A new wake detection methodology to capture wind turbine wakes using adaptive mesh refinement and Large Eddy Simulation

A new wake detection methodology to capture wind turbine wakes using adaptive mesh refinement and Large Eddy Simulation

U. Vigny[1,2], P. Benard[2], P. Tene Hedje[1], F. Houtin-Mongrolle[2], L. Bricteux[1], S. Zeoli[1]
[1]Université de Mons (UMONS), Polytechnic Faculty, Belgium.
[2]CORIA, CNRS UMR6614, Normandie Université, INSA and University of Rouen, 76801 Saint-Etienne-du-Rouvray, France.

Author contact email: ulysse.vigny@umons.ac.be

Abstract. The development of turbulent vortical wakes released downstream of wind turbines is crucial as it presents many technological implications for wind farm design and exploitation. The numerical prediction of these wakes constitutes a challenging problem as they involve the shedding of fine vortical structures, their instabilities, and interactions with a turbulent ambient flow. A Large Eddy Simulation (LES) approach allows capturing such flow phenomena, which implies a suitable mesh. Adaptive Mesh Refinement (AMR) is used to refine the mesh in the wind turbine wake to limit the computational cost. A methodology is developed to define and capture the wake envelope adequately. Three main parts of this methodology can be identified: The wind turbine wake detection, the target cell size required and adaptation frequency. The target cell size needed to properly capture the wind turbine wake is investigated in previous work [1], while this paper focuses on wind turbine wake detection. A strategy based on a progress variable with a source term in the rotor region is used to capture the wake. This variable is transported by the flow and thus defines the wake envelope. AMR is used to refine the mesh within this region. To validate the method, a comparison between an adaptive mesh case and a reference mesh case has been performed on a single rotor and a two aligned rotor configuration. For both, the wind turbine wake tracking method is effective. The progress variable is transported correctly and leads to a well-defined wake area. The mesh is refined adequately within it. The physical comparison between cases showed similar results, while the performance comparison showed a computational cost reduction of 30% in the single turbine configuration and 50% in the two turbines configuration. Therefore, our methodology coupled with adaptive mesh refinement can adequately capture wind turbine wake, define an accurate wake envelope and decrease the computational cost for the same physical precision.

1. Introduction

According to the current energetic and environmental challenges, maximizing the electric power generated in wind farms and minimizing wind turbine fatigue is a societal concern. The power production and the fatigue loads of a given wind turbine strongly depend on the incident wind turbulence or the wake coming from an upstream turbine. A critical physical phenomenon to study is thus the development of the turbulent vortical wake released downstream of a wind

turbine. The capture of these complex, three-dimensional, unsteady flows calls for a Large Eddy Simulation (LES) approach [2]. Nevertheless, LES requires a fine enough mesh to capture the wake meandering and instabilities. A compromise between computational cost and wake accuracy must be found. By following this tenet, adaptive mesh refinement (AMR) is applied to adapt the mesh in the wake [1]. AMR allows the adaptation of the mesh resolution dynamically during the computation. This technique features the following advantages compared to refined static meshes:

- The adaptive approach requires less a-priori knowledge of the solution.
- It requires significantly less meshing human efforts.
- It allows to save computational elements and thus reduce the computational cost.

AMR is widely used in CFD to simulate a broad range of physical phenomena where a local refinement is required, such as in shock waves [3], flames front [4], bluff body wakes [5], two-phase flows [6]. To the authors' best knowledge, only scarce literature exists concerning AMR methods applied to wind turbine problems. Most of the literature includes strategies based on cartesian grids where the mesh is locally refined based on the global truncation error solution as in [7, 8, 9]. Kirby et al. [9] proposed a mesh composed of two parts where the adaptation is only performed on the cartesian off-body mesh region while the region close to the bodies is fully unstructured. In contrast to these investigations, our strategy relies on the use of fully unstructured conformal tetrahedral LES grids to cope with both geometrical and unsteady effects, either due to the turbulent nature of the flow or the change of the wind turbine operating condition (e.g. turbulence, wind turbine movement...).

In a previous work [1], the goal was to use AMR to generate Eulerian elements only where the flow physics requires a finer mesh. In wind turbine simulations, it corresponds to wind turbine wakes. For that, a physical criterion was introduced to detect the area of interest and to calculate the required target cell size to capture the flow physics. This work limitation was that the strategy didn't work under turbulent flows. Indeed, the criterion was based on the velocity gradient and thus activated in the wake and all turbulent regions of the computational domain. This led to a very expensive mesh. Here, we extend this work while focusing on the detection part. To detect wind turbines wake, several methods are used in the literature [10, 11, 12, 13]. Usually, it is used to detect the position of the wake for turbines in yaw. These techniques are based on linear/angular impulse, velocity/momentum deficit and available power in the flow for run-time tracking of the wake centerline. These wind turbine wake detection methods are mainly limited by two factors: the computational cost and the accuracy under highly turbulent conditions. Thus, a new wake detection method is developed, satisfying both issues. Once the wake is adequately detected, the AMR strategy is applied within this envelope without impacting the other turbulent regions of the domain.

To validate this approach, two configurations are investigated. One is a single wind turbine case with dynamic yaw misalignment. The second case is based on the first one, with the addition of a downstream aligned wind turbine. For both configurations, two methodologies are compared. The reference case is computed on a mesh with a large refined area over the turbine position and the supposed wake position. The second case is based on the AMR strategy introduced in this work, where the mesh is progressively optimized during the simulation.

2. Methodology

The developed methodology is divided into three parts: the detection to capture the wake, the target mesh size definition and the adaptation frequency.

