

RESEARCH ARTICLE

Mesh-adaptive simulations of horizontal-axis turbine arrays using the actuator line method

Georgios Deskos  | Matthew D. Piggott 

Department of Earth Science and Engineering,
South Kensington Campus, Imperial College
London, London, UK

Correspondence

Georgios Deskos, Department of Earth Science
and Engineering, South Kensington Campus,
Imperial College London, London SW7 2AZ, UK.
Email: g.deskos14@imperial.ac.uk

Funding information

Imperial College London's Energy Futures Lab;
EPSRC, Grant/Award Number: EP/R007470/1
and EP/L000407/1

Abstract

Numerical models of the flow and wakes due to turbines operating within a real-scale offshore wind farm can lead to a prohibitively large computational cost, particularly when considering blade-resolved simulations. With the introduction of turbine parametrizations such as the actuator disk (AD) or the actuator line (AL) models, this problem has been partially addressed, yet the computational cost associated with these simulations remains high. In this work, we present an implementation and validation of an AL model within the mesh-adaptive three-dimensional fluid dynamics solver, Fluidity, under a unsteady Reynolds-averaged Navier-Stokes-based turbulence modelling approach. A key feature of this implementation is the use of mesh optimization techniques, which allow for the automatic refinement or coarsening of the mesh locally according to the resolution needed by the fluid flow solver. The model is first validated against experimental data from wind tunnel tests. Finally, we demonstrate the benefits of mesh-adaptivity by considering flow past the Lillgrund offshore wind farm.

KEYWORDS

actuator line model, Lillgrund offshore wind farm, mesh optimization, uRANS turbulence models

1 | INTRODUCTION

Turbine parametrization models (TPMs) such as the actuator line model (ALM) and the actuator disk model (ADM) exhibit a large number of advantages compared to blade-resolved simulations, both in terms of their respective computational efficiency but also as far as their implementation within a CFD solver is concerned. First, by using TPMs, the number of degrees of freedom needed by the fluid solver is significantly reduced since the boundary layer of the individual blades is no longer required to be resolved. Second, the introduction of the momentum source to represent the motion of the blades circumvents the need to use either a rotating or an overlapping mesh strategy to capture the motion of the turbine rotor. These two factors have rendered the use of TPMs a computationally affordable alternative approach for the modelling of large-scale wind farms. Hence, the actuator disk model was used in previous¹⁻⁵ to model the wake field and predict the power output of operating offshore wind farms (eg, Lillgrund and Horns Rev) while others⁶⁻⁸ undertook AL simulations to solve for the wake field as well as to obtain statistics for blade loads. In all of these studies, either a uniform mesh or a block-mesh strategy was employed within the presented simulations. Such an approach requires *a priori* knowledge of the wake length and width, or otherwise, a large volume of the computational domain to be assigned as the refined “wake region.” Moreover, as more realistic simulations are required for utility-scale wind farms, including changes in the wind (and therefore wake) direction, the “refined wake regions” will need to be expanded to a greater extent in order to provide the required resolution of such simulations. Undoubtedly, this approach is not optimal for the discretization of the domain, as even a moderate expansion (in the order of a few decades of metres) of the wake region can significantly increase the number of degrees of freedom. Inherently, some sort of flexible mesh adaptivity procedure (eg, the dynamic mesh optimization approach used here) that is employed during the course of the simulation would be an attractive approach to consider for the above described problem. This particular need has already been expressed in previous studies. For instance, Churchfield et al⁸ mentions that “... adaptive mesh refinement would be useful in providing higher resolution only where necessary, but may incur a run-time penalty in performing the refinement and the processor load balance ...” In a similar note, Nilsson et al⁴ also used different resolution grids and pointed out that the simulations using the

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2018 The Authors Wind Energy Published by John Wiley & Sons Ltd.

most refined grids were abandoned because of limitations on the available computational resource.

Combining TPMs with mesh-optimization techniques was first considered by Creech et al,⁹ and more recent implementations can also be found in Abolghasemi et al,¹⁰ and Creech et al¹¹ each employing a different TPM (actuator volume and actuator disk and actuator line based, respectively). Mesh-adaptivity has also been used in conjunction with blade-resolved simulations by others¹²⁻¹⁴ who employed an unstructured overlapping mesh strategy to obtain the near-body solution and either unstructured or Cartesian mesh-adaptive solvers for the off-body flow. Such dual mesh approaches have been effectively used to calculate wind turbine and rotorcraft wakes, and it has been postulated that the computational efficiency of the approach is thanks to the off-body dynamic mesh-adaptivity solver. Indeed, a significant amount of computational resources can be saved by using an optimal local refinement or coarsening of the mesh while at the same time maintaining the desired levels of solution accuracy. In the context of large-scale wind farm simulations, mesh-adaptivity was used by Creech et al¹⁵ and Kirby et al¹⁶ to simulate the wakes developed within the Lillgrund offshore wind farm. Both studies demonstrated the ability of a mesh-adaptive solver (either unstructured or Cartesian-based) to be coupled with TPMs or overlapping mesh strategies, respectively, and to be used as a high-fidelity, multiscale wind farm modelling tool.

In the abovementioned mesh optimization/adaptivity algorithms and studies the obtained solutions have not been compared with traditional static mesh solutions using the same solver in order to provide a rigorous estimate of either the potential accuracy gains and/or reductions in computational effort. Such questions are important in shedding light on the efficiency and accuracy of mesh-adaptive solvers, and more specifically on their applicability to wind energy research. To partially address these questions, we present herein the implementation and validation of an ALM, which employs dynamic mesh optimization techniques. The optimization of the mesh is achieved through a strategy, which allows control over both the numerical error and mesh size at run time.¹⁷⁻¹⁹ Both the ALM and mesh-adaptivity approach are developed within the open-source code Fluidity,^{18,20} which is a general purpose unstructured mesh-based finite element solver. In addition, the fluid flow is modelled in this work using an unsteady Reynolds-averaged Navier-Stokes (uRANS)-based approach, combined with the *k- ω* shear-stress transport (SST) turbulence model.²¹ Before proceeding to the large-scale simulations, the accuracy of the new ALM implementation in Fluidity (combined with the uRANS configuration) is investigated through comparisons with a series of wind tunnel tests^{22,23} for the power and wake of one and two turbines operating in-line. The model shows very good agreement with the rotor's thrust and power coefficients predictions as well as the wake field. These comparisons give us confidence that the model can predict the wake characteristics with high accuracy when real-world scale wind farms are considered. To demonstrate the efficiency of the mesh optimization approach, we compare the results from an adaptive mesh simulation with those from a static pre-refined mesh simulation for the Lillgrund offshore wind farm and data from the literature.²⁴

The paper starts with Section 2, which introduces the mesh-adaptive fluid solver, and Section 3, which discusses the implementation of the turbine parametrization. In Section 4, the two mesh approaches (fixed vs adaptive) are validated against the experimental data of Krogstad and Eriksen²² and Pierella²³ while in Section 5 simulation of the Lillgrund offshore wind farm are undertaken. The mesh optimization techniques are presented from the point of view of the same solver (Fluidity), and its contribution to increasing accuracy and reducing computational cost is finally discussed thereafter in Section 6.

2 | MODEL IMPLEMENTATION

2.1 | Unsteady RANS formulation of the governing equations

The wind turbine wakes are modelled using the uRANS equations, in which the velocity is decomposed into mean and fluctuating (turbulent) components. Within uRANS, the governing equations take the form

$$\nabla \cdot \bar{\mathbf{u}} = 0, \tag{1}$$

$$\rho \frac{\partial \bar{\mathbf{u}}}{\partial t} + \rho \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} = -\nabla \bar{p} + \mu \nabla^2 \bar{\mathbf{u}} - \nabla \cdot \overline{(\rho \mathbf{u}' \otimes \mathbf{u}')} + \mathbf{F}_T, \tag{2}$$

where $\bar{\mathbf{u}}$ is the time-averaged component of velocity, \mathbf{u}' is the fluctuating velocity component, \bar{p} is the mean pressure, μ is the dynamic viscosity of the fluid, and \mathbf{F}_T is a momentum source term computed at each time step from the ALM. The term $-\nabla \cdot \overline{(\rho \mathbf{u}' \otimes \mathbf{u}')}$ is a residual term from the application of the time-averaging operator on the outer product of the fluctuating velocity components, called the Reynolds stress tensor $\bar{\bar{\tau}}_R$. The presence of the Reynolds stress tensor in (2) introduces a number of additional unknowns, and therefore, the Boussinesq approximation is adopted to provide closure to the system of equations. That is, the Reynolds stresses are related to the time-averaged turbulence kinetic energy $k = \frac{1}{2} \overline{\mathbf{u}' \cdot \mathbf{u}'}$ and strain-rate tensor via

$$\overline{-\rho \mathbf{u}' \otimes \mathbf{u}'} = \bar{\bar{\tau}}_R = -\frac{2}{3} k \rho \mathbf{I} + \mu_T (\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T), \tag{3}$$

where μ_T is the eddy viscosity and \mathbf{I} the unit tensor. To compute k , we make use of the standard *k- ω* SST model proposed by Menter,²¹ which requires the solution of two additional transport equations:

$$\rho \frac{\partial k}{\partial t} + \rho \mathbf{u} \cdot \nabla k = \nabla \cdot ((\mu + \mu_T \sigma_k) \nabla k) + \bar{P}_k - \rho \beta^* k \omega, \tag{4}$$

and

$$\rho \frac{\partial \omega}{\partial t} + \rho \mathbf{u} \cdot \nabla \omega = \nabla \cdot ((\mu + \mu_T \sigma_\omega) \nabla \omega) + \left(\frac{\rho \alpha}{\mu_T} \right) \bar{P}_k - \rho \beta \omega^2 + 2(1 - F_1) \rho \sigma_{\omega 2} \frac{1}{\omega} \nabla k \nabla \omega, \tag{5}$$

where

$$\mu_T = \rho \frac{k}{\omega}, \quad (6)$$

is the eddy viscosity and F_1 a blending function defined in Menter.²¹ The tilded quantity \tilde{P}_k denotes a limiting turbulence kinetic energy production given by

$$\tilde{P}_k = \min(P_k, 10\rho\beta^*\omega), \quad (7)$$

which is applied to prevent the build-up of turbulent energy in stagnation regions.²⁵ Finally, the closure coefficients σ_k , σ_ω , α , β , and β^* are selected by linear interpolation using the blending function value F_1 . Further information for the model implementation can be found in the appendix of Abolghasemi¹⁰ and the references therein.

