

EULERIAN PRESSURE-VELOCITY/ LAGRANGIAN VORTICITY-VELOCITY COUPLING APPLIED TO WAKE AND FORCES CALCULATION FOR BIOFOULED TIDAL TURBINES

Ilan Robin^{1,*}, Anne-Claire Bennis¹ and Jean-Claude Dauvin¹

¹ Morphodynamique Continentale et Côtière (M2C, UMR CNRS 6143), Université de Caen
Normandie (Campus 1), 14000 Caen cedex, FRANCE.

*ilan.robin@unicaen.fr

Key words: Marine Renewable Energy, Tidal Turbine, Eulerian-Lagrangian Coupling

Abstract. Marine tidal turbines are subject to the environment in which they are deployed. In the natural environment, they are gradually colonized by sessile species. These fouling organisms modify the flow around the blades and in the wake of the tidal turbine. Unfortunately, they also complicate the numerical study of such tidal turbines by preventing the use of usual methods such as the Blade Element Method or the Lifting Line Theory. In this context, we propose to use an alternative solution, which combines an Eulerian code to study the near field with a Lagrangian code for the wake. After a short presentation of each code, the coupling method is detailed, and applied to the case of a tidal turbine with its own vertical axis. First results are shown and compared to a full Eulerian simulation. Although the data transmission between both codes works well, discrepancies were found due to abnormal increase of energy in the Lagrangian area. A solution is proposed and explained.

1 INTRODUCTION

Tidal turbine energy represents a potential of 120 GW (around 150 TWh per year). Even if it seems relatively low compared to other renewable energies, it is an important source for the energy mix that could cover one third of the European consumption. The potential is always calculated taking current technologies into account. That number could highly be increased by a technology breakout just like the oil peak with shale gas. Until then, tidal turbine installations require strong currents and a specific environment. The flow around a tidal turbine is highly turbulent, strong and complex (around 3m.s^{-1}), it is very difficult to reproduce realistic conditions in a laboratory. Hence, a numerical approach is a good alternative. Depending on the kind of turbine, vertical or horizontal axis, and its rotation speed, the wake of one blade can impact the others through swirling releases. These interactions between the flow and the blades generate important variations on the blade load. To answer these modeling challenges, numbers of methods have been developed to model rotors. The Actuator Disk Method (ADM) is efficient to evaluate a turbine's performances ([1]) but does not take into account any of its geometry apart from the turbine's radius. Other methods such as the Blade Element Momentum Method (BEMM) or the Lifting Line Theory Method (LLTM)([2]) are more precise but need strong hypothesis to be applied such as having an attached flow on the blades (Kutta hypothesis) or a low variation of thickness along them. But when it comes to tidal turbines, the biological factors of the environment should be taken into account. Only few weeks after their immersion in sea water, the blades are colonized by fouling animal organisms whose

size can be from some millimetres to some centimetres ([3],[4]). This phenomenon is called biofouling and causes complexities such as dynamic stall, separated flow and turbulent wake ([5]). To simulate the flow around biofouled turbines, the best option is to use classic full Navier-Stokes methods ([6]). In order to compute the forces on the structure as well as the wake, results of Reynolds-averaged Navier-Stokes (RANS) and Large Eddy Simulation (LES) have been compared in [7]. Even though RANS gives good results for constrain estimation, it often dissipates vortices, which leads rapidly to an early fading of the wake. LES has therefore been chosen for this work.

The most common way to apply the Navier-Stokes equations is to use a mesh and let the flow develop on it, using numerical schemes. This is an Eulerian approach. Eulerian specifications have been used for a long time, and a lot of different codes with good prediction results exist. However, the computational cost is high, due to the numerous grid points which are required for efficient computation. For example, mesh sensitivity studies show a convergence between 7 millions and 12 millions points for wind and tidal turbines. Some methods exist to limit the number of points such as adaptive grid methods or overset meshes, but that still needs heavy computations.

An alternative solution is to use Lagrangian methods. They are mesh free and describe the flow motion moving fluid particles. They need less computing time, but they have been ignored for a long time for the benefit of Eulerian methods. So, they are, up to now, not efficient to solve fluid-structure interactions or require an important amount of particles in the near-blade field. Therefore, we suggest the use of a hybrid Eulerian/Lagrangian method in this paper. This kind of approach has already been used like in [8] and [9]. Both codes employ LES to solve Navier-Stokes equations but under different forms. This work is a presentation of the coupling method developed in the M2C Caen laboratory. After a short introduction, Section 2 describes all the methodology and methods used to compute the results. The first results are given and discussed in Section 3. Section 4 is the conclusion of the paper.