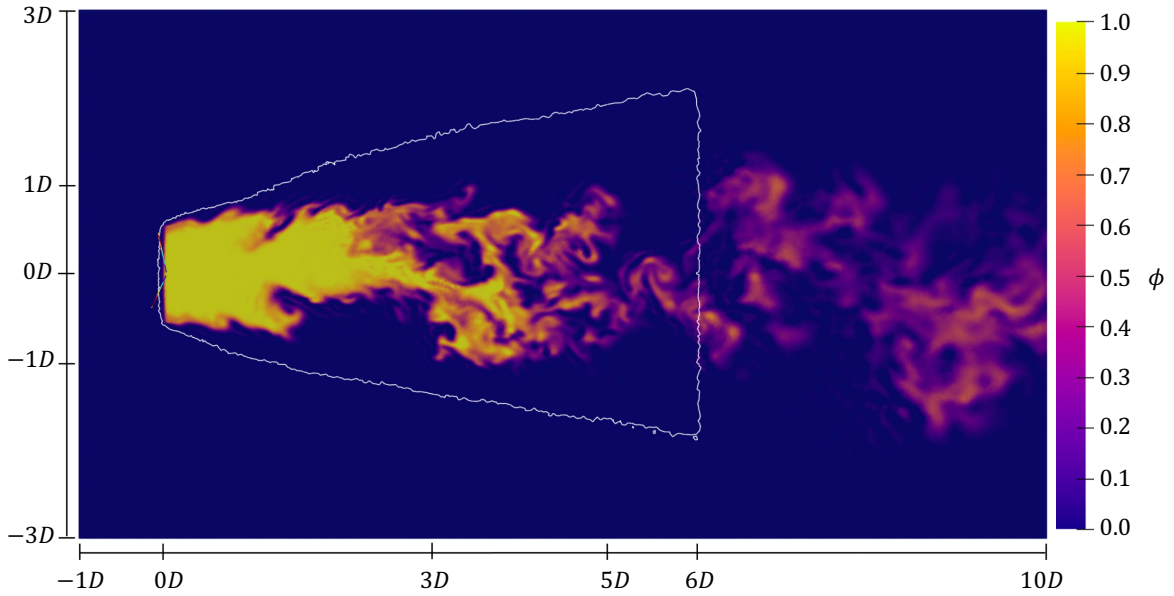

Figure 1. Mid-place passive scalar ϕ field highlighting the wind turbine and $\widehat{\phi} = 1$ iso-contour (in white) representing the mesh refinement envelope

• **Detection:** To properly capture wind turbine wakes we use a strategy based on a progress variable ϕ. A progress variable is a passive scalar with a source term transported on the Eulerian grid. Here, the source term is nonzero in the cylinder inscribed in the rotor region. The transport equation for a progress variable ϕ is

$$\frac{\partial \rho\phi}{\partial t} + \nabla \cdot (\rho \mathbf{u}\phi) = \nabla \cdot (\rho D_\phi \nabla \phi) + \dot{\omega}_\phi \,, \tag{1}$$

where ρ is the density, $\mathbf{u}$ the local velocity on the Eulerian grid, D_ϕ the diffusion coefficient of the variable and $\dot{\omega}_\phi$ its source term. A representation of the progress variable can be seen in fig. 1. This method captures the instantaneous wake. To have an average wake position, at every fluid iteration, the progress variable is overlaid with the previous one, leading to a 3D field called $\widehat{\phi}$. In other words, a thresholding method is applied: $\phi > 0.1$ leads to $\widehat{\phi} = 1$. Since $\widehat{\phi}$ is never reset, the wake envelope is only growing and converges to an area where the wind turbine wake is located. The $\widehat{\phi}$ quantity follows:

$$\widehat{\phi}(\mathbf{x}, t + dt) = \max\left(\widehat{\phi}(\mathbf{x}, t), \phi^\star(\mathbf{x})\right) \text{ with } \phi^\star(\mathbf{x}) = \begin{cases} 1, & \text{if } \phi(\mathbf{x}) > 0.1 \\ 0, & \text{if } \phi(\mathbf{x}) \leq 0.1 \end{cases} \tag{2}$$

Figure 1 shows that the instantaneous ϕ plane is highly turbulent and exhibits pockets while $\widehat{\phi}$ covers a larger area containing any wake position. Given the fact that turbine spacing in wind farms usually falls within the range of 3 to 10 rotor diameters [14], the authors chose to stop the wake analysis, and thus $\widehat{\phi}$, at $6D$ downstream the wind turbine. Likewise, the $\phi = 0.1$ value seems arbitrary and questionable, but testing different values has shown no significant influence on the wake envelope.

• **Mesh size:** To properly capture the flow physics, the cell size should depend on it. Since this work has already been performed [1] and for simplicity purposes, the cell size is imposed as a user-dependent target value. A future task will be to assemble both works.

• **Frequency:** AMR is occurring iteratively. The adaptation is triggered when the current mesh is too far from the objective mesh. To this end, a metric error is measured on each control volume as the local ratio between actual and target cell sizes. When this ratio exceeds a specified threshold value, 50%, on a sufficiently large number of elements, 0.2%, the AMR is triggered. In other words the AMR is triggered when 0.2% elements will have an error superior to 50%.

3. Simulation framework

3.1. Flow solver

The YALES2 flow solver is used [15, 16]. It is a massively-parallel finite-volume solver, specifically tailored for Large-Eddy Simulation, and relies on a 4th-order central numerical scheme for spatial discretization. The 4th-order Runge-Kutta method is used for the time integration [17]. This code solves the Navier–Stokes equations for incompressible turbulent flows on both structured and unstructured meshes. The library also features an actuator line model [18] as described and validated in [15]. The sub-grid scale model used here is the σ-model proposed by Nicoud et al. [19] which prevents sub-grid dissipation in the vortex cores. The mesh adaptation is performed using the MMG [20, 2] remeshing library.