2.2 | Numerical implementation and mesh optimization

The system of governing equations, including the additional scalar transport equations required for the turbulence modelling, has been discretized within the open-source code Fluidity.²⁰ Fluidity is a general purpose three-dimensional, unstructured mesh, finite element/control volume-based PDE solver^{17,19,26} with the ability to make use of optimization-based anisotropic mesh adaptivity. For our analysis, the continuity and momentum equations are discretized using mixed finite elements for which piecewise-linear discontinuous basis functions are used to represent velocity, while continuous piecewise-quadratic basis functions are used for pressure over tetrahedral elements (the so-called P1_{DG}-P2 element pair). This scheme is known to be a Ladyzhenskaya-Babuska-Brezzi stable combination²⁷ and to perform well for advection-dominated problems.²⁸ The formulation also uses a slope limiter²⁹ to ensure a robust solution for the velocity and pressure fields. The k - ω SST model makes use of a control volume-based discretization¹⁹ with flux limiters to help prevent oscillatory behaviour of the turbulent kinetic energy k . For time marching, the second-order accurate Crank-Nicolson scheme is used and is combined with an additional explicit subcycling approach for momentum advection.²⁰

The underlying unstructured tetrahedral mesh is also subject to optimization-based adaptivity algorithms, which are used in order to improve the quality of the mesh and provide higher or lower resolution at locations, which are identified by the solver. For instance, the introduction of the AL momentum source will create a requirement for a particular edge length over the assigned rotor volume. This is achieved here through the specification of a scalar field, which identifies the region that the rotating actuator lines will occupy during the simulation. At the same time, two additional fluid properties, the velocity vector field and the turbulence kinetic energy, (TKE) are also used to guide the mesh optimization process. This is achieved through the derivation of a metric tensor field. If we consider a single scalar field φ that we want to adapt our mesh to optimally resolve, we form a metric tensor, \mathcal{M}_φ , by first computing the Hessian, H_φ , of that scalar field and defining

$$\mathcal{M}_\varphi = \frac{1}{\epsilon_\varphi} |\bar{H}_\varphi|, \quad (8)$$

where ϵ_φ is a user-defined weight for field φ (ϵ_φ can in some sense be considered a requested error—a smaller value leading to a larger \mathcal{M}_φ and consequently a finer mesh). The Hessian encodes information about the curvature of the scalar field and includes both spatial and direction information; we desire finer mesh resolution at location and in directions where curvature is high, and coarser resolution where the solution is close to linear. $|\bar{H}_\varphi|$ indicates that we are interested in the magnitudes of curvatures when deciding on optimal mesh resolution and do not care about the sign. Once metric tensor fields have been calculated for all the scalar fields (for the velocity, vector we consider each component separately), we wish to adapt to, a final metric is obtained through superimposition of the individual metrics.³⁰ At this stage, additional constraints on the total number of elements in the calculation, and/or maximum and minimum edge length, and the maximum rate at which edge lengths can vary in space can all be incorporated. The metric can then be used to measure the length of vectors—primarily the edge lengths of elements. A perfectly optimized mesh in physical space is defined as one, which is made up of unit length edges in metric space. The inhomogeneous and directionally dependent nature of the Hessians, and hence the metrics, thus leads to a mesh which is variable in both space and direction (ie, is potentially anisotropic). Subsequently, an optimization functional is defined which measures how well the current mesh achieves this goal, and a series of topological operations are performed on the current mesh to improve this agreement. These operations include edge collapse and splitting, and face to face and face to edge swapping.^{19,30} Finally, conservative mesh to mesh interpolation is used to transfer solution data from the old to the new mesh.^{31,32} The entire mesh optimization process is conducted every user-defined number of time period, termed the adaptation period T_{adapt} . Further information on the mesh optimization process, including its parallelisation, may be found in other works^{17-19,30,33} and the references therein.

3 | TURBINE PARAMETRIZATION

Our ALM follows the standard approach of Sørensen and Shen³⁴ and Troldborg et al³⁵ in which the turbine blades are represented by rotating virtual lines—the actuator lines (ALs). Point forces are computed along each AL, (at each blade element's midpoint) using the relative velocity \mathbf{U} extracted from the fluid solver by evaluating the globally-defined finite element solution at these points, the solid body velocity \mathbf{U}_b of each point, and the lift and drag coefficients obtained from look-up tables using airfoil data for the blade element's respective profile. Extra care has been taken for the elements near the blade tip for which the tip loss correction model of Shen et al³⁶ is used. The tower behind the turbine is also incorporated by

adopting a model similar to Sarlak et al³⁷—the tower is represented by an actuator line with element having a cylindrical cross section of constant drag coefficient $C_D = 1$ and a time-dependent lift coefficient C_L , which is “tuned” in order to reproduce the Von Kármán street behind the cylinder,

$$C_L = A \sin(2\pi ft), \tag{9}$$

where A is the amplitude of the “dynamic lift force” and $f = 0.2 \times U_\infty/D_{tower}$ is the Strouhal number, based on the uniform velocity and the diameter of the tower D_{tower} . For all simulations presented in this work, the hub and nacelle of the turbines are not modelled. The final ALM forces are projected onto the fluid mesh and represented in the governing equations via the momentum source term \mathbf{F}_T . To ensure a smooth transition from a concentrated point AL force \mathbf{f}_{AL} to the source term \mathbf{F}_T , a smoothing interpolation function η_ϵ is used.

$$\eta_\epsilon = \frac{1}{\epsilon^3 \pi^{3/2}} \exp\left(-\frac{|r|^2}{\epsilon^2}\right), \tag{10}$$

where $|r|$ is the distance of the mesh point from the AL node and ϵ is a smoothing parameter selected after taking into account the mesh size, drag force C_D , and chord size c , and V_{elem} is the volume of the element in which the AL node lies,

$$\epsilon = \max\left[\frac{c}{4}, 4\sqrt[3]{V_{elem}}, \frac{cC_D}{2}\right], \tag{11}$$

consistent with the recommendations of Martínez-Tossas et al.³⁸

4 | MODEL VALIDATION

The newly implemented ALM is validated using data from a series of “Blind Test” workshops organized by NoWiTech and NoCOWE and that was obtained from the wind tunnel facilities of the Norwegian University of Science and Technology (NTNU),^{22,23} We will refer to the two papers that reported the data as “blind tests” or when mentioned individually as “blind test 1” (BT1) and “blind test 2” (BT2). The wind tunnel facility used for the two blind tests is 11.15 m long, 2.72 m wide, and 1.8 m high. In (BT1) a single turbine with rotor diameter $D = 0.894$ m, hub height equal to $H_{hub} = 0.817$ m is placed at a distance of $2D$ from the wind tunnel inlet. The turbine has three blades consisting of 14% NREL S826 airfoils, a tower (support structure) with diameter $D_{tower} = 0.11$ m and is designed to operate optimally for a tip speed ratio of $\lambda = 6$. For BT1, power and thrust coefficients are reported by Krogstad and Eriksen²² for a large range of tip speed ratios (5-11.5) while wake statistics including the stream-wise velocity deficit and the turbulence kinetic energy for $\lambda = 6$. On the other hand, BT2 involves wake predictions from two similar turbines (with slightly different diameters $D_1 = 0.944$ m and $D_2 = 0.894$ m

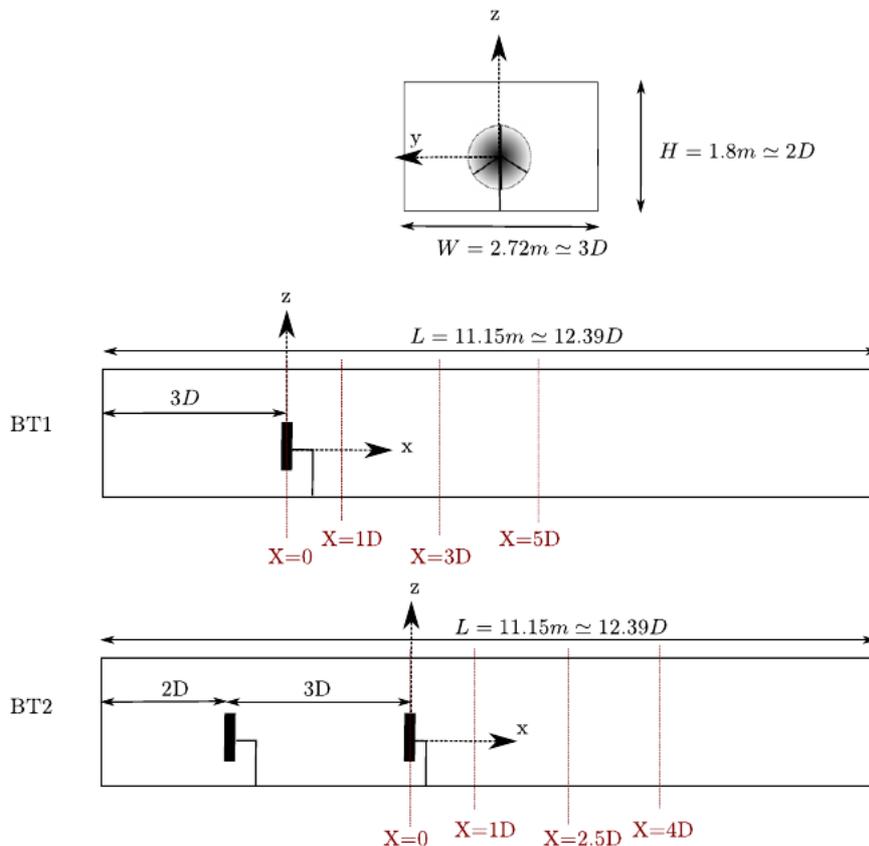


FIGURE 1 Schematic representation from the two “Blind Tests” set ups [Colour figure can be viewed at wileyonlinelibrary.com]

