13th EAWE PhD Seminar on Wind Energy
20-22 September 2017
Cranfield University, United Kingdom
Session XXXX Paper XXXXXXX

NUMERICAL SIMULATION OF FLOATING WIND TURBINE USING ANISOTROPIC MESH ADAPTATION

Louis DOUTEAU [2] **Luisa ROCHA DA SILVA** [1] **Hugues DIGONNET** [1]
Student,Louis.douteau@eleves.ec-nantes.fr

Thierry COUPEZ [1] **David LE TOUZE** [2] **Jean-Christophe GILLOTEAUX** [2]

[1] High Performing Computing Institute (HPCI), Centrale Nantes, France
[2] Research Laboratory in Hydrodynamics, Energy and Atmospheric Environment (LHEEA), Centrale Nantes

ABSTRACT

The demand in renewable energy has significantly increased over a few years and, consequently, the industrial production of renewable energy has considerably expended. In the meantime, wind energy has matured, and many wind farms have been installed, both onshore and offshore. Offshore wind is an emerging field where current fixed-bottom technology is limited when water depths exceeds 50 m. Consequently new solutions are currently being explored, with, in particular, the development of floating supports. Moving offshore enable to capture stronger and more constant winds, but may lead to higher CAPEX and OPEX. One way to reduce costs is to use numerical simulation for optimizing the whole structure from the mooring lines to the blades. The numerical tools involved enable a fine prediction of the behavior the structures have under a large span of conditions, e.g. the loads applied on the structures when extreme events occur. These results can lead to an adjustment of the security coefficients of the wind turbines, which can reduce the CAPEX costs.
This work focuses on a methodology enabling the simulation of one or several floating wind turbines using full-CFD, with an accurate representation of their respective geometries. The software library used is ICI-Tech, developed at the High Performing Computing Institute of Centrale Nantes. A monolithic approach is used, with a single mesh in the simulation, where all the interfaces are defined using modified level-set functions. The Navier-Stokes equations are solved using stabilized finite elements and the Variational MultiScale formulation. In order to largely reduce the computational costs, an anisotropic and automatic mesh adaptation is done, which enables to capture physical phenomena having different orders of magnitude. The first results of mesh immersion and aerodynamic simulations are presented.

INTRODUCTION

The development of renewable energies has significantly accelerated over a few years. Wind energy is particularly interesting to that extend with massive amount of knowledge gathered, as wind farms have been implanted onshore for more than a decade. However, the future of wind energy seems to be located offshore. Ground bound wind farms have been producing energy for a decade, especially in North Sea, but only few places are eligible to the implantation of wind turbines as low water depth is needed. To that extend, floating wind turbine (FWT) represent a major field of development for the wind industry. Prototypes are already under study off the coasts of Norway, Japan and Portugal, and a floating wind farm is to be installed offshore Scotland by the end of 2017.

Fast and accurate simulation methods are crucial for the development of floating wind energy. Simulating floating wind turbine under various aerodynamic and hydrodynamic cases, and especially extreme events, will enable a better dimensioning of both the turbine and the floating structure. This optimisation of the geometries currently used is critical for an industrial deployment of this sector, currently noncompetitive compared to more conventional energy sources. Motivated by the development of onshore wind energy, many authors studied the aerodynamic behavior of turbines, e.g. [1], often without representing exactly the blades to reduce the computational effort. The wake effect due to the wind

turbine is modeled, which leads to trustworthy aerodynamic results far from the turbine, but also to low precision data near the blades. This limitation can be overcome if full resolution of the flow equation are done in the fluid domain, and some studies consider this, e.g. [2], making them deal with computationally expensive simulations. Hydrodynamic effects also hold a huge influence in these computations, as [3] showed that a combination of high amplitude swell and calm wind can inverse the direction of the relative wind at the rotor. However, dealing with a wind turbine under aero and hydro effects requires high efforts in both development and computations. Consequently, only few authors interested themselves in coupled simulations, e.g. [4] or [5]. In a majority of cases, only a component was studied at a time, as did [6], [7] or [8]. The same situation is observed in the literature for experimental studies. Even if a lot of turbines have been studied in wind flumes, few experiments focused on both wind and waves effects using validated generators. The work in [9] and [10] can be noted, where only particular aerodynamics behavior have been studied.