3.2. Wind turbine modelling

In this work, the modeled wind turbine is the academic DTU10MW [21]. This turbine follows the technological evolution of offshore wind turbines, for which the rotor reaches diameters values of $100 - 200\,\mathrm{m}$. In the following, all quantities are scaled by the wind turbine diameter $D = 178.3\,\mathrm{m}$. The blades use multiple airfoils along the span with variable chord and twist [21]. The deformation of the blades is not taken into account in this study, which implies a strong hypothesis on the loads computation. Indeed, for such diameters, the flapwise blade deformation at the tip can reach up to tens of meters [22]. With this hypothesis, the authors choose to use the non-prebended blades with the designed cone angle of 2.5°. The rotation speed is imposed for the upstream wind turbine to obtain the design tip speed ratio $\lambda_{opt} = 7.5$, giving a Reynolds number of approximately $Re_{tip} \approx 6 \times 10^6$ at the blade tip. The DTU10MW controller developed in [21] is applied to the downstream wind turbine to obtain consistent results. The rotor blades are modeled as actuator lines (AL) [18] that compute the blade forces at each time step based on the inflow velocity and the angle of attack α. Then the lift C_L and drag C_D coefficients obtained from the tabulated airfoil properties [21] are used to compute the aerodynamic forces. This computation of lift and drag forces $\mathbf{F}_{2D} = (\mathbf{L}, \mathbf{D})$ is done at the AL location using two-dimensional airfoil theory. The blade forces are then regularized on the Eulerian grid by performing a mollification using an isotropic three-dimensional Gaussian kernel [18]. The Gaussian kernel width ϵ is related to the local grid cell size Δx by Eq. 3 and thus depends on the mesh.

$$\frac{\epsilon}{\Delta x} = 2 \tag{3}$$

Each blade is discretized using 50 sections, i.e 50 points per actuator lines. The use of the AL method allows to model the unsteady load distribution of the blades and the resulting induction on the flow without resolving the flow over the blade geometry.

3.3. Turbulence injection

As shown in [23], turbulence injection is a key parameter for high fidelity wind turbine simulations. Different turbulent inlet boundary conditions for LES exist. The four most renowned methods are the Wind Tunnel Replication Method [24], the Recycling Method [25], the Precursor Database Method [26] and the Synthetic Turbulence Generator Method [27]. The wind tunnel replication method is the most straightforward method, easy to implement with high accuracy, although it is time-consuming and computationally expensive. The synthetic

generator method is very effective and satisfies a Gaussian spectrum. Thus it does not apply to atmospheric boundary layer flows. The recycling and precursor database methods are similar in terms of flow-processing and computation cost since they both require a dedicated computation. Here, the precursor method was chosen for its cost. A single pre-processing computation is needed for every case in every configuration and thus reduces the global computational cost. The precursor computation is a half-channel flow of height H with periodic inlet and outlet boundary and a wall model at the bottom. It is driven using a constant pressure gradient forcing Eq. 4, where u_τ is the wall friction velocity.

$$\left(\frac{dP}{dx}\right)_f = \frac{u_\tau^2}{H} \tag{4}$$

It simulates a velocity profile similar to that in Eq. 5 where the roughness length is here taken as $z_0 = 0.02\,\mathrm{m}$ and $\kappa = 0.41$. u_τ is determined so that u_{ref} is near $10\,\mathrm{m/s}$ at hub height ($z_{ref} = 119\,\mathrm{m}$).

$$u(z) = \frac{u_\tau}{\kappa} \log\left(\frac{z + z_0}{z_0}\right) \tag{5}$$

2D cross-flow planes are then injected into the wind turbine computation as inlet velocity, accounting for turbulence and vertical logarithmic law velocity profile. Velocity profile and Turbulent Intensity (TI) are shown in fig. 2. The velocity profile shows good agreement between the theoretical and actual velocity profiles observed in the precursor computation. At hub height, the mean velocity, which will be called u_{ref} is $10.5\,\mathrm{m/s}$. The streamwise turbulent intensity is near 10% at hub height. The transverse turbulent intensity has a similar appearance with a lower value. At the bottom, the vertical turbulent intensity is minimum because of the boundary but near 5% at hub height.

3.4. Configuration

The methodology is applied to two configurations involving the DTU10MW [21] wind turbine. Configuration (A) investigates a single turbine with a time-varying yaw misalignment defined in Eq. 6. In configuration (B) a second wind turbine is added downstream the first turbine operating with a dynamic yaw misalignment.

$$\gamma = 15 \sin\left(\frac{\pi}{60} \times t\right) \tag{6}$$

The AMR methodology is compared to a reference case (refine REF) with uniform cell size in the wake region for both configurations. Figure 3 presents the mesh for both AMR and REF cases. Wind turbines are indicated as T for the configuration (A) and as $T1$ and $T2$ respectively for the upstream and the downstream wind turbine for the configuration (B). The domain is a $14D \times 10D \times 10D$ box for case (A) and $18D \times 10D \times 10D$ for case (B), with a $6D$ distance between both turbines. Such dimensions allow to properly study the wake and to prevent confinement effect due to the boundary proximity [15, 28, 29]. For both configurations, two regions are refined: the wind turbine wake and the rotor region. The wind turbine wake region is refined to capture the flow physics adequately. As said in section 2, for simplicity purposes, we impose a user-dependent target cell size, which will be constant in the wind turbine wake region. The rotor region is refined to mollify the Gaussian kernel properly. As mentioned in section 3.2, the Gaussian kernel width depends on the mesh, and thus, more refined mesh results in a smaller kernel size.

A third case, named coarse REF, is computed to measure the sensitivity of the results to mesh resolution. Its cell size is twice the one of the refine REF in every three directions. It is, therefore, a mesh with eight times fewer elements.