2 METHODOLOGY

2.1 Algorithms and equations

This study uses two coupled codes to solve fluid motion and forces on the blades. Each code solves a different domain depending of its strengths and weaknesses: OpenFoam (Eulerian) is used for the close rotor field and Dorothy (Lagrangian) is used to compute the far field.

2.1.1 The Eulerian Navier-Stokes code

OpenFoam is an open source code, which solves the three-dimensional Eulerian Navier-Stokes equations. It contains numerous solvers dedicated to different purposes. In this case, the `PimpleFoam` solver is used to solve Eq.(1) and Eq.(2). It is a transient solver for incompressible, turbulent flow of Newtonian fluids with a moving mesh.

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = \frac{1}{\rho} \nabla p + \frac{1}{\rho} \mathbf{f} + \nu \Delta \mathbf{u}, \quad (2)$$

where \mathbf{u} is the fluid velocity vector, t is the time, ρ is the fluid density, p is the pressure. \mathbf{f} represents the volumetric forces and ν is the kinematic viscosity.

Boundary conditions are given here after:

$$\mathbf{u}|_{\delta\Omega_1} = U_\infty \mathbf{x}_1, \quad (3)$$

$$p|_{\delta\Omega_2} = p(u_{Dorothy}), \quad (4)$$

$$\nabla \mathbf{u} \cdot \mathbf{n}|_{\delta\Omega_{3,4,5,6}} = 0, \quad (5)$$

$$\mathbf{u}|_{rotor} = 0, \quad (6)$$

where U_∞ is the inlet flow velocity and \mathbf{x}_1 is the unit vector in the x-direction. Eq.(3) is an inlet ($\delta\Omega_1$) velocity condition. Eq.(4) is a pressure outlet ($\delta\Omega_2$) condition that depends of Dorothy velocity ($u_{Dorothy}$) results (computed thanks to `PimpleFoam` algorithm). Eq.(5) is the slip conditions on the lateral limits ($\delta\Omega_{3,4,5,6}$) of the calculation domain. At least, Eq.(6) is a non-slip condition applied on the surface of the turbine named "rotor".

`PimpleFoam` uses a coupled pressure/velocity algorithm. The turbulence is solved using a Smagorinsky's scheme of LES (Large Eddy Simulation) model. This choice was guided due to the good capacity of the turbulence model for simulating the generation and transportation of vortices allowing to get inlet conditions for the second code (Dorothy). The convergence criterium is set to 10^{-6} . The code is parallelized in order to attribute around 15 000 nodes for each processor, which is defined as a good optimum in the OpenFoam documentation [10].

`sixDoFRigidBodyMotion` ([10]) is a `PimpleFoam` module designed to compute the fluid-structure interactions and mechanical forces acting on the structure. It is compatible with the AMI (Arbitrary Mesh Interface) allowing to set up a moving mesh in addition to a stationary mesh. In this case, the rotation speed is chosen to fix the Tip Speed Ratio (λ) defined as:

$$\lambda = \frac{|\Omega_x| R}{U_\infty} \quad (7)$$

where Ω_x is the angular velocity, R the radius of the tidal turbine.

`sixDoFRigidBodyMotion` also allows to define a rotation speed depending of the forces applied on the tidal turbine blades (flow-induced rotation).

The computation is initialized with OpenFoam to let the flow develop itself and let the first vortices reach the position from which vortex particles are generated ($\delta\Omega_E$) for transmission to the Lagrangian code. The computation respects the CFL condition which ensures the numerical stability of the code:

$$CFL = U_\infty \cdot \left(\frac{\Delta T_O}{\Delta x} \right) < 0.5, \quad (8)$$

where ΔT_O is the computational time step in OpenFoam and Δx is the length of the smallest cell in the OpenFoam domain.

2.1.2 The Lagrangian vortex code

Dorothy is an unsteady Lagrangian vortex code developed by the French Ifremer and University Le Havre Normandy LOMC laboratories. As OpenFoam, it solves the Navier-Stokes equations but uses a different formulation:

$$\nabla \cdot \mathbf{u} = 0, \quad (9)$$

$$\frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} + \nu \Delta \boldsymbol{\omega}, \quad (10)$$

$$\mathbf{u}|_{\Omega} = U_{\infty}, \quad (11)$$

where $\boldsymbol{\omega}$ is the vorticity. Eq.(11) shows that all particles in Dorothy moved by U_{∞} in the entire computation domain (Ω), regardless of their position in the wake.

