54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference
April 8-11, 2013, Boston, Massachusetts

AIAA 2013-1531

Aero-Structural Blade Design of a High-Power Wind Turbine

Bruno M. Tojo* and André C. Marta†

Center for Aerospace Science and Technology, Instituto Superior Técnico,
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal

The present work focused on the development of an aero-structural design framework for a high-power horizontal axis wind turbine. To achieve this, it was necessary to carefully characterize the blade, to develop a suitable fluid-structure interaction solver and, finally, to combine both with post-processing tools. A fluid-structure interaction solver was developed for *OpenFOAM*, dedicated to simulate wind turbine rotors. The fluid-structure coupling was achieved through a loose coupling strategy, which means that there are separated solvers for the flow and structure analysis, which are combined through the update of boundary conditions. To simulate the blades rotation movement, an approach based on the single rotating frame method was used, meaning that the whole domain rotated with the turbine rotor with a constant angular velocity. The simulations of the rotor produced valid and interesting results namely, correct flow fields and pressure distributions. Considering that it is expected horizontal axis wind turbine rotors to continue growing to even larger sizes than the one modeled, it was shown that blade displacements due to flow induced forces are definitely a problem that needs to be taken into account when designing new wind turbine blades.

I. Introduction

Following the recent advances made in terms of wind energy generation, and considering that next step for this industry is to invest in offshore developments, the work presented consists on the development of an aero-structural design framework for high-power horizontal axis wind turbines. This section includes a brief review of the state-of-the-art in aerodynamic, structural and coupled models for wind turbine blades.

A. Aerodynamics and Computational Fluid Dynamics (CFD)

CFD studies are always a clever way to introduce, test and design new blade models. Disgraskar[1] presented a project that is somewhat similar to the one presented in the current work using *OpenFOAM*.[2] For this, special boundary conditions were developed, such as cyclic grid interfaces and a multi-reference frame solver, to model the wind turbine as accurately as possible. It was possible to model the NREL Phase IV and DU wind turbines.

Also using the NREL turbine,[3] a comparative study is presented between the actuator disc theory, CFD analysis on a high resolution structured mesh with advanced turbulence and transition models, and experimental results, discussing the possibility of creating a hybrid code. It was concluded that the CFD simulation provided an excellent match with experimental data in the attached flow regime. However, the CFD and panel code over-predict peak lift and tend to underestimate stalled flow. The results presented support the use of a calibrated actuator disk methodology based on a normal fidelity CFD modeling approach. Implementing actuator momentum correlations obtained from an equivalent CFD analysis offers a potentially economical methodology for simulating complete multiple wind turbines within a specific environment.

Another CFD study of the NREL PHASE IV wind turbine is presented by Sezer-Uzol[4] using a finite-volume solver for the Navier–Stokes equations for compressible, unsteady or steady-state flows. Three

*MSc Student.

†Researcher, AIAA Member.

conditions were studied in which the inviscid results show that the flow is attached for the low velocity cases, with the case where yaw angle is not zero having an asymmetric wake structure, whereas there is a massive separation over the entire blade span in the post-stall case. Although on a much smaller scale, the mesh presented in this work served as inspiration for the mesh presented in the current report, as the outer mesh is represented by a cylinder and the region of the blades has a much smaller elements then the outside mesh, size restriction had to be applied as the computations presented in[4] were made using NREL and NASA clusters.

The project presented by Nilsson[5] proposed to study *OpenFOAM* as a tool for turbulent flow in water turbines. The aim was to validate *OpenFOAM* performance against both commercial CFD codes and experimental data. In that paper, the results obtained were as efficient as the ones achieved using commercial codes, fitting equally well to the experimental data. Based on this, *OpenFOAM* can confidently be used as a CFD tool to analyze and design turbines.

B. Structural Models and Finite Element Method (FEM)

As part of the design process, a wind turbine must be analyzed for the aerodynamic, gravitational, inertial and operational loads that will experience during its design life. Although structural analysis has always been important in the wind turbine blade design, mainly because of fatigue problems, as the size of blade become bigger, so do the concerns with blade structural design.[6,7]

Jensen et al.[8] described the structural testing and numerical simulation of a 34 m composite HAWT blade with a load carrying main spar. Their work starts with the same premise that the current paper starts, that is the fact that HAWT are becoming bigger with a considerable increase in rotor size, which means that structural problems on the blades need to be addressed. The finite element method (FEM) has been used traditionally to investigate global behavior in terms of, for example, eigen frequencies, tip deflections and global stress and strain levels. In their work, the FE model was calibrated with data from a full scale model, which means its accurate and it was possible to evaluate complex loads, modeling actual wind conditions, otherwise impossible on the physical model.

The design of a wind turbine structure involves many considerations such as strength, stability, cost and vibration. In,[9] an optimization model for the design of a typical blade structure of HAWT is presented and it is focused on the reduction of vibrations. A good design philosophy for reducing vibration is to separate the natural frequencies of the structure from the harmonics of rotor speed. The main spar was represented by thin-walled tubular beam composed of uniform segments, each of which had different cross-subsectional properties and length. The optimization variables were chosen to be the cross-subsectional area, radius of gyration and length of each segment. The optimal design is pursued with respect to maximum frequency design criterion. The problem was formulated as a non-linear mathematical programming problem solved by multi-dimensional search techniques. Structural analysis was restricted to the case of uncoupled flapping motion of the rotating blade, where an exact method of solution was given for calculating the natural vibration characteristics. The results presented showed that the approach used was efficient and improved the reference design.

Bechly[10] described the optimization on the structure of a HAWT blade. A computational code was developed to make a detailed finite element mesh from data obtained from blade element theory and panel code predictions. This preliminary blade design was done using NACA 4412 airfoils and varying twist and chord, similar to the method that will be explained further in this document. The blade was then constructed from fiberglass skin and additional stiffening was added at the hub connection. A detailed finite element mesh of the blade was then built using a propose written program. The finite element results, obtained using a commercial FEM software, compared well with static bending and twisting deflections of the blade and, according to the dynamic analysis, it was even possible to understand that the natural frequencies of the rotating blade are higher than non-rotating, due to stress-stiffening.

C. Fluid-Structure Interaction

Following the concerns explored in the previous subsections, fluid-structure interactions (FSI) are currently becoming more relevant in the design of new HAWT. This phenomenon is connected to the deflection of elastic materials when subjected to flow induced loads, as well as the alterations that this deflection produce on the flow, changing its characteristics.