where the subscripts 1 and 2 correspond to the “Front” and the “Rear” turbines, respectively) operating in-line. The front turbine is located at $2D_2 = 1.788$ m from the tunnel's entrance while the rear is placed at a distance $3D_2 = 2.686$ m from the first as shown in Figure 1. BT2 reports three operational scenarios, in which the front turbine operates with $\lambda_1 = \Omega R/U_\infty = 6$, and the rear with $\lambda_2 = 4, 7$, and 2.5 . In both BT1 and BT2, the mean free-stream velocity was measured to be 10 m s^{-1} whereas the vertical profile of the ambient turbulence intensity found to be nearly uniform with a value of $I \approx 0.3\%$. For our computations, we also assume a uniform velocity and TKE profiles with initial and inlet conditions $U = 10 \text{ m s}^{-1}$ and $k = 1.31 \cdot 10^{-3} \text{ m}^2/\text{s}^2$, respectively, while the turbulent frequency is also considered to be uniform and equal to $\omega = 0.5 \text{ s}^{-1}$. Finally, the tower model is enabled for all the wind tunnel simulations with the lift coefficient given a value of $A = 0.3$, consistent with the study of Sarlak et al.³⁷

4.1 | Mesh convergence study

The design power coefficient from the single turbine experiments (BT1) is used as a representative quantity for our mesh size and time step convergence studies. For the mesh convergence study, we consider six set-up cases by assuming two mesh types (uniform fixed and adaptive) and three minimum element edge lengths $h = 0.1 \text{ m}, 0.075 \text{ m}$, and 0.05 m , as shown in Table 1, while the same time step $\Delta t = 0.005 \text{ s}$ is used for all simulations. We should note here that the adaptive simulations are also restricted by a maximum edge length of 0.5 m . Additionally, all adaptive simulations were conducted using an adaptation period $T_{\text{adapt}} = 2.5 \text{ s}$. On the other hand, the time step convergence study considers three time step sizes $\Delta t = 0.02; 0.01; 0.005$, corresponding to approximately 25, 50, and 100 time steps per rotor revolution, and using the finest mesh ($h = 0.05 \text{ m}$) for both the fixed and adaptive mesh. The lift and drag coefficients for the 14% NREL S826 section are taken from Sarlak³⁹ and include data spanning a Reynolds number from 4×10^4 to 4×10^6 and angles of attacks from -10 to 25 as shown in Figure 2. The turbine performance and thrust coefficients are computed based on the uniform upstream velocity U_∞ and the nominal radius $R = D/2$ via

$$C_T = \frac{T}{0.5 \rho \pi R^2 U_\infty^2}, \quad (12)$$

$$C_P = \frac{P}{0.5 \rho \pi R^2 U_\infty^3}. \quad (13)$$

To obtain mean values for the power and thrust coefficients, we run long enough simulations and allow the turbines to undergo more than 100 revolutions. In Figure 3, the relative error in the predicted power coefficient is plotted against the average number of elements used in the simulations. We observe that for a given minimum element edge length, the two approaches (fixed and adaptive mesh) yield very similar relative errors. However, the adaptive mesh method required a smaller number of elements due to the optimum and flexible use of the underlying mesh within the computational domain. Therefore, within the ALM/uRANS configuration considered here, the minimum edge length in the mesh can be considered the primary factor determining the accuracy of the model, while adaptivity is used primarily to reduce the overall number of degrees of freedom, and therefore the associated computational cost. Later, we will further demonstrate the potential advantages of mesh-adaptivity by undertaking

TABLE 1 Table of simulation set-up cases

Case	Min. Mesh size h [m]	Average. Num. Elem.	Mesh type	Tag
01	0.1	234 098	Fixed	FC
02	0.075	561 611	Fixed	FM
03	0.05	1 699 635	Fixed	FF
04	0.1	169 939	Adaptive	AC
05	0.075	351 688	Adaptive	AM
06	0.05	698 236	Adaptive	AF

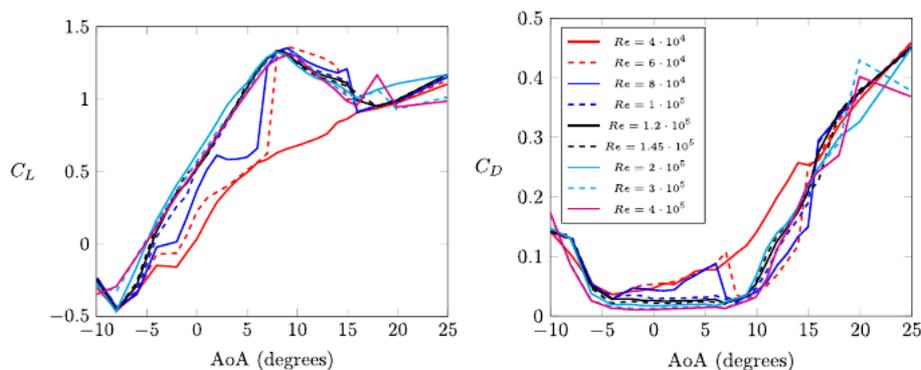


FIGURE 2 Lift and drag coefficients as a function of the angle of attack (AoA)³⁹ used for the model validation [Colour figure can be viewed at wileyonlinelibrary.com]

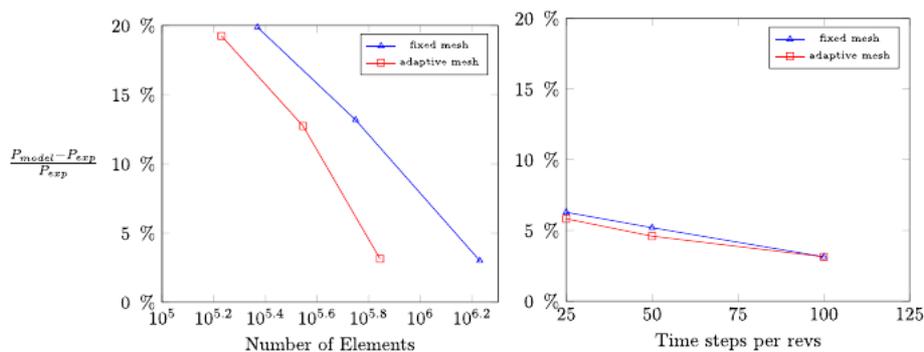


FIGURE 3 Convergence study with respect to spatial and temporal resolution [Colour figure can be viewed at wileyonlinelibrary.com]

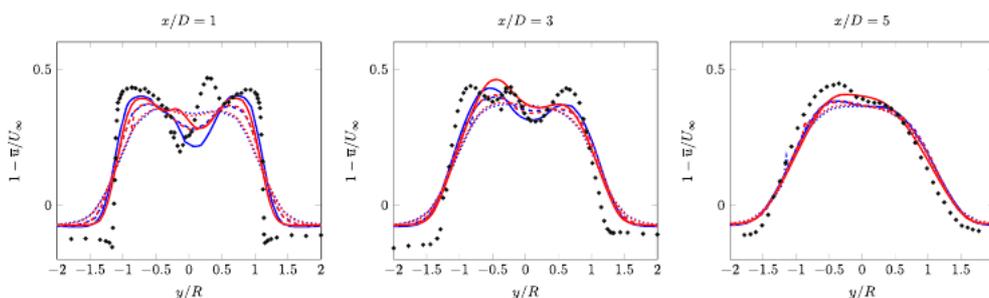


FIGURE 4 Blind Test 1: Horizontal mean stream-wise velocity profiles at $x/D = 1, 3,$ and 5 . The plotted lines correspond to: fixed coarse (FC) (.....), fixed medium (FM) (- - -), fixed fine (FF) (—), adaptive coarse (AC) (.....), adaptive medium (AM) (- - -), adaptive fine (AF) (—), and the symbols (●) to the experimental values reported by Krogstad and Eriksen²² [Colour figure can be viewed at wileyonlinelibrary.com]

large-scale wind farm simulations. In that case, the substantially fewer elements required during the spin-up period of the wake development leads to an important saving in CPU time. On the other hand, the temporal convergence study (right-hand side of Figure 3) shows that as we reduce the time step (and thus increase the number of time steps used during one rotor revolution), the accuracy of the model converges to the element-based maximum obtainable accuracy. Again, the plotted figures correspond to simulations using the minimum edge length $h = 0.05$ m. Therefore, the element edge length and the time step used in all simulations presented hereafter will be based upon this preliminary convergence study. We should also note that once we have selected an element edge length for our analysis, the magnitude requirement on the time step is entirely dictated by the ALM and not the stability of the fluid solver, which allows for a far more relaxed condition assuming a Courant number of near unity.

Lastly, the smoothing parameter ϵ varies with the element's edge length and for all simulations hereafter is taken equal to 2.5 times the edge length, which also satisfies Equation 11.