Boundary conditions are:

$$\boldsymbol{\omega}|_{\delta\Omega_E} = \boldsymbol{\omega}_{OF}, \quad (12)$$

$$\boldsymbol{\omega}|_{\delta\Omega_{out}} = 0, \quad (13)$$

where $\boldsymbol{\omega}_{OF}$ is the vorticity field from OpenFoam at the boundary $\delta\Omega_E$ and $\delta\Omega_{out}$ at the end of the computational domain.

The vorticity field is discretized with a Lagrangian transportation model. The Biot and Savart relation [11] gives the particle velocity generated by the particle interactions:

$$\mathbf{u}_{\omega,i} = \sum_j \mathbf{K}(\mathbf{X}_i - \mathbf{X}_j) \wedge \boldsymbol{\Omega}_j \quad (14)$$

$$\boldsymbol{\Omega}_i = \boldsymbol{\omega}_i V_i \quad (15)$$

where \mathbf{K} is a corrected Green kernel, $\mathbf{u}_{\omega,i}$ is the velocity of the i-particle, $\boldsymbol{\Omega}_j$ is the vortex intensity vector and \mathbf{X}_i and \mathbf{X}_j are respectively the position of the i and the j particles. V_i is the volume of the i-particle.

The velocity of each particle is given by:

$$\mathbf{U}_i = \mathbf{u}|_{\Omega} + \mathbf{u}_{\omega,i} - \mathbf{u}|_{\delta\Omega_E} \quad (16)$$

The velocity of each particle is therefore dependent of every particle in the computational domain. The main interest of Dorothy lies in its capacity to use the vortex particles as mesh and the user can delete low vorticity particles in the studied domain. A remeshing technique is used at given time steps to limit the number of particles. A more detailed description of the remeshing method is available in [11]. The numerical scheme for the time integration is a second order Runge-Kutta applied to both position and vorticity of particles. The viscous diffusion is modeled by a Particle Strength Exchange method (PSE) around the Gaussian kernel. The turbulence is computed thanks to the Mansour model [12].

Otherwise a tree code method is implemented in Dorothy ([11]) to reduce the computational time. Instead of taking each particle into account to solve the vorticity for the i particle, several zones are defined where equivalent particles are generated using all the local zone particle properties. The global particle computation concerns the equivalent particles.

2.1.3 Coupling strategy

An offline coupling between OpenFoam and Dorothy was implemented. Both models exchange, at given coupling time steps (ΔT_c), in two ways (see Fig.1): 1) OpenFoam sends to Dorothy the update vorticity field and 2) Dorothy sends to OpenFoam the updated velocity field.

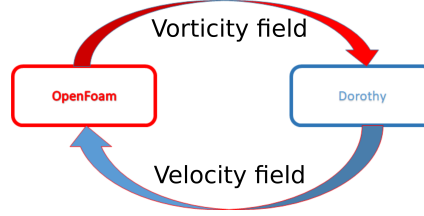


Figure 1: Scheme of the 2-ways coupling between OpenFoam and Dorothy

The computational domain is split into two different parts solved by each code and a joint part (buffer zone) where the flow characteristic is computed by both codes. The distribution along the x-axis is as follows:

- OpenFoam domain : $X \in [X_0; X_2]$, with $X_0 = -1.5\text{m}$ and $X_2 = 2.1\text{m}$ (see Fig.2, red rectangle).
- Dorothy domain : $X \in [X_1; X_3]$, with $X_1 = 0.5\text{m}$ and $X_3 = 25\text{m}$ (see Fig.2, blue rectangle).
- Buffer zone : $X \in [X_1; X_2]$ (see Fig.2, black rectangle).

In future works, the buffer zone may evolve into cylinders around each blade.

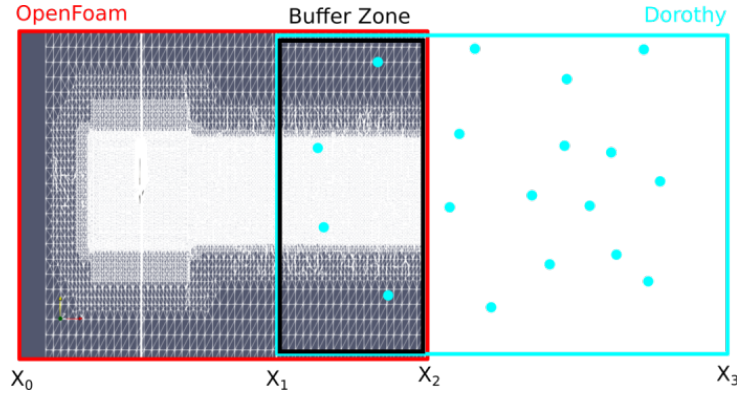


Figure 2: Bi-dimensional view of the computational domain, with the OpenFoam and Dorothy domains in red and blue rectangles, respectively, and the buffer zone in black rectangle.

