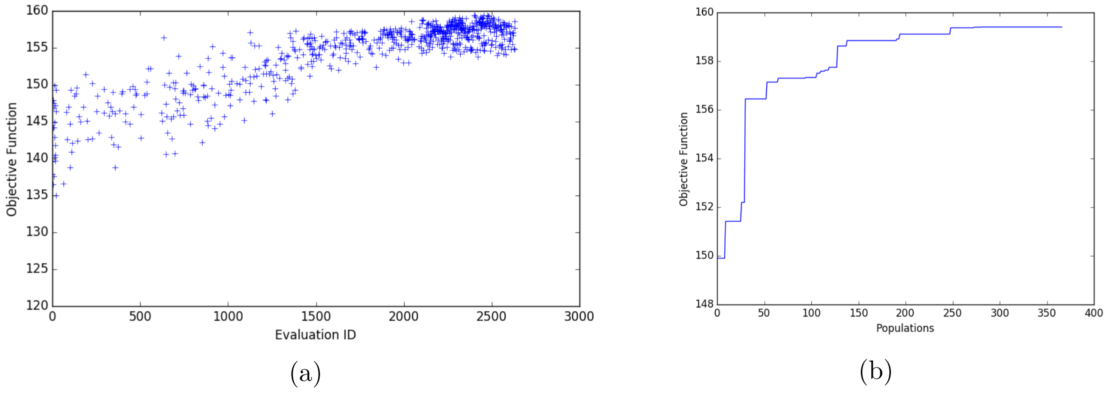
Figure 5 – Wind rose for the Bump terrain case

On this domain, 12 wind turbines will be placed inside the before mentioned 10 x 10 grid, located in the center of the domain. It was assumed that the disks representing the turbine would be aligned with the direction sector instead of with the local flow as it happens with real turbines. The population size for this case is 20 individuals and the roughness value z_0 of 0.01 m was adopted.

4.3.2 Results

The objective function value of each layout evaluation carried out by OpenFOAM is shown in Figure 6a. It is noticed that on 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 2000 and 2500 there is no significant change in the objective function value. This can be understood as an indication of convergence. Next, the best fitted individual of each population is analyzed in Figure 6b. A similar behavior can be seen in the populations, showing that the optimization can be considered as converged after approximately population 250, since there is no change in the objective value after that.

