Immersed boundary method for flow simulation around horizontal wind turbine

Aymen Jendoubi
Mechanical Engineering Department, Prince Mohammad Bin Fahd University, P.O. Box 1664
Al Khobar 31952 KSA
E-mail: ajendoubi@pmu.edu.sa

Abstract. Accurate simulation of wind turbine wakes is critical for the optimization of turbine efficiency and prediction of fatigue loads. These wakes are three-dimensional, complex, unsteady and can evolve in geometrically complex environments. Modeling these flows calls thus for high-quality numerical methods that are able to capture and transport thin vortical structures on an unstructured grid. In this work, an immersed boundary (IB) method for solving fluid flow problems of aligned wind turbines under uniform inflow is shown. A finite element method is used starting from a base mesh which does not represent exactly the blade of the wind turbine (non body-conforming mesh). At each time step, the base mesh is locally modified to provide a new mesh fitting the boundary of the rotor blades. The mesh is also locally improved using edge swapping to enhance the quality of the elements and ensure high accuracy of simulation. The methodology is assessed on two different test cases and validated with experimental results.

1. Introduction
Due to increasing environmental concern and oil price and concerns over limited fossil-fuel resources, more and more electricity is being generated from renewable sources. Wind energy is one of the most popular renewable energy resources all over the world. Wind turbine technology has gained great development over the last decades. The efficiency of the wind turbine blade determines the power performance of the wind turbine rotor.

Wind turbines that are most efficient had been designed by Cole Gustafson from Dakota State University. In his research, he showed that the horizontal axis wind turbine machines was more efficient than vertical axis one. Moreover, the blade span of horizontal wind turbines used was larger than vertical axis machines which limited the placement confined spaces.

Computational Fluid Dynamic (CFD) is a good method for investigating the aerodynamic performance of different types of blade shape [1]. The well-known CFD method to simulate the flow in an horizontal wind turbine is the body-fitted method where the mesh is built to fit as well as possible the geometry of the blade. Since the blade are in general moving or deforming, this requires a complete remeshing of the domain at each time step, which is a very difficult task in itself, and often to high computational cost and memory requirements. In order to avoid such problem, we investigate in this work the Immersed Boundary method (IB).

The IB method is often used in the numerical resolution of partial differential equations in a domain with moving obstacles. It has been studied in particular by: Peskin [2], Glowinski et al. [3], Ilinca et Hetu [4], Su et al. [5] and many others in recent years. In particular, this method is applicable for the fluid-structure interaction. Indeed, the problem to be solved is
extended from the fluid domain to the total fluid-solid domain, by extending the velocity field to the solid. A fluid problem is thus solved on the entire fluid-solid fictitious domain. In this way, the condition of rigidity of obstacles is no longer explicit and it should be added in the weak formulation. The calculation of the velocity field is done on the total domain, which has a simple and fixed geometry. The mesh is therefore fixed and must be calculated only once ([6]), which is not the case for the resolution of fluid flow problems with a mobile obstacle (rigid body).

The IB method was introduced by Peskin [2] who successfully applied it to simulate heart valves as well as many biomedical problems. Peskin considered that the complex geometry of the valve in the Cartesian mesh can be simulated by the generation of an external force field to mimic the immersed boundary. In the same context, Van Loon et al. [7] proposed an extended fictitious domain method with a local mesh matching algorithm to provide the necessary flexibility with respect to valve movement and deformation and to ensure the ability to impose the pressure exerted by the solid on the fluid.

Huang et al. [8] also adopted the fictitious domain method for the simulation of the interaction of a fluid with a deformable solid. In their formulation, they take into account both the dynamics of the fluid and the solid and precisely determine the free displacement of the flexible solid boundary. A Lagrangian mesh is used to discretize the solid domain and the solid motion equation is determined by the energy derivation method and is solved by an iterative method. The fluid-structure interaction is formulated by the direct forcing method as in [9], that is, by adding a force in the Navier-Stokes equations. This force is obtained on the Lagrangian mesh of the solid by the use of the equation of motion of the solid and it is diffused in all the field of the fluid by means of a function delta of Dirac.

Several other authors have adopted the IB method, among which Baaijens [10], Cottet et al. [11], Mittal et al. [12], Van Loon et al. [13], Lohner et al. [14], Ilinca et al. [4]. In this paper, we present an immersed boundary method [15] where a base mesh is locally modified in order to fit the boundary of the wind blade. We use this method to simulate numerically the wake of an horizontal wind turbine employing low Reynolds number airfoils, i.e. low wind speed profile such as in Saudi Arabia regions where the average wind speed is in range of 1 m/s to 5 m/s maximum.