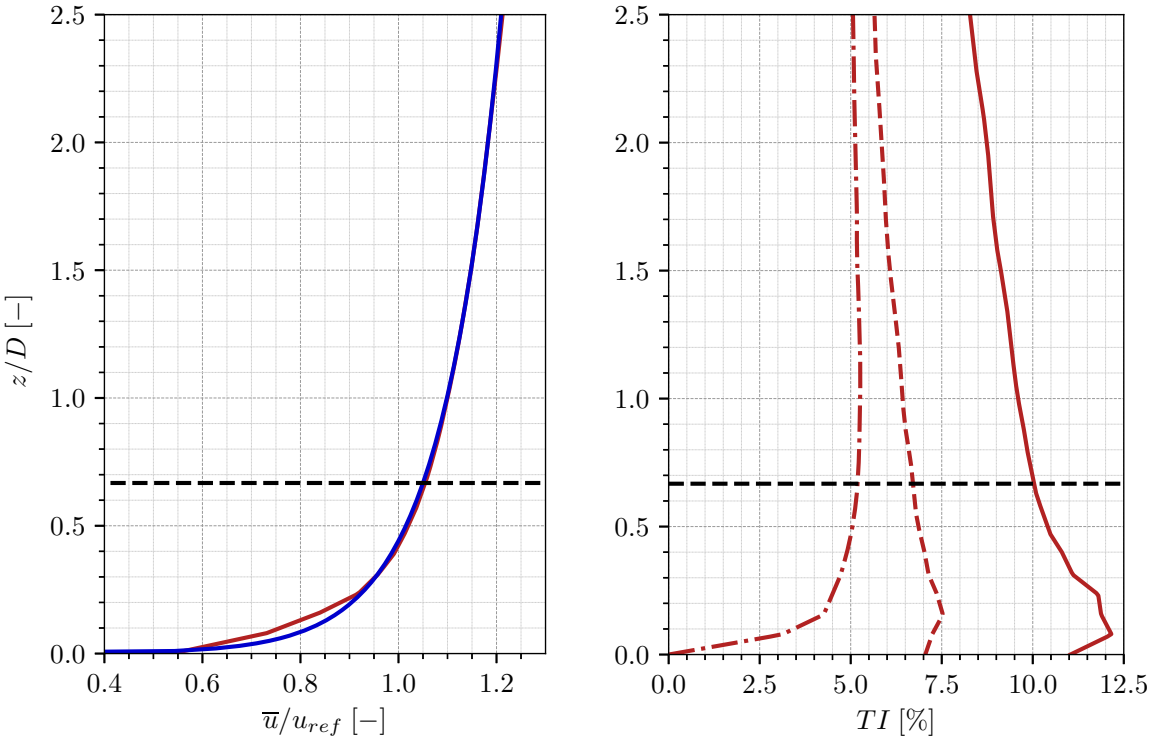

Figure 2. Velocity profiles and precursor turbulence intensity, zoomed in rotor region. Left : (—) for the precursor simulation, (—) for the theoretical velocity profile. Right : (—) for TI_X, (- - -) for TI_Y and (-·-·-) for TI_Z. (- - -) for the rotor hub position.

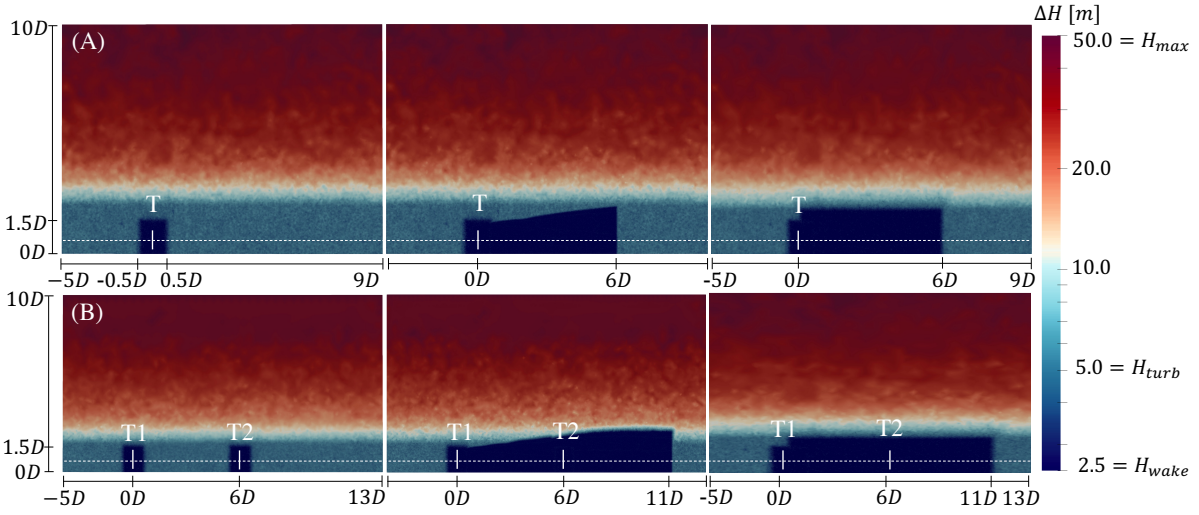

Figure 3. Mid-plane visualisation of the meshes with configuration (A) at the top and configuration (B) at the bottom. From left to right : initial AMR mesh, final AMR mesh, REF mesh.

4. Results

4.1. Flow dynamics analysis

For configuration (A), the mesh goes from 76.8 to 134 million tetrahedrons, while the REF mesh contains 160 million elements. For configuration (B), the mesh goes from 83 to 290