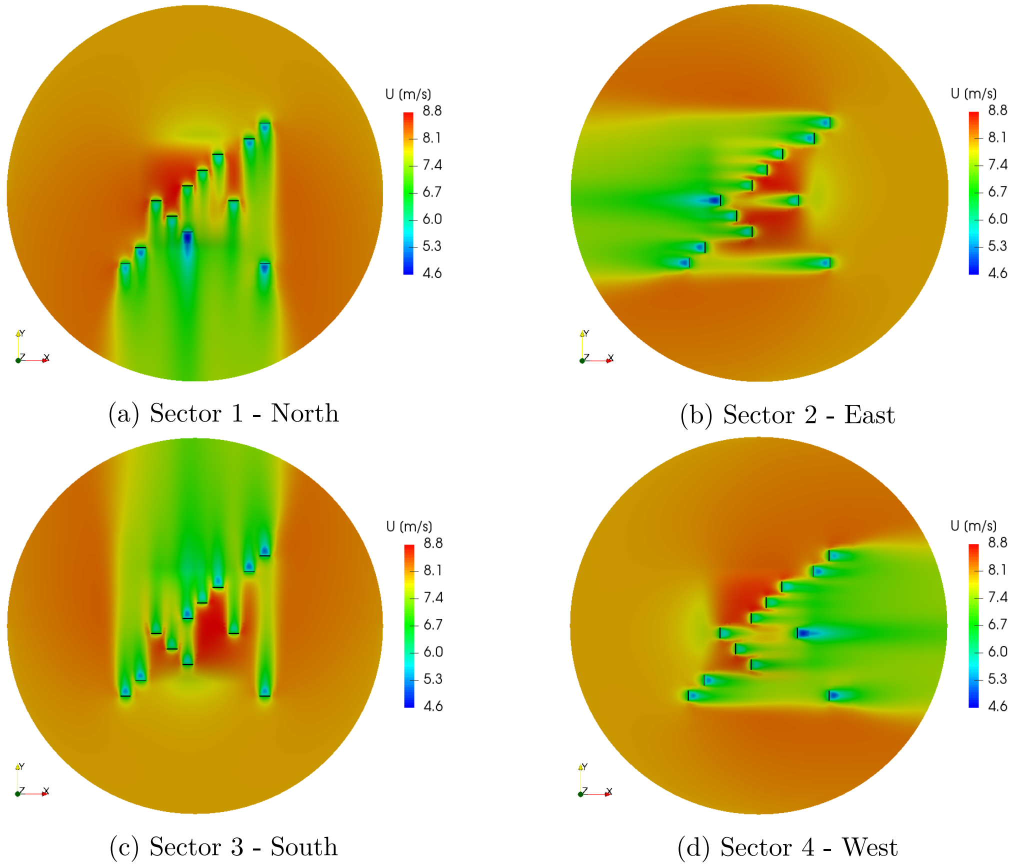


(a)

(b)

Figure 6 – Objective function versus evaluations and populations for Case 1

As a visual inspection in the CFD simulation, Figure 7 shows the wind velocity 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 this was achieved by an almost diagonal pattern in the turbine locations.



(a) Sector 1 - North

(b) Sector 2 - East

(c) Sector 3 - South

(d) Sector 4 - West

Figure 7 – Wind velocity of the best individual for the four different sectors, synthetic hill case.

4.4 Case 2: Askervein Hill

4.4.1 Topography description and considerations

The Askervein hill project aimed at gathering data on surface layer winds in the presence of a 116-meter-high hill (TAYLOR; TEUNISSEN, 1987). Several met masts containing cup and sonic anemometers were deployed near the hill and at a upstream site. The terrain upstream in plain and covered with low vegetation. The data collected was split into two-hour runs for which there was detailed description of the wind conditions. The run named "TU-03B" was chosen since the atmospheric stability was found to be nearly neutral and the wind direction was perpendicular to the major axis of the hill. The inflow conditions parameters were defined based on the measurements at the reference site a few kilometers upstream and are displayed in Table 3. The simulation domain around the Askervein hill can be seen in Figure 8

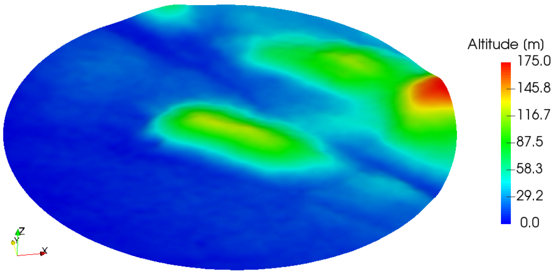


Figure 8 – Askervein hill area colored by altitude values

Table 3 – Summary of inflow and model parameters used in the Askervein hill simulations

Case	z_0 $[m]$	u^* $[m/s]$	sector $[deg]$	z_{ref} $[m]$	U_{ref} $[m/s]$
2.1	0.03	0.618	210	10	9.0
2.2	0.01	0.341	210	120	8.0
2.3*	0.01	-	210	120	-

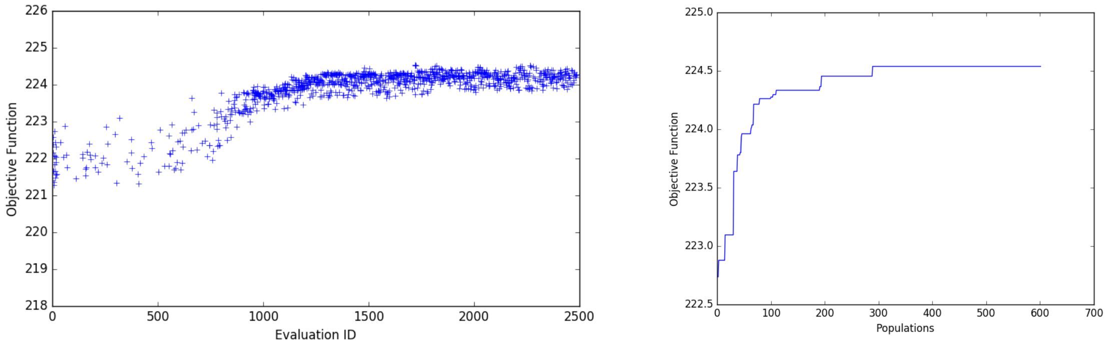
*This case contains simulation with the following reference wind speeds: 5, 7, 9, 11 and 13 m/s

The same considerations regarding domain size, grid of possible locations, number of turbines and population size from the previous case were considered. This case will be decomposed in three, varying only the inlet boundary condition.