The FSI computations are usually classified in weakly or strongly coupled. For weakly coupled systems, it is common to use partitioned solvers, in which the main problem is how to pass the information between the solvers. On the other hand, for strongly coupled ones, the only choice is to use a single solver that solves both fluid and structural problems, but which is much more complex than the partitioned one.[11]

Aerostructural analysis techniques are a specialization of more general FSI solution methods, where the fluid in question is air, and the structure represents a flexible component such as an aircraft wing or wind turbine blade. Aerostructural design optimization using high-fidelity models is a computationally intensive multidisciplinary design optimization (MDO) problem, usually using high-fidelity aerodynamic analysis in the aerostructural problem while using considerably smaller finite-element models. This is often justified since the primary area of interest in these studies is the aerodynamic performance of the aerostructural system.

Kennedy et al.[12] presented both aerostructural analysis and design sensitivity methods that are designed to be efficient when the aerodynamic and structural disciplines both require significant computational resources and time. The aerodynamic analysis uses a parallel panel code, coupled to a parallel finite-element code for the analysis of composite structures. This finite-element code is specifically designed, including geometric nonlinearity and has analytic design variable sensitivity analysis capability. The inter-disciplinary coupling is handled by passing pressure and displacement values through a parametric surface representation of the outer mold line of an aircraft.

Aerodynamic and structural optimization of HAWT blades has become a subject of considerable interest, as the ultimate goal is to reduce the cost of energy production. This design process is a multidisciplinary task, involving conflicting requirements, such as maximum performance with minimum loads and noise. Moreover, since the turbine operates in a wide range of wind conditions, it can only be optimized for one. In,[13] the optimization was achieved by modifying the twist and chord of the blade, as well as the airfoils. A minimum cost of energy was obtained, which is lower then current commercial turbines, but it was also concluded that, with the use of traditional airfoils, it is impossible to reduce the cost of energy from the one that is already achieved.

Downloaded by Andre Marta on April 15, 2013 | http://arc.aiaa.org | DOI: 10.2514/6.2013-1531

II. Methodology For Wind Turbine Aero-Structural Analysis

Following the strategy illustrated in Fig.1 and having decided that the toolbox to perform the CFD simulations would be *OpenFOAM*[2] which, as clarified above, it is a good option for fluid structure interaction simulations, as well as accurate simulations on wind turbines. It was then necessary to evaluate the other parts of the methodology, such as the parametrization of the blade, mesh and geometry generators and finally, post processing of the results once the simulations were performed.

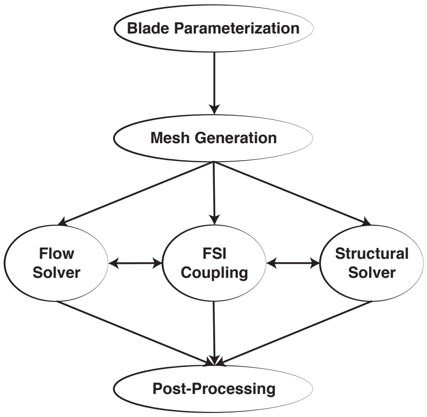


Figure 1. Schematic of WT aero-structural blade analysis.

A. Blade Parameterization

As a first approach, the surface will be modeled using a discretization presented by Kim et al.[14] This discretization starts with the selection of the airfoils, to which a series of coordinate transformations will be applied to obtain a HAWT blade. From the literature reviewed, DU airfoils seemed to be the obvious choice, unfortunately it was impossible to access these airfoils. As such, airfoils from NACA and S8xx series, with similar characteristics,[15] which were easier to access.

The power developed by a rotor at a certain wind speed depends on the relative velocity between the wind and the rotor tip. The ratio between these two speeds, called the tip speed ratio(λ), is defined as

$$\lambda = \frac{R\Omega}{U} == \frac{2\pi N_r R}{60U}\,, \tag{1}$$

where Ω is the angular velocity, R is the radius of the rotor and N_r is the rotational speed of the rotor. The design tip speed ratio is usually defined between 7 and 9 for large wind turbines. As λ gets bigger, the blade will be more slender and flexible, which has the advantage of reducing the load. However, it may cause interference problems between the blade and the tower at extreme wind conditions and the blade rotation speed may need to be increased to achieve the targeted output power. On the contrary, a smaller value will contribute for the blade being thicker and cause greater axial thrust forces.

The rotor geometry is periodic so it is recommended to use cylindrical coordinates. To define the 3D blade, it is convenient to define an non-dimensional parameter for the direction normal to the flow direction, used to define how the airfoil will be altered in a certain position, and defined as

$$\mu = \frac{r}{R}\,, \tag{2}$$

where r is the distance do the blade axis of rotation and R is the radius of the turbine.

As the blade is finite, it was necessary to define a blade tip loss factor,

$$f = \frac{2}{\pi} cos^{-1}\left(e^{1/((N/2)(1-\mu)/\mu)\sqrt{1+(\lambda_{design}\mu^2)/(1-a)}}\right)\,, \tag{3}$$

as it was first define by Prandlt and then modified by Glauert[16] in their aproach to the BEM theory. This definition depends on the flow induction factor a, which the exact value is given by

$$a = \frac{1}{3} + \frac{1}{3}f - \frac{1}{3}\sqrt{1-f+f^2}\,, \tag{4}$$

making it necessary to calculate this values using a recursive method.

Wind turbine blades are designed with tapered and twisted shapes to obtain maximum power output with high efficiency. In[14] several discretizations for the tapper across the blade were proposed, having all similar aerodynamic results, from which the best chosen to be applied in this work was

$$\frac{c}{R} = \frac{2\pi}{N} \frac{4\lambda^2\mu^2 a'}{\sqrt{(1-a)^2 + (\lambda\mu(1+a'))^2}}\,, \tag{5}$$

where N is the number of blades in the HAWT and C_l is the lift coefficient of the airfoil for the angle of attack α defined, which depends on another flow induction factor a',

$$a' = \frac{a(1-a/f)}{\lambda^2\mu^2}\,, \tag{6}$$

which also takes in account the tip speed ratio and the position in the blade.