The full resolution of the geometries, boundary layers and hydrodynamic effects in the simulation of a floating wind turbine is a real challenge. Events with very different orders of magnitude for both their characteristic time and length are observed in a large computational domain, e.g. between the propagation of the swell and the aerodynamic vortex at the trailing edge of the blades. This problem can be handled using actuator disk methods to approximate the small-scale aerodynamic effects observed at the rotor, e.g. in [11], or with several overlaid meshes having different levels of refinement, e.g. [4]. This work proposes an other methodology, based on a unique computational mesh and a monolithic approach. The simulated wind turbine is immersed into the computational mesh, and is defined using a level-set function, while the air/water interface is specified with mixing laws and level-set function. The Navier-Stokes are solved in all the domain using stabilized finite elements and a Variational-MultiScale (VMS) formulation. The computational mesh is adapted anisotropically and automatically during the simulation, with its deformations ruled by an error estimator. The generated mesh is well adapted to the simulation of floating wind turbines, as it is refined only at the interfaces and where small-scale events occur.

The methodology used in this work is detailled, with a focus on mesh immersion and on the simulation procedure. The first results obtained are presented.

METHODOLOGY

MESH IMMERSION

SIGNED-DISTANCE FUNCTION

The mesh immersion procedure is an efficient way to rebuild into a computational domain an object represented by a structure mesh. This method is based on the determination of the material properties at each point of the computational mesh. To do so, the distance from each point of the computational mesh to the boundaries of object is computed, and a signed distance function is introduced. The interior of the object is defined by all the points having a positive signed distance, and consequently, the results of the immersion directly depends on the computational mesh. In ICI-tech, a coupling between immersion and automatic mesh adaptation is done to guarantee a good positioning of the mesh, i.e. most of the points concentrate in the critical areas of the computation. Moreover the adaptation process is anisotropic, which enables the use of highly stretched cells at the interfaces detected in the domain and, consequently, reduces the number of points needed in the simulation and the computational effort.

$$\alpha = \bar{d}(x,\Gamma) = \begin{cases} d(x,\Gamma)\ if\ x \in \omega \\ -d(x,\Gamma)\ if\ x \notin \omega \end{cases}.$$

Equation 1. Signed-distance function.

$$H_\varepsilon(\alpha) = \frac{1}{2}\left(1 + \frac{u_\varepsilon(\alpha)}{\varepsilon}\right), \text{ with } u_\varepsilon(\alpha) = \varepsilon \tanh\left(\frac{\alpha}{\varepsilon}\right).$$

Equation 2. Smoothed Heaviside function.

The major steps of the immersion technique, initially proposed by [2], are presented in this work. The reconstruction of an object ω, whose boundaries are described by a mesh Γ, is immersed into a domain Ω. The signed-distance function α presented in Eq.1 is the most natural choice, and corresponds to the combination of a distance and a Heaviside function. However, this immersion procedure doesn't highlight well the boundaries between phases, and is not

appropriate for small objects or details. Moreover, the frontier between two phases obtained with this technique is very abrupt, and all the transition of phase has to be handled within a single mesh cell. Therefore, the stabilization of the numerical schemes used in the simulation is almost impossible. The introduction of a smoothed Heaviside function, visible in Eq.2, is done in the signed-distance function to improve the stability of the computation. The parameter ε introduced into the distance function softens the boundaries between phases in the computation, generating a transition area around the frontiers where mixing laws have a great influence. The smaller the ε is chosen, the thinner the representation of the object is, if an appropriated mesh is used.

To highlight the influence of the smoothing function, an example is given. The computation of the distance to an immersed object of approximative size *2x1* is done in Fig.1. The defaults of this method, previously overlooked, are notable here. The smoothed function, computed with $\varepsilon=10^{-2}$ and the adapted mesh shown in Fig.2, in given in Fig.3. The small size of the transition area, coupled with the high concentration of points in this zone, guarantees a good representation of the object, along with a better convergence of the computations.