2. Methodology
2.1. Governing equations
We consider the flow of an incompressible fluid. The governing equations in non dimensional form are:

\[
\begin{aligned}
\left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nabla p - \frac{2}{Re} \nabla \cdot (\dot{\gamma}(\mathbf{u})) &= \mathbf{f} \\
\nabla \cdot \mathbf{u} &= 0
\end{aligned}
\]

where \(Re\) denotes the Reynolds number, \(\mathbf{u}\) the velocity vector, \(p\) the pressure, \(\dot{\gamma}(\mathbf{u})\) the strain-rate tensor defined as \(\dot{\gamma}(\mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T)\) and \(\mathbf{f}\) a volumetric force vector (generally vanishing).

2.2. Finite element formulation
The Navier-Stokes equations are discretized using a second order \(O(h^2)\) Taylor-Hood \(P_2 - P_1\) element (see Brezzi and Fortin [16]). The Galerkin finite element formulation in the computational domain \(\Omega\) (possibly including rigid objects), with boundary \(\Gamma\) is given by:
\[\begin{aligned}
&\int_{\Omega} \left( \frac{3}{2\Delta t}\mathbf{u}^n \cdot \mathbf{w} + \frac{2}{Re} \hat{\gamma}(\mathbf{u}^n) : \hat{\gamma}(\mathbf{w}) - p\nabla \cdot \mathbf{w} + \left((2\mathbf{u}^{n-1} - \mathbf{u}^{n-2}) \cdot \nabla \mathbf{u}^n\right) \cdot \mathbf{w} \right) \, dv = \\
&\int_{\Gamma_1} f \cdot \mathbf{w} \, dv + \int_{\Gamma_1} t \cdot \mathbf{w} \, dS + \frac{1}{2\Delta t} \int_{\Omega} \left(4\mathbf{u}^{n-1} - \mathbf{u}^{n-2}\right) \cdot \mathbf{w} \, dv \\
&\int_{\Omega} q \nabla \cdot \mathbf{u}^n \, dv = 0
\end{aligned}\]

where symbols \( \mathbf{w} \) and \( q \) denote the appropriate weighting functions and \( t \) stands for a possible natural Neumann boundary of the form \( \mathbf{\sigma} \cdot \mathbf{n} = t \) on \( \Gamma_1 \), where \( \mathbf{\sigma} \) is the Cauchy stress tensor and \( \mathbf{n} \) the unit normal vector. For the time derivative, we use an implicit backward finite difference scheme of order 2 (BFD2) and thus, the global finite element approximation is therefore second order in both space and time.

### 2.3. Resolution method

A major difficulty for the numerical simulation of incompressible flows is that the velocity and the pressure are coupled by the incompressibility constraint. The fractional-step projection method of Chorin [17] and Temam [18] is the most frequently employed technique for the numerical solution of the primitive variable Navier-Stokes equations. This method is based on a rather peculiar time-discretization of the equations governing viscous incompressible flows, in which the viscosity and the incompressibility of the fluid are dealt within two separate steps. These methods proceed in a first step by calculating an intermediate velocity field \( \tilde{\mathbf{u}} \) using the momentum equation. This field does not check the incompressibility condition \( \nabla \cdot \tilde{\mathbf{u}} = 0 \), and it is therefore necessary to project it into a space where the vector are at zero divergence to obtain \( \mathbf{u}^{n+1} \). The algorithm of resolution can be described in 4 steps as follows:

- **Step 1**: viscous prediction
  \[
  \frac{\rho}{2\delta t} \left(3\tilde{\mathbf{u}}^{k+1} - 4\mathbf{u}^k + \mathbf{u}^{k-1}\right) - \nabla \cdot (\mu \hat{\gamma}(\tilde{\mathbf{u}}^{k+1})) + (2\mathbf{u}^k - \mathbf{u}^{k-1}) \cdot \nabla \tilde{\mathbf{u}}^{k+1} + \nabla p^k = f(\mathbf{t}^{k+1}),
  \]

- **Step 2**: projection
  \[
  -\Delta \phi^{k+1} = -\frac{3\rho}{2\delta t} \nabla \cdot \tilde{\mathbf{u}}^{k+1},
  \]

  where
  \[
  \nabla \phi^{k+1} \cdot \mathbf{n} \big|_{\Gamma_D} = 0 \quad \text{et} \quad \phi^{k+1} \big|_{\Gamma_N} = 0
  \]