At the locations for which the chord was calculated, it was also calculated the blade twist ϕ , which is defined by

$$tan(\phi) = \frac{1-a}{\lambda\mu(1+a')}\,, \tag{7}$$

which is then applied by rotating the airfoil at that position by a pitch angle θ which also depends on the angle of attack on the blade as is defined by

$$\theta = \alpha - \phi\,. \tag{8}$$

All these equations were programmed in a *python*[17] script that accepted regular airfoil files, being possible to define the airfoil in up to five radial subsections of the blade, and where the positions that were not defined by the user would be a result of a interpolation between the coordinates of the two prescribed airfoils that were closer to that location.

B. Mesh Generation

To have an acceptable mesh that produced good results in *OpenFOAM*, it was necessary to evaluate the capabilities and possibility of integration with the solver software of several mesh generators.

After selecting *ICEMCFD* as the mesh generator, a *python* script was developed and functions were defined to automatically generate the different commands, such as straight lines, splines and surfaces. This script was different for the solid and fluid meshes and the commands generated were then saved to *ICEMCFD* replay files. These codes generate the geometries defined for the fluid and solid domains and define the blocking for each of the geometries. The mesh is then computed and exported in *Fluent* format, and is copied to the case folder and converted to the *OpenFOAM* format using *fluent3DMeshToFoam*, which is the default mesh converter from *OpenFOAM* to convert three-dimensional *Fluent* meshes. All this process was programed to be automatic.

Using the capabilities of the software used and the code created, the point cloud created previously was uploaded, and using these points the blade starts to be formed by defining splines across the all points of the blade. The surfaces were programed to be created automatically from these lines, using as much automation as possible in order to be possible to change the geometry at any time without having to redefine all the surfaces.

C. Post Processing

To evaluate the quality of the design, it was necessary to calculate the power that could be extracted from the blades. The method achieved consists of using an *OpenFOAM* library that calculates forces and moments on a determined surface for each time step, which can be viscous or pressure forces, saving the results to a data file, which means it would give all the information needed to calculate the mechanical power generated by the blades. The information obtained has then to be processed: the correct couples of forces (viscous and pressure based) were added the power obtained from a rotating body was calculated using moment generated by the flow on the blades (Torque - T),

$$P = T \times \Omega \,, \tag{9}$$

where Ω is the rotation speed in rad/s. In addition, the drag/trust forces were also calculated, which are important as the structural elements in the HAWT must be able to support it, and the power coefficient (C_P) which served as a comparison between the results obtained, and the maximum theoretical possible and the values which are reachable with todays top commercial solutions.

III. FSI Solver Definition

A. icoFsiFoam

icoFsiFoam is the fluid-structure interaction (FSI) solver available in the standard distribution of *OpenFOAM*.[18] This solver is useful for weakly coupled systems with small structural deformations and laminar fluid flow. The structural part is based on *stressedFoam*, thus limited to linear stress strain relationships and small deformations. The fluid part of the solver is based on the *icoFoam* and is limited to Newtonian fluids in incompressible, laminar flow.

This solver consists, in fact, in having multiple solvers side-by-side, which means that each solver uses its own mesh, with access to its fields, material properties, solver controls, etc. The fluid-Structure coupling is achieved through boundary condition updates. In this case, this coupling is achieved by the following loop: the pressure data from fluid is moved to structure; the structural equations are solved; the displacement of the solid mesh is then transfered to fluid mesh; the fluid flow is solved with mesh motion and the loop restarts.

To test this solver, it was used a flow around a 2D structured mesh, and its structural counterpart. The flow has an uniform velocity inlet at $4m/s$ and, the results are presented in Fig.3 and Fig.4 for comparison with the developed solver.

B. simpleFsiFoam

Due to the major limitations of *icoFsiFoam*, as the impossibility to define a turbulent flow, it was necessary to develop a solver that could model the flow around a turbine blade more accurately. This was considered important due to the fact that the flow over wind turbine blade is always turbulent. It was decided to develop *simpleFsiFoam*, a FSI solver based on *simpleFoam*. This solver takes advantage of the simplicity of *simpleFoam*, as well as the extra functionalities that it has over *icoFoam*.

As simplicity and flexibility were important, the fluid-structure coupling was achieved in the same way that is done in *icoFsiFoam*, which means that there is still two different meshes, one for the fluid and the other for the structural solver. The meshes do not need to have the same amount of elements in the correspondent boundaries, which is ideal for the coupled optimization process.

As described earlier, the partitioned approach is being implemented for this work, employing a staggered solution procedure wherein each domain. This approach was also used because of the compatibility with black-box solvers, which means that even though this solver has been developed to use *stressedFoam* and *simpleFoam*, any of the solvers may be replaced by a similar one, without much change in the code, simply replacing the part were the calculations are made.

Each iteration begins with setting the pressure around the solid mesh, using the last iteration fluid information. The solid part is then solved and the displacement is applied, deforming both of the meshes, and finally the fluid is solved. After each iteration, the solver evaluates if it is time to save the solution, as it is possible to define in *controlDict* the writing interval, and increments time, until the end time is reached.

To define a case, it is necessary to define the two cases in the same folder and define symbolic links between them, as is shown in Fig.2, for the solver to get input to the solution of both the fluid and structural equation systems. The fluid-structure interface is defined after both the fluid and solid meshes have been created.

The basic layout is the same as in any *OpenFOAM* case, but much more information is required. In the *0* folder for the fluid is defined the initial and boundary conditions, using the *U* file to define velocity associated to each boundary; the *p* file to impose the pressure settings; and *nuT*, *nuTilda*, *k*, *omega* and *epsilon* are used to define the turbulence properties depending on the turbulence model to use. The file *motionU* contains the conditions for the deforming mesh motion, which is a requirement for the FSI analysis. Using *movingWallVelocity* as the boundary condition in *U* at the interface, the motion of the mesh at the interface is implemented as a Dirichlet boundary condition for the fluid velocity. For the solid mesh, in *0* the only property that has to be defined is the initial displacement which is set on the *u* file.