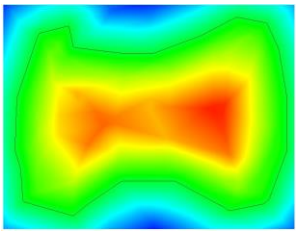

Figure 1. Signed-distance function.

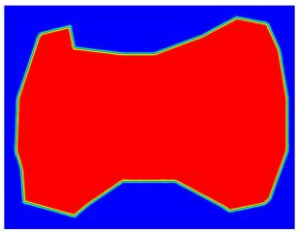

Figure 2. Smoothed Heaviside function.

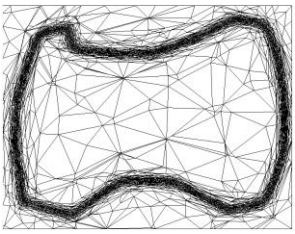

Figure 3. Adapted mesh.

The mesh reconstruction is expensive, computationally speaking, as, for each point of the domain, the closest distance to the immersed mesh has to be computed. If M cells compose the immersed mesh, for N points in the domain, the computations of $N \times M$ distances is needed. A tree data structure has been implemented to limit the complexity of the immersion, which can become at best $Nlog(M)$, but is strongly dependent on the distribution of the immersed mesh cells. For wind turbine immersion, the gain has been significant.

ANISOTROPIC AND AUTOMATIC MESH ADAPTATION

The quality of results directly depends on the quality of the mesh used for the reconstruction. In this work, the coupling of the immersion procedure with an automatic mesh adaptation, described in [13], is necessary to get a precise reconstructed object. A gradient operator is built at each point from the post-treated signed-distance results. If the Navier-Stokes equations are solved in the simulation, the velocity profiles are used in the construction of the gradient. Using this operator, an error estimator is built everywhere in the computational mesh. If the error gap between two adjacent points is too important, the integration of new points can be performed in order to control the error difference between neighbor nodes. A metric is built from the error estimator at each point, which allows local deformations of the mesh. The anisotropic characteristic of the adaptation is brought by the integration of this metric of stretching coefficients acting along each of the axis. This behavior authorizes the presence of highly stretched cells in the computational mesh, which enables a fine capture of the frontiers while using a relatively low number of points in the mesh.

The adaptation is always linked to the signed-distance function, and, consequently, is dependent on the ε chosen. Several parameters, including the minimum size of cells in the mesh, are specified to guarantee a coherent number of cells in the transition zone around the borders, contributing to a reduction of the computational effort while assuring convergence.

SIMULATIONS

The procedure of the simulation used in ICI-tech is detailed in Fig.4. While the immersion procedure uses the two stages on top of the diagram, i.e. the definition of level-set functions through the immersion and the mesh adaptation, the numerical simulation includes the resolution of the Navier-Stokes (NS) equations everywhere in the domain. However, the level-set functions only provide the phases in the computational domain, and mixing laws have to be applied to define the fluid and materials properties at each point. After the resolution, velocity and pressure values are

obtained. As previously explained, the velocity is used along this the reconstructed geometries to build the recovery gradient operator used in the mesh adaptation procedure. All those steps are massively parallelized, enabling to simulate computationally expensive situation, e.g. offshore wind turbines.