4.2 | Wind tunnel tests

Having established that the element edge length determines the accuracy of our model, we will present the wake predictions from BT1 for all six set-up cases described in Table 1 using the smallest time step $\Delta t = 0.005$ s, while results for the power and thrust coefficients as well as the wake predictions of BT2 are presented only the fine mesh-adaptive, in order to maintain clarity. Starting with the wake profiles, we present three horizontal profiles downstream of the single turbine at $x/D = 1, 3,$ and 5 for BT1 and the three horizontal profiles downstream of the rear turbine at $x/D = 1, 2.5,$ and 4 for only the first scenario ($\lambda_2 = 4$) from BT2. Figures 4 and 5 show the mean stream-wise velocity and the TKE for BT1, while Figures 6 and 7 show the mean stream-wise velocity and the stream-wise turbulent stresses $\overline{u'u'}$ for BT2. All quantities are time-averaged after a spin-up period, which is taken to be approximately 1 s and have been normalized by the upstream velocity U_∞ and presented as a function of the normalized horizontal distance y/R .

Similarly, to resolve the wake field in BT2, only the refined adaptive case was used, and for brevity, we present the wake predictions only from scenario 1. We should also note that in order to obtain the Reynolds stress $\overline{u'u'}$, we make use of the isotropy turbulence relation $k = 3\overline{u'u'}/2$. Such an assumption is not appropriate when the turbulence stresses are highly anisotropic, which may explain the large discrepancies in Figure 7 for $x/D = 1$. However, better estimates are obtained for the other two downstream profiles. Flow anisotropy was also found to affect the predictions of the wake for BT1. This is an inherent inability of all turbulence models, particularly for the estimation of TKE.⁴⁰

Looking at integrated rotor quantities such as the power and thrust coefficients, an overall good agreement is observed between the ALM predictions and the wind tunnel measurements. Results are shown for both BT1 and BT2 in Figure 8. The only large discrepancy that can be observed in Figure 8 is for scenario 2 of BT2 (rear turbine operating with $\lambda = 7$). In that case, the power coefficient exhibits a discrepancy of 457.14% in comparison with the measurements, which can be attributed primarily to the ALM limitation rather than the modelling of the wake. Indeed,

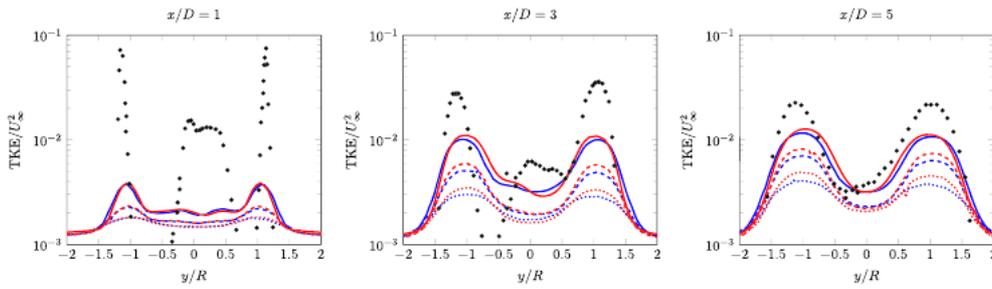


FIGURE 5 Blind Test 1: Horizontal TKE profiles at $x/D = 1, 3,$ and 5 . Line colouring the same as in Figure 4 [Colour figure can be viewed at wileyonlinelibrary.com]

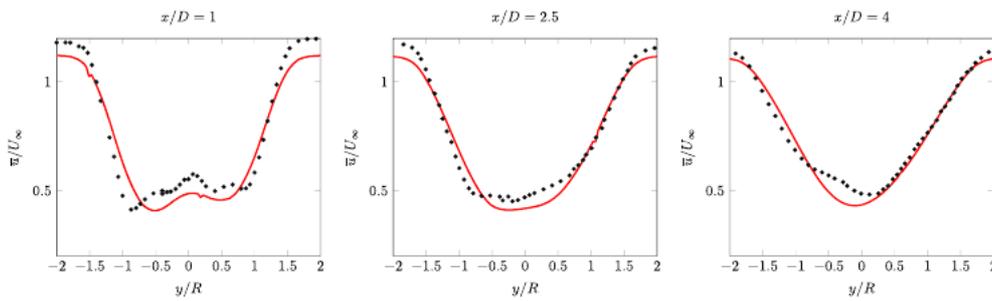


FIGURE 6 Blind Test 2: Horizontal mean stream-wise velocity profiles at $x/D = 1, 2.5,$ and 4 . Only the adaptive fine (—) solution and the experimental data (●) are shown [Colour figure can be viewed at wileyonlinelibrary.com]

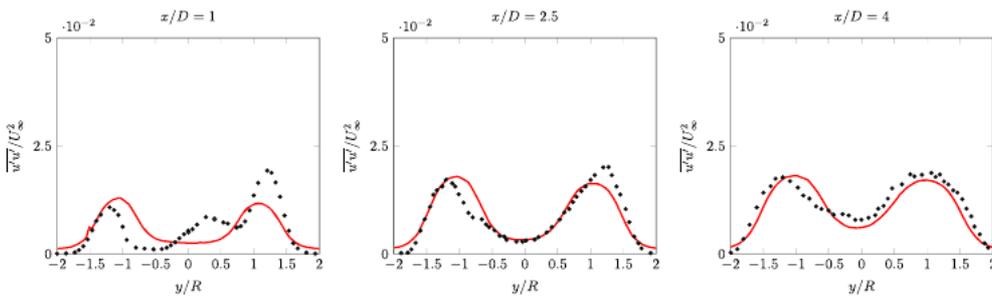


FIGURE 7 Blind Test 2: Stream-wise turbulent stress profiles at $x/D = 1, 2.5,$ and 4 . Only the adaptive fine (—) solution and the experimental data (●) are shown [Colour figure can be viewed at wileyonlinelibrary.com]

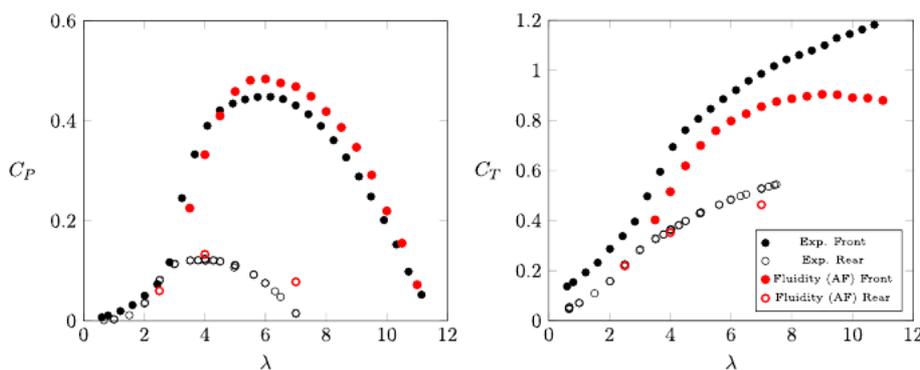


FIGURE 8 Power and thrust coefficient curves as computed by the present model using the adaptive fine mesh and as measured by Krogstad and Eriksen²² and Pierella et al²³ plotted against the tip speed ratio λ [Colour figure can be viewed at wileyonlinelibrary.com]

Pierella et al²³ reported this scenario as the most challenging one for TPMs. This is due to a non-uniform span-wise pressure experienced by the blades of the rear turbine, necessitating the use of blade-resolved simulations to accurately predict the lift and drag coefficients of the individual blade elements.

Still, the results from both BT1 and BT2 give us confidence that the proposed adaptive methodology is both faster and more accurate than the fixed mesh one. However, applying mesh adaptivity in laboratory scale (wind tunnel) experiments cannot demonstrate its full potential. This is due

to the high blockage created in the wind tunnel which necessitates the use of the finest mesh resolution almost everywhere in the domain. To better demonstrate the potential benefits of mesh adaptivity, we present in the next sections simulations for the Lillgrund offshore wind farm and discuss some options that can further reduce the computational time.

5 | THE LILLGRUND OFFSHORE WIND FARM

5.1 | Parametrization of the wind farm

The Lillgrund offshore wind farm is located near the southern coast of Sweden and has recently attracted the interest of the research community as a benchmark for numerical model validation.^{4,8,15,41} Model validation is achieved through comparison with supervisory control and data acquisition (SCADA) system measurements and it primarily tests the ability of the model to predict power losses along a row of turbines for different wind directions. Part of the SCADA data are accessible via an online technical report²⁴ while the complete information used for the validation can be extracted from the validation studies just mentioned, and in particular from the studies of Nilsson et al.⁴ and Creech et al.¹⁵ The wind farm consists of 48 Siemens SWT93-2.3 MW (Siemens Wind Power, Hamburg, Germany) distributed in eight rows (A-H) as shown in Figure 9. For our simulations, we consider the Southwestern statistically dominant wind direction ($43^\circ / 222^\circ$), and we align the coordinate axis system (x - y) with the rows A, B, etc resulting in a layout where the aligned row turbines have a 4.3 rotor diameter spacing.

To parametrise the turbines making up the wind farm, we need to make a number of assumptions as their exact blade geometry and airfoil characteristics are not publicly available. The available information does include the turbines' rotor radius $R = 46.5$ m, the hub height $H_{\text{hub}} = 65$ m, and the thrust and power coefficients as a function of wind speed.²⁴ For the rest of the turbine parameters, we adopt the approach of Nilsson et al.⁴ who considered a downscaled version of the conceptual NREL 5 MW turbine as presented by Jonkman et al.⁴² and confirmed its suitability by comparing the thrust and power (through torque) output for different wind speeds. Here, however, we consider an up-scaled version of the "Blind Test" turbine, scaled up by approximately a factor of 100. To confirm the validity of our choice, we present in Figure 10 the thrust and power coefficients for velocities varying from 5 m/s to 11 m/s, similar with.⁴ So far in Section 4.2, we have expressed both the thrust and power coefficient as a function of the tip speed ratio λ . As an active control strategy, we assume that the turbine will simply adjust its angular velocity in accordance with the optimum tip speed ratio, which was found to be around $\lambda = 6$ as shown in Figure 8. This assumption is justified in part by the good agreement between the up-scaled "Blind Test" turbine and the Siemens SWT-93 manufacturer's curve shown in Figure 10. Here, it is worth emphasizing that a recent investigation by Deskos et al.⁴³ showed that when wake predictions are made the need for mesh resolution is primarily driven by the velocity deficit, which inherently relates to the thrust coefficient. Therefore any turbine parametrization that accurately captures the thrust force would result in a similar computational mesh.