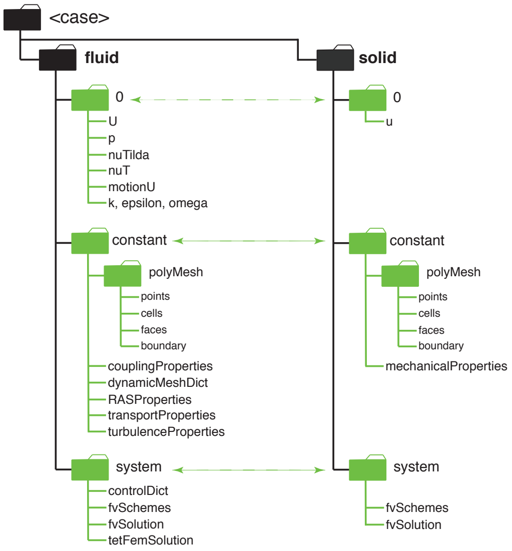


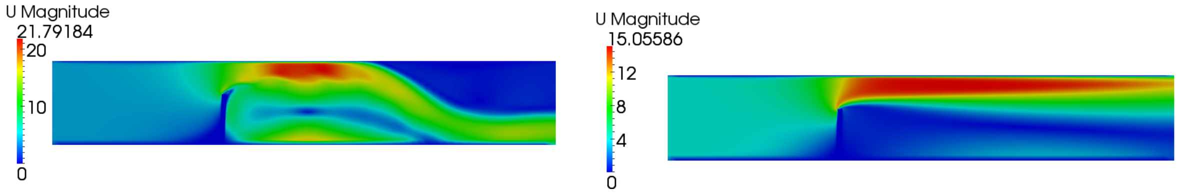
Figure 2. Schematic of the definition of a case for *simpleFsiFoam*

In the the *constant* directories most of the information contained is about the meshes and the material properties, as explained previously, the *polyMesh* folders contain all the necessary information about the meshes, as for the fluids transport properties, such as density and viscosity, are defined in the file named *transportProperties*, while the solids material properties like density, Young modulus, possion coefficient etc., are defined in *mechanicalProperties*. As the names denote, the turbulence properties and model are defined using the *RASProperties* and *turbulenceProperties* files. In the *constant* of the fluid part are also defined most of the settings governing the fluid-structure interaction are defined: in *couplingProperties* are defined the coupled boundaries and the moving mesh patch; the settings for the solution of this patch motion are defined in *dynamicMeshDict*.

The *system* directories are where the solution procedure is setup. Settings such as end time, time step and time interval for output to be written are all defined in *controlDict*, also in this file is defined if some none default library is going to be used, as is the case with the library that calculates the forces or the special libraries for turbo-machinery applications. The files *fvSchemes* contain information about which differencing schemes to use and the files *fvSolution* the settings for convergence tolerances.

In the development versions, *OpenFOAM* has a vertex-based, unstructured mesh motion solver, which is used in this case. This solver executes Laplace face decomposition and performs very well for many problems and it is easy to implement because only motion on the tracked interface is required, but it does have limited functionality. In particular, in its current form it does not function properly for parallel solutions and it does not support periodic (cyclic) boundaries. This is a major limitation to the use of *OpenFOAM* as a fluid-structure interaction tool for turbo-machines, as this machines usually have periodic geometries, which means that a big part of the computing power is wasted because of the impossibility to use periodic boundaries. Furthermore, in the special case of the horizontal axis wind turbine where 3D effects are very important, the analysis would really benefit from a finner mesh that could only be feasible with the use of parallelization.

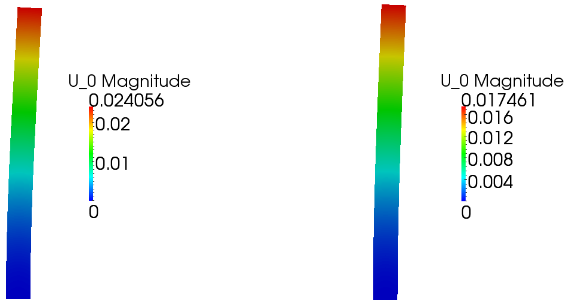
The same case that was tested with *icoFsiFoam* was tested with the new solver. The results were similar but, as in the case where it was used *simpleFoam*, the decrease in velocity due to turbulence is evident.



(a) *icoFsiFoam* **(b)** *simpleFsiFoam*

Figure 3. Velocity arround a 2D beam using FSI solvers.

Comparing the results of the deformation of the beam between the two fluid-structure interaction solvers in Fig.4, it is possible to observe that the laminar solver produces a larger deformation of the beam, as the velocity is bigger, so is the pressure differential between the two sides of the blade.



(a) *icoFsiFoam* **(b)** *simpleFsiFoam*

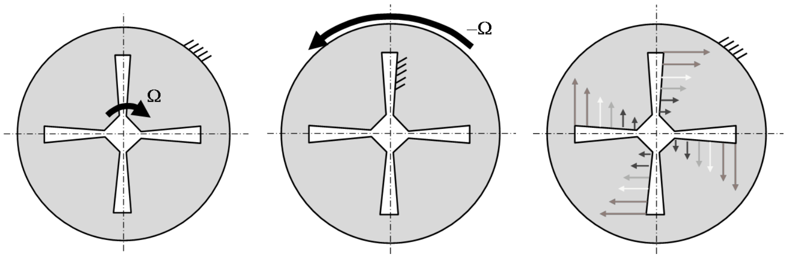
Figure 4. Displacement caused by the flow around a 2D beam

C. fsisimpleSRFFoam

After trying to obtain results by applying a series of rotating velocity boundary conditions, using *simplefsiFoam*, this revealed to be impossible. To obtain better results for the HAWT blades simulations, and considering the limitations in the computing power available, it was necessary to find a solution that brought better results and was computationally cheap.

The solution to this problem was obtained by using a single rotating frame (SRF) solver, since a rotating frame of reference is often used to model flows in rotating machines. In this solver the flow is unsteady in an inertial frame because the blades or rotors sweep the domain periodically as shown in Fig.5(a). However, it is possible to perform the calculations in a domain that moves with the rotating coordinate which is fixed on the rotating part, and for this the angular velocity must be set as symmetric to the real velocity (Fig.5(b)). The flow is steady relative to the rotating frame, which reduces the computational cost needed for a more accurate analysis. When transposing the results for the inertial referential, it appears as the blades are moving(Fig.5(c)).

This approach is appropriated when the flow at the boundary between the rotating parts and the stationary parts is weakly affected by the interaction. It provides a reasonable time-averaged simulation result for many applications. Rotating frames do not physically rotate anything and therefore do not show transient effects due to the real motion and the Coriolis and centrifugal forces are treated as body forces. Any problems where transient effects due to rotor-stator interaction are small are candidates to use the rotating frame of reference approach.