During the iterative process, OpenFoam was run synchronously with Dorothy. The coupling algorithm is:

- Step 1: OpenFoam running until it has the computation of the flow characteristics at the data extraction position.
- Step 2: Computation of the vorticity from the velocity computed by OpenFoam.

Step 3: Dorothy running. New vortex particle generation and computation of the fluid vorticity.
 Step 4: Computation of the velocity field from Dorothy vorticity.
 Step 5: Injection of the velocity field from Dorothy in OpenFoam via boundary conditions, and go to Step 1.

Exchanges occur at each coupling time step ($\Delta T_c = 0.01s$). The OpenFoam vorticity field is computed using the curl function applied to the velocity:

$$\boldsymbol{\omega} = \text{curl}(\mathbf{u}) = \nabla \times \mathbf{u} \quad (17)$$

The vorticity, taken at X_1 , is injected as particles in Dorothy at the same position as shown on Fig.3. Every ΔT_c , new particles are generated this way. These particles describe the wake and go through a buffer zone where the two codes are running.

The velocity field at X_2 is computed (from Dorothy) using the boundary conditions and the inverse curl function (Fig.3). Then, this velocity field is interpolated on the OpenFoam mesh and injected in OpenFoam as an outlet velocity condition. The nearest cell method was used to simplify calculations but an interpolation impact study should be carried to compare the interpolation method consequences on the wake development. The coupling algorithm forces PimpleFoam to iterate at fixed time until it reaches a stabilization between pressure and velocity.

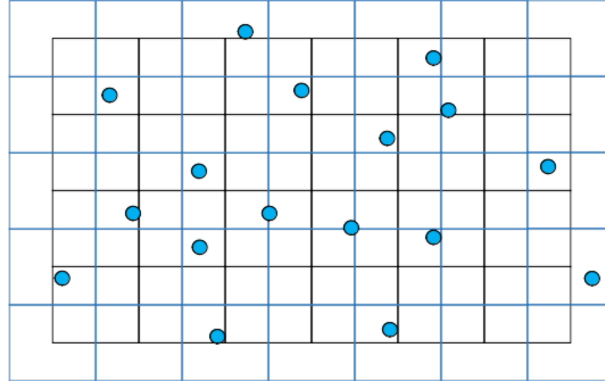


Figure 3: Dorothy provisory mesh (blue grid), OpenFoam mesh (black grid) and Dorothy particles (blue circles) at X_2 position

Note that the new generated particles in Dorothy are added to the existing ones at each ΔT_c . That can lead to a high amount of particles inducing a rapid increase of computation time. So, Dorothy needs to remesh the wake at each coupling time step. The remeshing algorithm had already been implemented and is more described in [11].

2.2 Test case: a horizontal axis tidal turbine

The horizontal axis tidal turbine developed by Ifremer and LOMC is used ([13]). The blade profiles are NACA63418. The rotor and the hub radius are $R = 350$ mm and $R_h = 46$ mm, respectively. The turbine geometry does not include a hub to reduce computational time (see Fig. 4).