4.4.2 Results for Case 2.1: Askervein Hill project data

The first case considering this topography was built using the experimental data as the inlet condition of the domain, as shown in Table 3. In the same way as Case 1, the objective function of each evaluation (Figure 9a), the best fitted individual of each population (Figure 9b) and the visual inspection of the CFD simulation (Figure 10) were used to drawn conclusions about this test case.

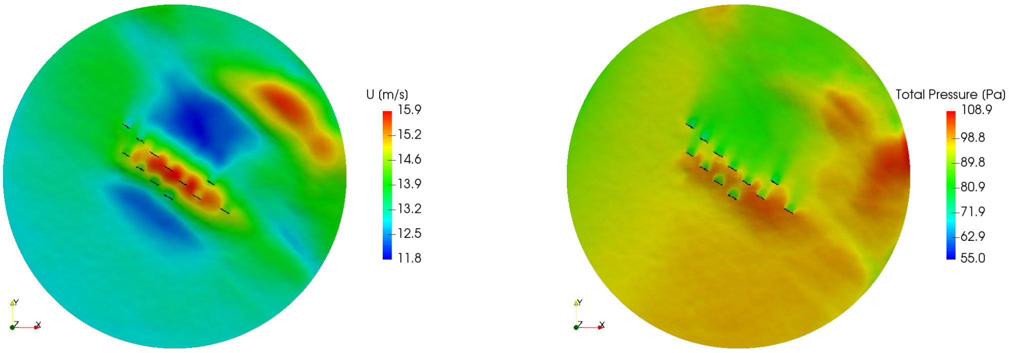
The same behavior of the previous test case can be noticed. After evaluation number 1300, there was a clear stagnation of the evaluation, with no expressive changes in the objective function. Regarding the populations, the objective function of best fitted individual suffered no expressive changes either.



(a)

(b)

Figure 9 – Objective function versus evaluations (a) and populations (b) for Case 2.1



(a)

(b)

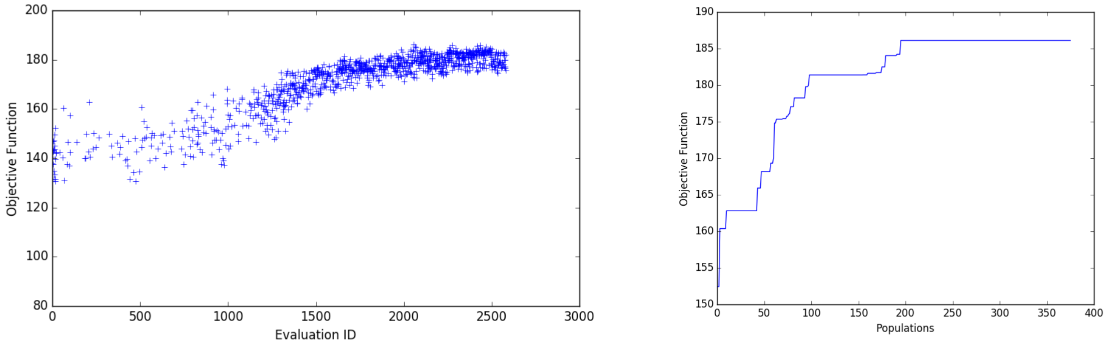
Figure 10 – Wind velocity (a) and total pressure (b) at 120 m from the ground of the best individual for Case 2.1

Figure 9 shows that for this set of inlet conditions, the objective function variation throughout the optimization changed from about 221 to 225 GWh/year. It was concluded that due to the high wind speeds, even after energy is extracted from the fluid in the wakes, the wind velocity is still high enough to keep the turbines operating close to the rated power, so any possible location could be already considered close enough to the optimum design. Due to this observation, two new scenarios were tested in order to explore the algorithm behavior under different conditions, as described next.