(a) Inertial referential **(b)** Rotating referential **(c)** Equivalent motion of the blades

Figure 5. Single rotating frame of reference.[19]

Using the same strategy as it was explained in Sec.B, it was developed another FSI solver for *OpenFOAM*, now using as base the Single Rotating Frame Solver(*simpleSRFFoam*). This was the solver used in the final simulations, which produced relatively good results. In the definition of the case, the only differences to the *simpleFsiFoam* case is that the rotational speed must be defined, in the *constant* folder, in a file called *SRFProperties*, and the boundary conditions for the velocity are now defined in the *Urel* file, where the velocity in the boundaries may be defined as relative or absolute. As stated before, all the calculations are made in the rotating referential and, only at the end, properties are transposed to the inertial referential. In this the absolute velocity is called *Uabs*.

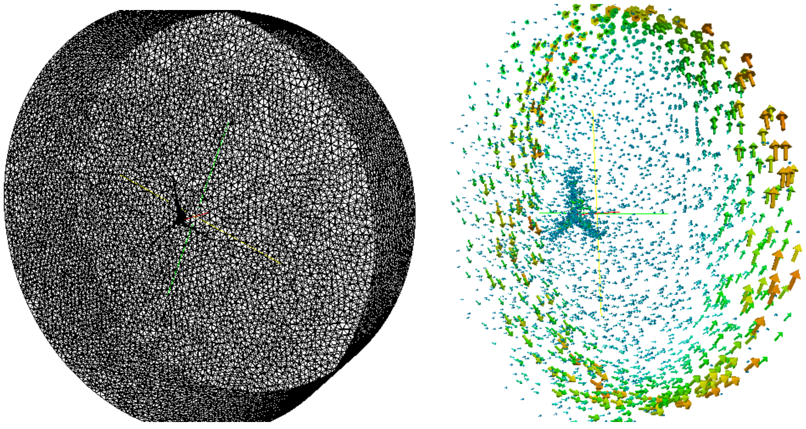
IV. Wind Turbine Definition and Simulations

The preliminary simulation results that were considered important to reach to the FSI simulation, as well as the results for this simulation are presented in this section.

A. Wind Turbine Flow Simulation

Using *simpleSRFFoam* it was then possible to simulate correctly the flow around the blades. The mesh used is presented in Fig.6, where the angular velocity was set to $-15.11rpm$ and the inlet axial velocity was set to $12.1m/s$. The results for the velocity vector field (Fig.6(b)), were as expected in a rotor of a wind turbine, and the convergence was achieved quickly and even the values for mechanic power produced, from the pressure differential between the side of the blades, seemed promising.

After defining the blades, the last concern was a way to connect them to the central hub. To give a more



(a) Mesh **(b)** Velocity vector plot

Figure 6. SRF simulation of the rotor.

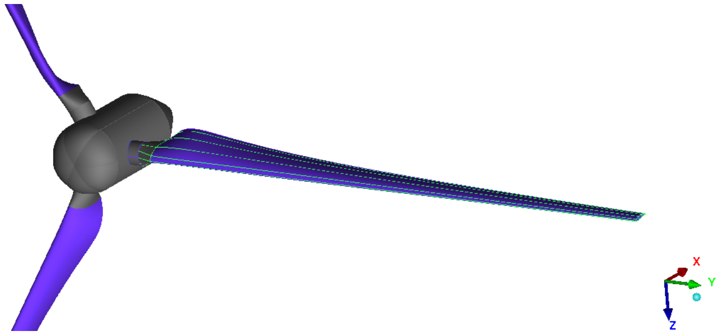


Figure 7. Final geometry of the HAWT rotor.

realistic feeling to the flow, the blades were joined to it with simple cylindrical connections. The hub and the connections are presented in gray in Fig.7. This is not a perfect solution as, in a real wind turbine, only a very small portion of the hub actually rotates, as the rest is used to store the electric generator, but this solution solved the problem explained before and it seemed to generate a wake much like it was expected. The dimensions and characteristics of the final simulation are presented in Tab.1.

Table 1. Final design parameters of a HAWT blade.

Rated wind Speed	$12.1 m/s$
Diameter	$103.8m$
Number of Blades	3
λ_{design}	7.5
Rotor Speed	$15.11rpm$
Angle of Attack	2°
Hub Diameter	$3m$
Hub length	$10m$

The outer surfaces of the mesh, where the inlet and outlet of the flow are located, were defined by a third of a cylinder defined using the dimensions of the blade as input parameters and then multiplying them to obtain a volume big enough so the outer boundaries did not affect the flow around the wall. It was first tried to run the simulations with only one sector of the rotor, using periodic boundaries, but it revealed to be unfruitful as the simulations had convergence problems. Using *ICEMCFD* geometry functions, the points, lines and surfaces created were copied and rotated to angles of 120^o and -120^o, forming the complete fluid domain presented in Fig.8.

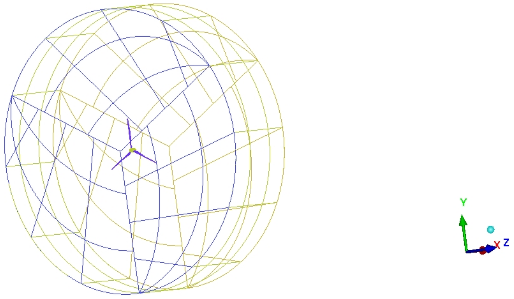


Figure 8. Fluid domain of the final fluid mesh.

The blocking on this geometry was done very carefully to obtain a good discretization of the domain. This geometry was then partitioned in 73 blocks, that defined all the fluid around the three blades and the hub.

The mesh generated was chosen to be tetrahedral, to model the blade with better detail, and the element size was tunned so that it was possible to obtain good results with as small as possible computational power. The rotor region was defined with more definition to account for boundary layer effects and obtain better results for the fluid structure coupling. The elements in this area are around five times smaller then the ones on the rest of the mesh. In an effort do maintain the simulation as quick as possible the mesh was restrained at a little above one million cells, which give it good definition. The final fluid mesh can be observed in detail in Fig.9.