5.2 | Mesh strategies

For the simulations presented here, two mesh strategies are investigated, namely, the fixed (but pre-refined, ie, variable resolution) and fully dynamically adaptive meshing. Considering that the problem at hand is inherently multi-scale and a number of length scales, spanning from the individual turbine wake's turbulence length scale to the far larger atmospheric mesoscales usually in the order of $\mathcal{O}(200$ m), need to be resolved, the two selected meshing strategies are the following:

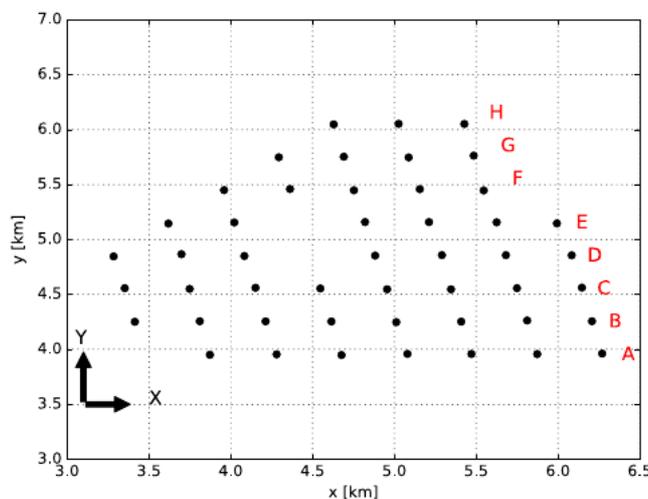


FIGURE 9 Layout of the Lillgrund offshore wind farm. Rows A, B, etc are aligned with the x -axis of the coordinate system [Colour figure can be viewed at wileyonlinelibrary.com]

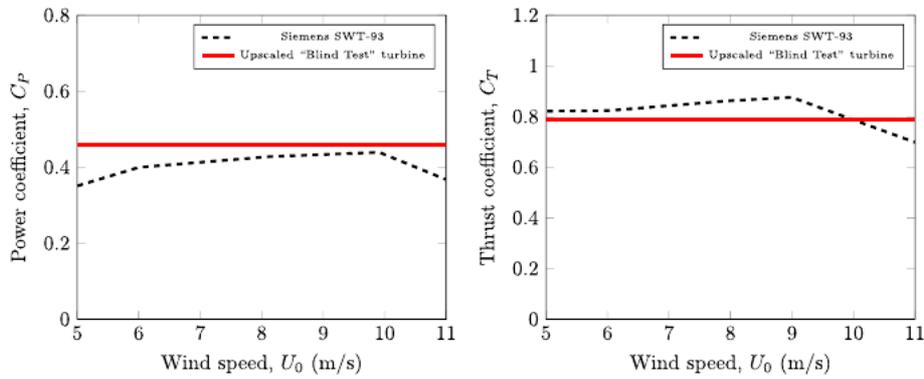


FIGURE 10 Comparison of between the real power and thrust coefficient (C_p and C_T) and the up-scaled “Blind Test” turbine as a function of the upstream wind speed [Colour figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com)]

1. a large domain of dimensions $10 \text{ km} \times 10 \text{ km} \times 1 \text{ km}$ is considered with a mesoscale resolution $\mathcal{O}(200 \text{ m})$ everywhere except for an inner region with prescribed resolution using an edge length equal to $h = 20 \text{ m}$ (in the subsequent discussions this is referred to as the pre-refined region [PRR]), and
2. a large domain of the abovementioned dimensions and mesoscale initial resolution everywhere without the *a priori* chosen inner region refinement, but with mesh-adaptivity enabled and operating over the whole domain and a minimum allowed edge length of $h = 20 \text{ m}$ specified.

The simulations were performed for a case with a PRR assigned in an area of $4 \text{ km} \times 6.5 \text{ km} \times 0.16 \text{ km}$ (see Figure 11), and three adaptive simulations in which different frequencies of mesh adaptations ($T_{\text{adapt}} = 2.5 \text{ s}, 5 \text{ s}, \text{ and } 10 \text{ s}$) were applied. The selection of the PRR was performed through a “trial and error” approach as the width and the height of the individual turbine wakes were not known *a priori*. Therefore, our goal was to create a domain which has the refined region covering the wake region but does not extend excessively beyond this point. An alternative approach (which would arguably represent a fairer comparison to the adaptive approach) would be to assume no *a priori* knowledge and simply use the minimum edge length over the entire domain, but this would of course result in a huge problem size. For the mesh adaptive simulations, the adaptation period is selected from a larger number of user-defined parameters available to us such as the elements' maximum aspect ratio, the minimum/maximum element ratio, etc. This option is considered in order to understand its impact on reducing the overall CPU time in the present uRANS set-up. The selection of mesh adaptivity frequency as a key parameter is motivated by the computational cost (runtime penalty) associated with it, particularly during the interpolation process. Indeed, mesh adaptations can be seen as a potential bottleneck step in the computations, as load-rebalancing represents a data migration overhead and the application of relatively costly interpolation methods to be used (eg, Galerkin projection) to transfer the information from the previous mesh to the new one. The application of a Galerkin projection method is necessitated here by the use of a discontinuous function space, as the recent study of Farrell and Maddison³² showed that a consistent interpolation is not suitable for discontinuous fields. For the present simulations, we consider the wind speed to be constant, and therefore, the required frequency of the mesh adaptation is not expected to vary with time, except for when the wakes have converged. However, if longer simulations are to be undertaken in which the wind speed is expected to vary with time, the frequency of the adaptations may be adjusted during the course of the simulation to better capture the wake dynamics. It should be noted of course that a varying wind direction would also necessitate a larger PRR in the fixed mesh case, leading to substantial additional computational costs. Finally, other parameters such as the element edge length size h have been excluded as in Section 4, we showed that there is a direct correlation between the size of the underlying mesh and the model's accuracy. Thus, changing the frequency with which these adaptations occur is a key remaining parameter, which can be varied in order to reduce the overall CPU time while maintaining a similar accuracy for the turbine performance and wake predictions. The latter will also be reassessed later in this section.

5.3 | Simulation setup

Moving on to the simulation set-up, for the purpose of these simulations, we assume uniform initial and inlet mean velocity and TKE profiles ($U_0 = 8 \text{ m s}^{-1}$ and $k = 0.31 \text{ m}^2/\text{s}^2$, which corresponds to a turbulence intensity $I = 5.7\%$). On all other boundaries, we apply free-slip velocity and zero gradients for all other quantities ($\partial/\partial n(u, v, w, k, \omega) = 0$). The selected inlet and boundary conditions considered herein are not representative of the levels of shear that the turbine wakes will experience within a realistic atmospheric boundary layer. However, as a first approach, we will demonstrate the benefits of mesh adaptivity by imposing a uniform incident velocity and TKE profile and therefore based on the selected adaptation metrics (velocity and TKE) not need to refine outside the wake region. The accuracy of our choice is also discussed later, when the numerical results are compared with the observed SCADA data and some discrepancies are observed. It should be remembered that the focus of this study is to compare the PRR solutions with the mesh-adaptive ones, and the observed data is added only to show the overall performance of the model. Future studies will consider more complex scenarios in which the mesh will also be further controlled in a manner that allows for anisotropic gradation in the vertical direction only, outside the wake region. Nevertheless, to obtain a quasi-steady solution for the far wake field we time-march the solution with a constant time step of $\Delta t = 0.1 \text{ s}$ and the solution for a total number of 4000 time steps. The final wake solutions are shown in Figures 12 and 13, for

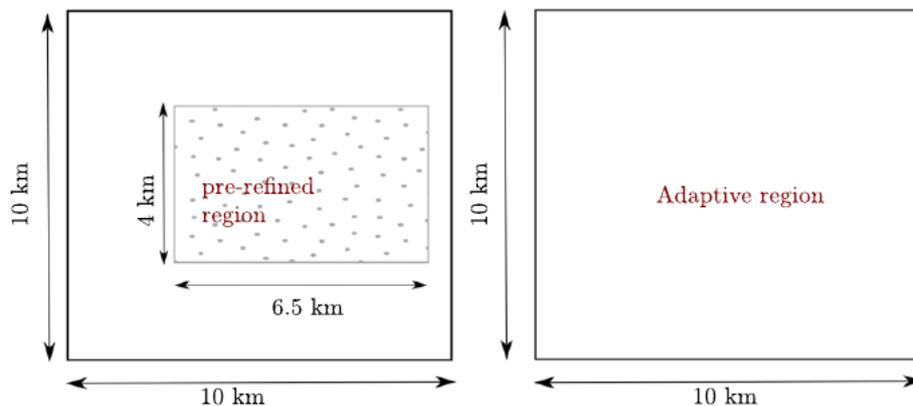


FIGURE 11 Schematic representation of the two mesh strategies considered for the Lillgrund offshore wind farm test case. The pre-refined region is centred around the middle of the y-axis while it starts 3 km from the domain inlet and ends 0.5 km before its outlet [Colour figure can be viewed at wileyonlinelibrary.com]