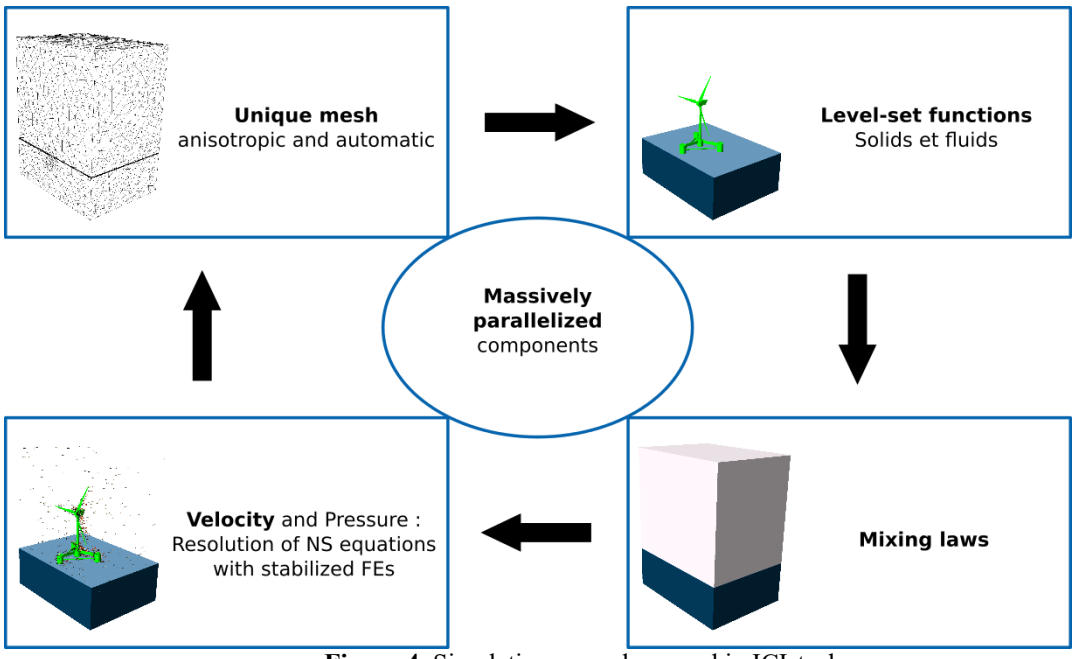

Figure 4. Simulation procedure used in ICI-tech.

The mixing laws are defined using an interpolation based on the material properties of the adjacent phases. For two materials of properties $\eta_1$and η_2, the material properties at current point is given by Eq.3. When more than 2 phases are defined in the computation, this interpolation is applied successively between the remaining phases. As an example, for a floating wind turbine simulation, the three phases defined are the air, the water and the wind turbine. The air water interface is defined first, and the results obtained are then interfaced with the floating structure.

$$\eta = \eta_1 H_\varepsilon(\alpha) + \eta_2\big(1 - H_\varepsilon(\alpha)\big).$$

Equation 3. Signed-distance function.

The NS equations are then solved using a Variational MultiScale (VMS) formulation, which corresponds to an implicit Large Eddy Simulation paradigm. The weak form of the NS equations shows in Eq.4, for $\boldsymbol{v} : [0,T] \to V$ fluid velocity and $p : [0,T] \to Q$ pressure. The contributions of all the variables are separated into coarse scale and unresolved scale, giving for velocity $\boldsymbol{v} = \boldsymbol{v}_h + \boldsymbol{v}'$ with $\boldsymbol{v}'$ small scales term. The functional spaces are separated too, giving for velocity spaces $V = V_h \oplus V'$. When replaced into Eq.4, the VMS formulation appears, as presented in Eq.5, along with stabilization terms τ and residuals $\boldsymbol{\mathcal{R}}$. Those terms are described in [13].

$$\forall(\mathbf{w}, q) \in V_0 \times Q,$$
$$\begin{cases} \rho(\partial_t \mathbf{v}, \mathbf{w}) + \rho(\mathbf{v} \cdot \nabla \mathbf{v}, \mathbf{w}) + 2\nu\varepsilon\mathbf{v} : \varepsilon\boldsymbol{w} - (p, \nabla \cdot \mathbf{w}) = (\mathbf{f}, \mathbf{w}) \\ (\nabla \cdot \mathbf{v}, q) = 0 \end{cases}$$

Equation 4. Weak form of Navier-Stokes equations.