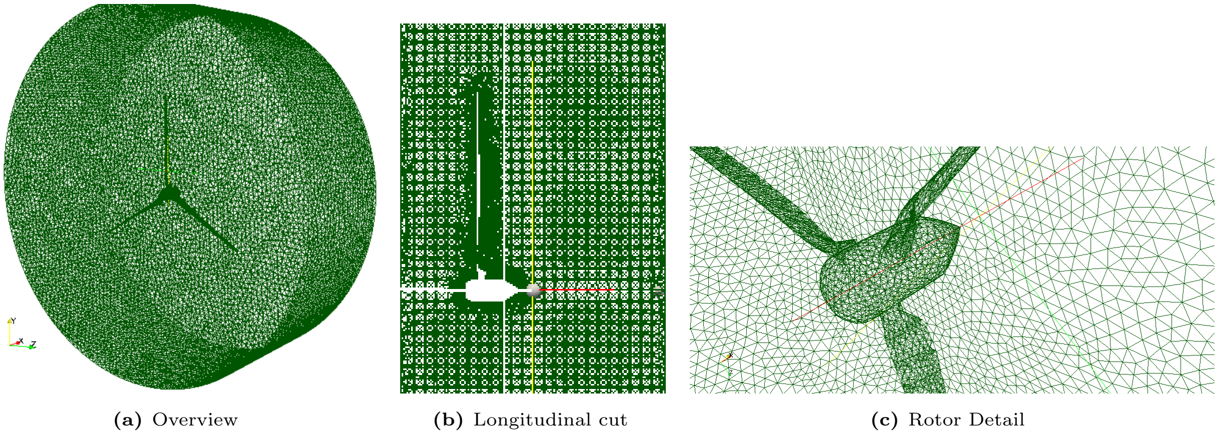

(a) Overview **(b)** Longitudinal cut **(c)** Rotor Detail

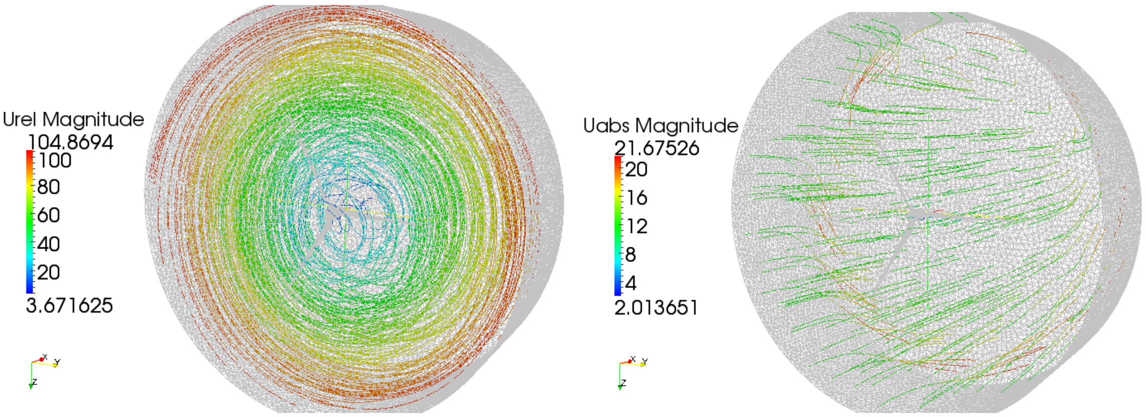
Figure 9. Final fluid mesh details.

This mesh was also tested through a *simpleSRFFoam* simulation using the same conditions described above. The results for the simulation are presented in Fig.10, the velocity field is now exactly as it was expected, rotating at constant speed and with visible axial velocity, consistent with the previous cases.

If it is considered that the values are symmetric to the real ones due to the use of the rotating frame, the pressure on the rotor surface (Fig.11) also shows a distribution consistent with what is expected from a HAWT blade, with a clear pressure differential between the two sides, presenting a suction peak near the leading edge.

B. Wind Turbine Blade Structural Simulation

At this preliminary stage of work, it was necessary to simplify the structural model due to time restrictions. The contributions of the leading and trailing edge of the blade were eliminated to have a more regular mesh. As it seemed that the problems with the *ICEMCFD* meshes in *stressedFoam* were related to the unstructured meshes, it was created a structured mesh for the wind turbine blade that could represent this blade accurately using only hexahedral elements.



(a) Stream lines for the relative velocity

(b) Stream lines for the absolute velocity

Figure 10. Final results for the flow simulations.

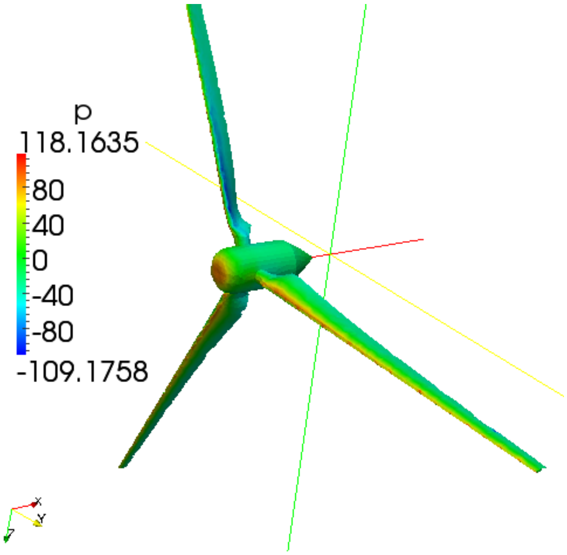


Figure 11. Pressure on the rotor.

The blade is then defined as a solid with hexahedral elements seen in Fig.12, with similar structural characteristics to aluminum. This is a very simplified solution but acceptable as most of the structural loads of the blade are supported by webs in the center part of the blade.[8]

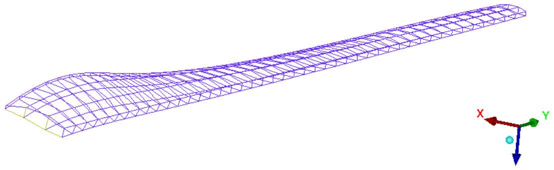
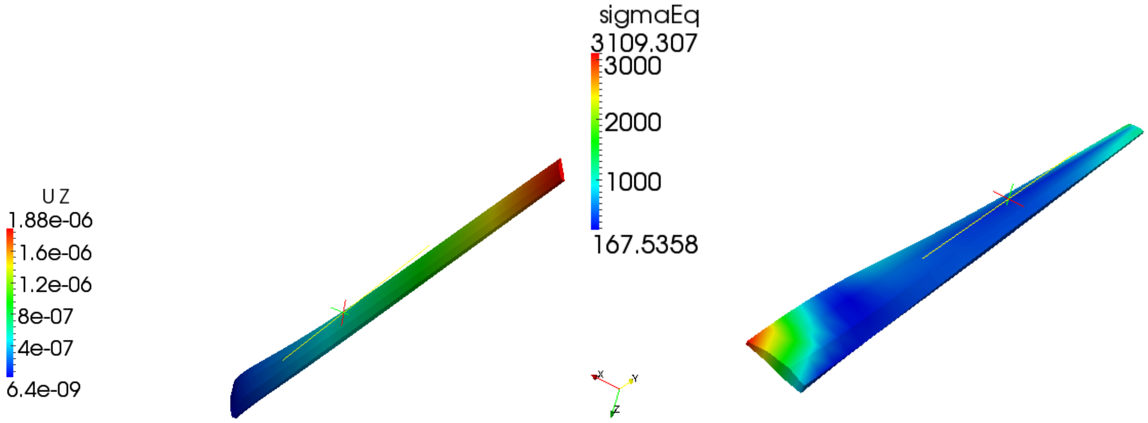


Figure 12. Blade structural mesh.