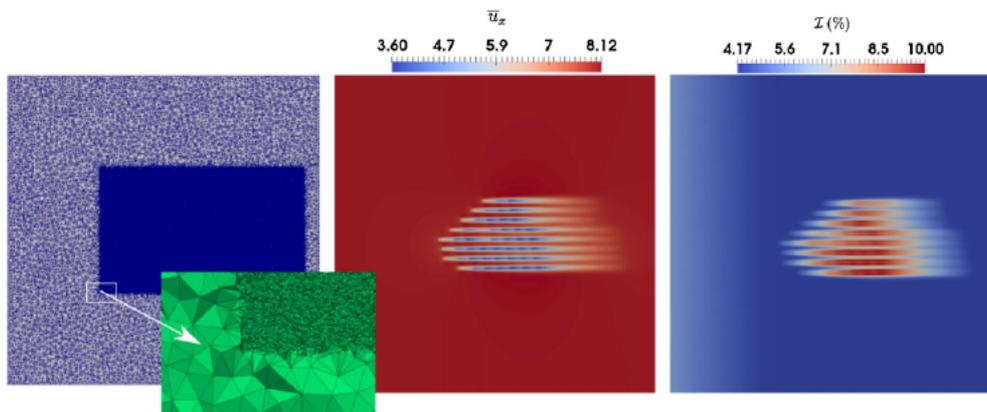


FIGURE 12 Fixed mesh simulations: From left to right a horizontal cross-section at hub height of the underlying mesh, the stream-wise velocity u_x and the turbulent intensity TI are shown [Colour figure can be viewed at wileyonlinelibrary.com]

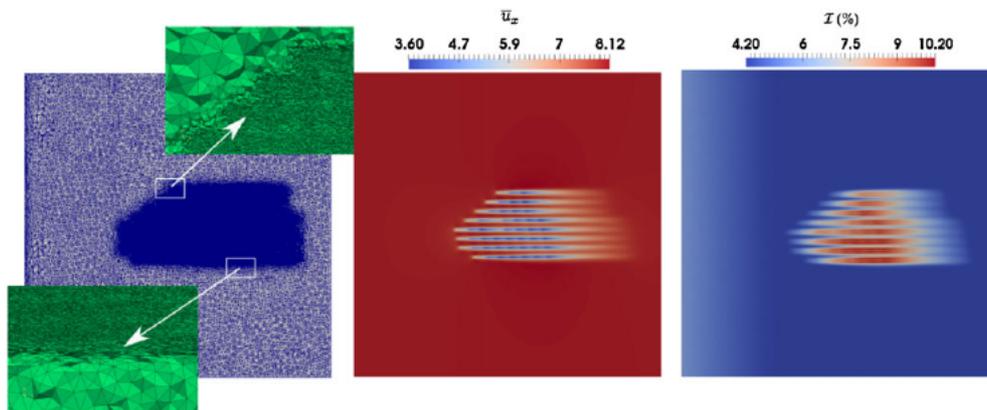


FIGURE 13 Adaptive mesh simulations ($T_{\text{adapt}} = 2.5$ s): From left to right a horizontal cross section at hub height of the underlying mesh, the stream-wise velocity u_x and the turbulent intensity TI are shown [Colour figure can be viewed at wileyonlinelibrary.com]

both the fixed and adaptive mesh (using the smallest adaptation period $T_{\text{adapt}} = 2.5$ s) simulations. In addition, the power estimates normalized by the median of the production of the front row turbines P_{mid} (similar to quantities considered by Nilsson et al⁴) are presented and compared against the measured SCADA data in Figure 14.

The two approaches' (adaptive and fixed mesh) results exhibit very similar behaviour both qualitatively and quantitatively as observed in Figures 12 and 13. An important feature in both solutions is the dissipation of the inlet TKE due to the assignment of a turbulence frequency required for the dissipation of the turbine wakes. The increased dissipation creates a moving front for the inlet TKE which however disappears after about 2000 time steps. This impacted on the adaptive mesh since the TKE field was used within the definition of the metric controlling mesh optimization. The impact of the TKE inlet front is discussed in more detail in the next subsection. Next, the power production trends are also found to agree

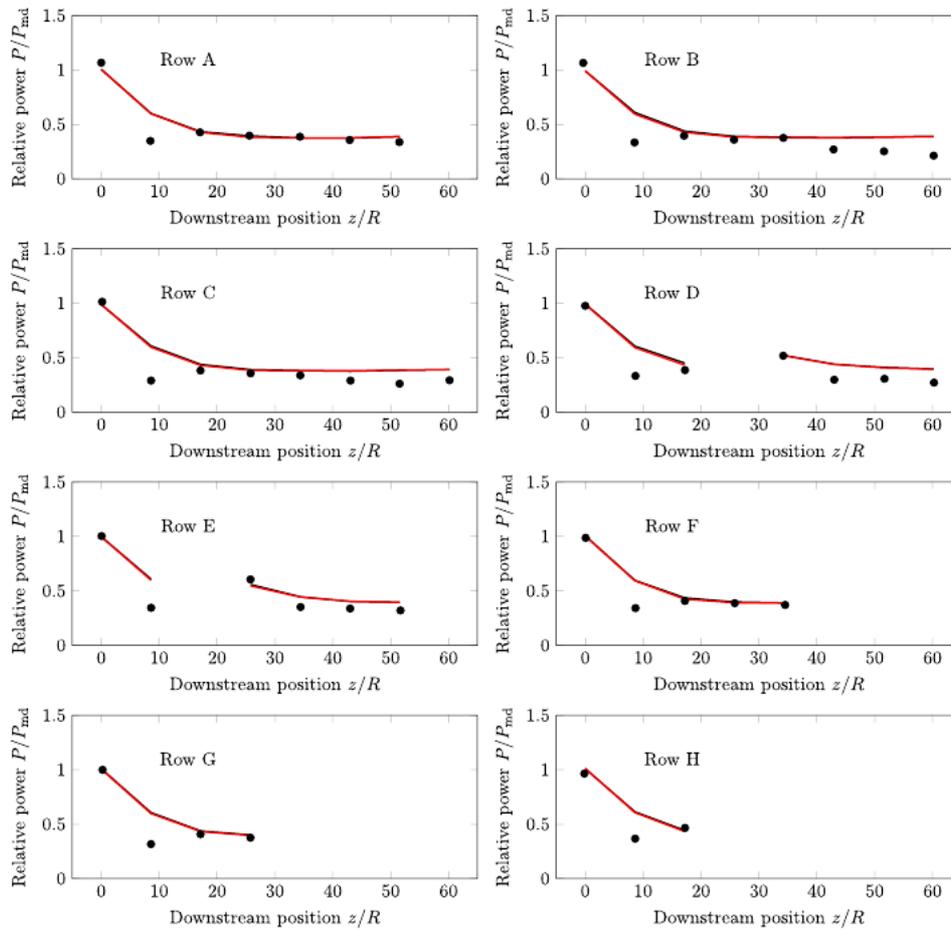


FIGURE 14 Normalized power computed by the fixed-mesh (black line) and the mesh-adaptive (red line) simulations for turbines in Row A-H and compared against the measured data (dots) of Dahlberg²⁴ [Colour figure can be viewed at wileyonlinelibrary.com]

well with measurements as shown in Figure 14. The only major discrepancy observed in the plots (rows A-H) is the over-prediction of the relative power of the second turbine. This discrepancy may be attributed to our selection of uniform inflow conditions, as the amount of computed shear and wake asymmetry will be underestimated. Similar trends were also observed by others^{4,8,15} who adopted a log-law profile, although the discrepancies between the experimental values and their LES results appear to be much smaller. Nevertheless, the power production as predicted by the three adaptive simulations at the final time level are essentially identical and therefore only one of the three is plotted in Figure 14. The mean relative error over all turbines between the CFD simulations' power prediction and the measured data amounts to 7.38% for all three adaptive mesh simulations while it takes the value 8.02% for the fixed-mesh ones. This result confirms our initial hypothesis that varying the frequency of mesh adaptations will not affect the accuracy of the model predictions in the case of the uRANS approach employed here. On the other hand, a small difference between the fixed-mesh and the adaptive mesh simulations' power predictions is observed; this can be attributed to the ability of the adaptive simulations to better resolve the wake field by applying an optimum element aspect ratio, as was also shown for the wind tunnel tests.

5.4 | Computational efficiency

Returning to a key objective of our investigation, which is to examine the computational efficiency of the two proposed mesh strategies, it was hypothesized that the computational cost of the adaptive simulations should be lower than that for the fixed pre-refined mesh simulations, as well as the computational cost becoming smaller as the adaptation period is increased. In addition, the same or similar accuracy should be achieved when the same minimum edge length is used. To test our hypothesis, we conducted simulations using one fixed and three adaptive meshes. The simulations were run in parallel using MPI on a total of 80 processing cores (four nodes each of 20 cores) on the cx1 cluster at Imperial College London. The pre-refined simulations required approximately 86 hours to complete which is considered as a reference value. We should mention here that the absolute value of the required CPU time would differ for different algorithms or code implementations. For this reason, we present the CPU times for the adaptive mesh simulations normalized by the CPU time for the pre-refined mesh case. Here, the term "CPU time" refers to the wall-clock time multiplied by the number of processing cores. This CPU time is dominated by the actual run time of the jobs and auxiliary computational procedures such as the decomposition and re-partition of the mesh were found to result in trivial computational times in comparison. With this in mind, we start by presenting the temporal evolution of the number of fluid mesh elements in Figure 15; the number of elements are seen to follow a similar trend for all three adaptive cases.