4.4.3 Results for Case 2.2: Askervein Hill with standard conditions

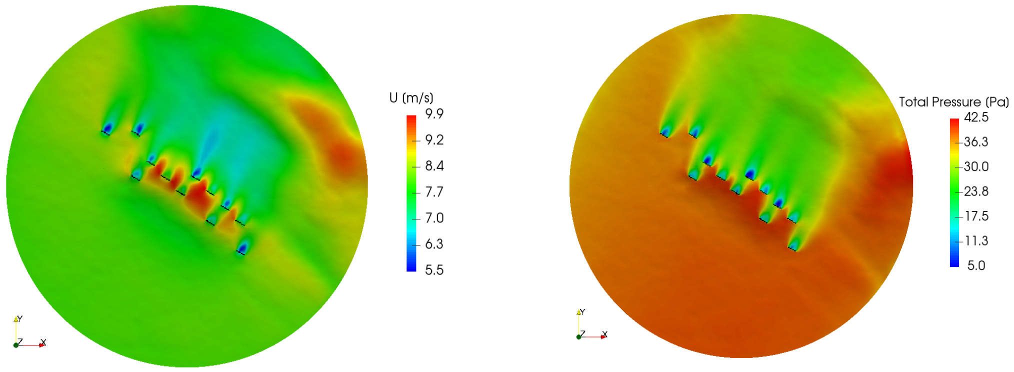
For this case, a wind velocity of 8.0 m/s, which represents a standard condition in the market, was used. With this new value it is possible to see that there is a substantial variation in the objective function, showing that not all locations will be able to extract a high energy as in Case 2.1. Figure 11a shows the objective function of each evaluation, Figure 11b shows the best fitted individual of each population and finally Figure 12 shows the visual inspection of the CFD simulation.

Notwithstanding the difference in the inlet conditions, a similar convergence behavior was noticed in this case, showing that the optimization reached a converged state after approximately evaluation 2050, corresponding to population 195. Figure 12 shows the wind velocity and total pressure at 120 m from the ground for this case.



(a) (b)

Figure 11 – Objective function versus evaluations (a) and populations (b) for Case 2.2



(a) (b)

Figure 12 – Wind velocity (a) and total pressure (b) at 120 m from the ground of the best individual for Case 2.2

4.4.4 Results for Case 2.3: Askervein Hill with frequency distribution

For the last test, we decided to simulate a set of different inlet wind velocities following a Weibull distribution for the direction sector already being considered. 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 speeds according to Figure 13.

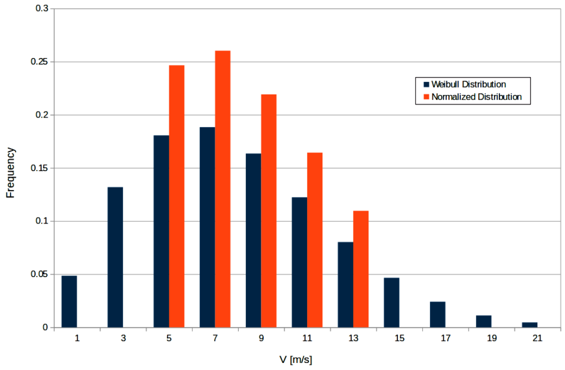
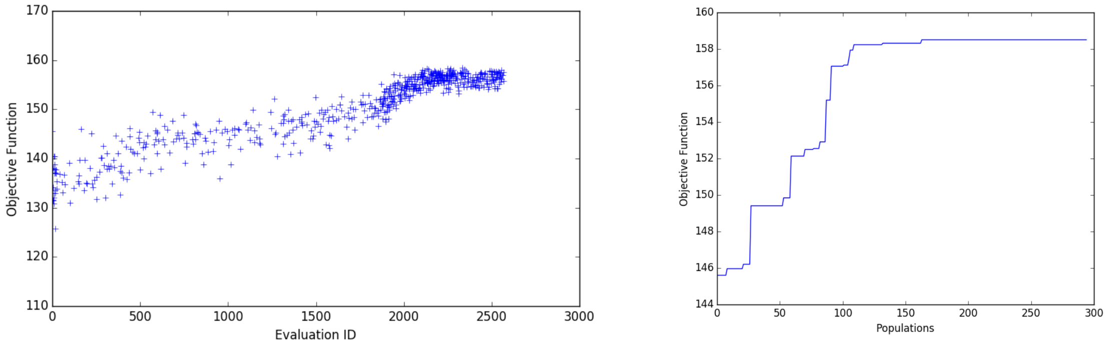
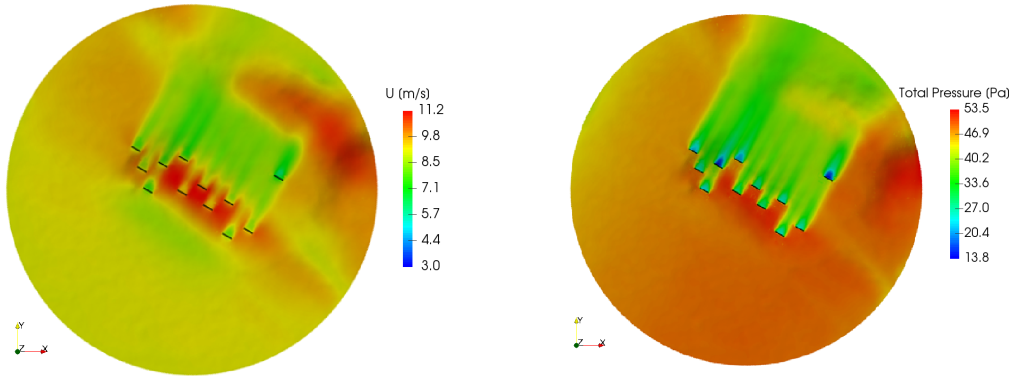