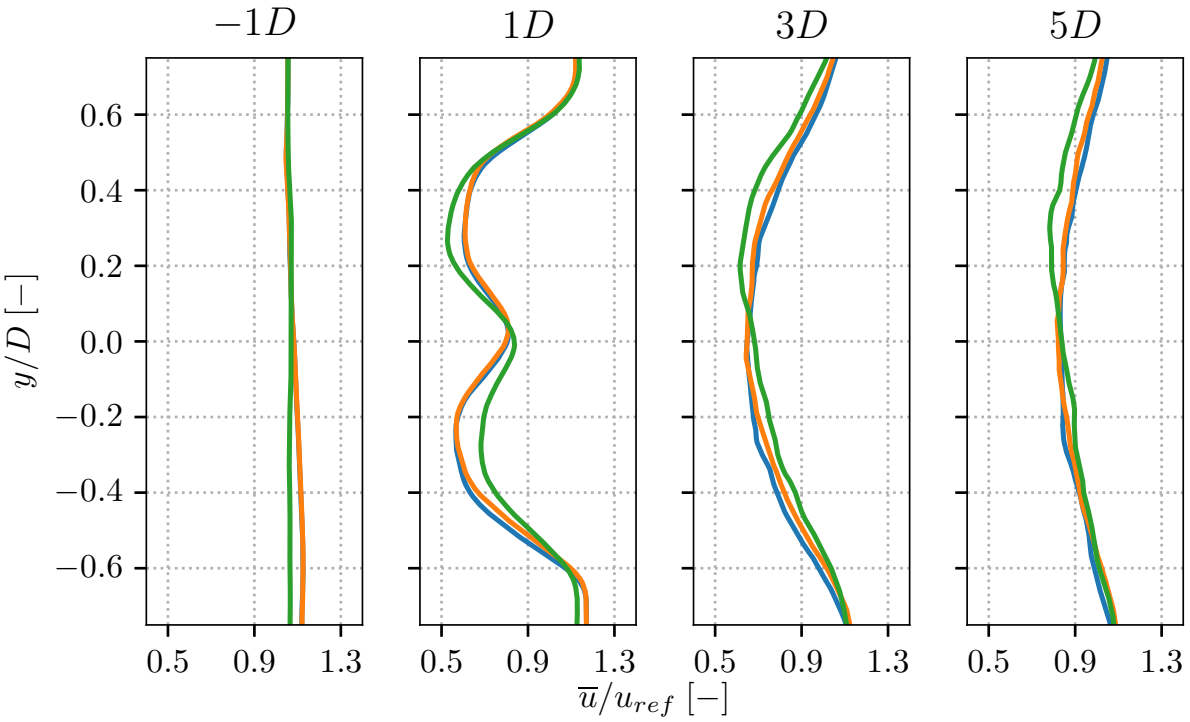

Figure 4. Mean streamwise velocity at various stream position in transversal (y) direction for (A) configuration. (—) for the AMR, (—) for the refine REF and (—) for the coarse REF.

million elements, while the REF mesh has 500 million tetrahedrons.

In the rotor region the cell size is $\Delta h_{rotor} = 2.5\,\mathrm{m}$ which leads to a 72 points per blade discretization. The cell size is the same for the wake region. The background cell size at hub height is twice bigger: $2\Delta h_{rotor}$. This cell size is the minimum required, according to the Shannon sampling theorem, to capture the precursor inflow velocity and turbulence properly. In the vertical direction, the mesh size increases linearly, up to $20\Delta h_{rotor}$. As mentioned in section 3.4, coarse REF mesh size is twice bigger as the finer REF. Therefore, the cell size in the refined region is $2\Delta h_{rotor}$ while the background cell size at hub height is $4\Delta h_{rotor}$.

Each case, on each configuration, is performed in two parts. The first is the convergence part, where the flow reaches a "steady" state; during this part, the inlet turbulence and the generated wind turbine wakes are slowly transported in the domain. Once convergence is achieved, the second part consists in accumulating statistics over the flow quantities. For the AMR case, the first part also allows obtaining a converged wake envelope and, consequently, refining the mesh in the wake area. During the statistic accumulation time, the adaptation process isn't triggered anymore. At every iteration, the wind turbine wake is within the wake envelope. Therefore statistic accumulations are made on a constant mesh.

Fig. 4 and Fig. 5 present the behavior of the time-averaged streamwise velocity in transversal direction at various streamwise positions for both (A) and (B) configurations, respectively. The three cases (AMR, refine REF and coarse REF) mean velocity profiles present a good agreement: only some minor discrepancies can be observed in the downstream region is near identical. Mean streamwise velocity profiles indicate that the wake velocity deficit recovers at a similar position in the three cases. The relative L2 norm error on horizontal mean velocity profiles quantifies the discrepancies to the refine REF case set as the reference case. Results are shown in Tab. 1.

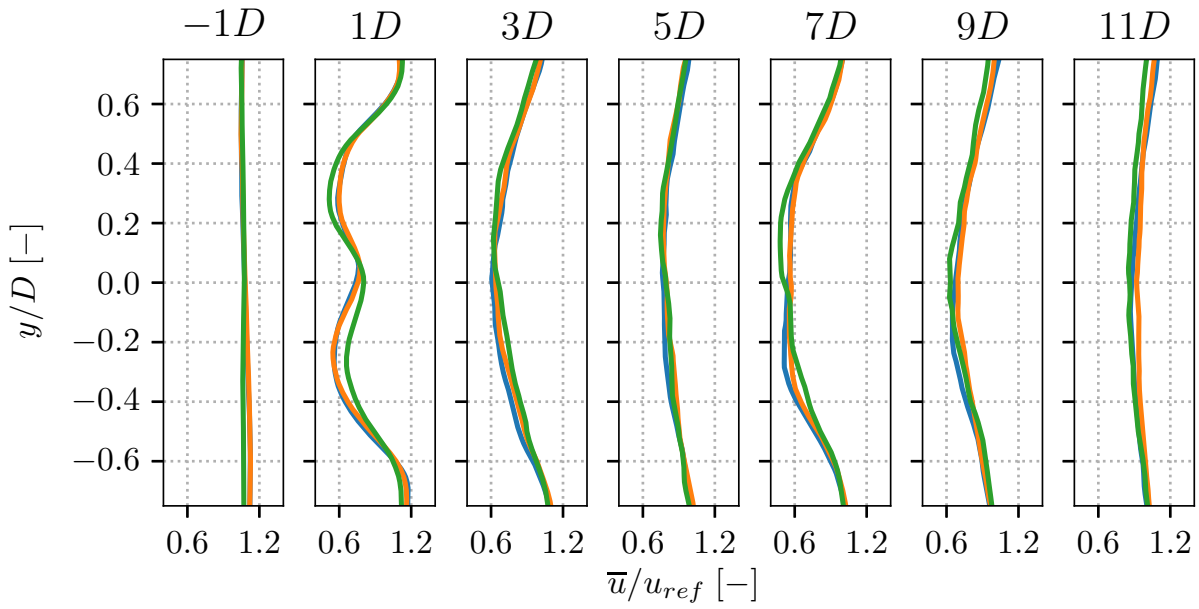

Figure 5. Mean streamwise velocity at various stream position in the transverse (y) direction for (B) configuration. (—) for the AMR, (—) for the refine REF and (—) for the coarse REF.