- **Step 3**: pression correction
  \[
  p^{k+1} = p^k + \phi^{k+1},
  \]

- **Step 4**: update the velocity
  \[
  \mathbf{u}^{k+1} = \tilde{\mathbf{u}}^{k+1} - \frac{2\delta t}{3\rho} \nabla \phi^{k+1},
  \]

Various variants of this method have been proposed over the years ([19, 20, 21, 22, 23]). For an interesting overview, we also refer to Guermond and coauthors [24, 23].
2.4. Immersed boundary method

Structured meshes allow a simple and precise discretization of the equations, but they are not generally adapted to the treatment of industrial cases involving obstacles or interfaces of complex shapes. In order to improve the taking into account of complex interfaces and to ensure the accuracy of the results given by the fictitious domain method, we have developed a mesh cutting method. For any mesh not containing an obstacle beforehand, we give ourselves a function \( f \) defining any known geometry: a circle, a triangle, an ellipse, a rectangle for example. The principle is to define a signed distance function in the computation domain whose zero isovalue is the interface that one will seek to determine. This function is chosen negative within a subdomain (the solid) and positive everywhere else.

Consider the case of a function \( f \) whose zero isovalue defines a NACA profile. An enlargement of the mesh around the wing is presented in figure 1. By traversing all the edges of the mesh, one seeks with the help of a method of bisection an intersection of the edges of the mesh considered with the isovaleur zero of the function \( f \) thus given. Once an intersection point is found, we add a node to the mesh at this point, that is to say on the edge containing the point of intersection and we build a shell around the edge of the mesh. We repeat the last step on all the edges containing the points of intersection of the mesh with the function \( f \). The addition of nodes on the interface as well as the refinement of the edges are represented in figure 2 and the mesh thus constructed in this case is represented in figure 3. Note that the technique is the same for cutting other shapes, just give the function \( f \) defining the geometry of the obstacle we want to cut.

\[
\partial \Omega_r
\]

Figure 1: Close view to the mesh around the NACA.

\[
\partial \Omega_r
\]

Figure 2: Mesh with nodes added.
Despite the introduction of several criteria on the method of edge cutting, elements of poor quality can appear in the mesh. Improving the quality of the elements therefore means trying to get as close as possible to equilateral triangles or regular tetrahedra. The quality of a mesh (or sub-mesh) is defined as the minimum quality of its elements. To improve the quality of the elements after edge cutting method, we introduce the notion of edge swapping (see [25, 26]). More details about the immersed boundary used in this paper are given in a previous work [15].

2.5. Algorithm
The complete algorithm requires a (fixed) starting base mesh $M_{\text{base}}$ and can be described as follows.

(i) For a given time $t^n$:
   1.1 Knowing the function $f$ defining the geometry of the obstacle at time $t^n$, perform the cutting edge method of the base mesh $M_{\text{base}}$. We denote the new mesh $M_{t^n}$.
   1.2 Apply edge swapping and a new refinement iteration to improve $M_{t^n}$.
   1.3 Reinterpolate the velocity and pressure fields from previous time steps ($u_{n-1}$, $u_{n-2}$ and $p_{n-1}$) on the new mesh $M_{t^n}$.
   1.4 Apply the appropriate boundary conditions at time $t^n$. In particular, Dirichlet boundary conditions are imposed inside the object and on all nodes on its boundary, including the ones recently introduced.
   1.5 Solve the Navier-Stokes equations using the fractional–step projection method and compute the solution ($u_n$, $p_n$)

(ii) Go to next time step.

Note that, at each time step, the cutting edge method (see step 1.1) is performed on a base mesh $M_{\text{base}}$ adding a new nodes in the mesh $M_{t^n}$ and thus the previous solution must be interpolated on the new mesh at each time step. Interpolation is done in a classical manner. Using the coordinates of such a new node in mesh $M_{t^n}$, we determine which element of the old mesh contains it and then simply use the Lagrange basis functions to interpolate. The step 1.2 in the algorithm do not increase the size of the system to be solved numerically since there is no nodes to be inserted on the mesh but just used here to enhance the quality of the elements.