Using this mesh, and applying a pressure of $600Pa$ in the z direction at the tip, the results obtain are presented in Fig.13. The displacements of this simplistic, over-designed, solid blade core structure were extremely small, as expected. As for the stress, the results obtained were also physically consistent and show that the major concentration is in the thinner portion in the clamped top.



(a) Displacement z

(b) Equivalent stress

Figure 13. Simulation of the blade using *stressedFoam*.

This model was used in the following FSI studies but a more realistic internal blade structure will be included in the final version of this paper.

C. FSI Simulation and Results

After all preliminary studies concluded, the results for the fluid-structure interaction in a HAWT are presented in this subsection. The meshes, materials properties and fluid boundary conditions were the same as before.

The results obtained for the deformation are shown in Fig.14. This displacements, considering all the simplifications that were made, are impressive. If a solid metal beam dislocates more than $2cm$ just by the effects of the flow, it is possible to infer that these displacements, can indeed become a problem. Furthermore, it is also notable that the major displacements happen in the directions normal to the flow. This clearly shows that the solver is being capable of passing the single rotation frame results to the structural domain.

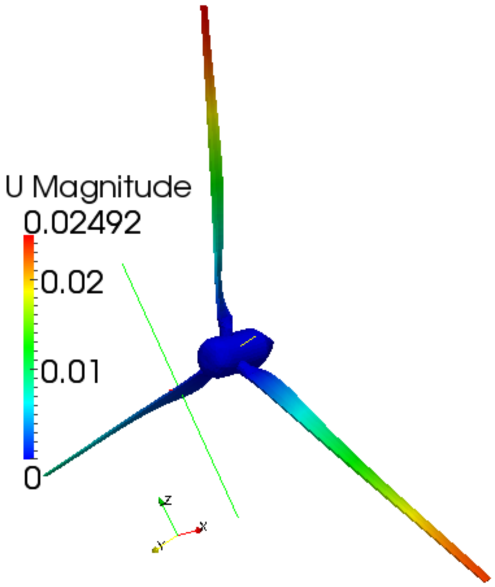
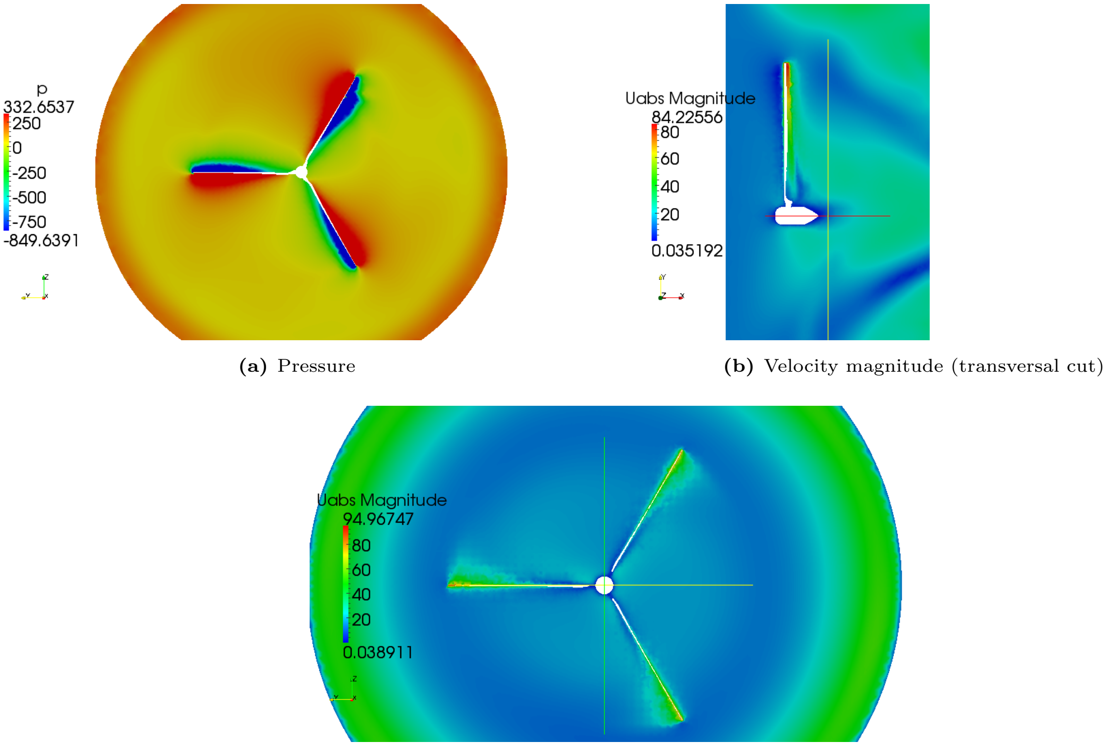


Figure 14. Displacement in the blades due to the flow.