For the (A) configuration, the AMR case has an error smaller than 1%, while the coarse REF case has around 18%. For the (B) configuration, the AMR case has an error of around 2.4%, while the coarse REF case has around 10%.

The L2 norm error reaches 1.4% for the AMR case and 19% for the coarse REF case on mean vertical streamwise velocity profiles. Since vertical and transverse direction velocities show similar results, profiles are not shown here for the sake of brevity. These results prove that the AMR strategy leads to quasi-identical time-averaged velocity.

Another measured quantity is the Turbulent Kinetic Energy (TKE), calculated as:

$$TKE = \frac{1}{2}\left(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}\right) , \tag{7}$$

where u', v' and w' are the variance of the three velocity components, also known as the root mean square velocity fluctuation. TKE can have many sources, such as fluid shear, friction, and buoyancy. Turbulent kinetic energy is then transferred down the turbulence energy cascade and is dissipated by viscous forces at the Kolmogorov scale. Therefore, the mesh size should influence the TKE.

Figures 6 and 7 present the behavior of the TKE in the transverse direction for (A) and (B) configurations, respectively. The AMR and refined cases TKE profiles exhibit similar trends in both configurations. However, the coarse REF case TKE is significantly weaker. Results are shown in Tab. 1. The AMR case shows an error around 5.5% while the coarse REF case reaches 60% for configuration (A). For configuration (B), similar behavior is noted. The AMR detection strategy demonstrates its reliability with these low differences on TKE, which can be complex to predict.

The similar results between the AMR and refined REF cases showed that they both have the same precision. On the other hand, the coarse REF case shows moderate deviations in the

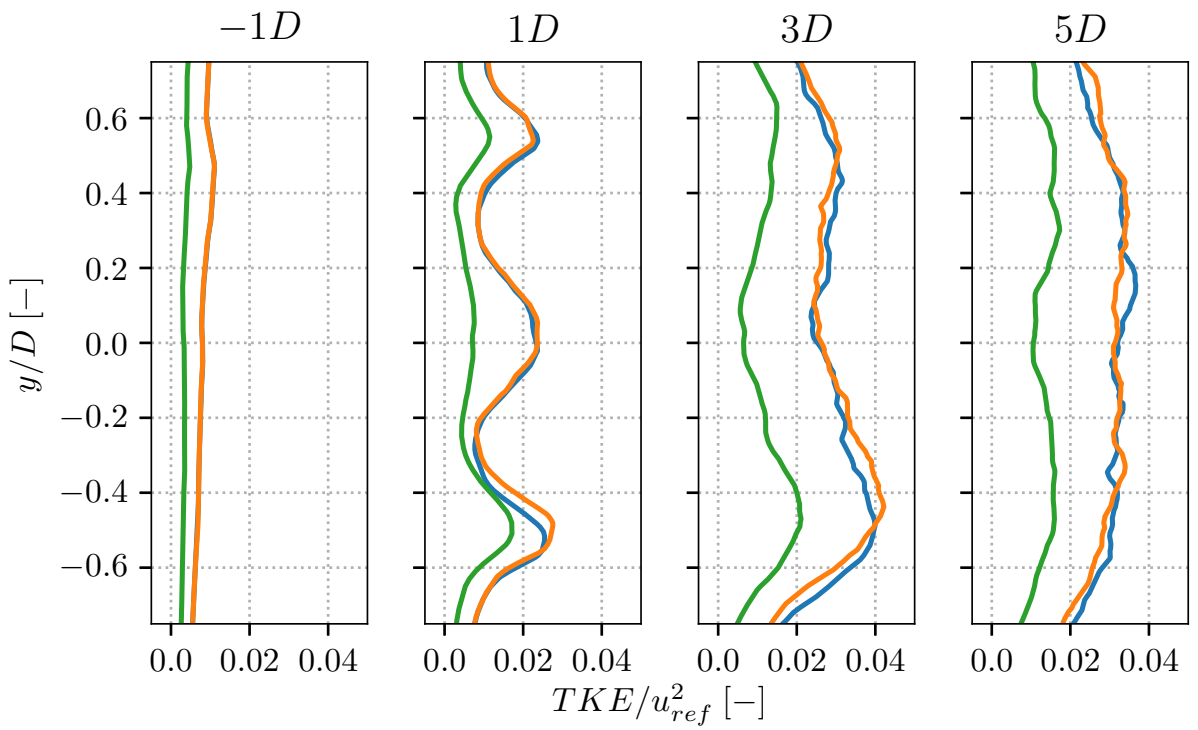

Figure 6. TKE at various stream position in the transverse (y) direction for (A) configuration. (—) for the AMR, (—) for the refine REF and (—) for the coarse REF.