$$\forall(\mathbf{w}_h, q_h) \in V_{0,h} \times Q_h,$$
$$\begin{cases} \rho(\partial_t \mathbf{v}_h, \mathbf{w}_h) + (\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h) - \sum_{K \in \mathcal{T}_h} (\tau_K \mathcal{R}_{\mathrm{M}}, \rho \mathbf{v}_h \nabla \mathbf{w}_h)_K \\ +2\mu\varepsilon(\mathbf{v}_h) : \varepsilon(\mathbf{w}_h) - (p_h, \nabla \cdot \mathbf{w}_h) + \sum_{K \in \mathcal{T}_h} (\tau_C \mathcal{R}_C, \nabla \cdot \mathbf{w}_h)_K = (\mathbf{f}, \mathbf{w}_h) \\ (\nabla \cdot \mathbf{v}_h, q_h) - \sum_{K \in \mathcal{T}_h} (\tau_K \mathcal{R}_{\mathrm{M}}, \nabla q_h)_K = 0 \end{cases}$$

Equation 5. Navier-Stokes VMS formulation.

The NS solver also requires the imposition of boundary conditions, which is handled in this work with Lagrange multipliers counterbalancing the movements and deformations observed in the domain.

RESULTS

RECONSTRUCTION OF FLOATING WIND TURBINES

A first test case focuses on the immersion of a single floating wind turbine, which was defined in [10]. This prototype is a *1/100* scaled reproduction of a *5MW* wind turbine, mounted on a Dutch Trifloater floating structure. The height of the full prototype is approximatively *1.5m*, and the blades are about *0.55m* long. A surface mesh composed of *74250* oriented triangles representing the full structure, drawn in Fig.5 is immersed into a computational domain of size *2x2.5x2*, which contains *117150* nodes and is sliced in Fig.6. The parallelization of the code was used, as *8* cores were used during the immersion. A precision in the order of the millimeter is reached, with a $\varepsilon = 10^{-3}$. As a matter of comparison, to get a similar precision with a regular mesh in a square surrounding the rotor, of dimensions *1.1x0.1x1.1*, around *120* millions of points would be needed. This corresponds to a precision in the order of the decimeter for a full scale machine, which is low to run computation but is enough for visualization only, as shown in Fig.7.

Figure 5. Immersed mesh.

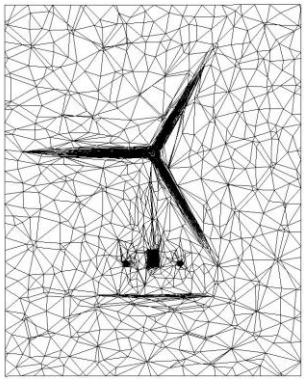

Figure 6. Adapted computational mesh.

Figure 7. Reconstructed wind turbine.

A hundred wind turbines have been randomly disposed in a *50m x 50m* computational domain in a second test case. This test case is only a demonstrator aiming to prove the scalability of the immersion procedure, when the the same precision of $\varepsilon = 10^{-3}$ is reached. While *8* cores were used for the reconstruction of one wind turbine, *480* cores were used for the immersion of a hundred wind turbines. The computational mesh features approximatively *18* millions points, while the immersed meshes were composed of about *7* millions points. The number of nodes in the mesh have increased by a factor of *150*, while the number of turbines have only be increased by a factor of *100*. However, the scalability remains satisfying, especially because the bigger computational domain can partly explain this rise. The results of the immersion are presented in Fig.8, and a zoom is proposed in Fig.9. The second figure shows the unique mesh used in the simulation, and how the adaptation procedure is used for the immersion of several turbines. The wind turbines are colored depending on the core used for the local immersion computations. This highlights the fact that, aside the computational resources, no development efforts are needed for the reconstruction of several wind turbines.

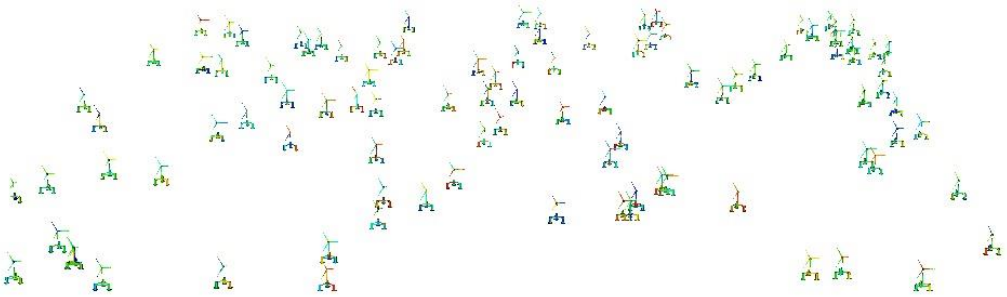

Figure 8. Immersed mesh.