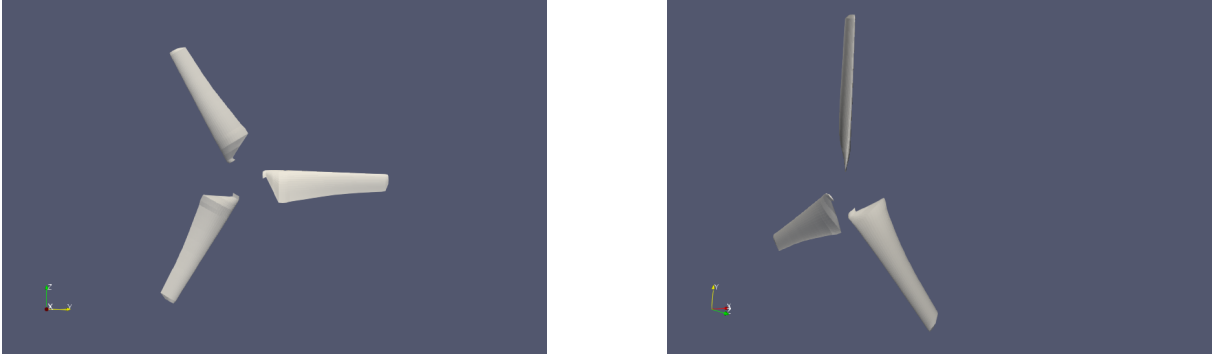


Figure 4: Three-dimensional geometry of the turbine without hub: front (left panel) and sideways (right panel) views.

As explained in section 2.1.3, the coupling strategy implies that the computational domains, chosen to limit the side effects, have different numerical characteristics. The OpenFoam domain is 5 rotor diameters long and 4 diameters large while Dorothy extends the domain with the same width but increasing its length to 20 diameters.

The OpenFoam domain is meshed thanks to the module snappyHexMesh ([10]) that allows to generate a refined mesh. It is very useful for complex geometry like a tidal turbine, especially with twist. The mesh includes a static and a rotating part linked by an Arbitrary Mesh Interface (AMI) ([7]). The rotating part is refined to preserve the quality of the evaluation of the forces and the generation of the vortices. After a mesh convergence study, the mesh described hereafter has been chosen : 9 051 299 points, 7 948 888 cells with a size of 7.10^{-4} m^3 and 3.10^{-10} m^3 for the biggest and the smallest cells, respectively.

The mesh around the blades is shown in Fig.5.

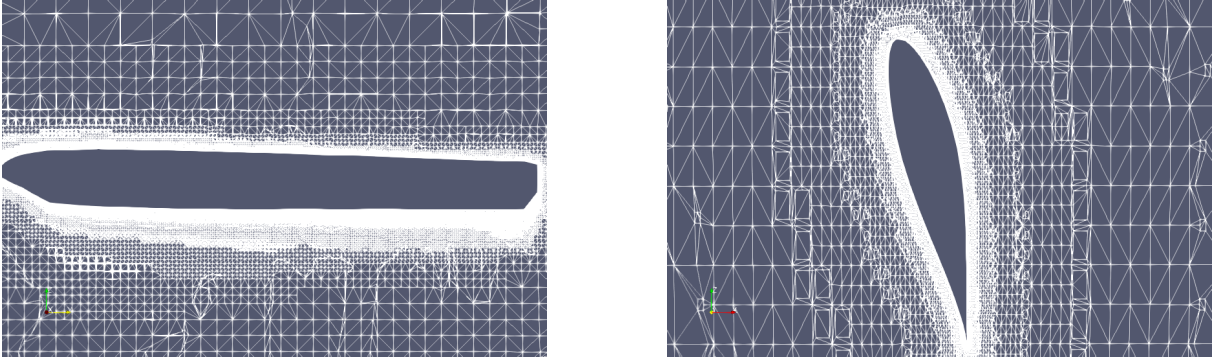


Figure 5: Refined mesh around one blade: face (left panel) and side (right panel) views. Irregularities are due to the cutting plan.

3 RESULTS AND DISCUSSION

Using the coupling method, it was possible to transfer the flow characteristics from OpenFoam to Dorothy as shown on Fig.6. Only the particles holding the strongest vorticity are kept. The particles are arranged like a cylinder behind the turbine which is due to the helicoidal wake generated by its rotation. Even though the slip condition applied to the edges of the domain of OpenFoam should prevent this, some particles remain on the walls in Dorothy. To avoid this phenomenon, several solutions are being considered: either removing the very particles with little energy as soon as they are generated, or filtering them with respect to their position in space. If the particles are too far from the centre, they are erased. However, the shape of the tidal turbine wake is well preserved.