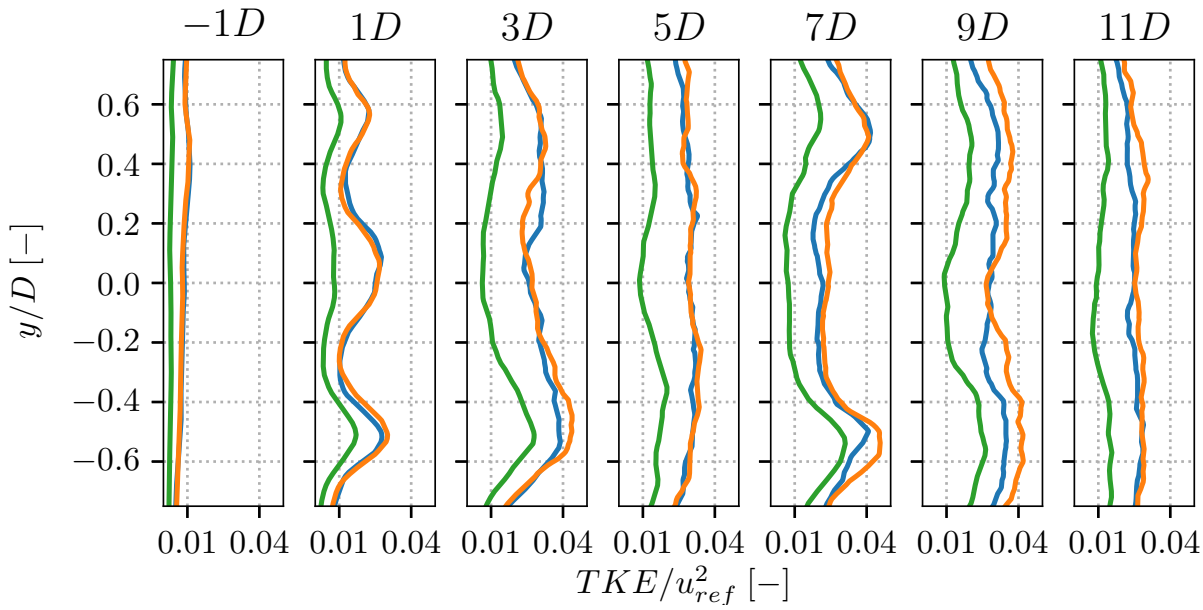

Figure 7. TKE at various stream position in the transverse (y) direction for (B) configuration. (—) for the AMR, (—) for the refine REF and (—) for the coarse REF.

Config	cases	$\overline{u}_y$ [%]	TKE [%]	ΔC_P [%]	ΔC_T [%]
(A)	AMR	1	5.5	0.8	0.18
	coarse REF	18	60	4.2	1.37
(B)	AMR	2.4	11.3	0.77	0.23
	coarse REF	10	72	3.7	1.57

Table 1. relative L2 norm error recapitulation table for mean velocity, TKE, power and thrust coefficient of both AMR and coarse REF cases, compared to the refine REF, in both (A) and (B) configuration.

mean velocity and strong discrepancies on the TKE. Therefore, results are strongly sensitive to the mesh resolution and sufficient refinement is needed for the physical precision.

Power C_P and thrust C_T coefficients are also compared. Results are shown in Tab. 1. For the configuration (A), coefficients difference between the AMR case and the refine REF case respectively are $\Delta C_P = 0.8\%$ and $\Delta C_T = 0.18\%$. The refine REF case and the coarse REF case respectively are $\Delta C_P = 4.2\%$ and $\Delta C_T = 1.37\%$. The configuration (B) shows similar results. Thus, the AMR and the refine REF cases have similar results, while the coarse REF tends to be less precise.

4.2. Computational cost comparison

Mesh adaptation is applied to reduce the mesh size and thus the computational cost. The computational cost comparison is performed for both convergence and statistic accumulation. Computational performances are summarized in Tab. 2.

Config	cases	C/S	#elem $[\times 10^6]$	#proc	timestep [ms]	adapt time [%]	khCPU
(A)	REF	CONV	160	1024	32.2	-	12.1
		STAT	160	1024	32.5	-	12.1
	AMR	CONV	77-135	1024	46.2	6.9	8
		STAT	135	1024	45.0	0.0	8.3
(B)	REF	CONV	500	2048	40.8	-	80
		STAT	500	2048	41.0	-	94
	AMR	CONV	83-290	1024	34.5	4.1	45
		STAT	290	1024	45.8	0.0	38

Table 2. computational performances recapitulation table of both refine REF and AMR cases in both (A) and (B) configuration. Each (A) configuration simulation represent 720 s physical time while each (B) configuration simulation represent 1440 s physical time.

For the configuration (A), refine REF case costs 24.2 khCPU, while the AMR case costs 16.3 khCPU, leading to a 30% gain. Both the convergence part and the statistic accumulation part gain are similar. In the AMR case, a similar cost is found between convergence and statistic accumulation parts because the adaptation process represents only 6.9% of the computation time. This emphasizes that the mesh contains fewer elements at the beginning of the simulation before reaching its final state. For the configuration (B), the refine REF case costs 174 khCPU and the AMR case 83 khCPU, representing a 50% gain. The gain is here bigger due to the higher

difference in the number of elements for the REF case: the mesh contains 42% fewer elements, while this gap was 16% on the (A) configuration. It must also be noted that the REF simulation had to be performed on more processors because of this difference in mesh size.

The computational cost difference is mainly due to two factors for both configurations: the mesh size and the timestep. The timestep is not imposed but computed via the CFL stability condition. A high skewness cell associated with a small cell size and a high local velocity leads to a smaller timestep. Both cases have a similar level of skewness, with a 0.8 maximum, but without the guarantee that these cells are in the same region, with similar cell size and local velocity. Therefore, a similar timestep for both configurations is not guaranteed. This difference is unfortunate because it prevents us from having a straightforward comparison between mesh size and computational cost. Nevertheless, the timestep difference cannot explain the computational cost difference on its own. The configuration (A) shows that for a 30% timestep difference, there is a 33% computational cost difference. The configuration (B) shows that for a 1.8% timestep difference, there is a 52% computational cost difference, which is therefore mainly due to a 42% mesh size difference. Thereby, AMR does decrease the computational cost and the mesh size. It also allows for a less user-dependent mesh which may imply better quality.

5. Conclusions and further work

This work objective was to show that adaptive mesh refinement could minimize computational cost and, therefore, be useful in wind turbine computation for the same physical precision. For this purpose, an adaptive mesh refinement methodology has been developed. Based on a progress variable transported in the wake with a source term in the rotor region, it allows to adequately capture the wind turbine wake within which the mesh is refined. Results have shown that the AMR case exhibits a similar physical precision for both single and two-turbines configurations as the REF case. Moreover, the AMR case showed a 30% computational cost reduction in configuration (A) and 50% in configuration (B). Therefore, this methodology coupled with adaptive mesh refinement has two advantages:

- Proper capture of the wind turbine wake and an accurate wake envelope definition.
- Cost reduction for the same physical precision.