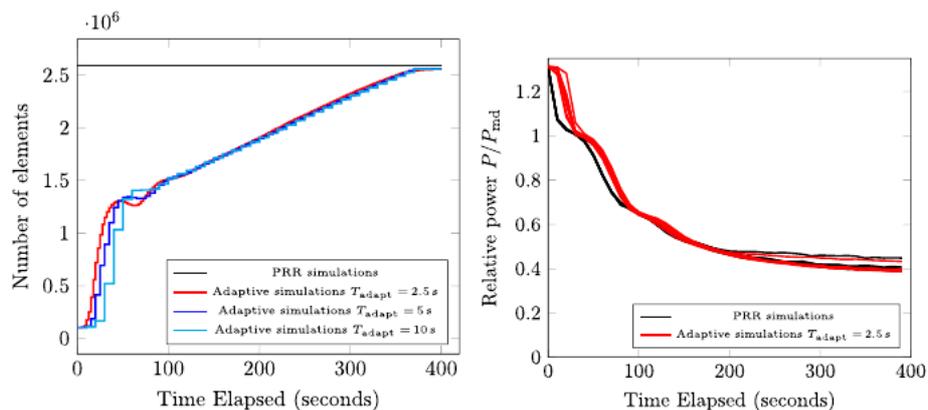


FIGURE 15 Left: Number of tetrahedral elements used for the underlying mesh as a function of the elapsed simulation time, Right: Convergence of the relative power of the rear row of turbines—individual lines for each turbine are plotted here [Colour figure can be viewed at wileyonlinelibrary.com]

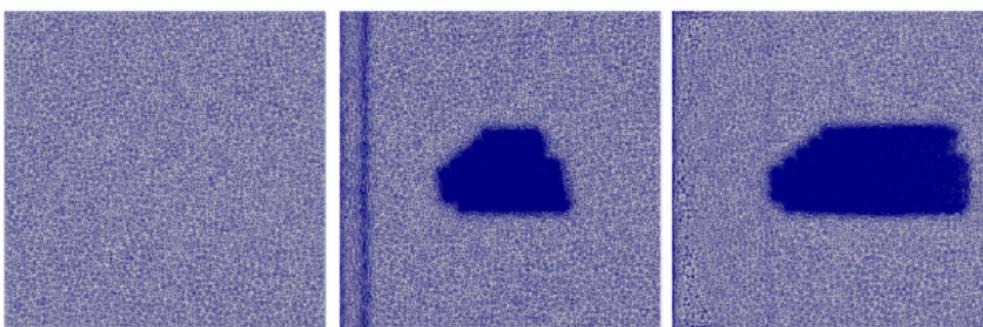


FIGURE 16 From left to right: A horizontal slice at hub height ($H = 65$ m) through the adaptive mesh at 0 s, 100 s, and 400 s, and having in total 100 760, 1 649 459, and 2 558 880 elements in the computational domain, respectively [Colour figure can be viewed at wileyonlinelibrary.com]

The simulations begin with a spin-up of the mesh in the vicinity of the turbines and a moving front starting from the inlet. The latter, is a result of the dissipation of the turbulent kinetic energy in the inlet and disappears after about 50 s. After this spin-up period, and between 50 s to 100 s, the number of elements are slightly reduced due to the “dissipation” of the moving front, after which the element counts start to increase again with a constant rate after 150 s. Finally, the number of elements becomes constant after approximately 380 s at which time a steady state solution has been reached for the far field. A similar trend for the count of elements/degrees of freedom was also obtained by Kirby et al.¹⁶ In their simulations of the full Lillgrund wind farm, they observed a strong linear spin-up curve due to wake transients, and subsequently a flattened peak where the wakes have started interacting with each other, and small variations in the mesh due to wake-wake and turbine-wake interactions were found. Such fluctuations are not observed however in the dynamic mesh evolution of the present simulations, which can be attributed to the low-pass temporal filtering incurred by the uRANS equations. On the right hand side of the same figure, we have plotted the evolution of the relative power (P/P_{md}) for all the back-row turbines against the elapsed time, for one mesh-adaptive case ($T_{adapt} = 2.5$ s) and the PRR simulation. It can be observed that the relative power of the individual turbines converges to near its final value long before 380 s. The evolution of the adaptive mesh is presented via horizontal and vertical slices through the domain in Figures 16 and 17. Note that slices through completely unstructured tetrahedral meshes do result in complex polygons and slivers, but these images do convey the spin up of the mesh as it resolves the developing wakes. Zoomed pictures of the vertical profiles are also shown in Figure 17 to demonstrate the element gradation (particularly the higher aspect ratio of the elements) near the edge of the wake field.

Element-wise, the plots (Figure 15) of the three element counts for the adaptive mesh simulations remain well below that for the pre-refined simulation line until the end of the simulations, and thus we may argue that since the adaptive mesh simulations always use a smaller number of elements than the fixed-mesh ones, and that the Fluidity solver scales approximately linearly with the number of elements, then the adaptive mesh simulations should require a much lower CPU time. For the adaptive mesh simulations, however, additional CPU “penalties” are imposed due to the various stages of the mesh adaptation procedure itself. Thus, the resulting CPU time can be significantly impacted by the frequency of these mesh adaptations. In addition, there is also potential for unequal balancing of the elements across the MPI processes (although a dynamic load balancing step is incorporated within the parallel mesh optimization procedure). The fact that a fixed number of processing cores is utilised throughout the simulation, and that this number is unlikely to be optimal from a parallel scaling perspective given the significant changes in element count during the course of the adaptive mesh simulations, can also impact on computational efficiency gains. The actual CPU required by our numerical simulations confirm our initial hypothesis. For the pre-refined simulations a total of 6880 CPU hours were required, while 5641.6, 4974.24, and 4747.19 CPU

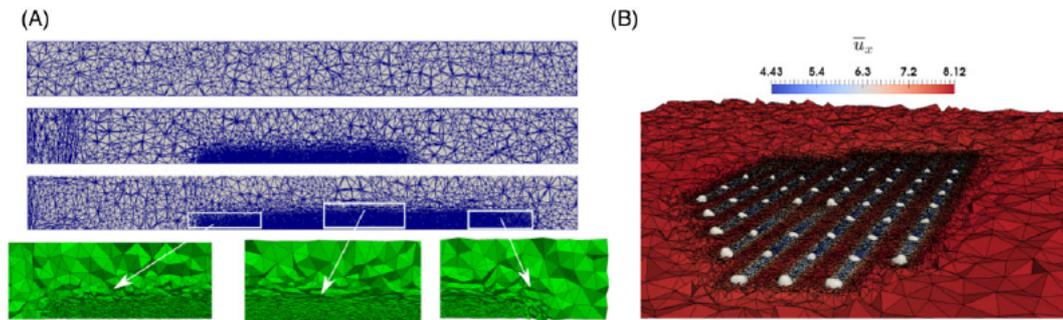


FIGURE 17 Left: A vertical slice along the x -axis passing through the centre of the domain ($y = 5$ km) for the adaptive mesh at 0 s, 100 s, and 400 s (from top to bottom) having in total 100 760, 1 649 459, and 2 558 880 elements in the computational domain, respectively, and zoomed-in snapshots from the final adaptive mesh ($t = 400$ s at different locations). Right: A three-dimensional view of the computational mesh and streamwise velocity \bar{u}_x at $t = 100$ s using a crinkle slice passing through hub height. Iso-contours of the magnitude of the turbine source term F_T are also shown to identify the location of each turbine. The iso-contour level is chosen small enough such that all turbines are visible [Colour figure can be viewed at wileyonlinelibrary.com]

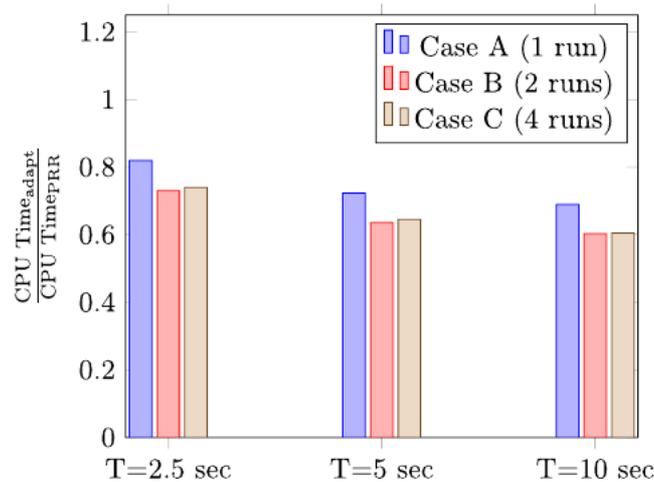


FIGURE 18 CPU time required by the adaptive simulations in the three scenarios (A, B, and C), normalized by the CPU time required for the PRR simulation [Colour figure can be viewed at wileyonlinelibrary.com]

hours were required for the adaptive mesh simulations using mesh adaptation periods of ($T_{\text{adapt}} = 2.5$ s, 5 s, and 10 s), respectively. From these results, we may make three observations. First, by switching to the adaptive approach the CPU time is reduced by 18% even when making relatively frequent mesh adaptation operations. Second, by reducing the frequency of mesh adaptations additional reductions in the CPU time are achieved to 27.7% and 31%, respectively. Finally, the number of 80 processors, while it can be optimal in terms of scalability in the case of the pre-refined region, it is not always so for the adaptive ones. This is due to the fact that during the spin-up period, a relatively small number of elements will be distributed across a large number of processors and the computational cost will be dominated by the MPI communications between the partitions.

In order to further optimize the CPU usage of the adaptive simulations, we have re-run the same simulations but this time starting the parallel computation on a smaller number of processing cores ($N_{\text{proc}} = 20$), and as the overall problem size grows we start and stop the simulation (using checkpoints) and increase the number of cores to the final 80 as the element counts increase. We should emphasize here that prior knowledge of the final mesh from the original adaptive simulations using 80 processing cores was essential in better designing the decomposition of the domain when checkpoints were used. Here we present three cases, the first one (case A) is the original case in which, the simulations were run on 80 processors for the whole time period, the second one (case B) starts with 20 processors, then after the first spin-up time ($t = 100$ s) the simulation is stopped and restarted with 80, while the last one (case C) uses 20, 40, 60, and 80 at the periods 0 to 100, 100 to 200, 200 to 300, and 300 to 400 s, respectively. The cumulative CPU time for the three cases is shown in Figure 18. It is observed that cases B and C result in a smaller overall CPU time than case A. In addition, while in case C the decomposition/repartition is performed four times, this does not lead to smaller CPU times. Instead, case B seems to behave better, although by a small amount. Based on this, we may infer that if a moderately large number of elements (≈ 25 000) is used per processor after the first spin-up time ($t = 100$ s), changing the number of processors more often does not lead to significant changes in the observed CPU times. This might be due to the fact that Fluidity exhibits excellent scalability properties (both strong and weak) when a small number of processors is used, and enough elements are distributed to each processor.