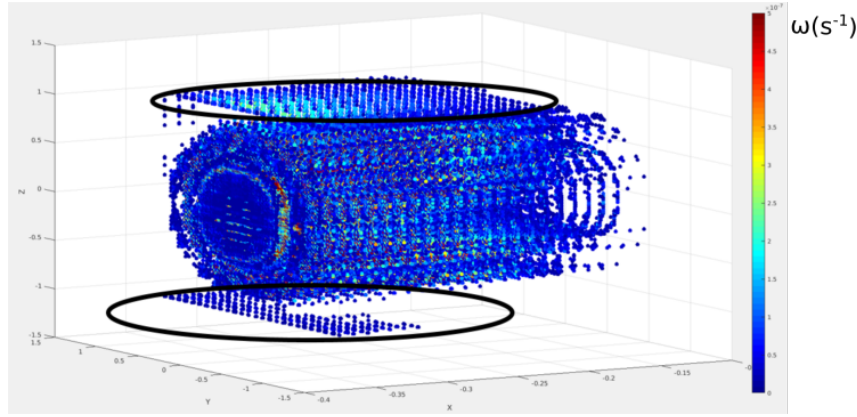


Figure 6: Simulation of the turbulent wake with vorticity particles. The black ellipses show particles which have been artificialy generated by interactions with the boundaries.

At X_1 , the reconstructed vorticity field in OpenFoam corresponds to the one interpreted by Dorothy (Fig.7). As explained in section 2.1.2, Dorothy is based on the vorticity intensity and every vorticity of each particle should be multiplied by its volume. The assigned volume is the volume of the generating cell in OpenFoam.

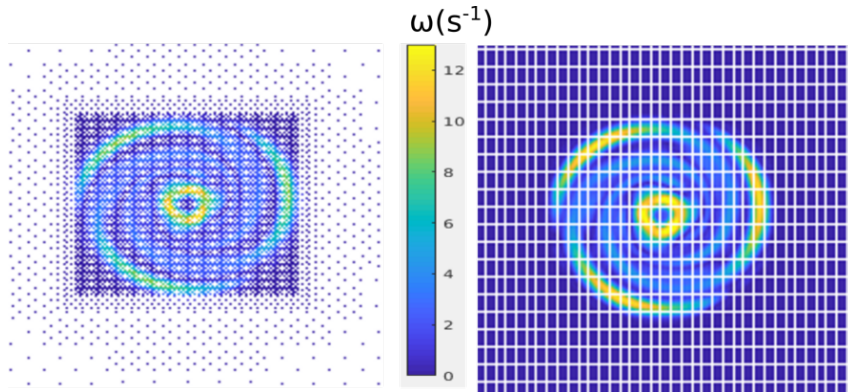


Figure 7: Vorticity field in the X_2 -plan interpreted by OpenFoam (left) and Dorothy (right)

Nevertheless, each extracted plan allows the information to pass from one code to the another, an accumulation of vorticity causes an over-estimation of the energy in the wake. So, the velocity and the vorticity, as shown in Fig.8 and 9, are overestimated in Dorothy.

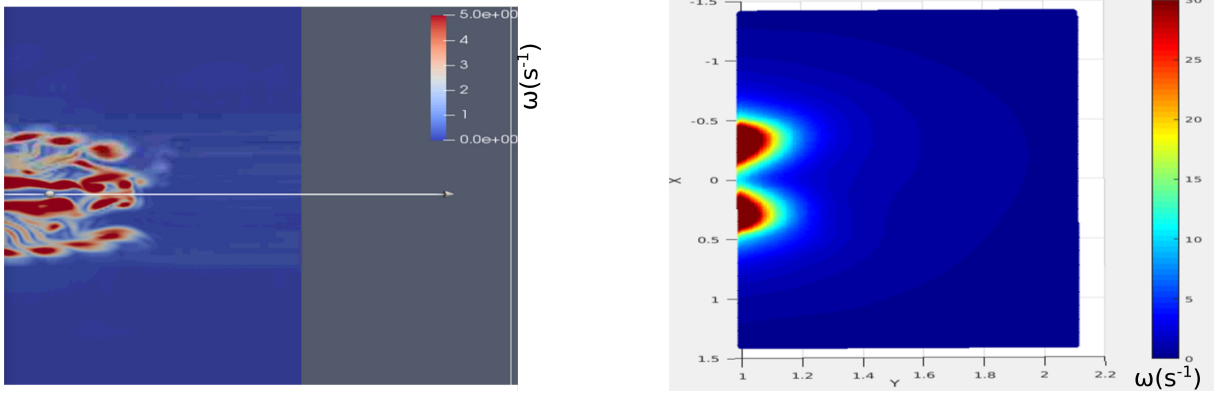


Figure 8: Vorticity field of the wake in OpenFoam (left panel) and Dorothy (right panel).