However, even if preliminary results are promising, this work shows some limits and needs deeper investigations. This adaptation process is closer to an adaptive mesh convergence than adaptive mesh refinement since the adaptation process does not occur in the statistic accumulation run. Also, the user-dependent target cell size is not correlated to the flow physics. Nevertheless, this work is part of a more global study that already introduces a target cell size criteria depending on the flow physics [1]. Thus, assembled, these two works would detect the wake area and dynamically refine the mesh at required positions to predict the flow characteristics.

6. Acknowledgment

We acknowledge PRACE for awarding us access to Joliot-Curie at GENCI@CEA, France.

References

[1] Zeoli S, Balarac G, Bénard P, Georis G, Houtin-Mongrolle F and Bricteux L 2020 Large eddy simulation of wind turbine wakes using adaptative mesh refinement *Journal of Physics: Conference Series* vol 1618 (IOP Publishing) p 062056
[2] Benard P, Balarac G, Moureau V, Dobrzynski C, Lartigue G and d'Angelo Y 2016 *International journal for numerical methods in fluids* **81** 719–740
[3] Berger M J and Colella P 1989 *Journal of computational Physics* **82** 64–84
[4] Bénard P, Lartigue G, Moureau V and Mercier R 2019 *Proceedings of the Combustion Institute* **37** 5233–5243

[5] Rossinelli D, Hejazialhosseini B, van Rees W, Gazzola M, Bergdorf M and Koumoutsakos P 2015 *Journal of Computational Physics* **288** 1–18
[6] Fuster D, Agbaglah G, Josserand C, Popinet S and Zaleski S 2009 *Fluid dynamics research* **41** 065001
[7] Angelidis D and Sotiropoulos F 2015 Simulation of wind turbine wakes on locally refined cartesian grids *33rd Wind Energy Symposium* p 1471
[8] Deiterding R and Wood S L 2016 Predictive wind turbine simulation with an adaptive lattice boltzmann method for moving boundaries *Journal of Physics: Conference Series* vol 753 (IOP Publishing) p 082005
[9] Kirby A C, Brazell M J, Yang Z, Roy R, Ahrabi B R, Stoellinger M K, Sitaraman J and Mavriplis D J 2019 *The International Journal of High Performance Computing Applications* **33** 897–923
[10] Oberlack M, Peinke J, Talamelli A, Castillo L and Hölling M 2012 *Progress in turbulence and wind energy IV: proceedings of the iTi conference in Turbulence 2010* (Springer)
[11] Vollmer L, Steinfeld G, Heinemann D and Kühn M 2016 *Wind Energy Science* **1** 129–141 URL `https://wes.copernicus.org/articles/1/129/2016/`
[12] Howland M F, Bossuyt J, Martínez-Tossas L A, Meyers J and Meneveau C 2016 *Journal of Renewable and Sustainable Energy* **8** 043301
[13] Coudou N, Moens M, Marichal Y, Van Beeck J, Bricteux L and Chatelain P 2018 Development of wake meandering detection algorithms and their application to large eddy simulations of an isolated wind turbine and a wind farm *Journal of Physics: Conference Series* vol 1037 (IOP Publishing) p 072024
[14] Porté-Agel F, Bastankhah M and Shamsoddin S 2020 *Boundary-Layer Meteorology* **174** 1–59
[15] Bénard P, Viré A, Moureau V, Lartigue G, Beaudet L, Deglaire P and Bricteux L 2018 *Computers & Fluids* **173** 133–139
[16] Moureau V, Domingo P and Vervisch L 2011 *Comptes Rendus Mécanique* **339** 141–148
[17] Kraushaar M 2011 *Application of the compressible and low-Mach number approaches to Large-Eddy Simulation of turbulent flows in aero-engines* Theses Institut National Polytechnique de Toulouse - INPT URL `https://tel.archives-ouvertes.fr/tel-00711480`
[18] Sorensen J N and Shen W Z 2002 *J. Fluids Eng.* **124** 393–399
[19] Nicoud F, Toda H B, Cabrit O, Bose S and Lee J 2011 *Physics of fluids* **23** 085106
[20] Dobrzynski C and Frey P 2008 Anisotropic delaunay mesh adaptation for unsteady simulations *Proceedings of the 17th international Meshing Roundtable* (Springer) pp 177–194
[21] Bak C, Zahle F, Bitsche R, Kim T, Yde A, Henriksen L C, Hansen M H, Blasques J P A A, Gaunaa M and Natarajan A 2013 The dtu 10-mw reference wind turbine *Danish Wind Power Research 2013*
[22] Manolas D, Riziotis V and Voutsinas S 2015 *Journal of Computational and Nonlinear Dynamics* **10** 041008
[23] Wu X 2017 *Annual Review of Fluid Mechanics* **49** 23–49
[24] Mann J 1998 *Probabilistic engineering mechanics* **13** 269–282
[25] Xiao F, Dianat M and McGuirk J 2010 A recycling/rescaling method for les inlet condition generation *Proceedings of the 8th International ERCOFTAC Symposium on Engineering Turbulence Modelling and Measurements, Marseille, France* pp 510–515
[26] Stevens R J, Graham J and Meneveau C 2014 *Renewable energy* **68** 46–50
[27] Zhong J, Cai X and Xie Z T 2021 *Geoscientific Model Development* **14** 323–336
[28] Moens M, Duponcheel M, Winckelmans G and Chatelain P 2018 *Wind Energy* **21** 766–782
[29] Houtin—Mongrolle F, Bénard P, Lartigue G and Moureau V 2021 A level-set framework for the wind turbine wake analysis: from high-fidelity unsteady simulations to 1d momentum theory *Journal of Physics: Conference Series* vol 1934 (IOP Publishing) p 012011