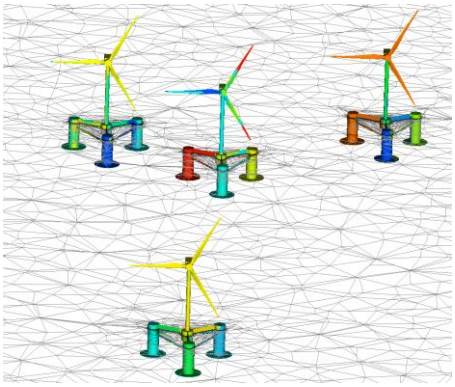

Figure 9. Immersed mesh.

These demonstrative cases showed the possibilities provided by mesh immersion coupled to anisotropic mesh adaptation. The parameter ε enables to control the targeted precision for the reconstruction, while enabling the local convergence of the NS resolution. The flexibility of this procedure is an advantage in the context of full CFD wind turbine simulation, as the geometries can be efficiently represented at a relatively low computational cost and the rotor rotation can be captured by the mesh adaptation procedure.

AERODYNAMIC RESULTS

Aerodynamic simulations have been launched on the rotor of the wind turbine studied in the Reconstruction section. The rigid rotor has been considered, and both fixed and prescribed rotation cases have been studied. The rotor, reconstructed with $\varepsilon = 10^{-4}$, is presented in Fig.10. This ε corresponds to a precision of *1/10* millimeters in the simulation, or of one centimeter for a full scale wind turbine. A constant and uniform ambient wind is considered, of velocity *1m/s*, which gives a Reynolds number at blade slice of approximatively *2300*. The low Reynolds number is due to a scaling of the wind turbine simulated based on a Froude similitude, without any low Reynolds profiles. Consequently, aerodynamic validation will not be performed using this prototype.

The computation mesh used *240000* nodes (cf. Fig.11), and the simulation was launched on 24 cores. Air flows have been studied around the blades when the rotor is fixed, as shown in Fig.12. The velocity profiles are investigated using the vectors and background color displayed. Even if the Reynolds number of the flow is not appropriated for wind turbine simulation, a recirculation phenomena and vortex can be observed behind the blade.

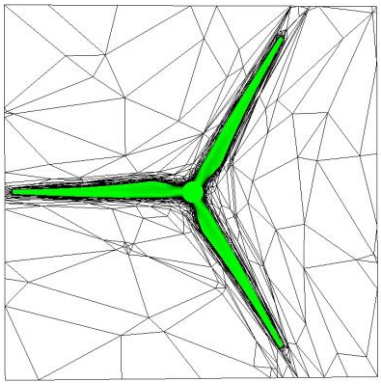

Figure 10. Reconstructed rotor.

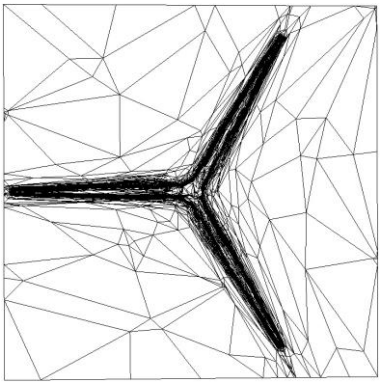

Figure 11. Adapted computational mesh.