Figure 13 – Original and normalized wind distribution



(a) (b)

Figure 14 – Objective function versus evaluations and populations



(a) (b)

Figure 15 – Wind velocity and total pressure at 120 m from the ground of the best individual for Case 2.3 and 9 m/s

Once more, it is possible to see in Figure 14 that the optimization algorithm performed as expected, reaching a converged state after evaluation 2100 and population 160. The optimal distribution of turbines can be seen in Figure 15. It is interesting to notice that the distribution is somewhat between the two previous results: not so spread out as in case 2.2, and not so concentrated on the hill as in case 2.1. This shows the importance of considering the wind probability distribution when deciding the layout of a wind farm.

5 Conclusions

In this paper, we presented a methodology for wind farm layout optimization using CFD modeling and genetic algorithm. When compared to the other methodologies usually employed in the industry, this approach takes into account more of the physics of the flow, considering the interference between turbines and terrain. So it supposedly yields more accurate results and should decrease the level of uncertainty in the design of a wind farm.

Four cases of optimization considering different topographies and wind conditions were tested in order to achieve the best layout in terms of AEP for a fixed number and model of wind turbines. 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.

A CFD-based optimization usually requires more computational power and time than the traditional semi empirical and approximate analytical models, but there is a significant improvement in terms of how the wake development and interaction with the terrain impact the solution, being able to handle more complex scenarios of turbines positioning and topography.

The next steps of this work would be a further validation of the power curve obtained using the ADM for all operational wind speeds and also to consider multiple wind directions on a real terrain, flat or complex.

Acknowledgments

The authors would like to thank Siemens Gamesa for providing information about the wind turbine chosen for the analysis. The Brazilian National Council for Scientific and Technological Development (CNPq) is also highly regarded for their financial support.

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.

BONANNI, A. et al. Wind Farm Optimization based on CFD model of wind turbine. n. 4.

GONZÁLEZ, J. et al. A review and recent developments in the optimal wind-turbine micro-siting problem. *Renewable and Sustainable Energy Reviews*, v. 30, p. 133–144, 2014.

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 k-ϵ\{lunate\} 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.

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.

OPENFOAM Documentation. <https://www.openfoam.com/documentation/>. Accessed: 2019-02-20.

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 k-??{lunate} 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 Optimization with Wakes from Fluid Dynamics Simulations. *European Wind Energy Conference and Exhibition 2014*, n. July 2016, p. 1–10, 2014. Disponível em: <http://www.ewea.org/annual2014/conference/mobile/speakerdetails.php?id=202{&}da>.

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.

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

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

WELLER, H. G. et al. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, v. 12, n. 6, p. 620, 2002. ISSN 08941866.