3. Numerical results
We present two numerical results related to horizontal axis wind turbines (HAWTs, as some people like to call them) because it represent the main goal of this work.
3.1. Flow around and airfoil NACA4418
Most HAWTs have blades designed as airfoils similar to aircraft wings. Thus, the first case
investigated to assess the proposed method is representative of the Tjaereborg wind turbine [27, 28]. It has a diameter \( D = 61 \) \( m \) and the rotor blade profiles are modeled using NACA4418 airfoils as in [29], with a chord length decreasing linearly from hub to tip. The wind speed \( U_\infty \) and the rotational speed are set so that the tip speed ratio \( TSR = \lambda = \frac{\Omega R}{U_\infty} = 7.07 \). Output boundary is treated with simple out ow conditions of zero streamwise gradients and constant pressure. The aim of this investigation is to validate the fact that the numerical IB method proposed for unstructured grids perform good results compared to those presented in [29].

3.1.1. Computational setup

The geometry of the problem is presented in Figure 4. To apply the IB method proposed in this paper we need to define the function \( \phi \) as discussed in section 2.4. The expression of the NACA profile is:

\[
(y)^2 = \left( \frac{0.15c}{0.2} \right)^2 \left[ 0.2969 \sqrt{\frac{x-1/3}{c}} - 0.1260 \left( \frac{x-1/3}{c} \right) - 0.3537 \left( \frac{x-1/3}{c} \right)^2 + 0.2843 \left( \frac{x-1/3}{c} \right)^3 - 0.1015 \left( \frac{x-1/3}{c} \right)^4 \right]^2 \quad (5)
\]

which is represented in Fig 4. Thus, its boundary is given by the level set function:

\[
\phi(x, y) = (y)^2 - \left( \frac{0.15c}{0.2} \right)^2 \left[ 0.2969 \sqrt{\frac{x-1/3}{c}} - 0.1260 \left( \frac{x-1/3}{c} \right) - 0.3537 \left( \frac{x-1/3}{c} \right)^2 + 0.2843 \left( \frac{x-1/3}{c} \right)^3 - 0.1015 \left( \frac{x-1/3}{c} \right)^4 \right]^2 = 0
\]

The initial (non body-fitted) mesh is presented in Figure 5 and contains 86 580 nodes, a number favorably comparable to the meshes used in [30] (120 000 nodes) and [31] (more than 200 000 nodes).

Applying the IB method, we use the function \( \phi(x, y) \) to cut the mesh and the result is given in figure 6.
3.1. Results

Figure 7 presents the vorticity field generated along the blades. These helical vortical structures are generated at the rotor tip and hub where the circulation gradient is the highest. Each helical structure is subject to growing instabilities and further transition which clearly shown in the construction of a von Kármán vortex street (see figure 8). This flow visualization also demonstrates the ability of the proposed IB method to capture large vortex over a long distance unless turbulence model was not used in this case since the Reynolds number $Re$ calculated is $Re = 1100$ where $Re = \frac{\rho U c}{\mu}$.

3.2. Four-Bladed Concept Wind Turbine

The second case investigated in this paper is a four-Bladed concept wind turbine. In general, HAWT usually have two or three rotor blades. A turbine with two rotor is faster and cheaper, but it flickers more than the rotor with three blades and is less efficient. Three blade rotors operate more smoothly and are therefore less disturbing. For this study, we choose to study numerically a four-Bladed concept wind turbine which is a new design studied theoretically in the following section.

3.2.1. Theoretical modeling

Theoretical analysis is conducted for the calculus of the theoretical power generation. In fact, wind power depends on the velocity and the mass flow rate

$$P = \frac{1}{2} m v^2 = \frac{1}{2} \rho A v^3$$
Taking in consideration the turbine Power coefficient, power in the wind is calculated using the following equation:

$$ P = \frac{1}{2} \rho A v^3 C_p $$

Where:

- $P$: Power in watts
- $\rho$: Air density At sea level air density is approximately $1.2kg/m^3$
- $A$: Turbine area in $m^2$, which can be calculated from the length of turbine blades. In this study, the turbine high is $0.9m$ and width is $1.25m$. Therefore, area is $A = 1.125m^2$
$v^3$: wind speed, which is the velocity of the wind in m/s. 
$C_p$: Power coefficient, usually varies according to wind turbine design, ranging between 0.05 and 0.45. In this case, we choose $C_p = 0.2836$ based on the selected angle $160^\circ$ ([32]).

To verify this theoretical results a numerical simulation based on IB method was carried out under the following computational setup.