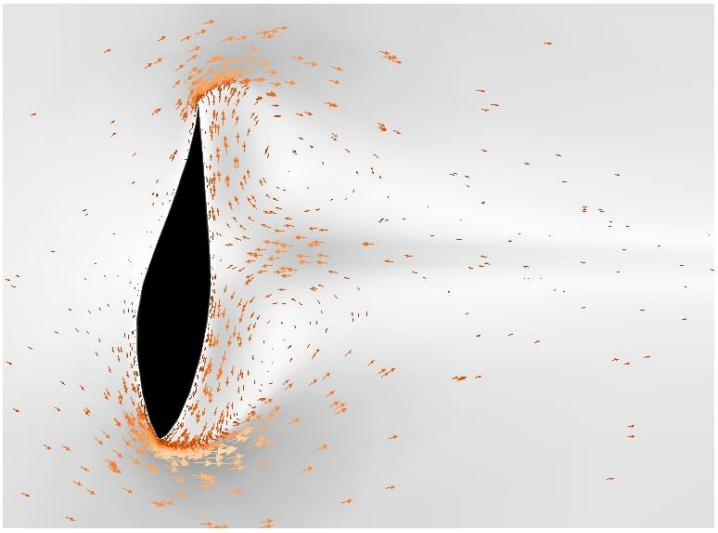

Figure 12. Air flow around a blade for a fixed rotor.

A prescribed rotation movement was then simulated, with the same precision as in the fixed rotor case, i.e. $\varepsilon = 10^{-4}$. A time step $\Delta t = 10^{-4}$s was chosen, and a rotation angle $\theta = 0.01°$ was applied at each increment. This corresponds to a rotation speed $\dot{\theta} = 16.7$ revolutions per minute (RPM), which corresponds approximatively to the highest velocity a wind turbine rotor can reach. The very short time step guarantees that blade displacements stay in the transitional and densely meshed area, which facilitates the mesh adaptation procedure and the convergence in the NS resolution. On the other side, long term simulations, in this case *36000* time steps, are required to simulate a full rotor revolution. Fig.13 presents the reconstructed rotor after about *800* increments, i.e. a rotation of *8°*. The adapted mesh is displayed in Fig.14, where the positioning of the nodes around the geometry can be noted. Air flows have been studied all along the computation, and are presented at the corresponding time increment in Fig.15. The evolution of the aerodynamics around the blade is generated by the downwards movement of the blade, and eddy detachments are observed.

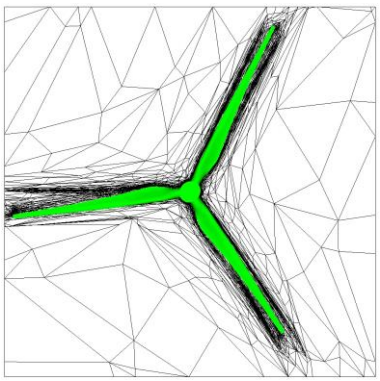

Figure 13. Reconstructed rotor.

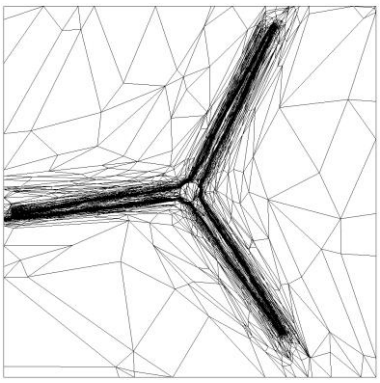

Figure 14. Adapted computational mesh.

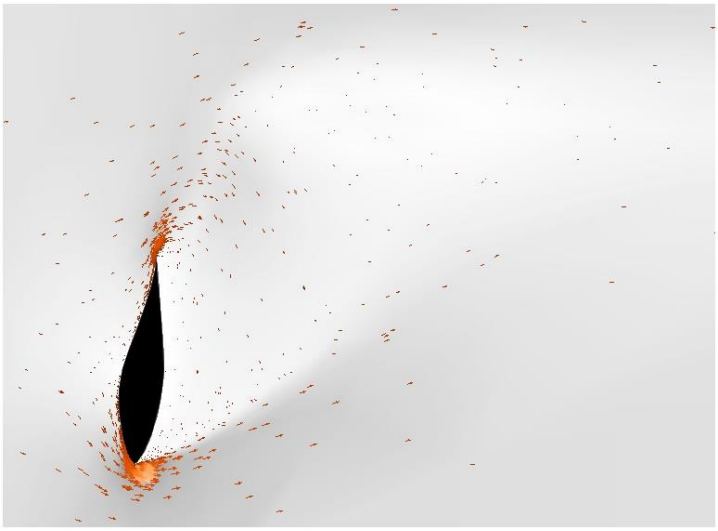

Figure 15. Air flow around a blade for a fixed rotor.