In the flow domain, the results shown in Fig.15 are also interesting: the pressure differential between the sides of the blade deforms it and pushes it, producing power. The velocity profile also showed physically



(a) Pressure

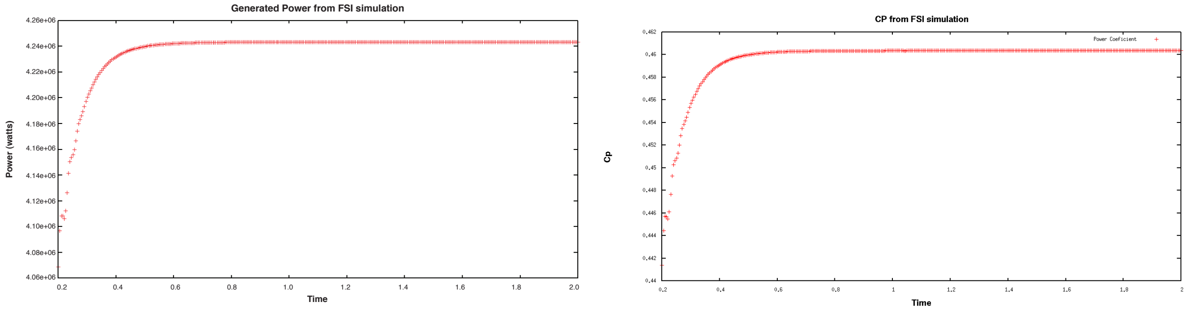
(b) Velocity magnitude (transversal cut)

(c) Velocity magnitude (axial cut)

Figure 15. Flow results on the FSI simulation of the HAWT blades.

correct results, the domain is definitely moving at constant rotational speed as, it is visible in Fig.15(b), the velocity increases linearly from the root of the blade to achieve the maximum value in the tip.

Using the script described in Sec.C, the values for the mechanical power and power coefficient were calculated and presented in Fig.16. The results are probably a bit over predicted since the *forces* script



(a) Power, converged at $4.24MW$

(b) C_p, converged at 0.46

Figure 16. Charts of the power produced in the FSI simulation.

in *OpenFOAM* might not be very accurate. Nevertheless, the turbine in which the parameterization was based[14] was projected to produce $5MW$ so the results obtained are in concordance with this.

The values obtained of $4.24MW$ for produced power and C_p of 0.46 are consistent with current values for high power wind turbines but are probably too optimistic for a design in this early stage. Anyway, this demonstrates that the method for designing the blades for wind turbine works and produces acceptable results.

V. Conclusions and Future Work

The preliminary results obtained are a clear indicator that the aero-structural design of high-power wind turbine blades is possible using open source software. The development of a new *OpenFOAM* multi-disciplinary solver that accounts for the fluid-structure interaction for a WT blade rotor under the assumption of incompressible, turbulent flow, with rotating parts, was completed.

Future work will focus on two points. First, further development of the structural solver and structural model of the blade core are needed, to accurately represent spars and ribs, to form a boxed structure, and blade skin. The second area of development is in the use of the aero-structural analysis developed in a multi-disciplinary optimization framework to perform some optimization exercises to maximize the output power or reduce blade mass, with constraints on maximum stress and blade tip deflection.

References

[1] Digraskar, D. A., *Simulations of Flow over Wind Turbines*, Mechanical and industrial engineering, University of Massachusetts Amherst, MA,USA, May 2010.

[2] OpenCFD, "OpenFOAM - The Open Source CFD Toolbox - User's Guide," OpenCFD Ltd. version 1.6, http://www.openfoam.org/archive/1.6/docs/user/, April 2009.

[3] Hartwanger, D. and Horvat, A., "3D Modelling of a Wind Turbine using CFD," *Proceedings of the NAFEMS Conference*, Gloucestershire, UK, June 2008.

[4] Sezer-Uzol, N. and Long, L. N., "3-D Time-Accurate CFD Simulations of Wind Turbine Rotor Flow Fields," *Proceedings of the* 44th *AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, Jan. 2006, AIAA Paper 2006-394.

[5] Nilsson, H., "Evaluation of OpenFoam for CFD of turbulent flow in water turbines," *Proceedings of the* 23rd *IAHR Symposium*, Yokohama, Japan, Oct. 2006.

[6] Herbert, G. J., Iniyan, S., Sreevalsan, E., and Rajapandian, S., "A review of wind energy technologies," *Renewable and Sustainable Energy Reviews*, Vol. 11, No. 6, 2007, pp. 1117–1145.

[7] Ashwill, T. D., "Materials and Innovations for Large Blade Structures: Research Opportunities in Wind Energy Technology," *Proceedings of the* 50th *AIAA Structures, Structural Dynamics & Materials Conference*, Palm Springs, CA, USA, May 2009, AIAA Paper 2009-2407.

[8] Jensen, F., Falzon, B., Ankersen, J., and Stang, H., "Structural testing and numerical simulation of a 34m composite wind turbine blade," *Composite Structures*, Vol. 76, 2006, pp. 52–61, ICCM-15 Fifteenth International Conference on Composite Materials.

[9] Maalawi, K. Y. and Negm, H. M., "Optimal frequency design of wind turbine blades," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 90, No. 8, 2002, pp. 961–986.

[10] Bechly, M. and Clausen, P., "Structural design of a composite wind turbine blade using finite element analysis," *Computers & Structures*, Vol. 63, No. 3, 1997, pp. 639–646.

[11] Bungartz, H. J. and Schäfer, M., *Fluid-Structure Interaction: Modelling, Simulation, Optimisation*, Lecture Notes in Computational Science And Engineering, Springer-Verlag, 2006.

[12] Kennedy, G. J. and Martins, J. R. R. A., "Parallel Solution Methods for Aerostructural Analysis and Design Optimization," *Proceedings of the* 13th *AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, USA, Sept. 2010, AIAA Paper 2010-9308.

[13] Fuglsang, P. and Madsen, H., "Optimization method for wind turbine rotors," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 80, No. 1, 2, 1999, pp. 191—206.

[14] Kim, B., Kim, W., Bae, S., Park, J., and Kim, M., "Aerodynamic design and performance analysis of multi-MW class wind turbine blade," *Journal of Mechanical Science and Technology*, Vol. 25, No. 8, 2011, pp. 1995–2002.

[15] Van Rooij, R. P. J. O. M. and Timmer, W. A., "Roughness Sensitivity Considerations for Thick Rotor Blade Airfoils," *Journal of Solar Energy Engineering*, Vol. 125, No. 4, 2003, pp. 468–478.

[16] Glauert, H., *Windmills and fans - Aerodynamic theory*, Julius Springer, 1935.

[17] "Python Reference Manual," PythonLabs, Virginia, USA, 2001. Available at www.python.org.

[18] *The OpenFOAM®Extend Project*, http://www.extend-project.de/, 1st ed., 2012.

[19] *Autodesk® Algor® Simulation User's Guide*, Available at http://download.autodesk.com/us/algor/userguides/mergedProjects/setting_up_the_analysis/fluid_flow/loads_and_constraints/rotating_frame_of_reference.htm, build 15 ed., Apr 2009.