### 3.2.2. Computational setup

The four-bladed design of a HAWT rotor blade can be expressed in polar coordinates (at time $t = 0$) by:

$$
\begin{align*}
  z^0(\theta) &= (\alpha + \beta \cos(4\theta)) \cos(\theta), \\
  y^0(\theta) &= (\alpha + \beta \cos(4\theta)) \sin(\theta)
\end{align*}
$$

where $\alpha = \frac{1}{5}$ and $\beta = \frac{1}{10}$. In cartesian coordinates, this corresponds to

$$
(y^0)^2(z^0)^2 - \frac{5}{4}((y^0)^2 + (z^0)^2)^2 \left(0.3 - \sqrt{(y^0)^2 + (z^0)^2}\right) = 0 \quad (6)
$$

As illustrated in Figure 9, the object is rotating clockwise around the $x$-axis with angular velocity $\omega$. At a given time $t$, the boundary of the four-bladed concept Wind Turbine is given by the level set function:

$$
\phi(x, y, z, t) = (y^0)^2(z^0)^2 - \frac{5}{4}((y^0)^2 + (z^0)^2)^2 \left(0.3 - \sqrt{(y^0)^2 + (z^0)^2}\right) = 0
$$

where

$$
\begin{align*}
  z^0 &= z \cos(\omega t) - y \sin(\omega t) \\
  y^0 &= z \sin(\omega t) + y \cos(\omega t)
\end{align*}
$$

This last expression is a counterclockwise rotation of the point $(z, y)$ to put it back in its original position. Eq. (6) can then be used to determine the boundary of the geometry. From (7), one easily gets

$$
\begin{align*}
  z(t) &= z^0 \cos(\omega t) + y^0 \sin(\omega t) \\
  y(t) &= -z^0 \sin(\omega t) + y^0 \cos(\omega t)
\end{align*}
$$

which gives the position of a point on the rotor blade at any time to apply adequately the boundary conditions.
This four-bladed turbine is rotating inside a cylinder of diameter $D = 1$. Computations were carried out for $Re = 50$ and $St = 10$. The IB method was applied on an unstructured tetrahedral mesh having 115,372 elements (20,038 nodes and 136,629 edges). Cross section of the mesh on a plane passing through the rotating blades is presented in Figure 10. Note that there is not regular triangular mesh but merely the intersections of three-dimensional meshes with the plane. The mesh for the IB method is illustrated after edge division and is slightly more refined in the region where we need to rotate the rotor. Note also that all nodes inside the moving rotor blade will be eliminated from the linear systems, thus reducing the computational cost.

3.2.3. Results  The velocity profile shown in figure 11. Isolines of the velocity component in the flow direction is presented in figure 12 and a close view of the velocity vecteurs is also shown in figure 13.

Qualitatively, The helical vortical structure that is usually observed experimentally downstream the rotor is clearly shown in figure 13. Quantitatively, the numerical results obtained was used to calculate the power gained and compared to those obtained experimentally as it is shown in table 1.

It is shown this table that there is a gap between the theoretical and experimental output power, where a several factors have affected clearly on the actual performance, these factors are due to external factors, lack of resources, process, geometrically, or due to human error. Also,
Table 1: Four-Bladed Concept Wind Turbine: comparison of the numerical results at different Wind speed values.

| Wind speed (m/s) | Theoretical power (W) | Numerical power (W) | Experimental power (W) |
|------------------|-----------------------|---------------------|------------------------|
| 2.0              | 2.0                   | 1.92                | 1.8                    |
| 3.0              | 5.0                   | 4.25                | 4.22                   |
| 4.0              | 12.0                  | 11.12               | 11.03                  |
it is shown that the numerical simulation of such a problem present results comparable to those obtained experimentally. The gap between the results is due the mesh considered, more accurate results can be obtained for more fine meshes.

4. Conclusions
In the present investigation, we have presented an immersed boundary method where a base mesh is locally modified in order to fit the boundary of the Wind Turbine. Thanks to the use of unstructured grids and the proposed method, such complex geometries can be handled and solved using a fractional-step projection scheme. The obtained CFD results are compared with the experiment. The results show that the method works remarkably well in various situations of Horizontal wind Turbine. These simulations can be further exploited to get more insights into wind turbine wakes and their interactions with complex environments and to validate design tools based on simpler models.

Acknowledgments
The author would like to thank the Mechanical Engineering Department at Prince Mohammad Bin Fahd University, Al Khobar City, Kingdom of Saudi Arabia for all supports into pursuing this work.