However, even if experiments have been performed on this scaled wind turbine in the wind flume of Luminy, see [10], the behavior of the wind turbine have been studied under wave only and wind/wave conditions. Moreover, ICI-tech has not been validated for aerodynamic effects. Consequently, this rotor only simulation needs to be seen as a demonstrator test case.

CONCLUSIONS

This paper intended to show how ICI-tech will handle floating wind turbine simulations, as the mesh immersion and automatic adaptation procedures have been presented. An aerodynamic demonstrator test case has been introduced, and the air flow around the blade has been studied.

Even if the test case results are interesting, a validation of the aerodynamic code will have to be performed. Moreover, while this paper focused only on the aerodynamic effects acting on the rotor, the hydrodynamic effects will have to be taken into account. Wave generation mechanisms are currently studied, and hydrodynamic validation will then be performed.

REFERENCES

[1] Hansen, Martin Otto Laver, et al. "State of the art in wind turbine aerodynamics and aeroelasticity." *Progress in aerospace sciences* 42.4 (2006): 285-330.

[2] Bazilevs, Y., et al. "Novel structural modeling and mesh moving techniques for advanced fluid–structure interaction simulation of wind turbines." *International Journal for Numerical Methods in Engineering* 102.3-4 (2015): 766-783.

[3] Sebastian, Thomas, and Matthew Lackner. "Offshore floating wind turbines-an aerodynamic perspective." *49th AIAA Aerospace Sciences Meeting, Orlando, Florida*. 2011.
[4] Quallen, Sean, and Tao Xing. "CFD simulation of a floating offshore wind turbine system using a variable-speed generator-torque controller." *Renewable Energy* 97 (2016): 230-242.
[5] Leble, Vladimir, and George Barakos. "Demonstration of a coupled floating offshore wind turbine analysis with high-fidelity methods." *Journal of Fluids and Structures* 62 (2016): 272-293.
[6] Wu, Chih-Hua Keni, and Vinh-Tan Nguyen. "Aerodynamic simulations of offshore floating wind turbine in platform-induced pitching motion." *Wind Energy* 20.5 (2017): 835-858.
[7] Tran, Thanh-Toan, and Dong-Hyun Kim. "The platform pitching motion of floating offshore wind turbine: a preliminary unsteady aerodynamic analysis." *Journal of Wind Engineering and Industrial Aerodynamics* 142 (2015): 65-81.
[8] Yan, J., et al. "Computational free-surface fluid–structure interaction with application to floating offshore wind turbines." Computers & Fluids 141 (2016): 155-174.
[9] Courbois, Adrien. Étude expérimentale du comportement dynamique d'une éolienne offshore flottante soumise à l'action conjuguée de la houle et du vent. Diss. Ecole Centrale de Nantes (ECN)(ECN)(ECN)(ECN), 2013.
[10] Lacaze, Jean-Baptiste, et al. "Small scale tests of floating wind turbines in the wind and wave flume of Luminy." 14th journées d'hydrodynamique, Val de Reuil, France, 18-20 Nov 2014. 2014.
[11] Sanderse, Benjamin, S. P. Pijl, and Barry Koren. "Review of computational fluid dynamics for wind turbine wake aerodynamics." Wind energy 14.7 (2011): 799-819.
[12] Coupez, Thierry, Luisa Silva, and Elie Hachem. "Implicit boundary and adaptive anisotropic meshing." *New Challenges in Grid Generation and Adaptivity for Scientific Computing*. Springer International Publishing, 2015. 1-18.
[13] Coupez, Thierry, and Elie Hachem. "Solution of high-Reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing." *Computer Methods in Applied Mechanics and Engineering* 267 (2013): 65-85.