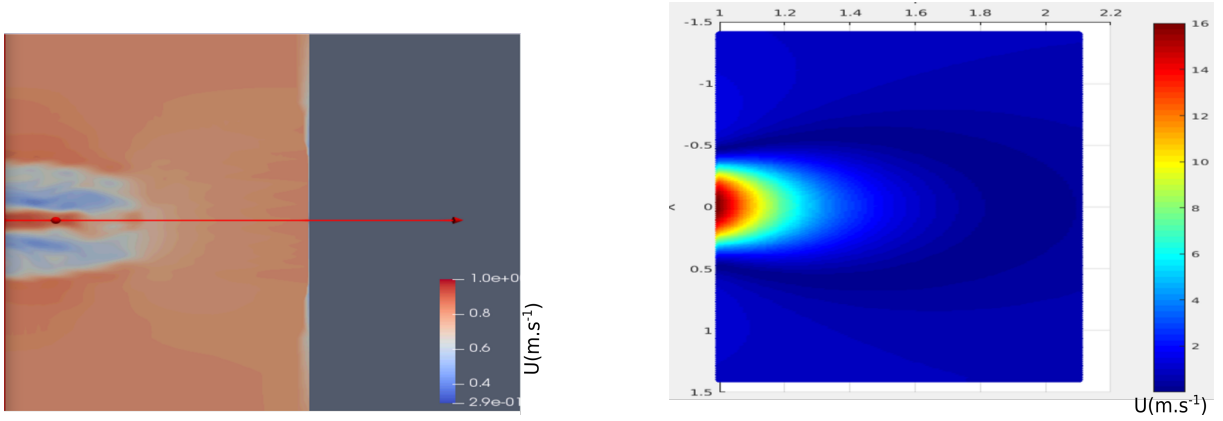


Figure 9: Velocity field of the wake in OpenFoam (left panel) and Dorothy (right panel).

All Dorothy particles are instantly carried by the flow although they already carry the vorticity information from OpenFoam, leading to a sudden acceleration.

A possible way to fix this issue is to implement a numerical source that would aim to suck the particles at a velocity $-U_{\delta\Omega E}$ in the generation plane. To avoid too strong interactions between the particles and the source (the effects apply in $\frac{1}{R^2}$, where R is the distance between two particles), the source should be placed at $\frac{\Delta x}{2}$ upstream of the plane as shown in Fig.10.

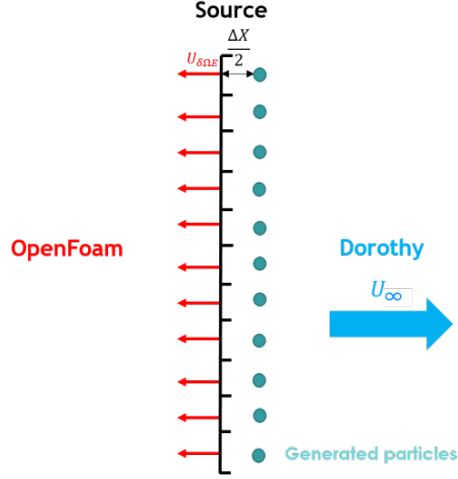


Figure 10: Scheme describing the adding of a source represented by $U_{\delta\Omega E}$ (red arrows). Dorothy particles (blue circles) and the OpenFoam boundary (black line) are represented.

The full Eulerian computation also shows that the tidal converter on performance does not depend on the downstream solution. The study reveals that the power coefficient (C_p) is independent from the mesh outside the AMI (Fig.11). This led to think that the two-way coupling can be optional. The downstream information does not seem to go all the way back to the turbine. There is therefore no need to transfer data from Dorothy to OpenFoam. A one-way coupling can be efficient and costs less computing time as long as the OpenFoam domain is big enough.

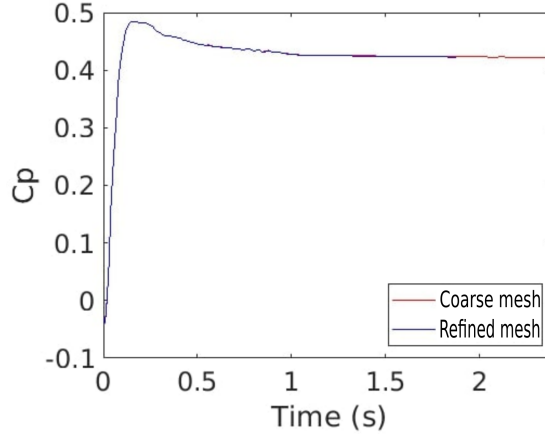


Figure 11: Time evolution of the power coefficient (C_p) with coarse (red line) and refined (blue line) mesh refinement in the wake.