References
[1] Hsiao F B, Bai C J and Chong W T 2013 Energies 6 2784 – 2803
[2] Peskin C 2002 ACTA Numerica 11 479–517
[3] Glowinski R, Pan T W, Hesla T, Joseph D and Periaux J 2000 J.Comput.Phys. 169–363
[4] Ilinca F and Hetu J F 2011 International Journal for Numerical Methods in Fluids 65 856–875
[5] Su S W, Lai M C and Lin C A 2007 Computers and Fluids 36 313–324
[6] Valette R, Bruchon J, Digonnet H, Laure P, Leboeuf M, Silva L, Vergnes B and Coupez T 2007 Mécanique et Industries 8 251–258
[7] Van Loon P, Anderson P and Van de Vosse F 2006 Journal of Computational Physics 217 806–823
[8] Huang W X and Sung H J 2009 Computer Methods in Applied Mechanics and Engineering 198 2650–2661
[9] Noor D, Chern M J and Horng T L 2000 Computational Mechanics 169–363
[10] Baaijens F P, Selen S H A, Baaijens H P W, Peters G W M and Meijer H E H 1997 Journal of Non-Newtonian Fluid Mechanics 68 173 – 203 ISSN 0377-0257 papers presented at the Polymer Melt Rheology Conference
[11] Cottet G H and Maitre E 2004 C. R. Acad. Sci. Paris 338 581–586
[12] Mittal R and Iaccarino G 2005 Annual Review of Fluid Mechanics 37 239–261
[13] Van Loon R, Anderson P, van de Vosse F and Sherwin S 2007 Comp. & Struct. 85 833–843
[14] Lohner R, Appanaboyina S and Cebral J 2008 International Journal for Numerical Methods in Fluids 57 13–30
[15] Jendoubi A, Yakoubi D, Fortin A and Tibirna C 2014 International Journal Numerical Methods in Fluids 75 63–80 ISSN 1097-0333 URL http://dx.doi.org/10.1002/fld.3884
[16] Brezzi F and Fortin M 1991 Mixed and Hybrid Finite Element Methods (Springer Series in Computational Mathematics vol 15) (New York: Springer-Verlag)
[17] Chorin A 1968 Math. Comp. 22 745–762
[18] Temann R 1968 Bulletin de la Societ Mathematique de France. 96 115–152
[19] Bell J, Colella P and Glaz H 1989 J. of Comp. Physics. 85 257 – 283
[20] Rannacher R 1992 On Chorin’s projection method for the incompressible Navier-Stokes equations The Navier-Stokes Equations II Theory and Numerical Methods (Lecture Notes in Mathematics vol 1530) ed Heywood J, Masuda K, Rautmann R and Solonnikov V (Springer Berlin Heidelberg) pp 167–183 ISBN 978-3-540-56261-0
[21] Shen J 1996 Math. Comp. 65 1039–1065 ISSN 0025-5718
[22] Guermond J L 1997 C. R. Acad. Sci. Paris Sér. I Math. 325 1329–1332 ISSN 0764-4442
[23] Guermond J L and Quartapelle L 1998 Internat. J. Numer. Methods Fluids 26 1039–1053 ISSN 0271-2091
[24] Guermond J L, Minev P and Shen J 2006 Comput. Methods Appl. Mech. Engrg. 195 6011–6045 ISSN 0045-7825
[25] Bois R, Fortin M and Fortin A 2012 Computer Methods in Applied Mechanics and Engineering 209-212 12–27
[26] Bois R, Fortin M, Fortin A and Cout A 2012 European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique 21 72–91
[27] Mikkelsen R 2003 Actuator disc methods applied to wind turbines Ph.D. thesis Technical University of Denmark
[28] Troldborg N 2008 Actuator Line Modeling of Wind Turbine Wakes Ph.D. thesis DTU-MEK Denmark
[29] Peet Y, Fischer P, Conzelmann G and Kotamarthi V 2013 Actuator line aerodynamics model with spectral elements 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
[30] Cori J F 2011 Analyse et caractérisation d’interactions fluide-structure instationnaires en grands déplacements. Ph.D. thesis Département de génie mécanique, cole Polytechnique de Montréal
[31] Kinsey T and Dumas G 2008 AIAA Journal 46 1318–30
[32] Nader N and Jendoubi A 2018 Study of a vertical axis wind turbine for low speed regions in saudi arabia Proceedings of the 5th International Conference of Fluid Flow, Heat and Mass Transfer (FFHMT’18)