4 CONCLUSION AND PERSPECTIVES

An Eulerian-Lagrangian coupled code has been developed and evaluated. It combines OpenFoam, an Eulerian code, and Dorothy, a Lagrangian code, and benefits from advantages of both codes. OpenFoam code is not limited by Kutta conditions, but requires high computing resources. Dorothy cannot effectively take into account fluid-structure interactions but is cheaper in terms of computational resources. A third party system has been designed to transfer them from one code to the other. The transferred data is consistent. However, the accumulation of particles in Dorothy creates an increase of the flow velocity. A "source" type solution is being considered. It needs to be integrated into the Dorothy code. This allows to correct the results obtained by the coupling code and also to gain in computing performance. A one-way coupling is considered to reduce computing time without changing performance results.

This work will be used to study the impact of biofouling on the performance and the wake of tidal turbines. It will enable the study of numerous species implantation configurations with a reduced calculation time. Thus, we will be able to parameterize the problem and vary the size, the density and even the mass of the colonizing species. More generally, impurity and roughness studies are facilitated by coupling algorithms.

ACKNOWLEDGMENTS

Researches were supported by the "Région Normandie" throughout a PhD grant and by the "Université de Caen Normandie". The authors will provide OpenFoam and Dorothy files and results data on request. Authors are grateful to the "Centre Régional Informatique et d'Applications Numériques de Normandie" (CRIANN) for the calculation facilities and technical support. Thank you to Patrice Robin for English reviewing.

REFERENCES

- [1] M. Shivesgaga and C. Crawford, "Tuned actuator disk approach for predicting tidal turbine performance with wake interaction," *International Journal of Marine Energy*, vol. 17, pp. 1 – 20, 2017.
- [2] D. Marten, M. Lennie, G. Pechlivanoglou, C. N. Nayeri, and C. O. Paschereit, "Implementation, Optimization and Validation of a Nonlinear Lifting Line Free Vortex Wake Module Within the Wind Turbine Simulation Code QBlade," in *ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*, vol. 9, 2015.
- [3] A. Foveau and J.-C. Dauvin, "Surprisingly diversified macrofauna in mobile gravels and pebbles from high-energy hydrodynamic environment of the 'Raz Blanchard' (English Channel)," *Regional Studies in Marine Science*, vol. 16, pp. 188 – 197, 2017.
- [4] H. Titah-Benbouzid and M. Benbouzid, "Marine Renewable Energy Converters and Biofouling: A Review on Impacts and Prevention," in *EWTEC 2015*, pp. 09P1–4–2, 2015.
- [5] J. Orme, I. Masters, and R. Griffiths, "Investigation of the effect of biofouling on the efficiency of marine current turbines," in *Proc. MAREC*, pp. 91–99, 2001.
- [6] Y. Liu, Q. Xiao, A. Incecik, C. Peyrard, and D. Wan, "Establishing a fully coupled CFD analysis tool for floating offshore wind turbines," *Renewable Energy*, vol. 112, pp. 280–301, 2017.
- [7] A. Rivier, A.-C. Bennis, G. Jean, and J.-C. Dauvin, "Numerical simulations of biofouling effects on

- the tidal turbine hydrodynamic,” *International Marine Energy Journal*, vol. 1, pp. 101–109, 2018.
- [8] B.-Y. Min and L. N. Sankar, “Hybrid navier-stokes/free-wake method for modeling blade-vortex interactions,” *Journal of Aircraft*, vol. 47, pp. 975–982, 2010.
- [9] P. Anusonti-Inthra and M. Floros, “Coupled cfd and particle vortex transport method: Wing performance and wake validations,” in *38th Fluid Dynamics Conference and Exhibit*, 2008.
- [10] C. J. Greenshields, *OpenFoam User Guide version 6*, 2018.
- [11] G. Pinon, *Modélisation numérique de jets transverses : application aux cas des inverseurs de poussée d’un avion en phase d’atterrissage*. PhD Thesis, 2005.
- [12] N. N. Mansour, P. Moin, W. C. Reynolds, and J. H. Ferziger, “Improved methods for large eddy simulations of turbulence,” in *Turbulent Shear Flows I*, pp. 386–401, Springer Berlin Heidelberg, 1979.
- [13] P. Mycek, B. Gaurier, G. Germain, G. Pinon, and E. Rivoalen, “Experimental study of the turbulence intensity effects on marine current turbines behaviour. part i: One single turbine,” *Renewable Energy*, vol. 66, pp. 729–746, 2014.