6 | DISCUSSION

In this work, we presented the implementation and validation of a uRANS-based, mesh-adaptive ALM which is able to optimize the number of elements/cells that are required to resolve the wake field within a full-size wind farm. Our ALM implementation is based on the original model of Sørensen and Shen³⁴ with additional models being used to account for blade end effects and the impact of the tower shadow on the near wake field. The fluid flow is resolved using a uRANS formulation of the governing equations and the $k-\omega$ SST turbulence closure model. The developed Fluidity-ALM model was validated against wind tunnel measurements before attempting full-scale wind farm simulations. For the wind farm simulations, we selected the Lillgrund offshore wind farm for which numerous previous studies and observed data are available. The effect of mesh-adaptivity on reducing the computational cost was also examined with particular emphasis being given on processors' load balancing, especially during the transient period where the number of elements follows a linear trend.

More specifically, for the developed model a detailed mesh convergence study was undertaken for both the fixed and the adaptive mesh approaches showing that for a given edge length h the two mesh strategies yield similar results for the integrated rotor characteristics such as the power or thrust coefficient. Moreover, as the element's edge length is reduced, the model predictions for the power coefficient C_p converge to the experimental reference values. Looking at the wake solutions on the other hand, mesh-adaptivity provides better estimates for the wake field, particularly in regions where large shear (velocity gradients) exist. This is due to the ability of the mesh optimization algorithm to identify these regions and allow elements with large aspect ratios to align with the underlying gradients in solution fields. A fixed mesh does not have such an ability and therefore a better solution at these regions can be obtained only by further refining the mesh locally. The validation of the model using the two Blind Tests (BT1 & BT2) shows that both the near and far wake field can be accurately predicted given that enough resolution is used. More satisfactory, however, are the predictions from the second validation case, BT2. Deep array wake modelling has been a great challenge for many wake models and it is a crucial step towards accurately predicting offshore wind farm power output. The results from the comparison with BT2 give us confidence that, although only two turbines in-line are used, we are capable of obtaining high-fidelity wake solutions for the back row turbines, when both operate at peak conditions (optimal tip speed ratio).

Looking at the Lillgrund offshore wind farm simulation results, the power predictions from each row (A–H) agree well with measured data from Dahlberg.²⁴ The largest discrepancy between the model predictions and the measured data is observed on the second turbine in each row. This systematically appears in all simulations, and is believed to be due to the uniform velocity profile chosen for the simulations. Nevertheless, the two modelling approaches using the fixed and the adaptive mesh strategies yield almost identical results for the power coefficients. This re-affirms our conclusions from the mesh convergence study which states that the turbine predictions are not dependent on the mesh strategy but rather the edge-length used at the location of the rotor. As far as the wake field is concerned, although there is lack of data for comparison, the accuracy of the power coefficients predictions suggests that the model is able to capture the magnitude of the wake-deficits along each row, at least in a time-averaged sense. In addition, mesh adaptivity was found to be a favourable choice leading to a reduction in the overall computational cost while maintaining the same accuracy. Moreover, by changing the adaptation period T_{adapt} we were able to further reduce the overall computational cost without compromising the model's accuracy. An additional reduction of the computational cost was also observed when the adaptive simulations were initialized on a smaller number of CPUs, with this number increased in response to the spin-up of the computational mesh. The adaptation period was considered to be constant during each individual simulation. Again, this is not necessarily an optimal approach, since the frequency of the adaptations should always be based on the state of the dynamics. However, such an approach requires relatively complex error measure designs which are beyond the scope of the present work. For an effective variable-frequency mesh-adaptive approach, a "goal-based" *a posteriori* error measure would be good to consider. This seeks to generate the optimal mesh at every instance purely for maximizing the accuracy in a user-defined "goal" (e.g. power or thrust), and thus resolution is not wasted at locations and times where it does not contribute to this goal.

Regarding the limitations of the present approach, we should begin by discussing the validity of our choice to use a uniform inlet velocity with slip conditions on both the bottom and the top of the domain instead of a log-law profile and a rough wall model at the bottom. Such a simplification of the flow conditions were found to affect the ability of the model to predict the power output in the large-scale simulations, particularly for the second turbine of each row. Future formulations of the mesh-adaptive solver should consider a boundary layer, although extra care should be taken to avoid extensive refinement near the bottom of the domain via for example using a vertically variant mesh gradation technique. In addition, by adopting a uRANS framework, many of the higher frequency interactions between the flow and the turbines were ignored and thus limiting the information that can be extracted from the simulations. To capture such effects, turbulence-resolving simulations need to be undertaken. Creech et al¹⁵ studied the Lillgrund offshore wind farm using the Fluidity solver (although using different discretization options) with an LES formulation and mesh adaptivity and found good agreement with SCADA data. However, their study did not examine the efficiency of mesh-adaptivity, e.g. through comparisons with fixed-mesh simulations. Inherently, an LES solver which employs mesh-adaptivity techniques will impose additional constraints on the frequency of mesh adaptations and the elements' aspect ratios, while a far smaller edge lengths and time step will be required. All these additional factors put the efficiency of coupling mesh adaptivity with an LES solver and conducting wind farm simulations in question and therefore additional studies will be needed to assess its applicability. On the other hand, coupling a uRANS solver with mesh adaptivity for wind energy problems may be seen as more appropriate, particularly for regional scale simulations that focus for example on the interaction of adjacent wind farms,⁴⁴ or when the wind farm simulations are used within an adjoint-based optimization algorithm.⁴⁵

Based on the above, the mesh-adaptive uRANS framework was shown to be an appropriate and efficient tool for large-scale wind farm simulations. The observed computational efficiency of the present simulations suggests that the same approach may be applied to more complex

configurations such as those of onshore wind turbines placed over an uneven terrain and that mesh-adaptivity could be used to solve for the ensemble-averaged flow quantities (mean velocity, TKE) while requiring substantially fewer CPU hours.

ACKNOWLEDGEMENTS

The authors would like to thank Dr James Percival for advice given on this work, and acknowledge funding from Imperial College London's Energy Futures Lab and the EPSRC (grant numbers EP/R007470/1 and EP/L000407/1). In addition, several insightful comments and meaningful suggestions provided by the anonymous reviewers helped us improve the quality of the presentation of the results. Support for parallel computations was provided by Imperial College's Research Computing Service.

ORCID

Georgios Deskos  <http://orcid.org/0000-0001-7592-7191>

Matthew D. Piggott  <http://orcid.org/0000-0002-7526-6853>

REFERENCES

1. Ammara I, Leclerc C, Masson C. A viscous three-dimensional differential/actuator-disk method for the aerodynamic analysis of wind farms. *J Sol Energy Eng.* 2002;124(4):345-356.
2. Jimenez A, Crespo A, Migoya E, Garcia J. Advances in large-eddy simulation of a wind turbine wake. *Journal of Physics: Conference Series.* 2007;75(1):012041.
3. Calaf M, Meneveau C, Meyers J. Large eddy simulation study of fully developed wind-turbine array boundary layers. *Phys Fluids B.* 2010;22(1):015-110.
4. Nilsson K, Ivanell S, Hansen KS, et al. Large-eddy simulations of the Lillgrund wind farm. *Wind Energy.* 2015;18(3):449-467.
5. Wu Y-T, Porté-Agel F. Modeling turbine wakes and power losses within a wind farm using LES: an application to the Horns REV offshore wind farm. *Renewable Energy.* 2015;75:945-955.
6. Lu H, Porté-Agel F. Large-eddy simulation of a very large wind farm in a stable atmospheric boundary layer. *Physics of Fluids.* 2011;23(6).
7. Churchfield MJ, Lee S, Michalakes J, Moriarty PJ. A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics. *J Turbul.* 2012;13:N14.
8. Churchfield MJ, Lee S, Moriarty PJ, et al. A large-eddy simulation of wind-plant aerodynamics. In: 50th AIAA Aerospace Sciences Meeting Nashville; 2012; Tennessee. 1-17.
9. Creech ACW, Fröh W-G, Clive P. Actuator volumes and hr-adaptive methods for three-dimensional simulation of wind turbine wakes and performance. *Wind Energy.* 2012;15(6):847-863.
10. Abolghasemi MA, Piggott MD, Spinneken J, Viré A, Cotter CJ, Crammond S. Simulating tidal turbines with multi-scale mesh optimisation techniques. *J Fluids Struct.* 2016;66:69-90.
11. Creech ACW, Borthwick Alistair GL, Ingram D. Effects of support structures in an LES actuator line model of a tidal turbine with contra-rotating rotors. *Energies.* 2017;10(5):726.
12. Potsdam M, Mavriplis D. Unstructured mesh CFD aerodynamic analysis of the NREL Phase VI rotor. In: 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, American Institute of Aeronautics and Astronautics; 2009.
13. Wissink A, Potsdam M, Sankaran V, Sitaraman J, Yang Z, Mavriplis D. A coupled unstructured-adaptive cartesian CFD approach for hover prediction. In: American Helicopter Society 66th Annual Forum; 2010; Phoenix, AZ.
14. Wissink AM, Kamkar S, Pulliam TH, Sitaraman J, Sankaran V. Cartesian adaptive mesh refinement for rotorcraft wake resolution. In: 28th AIAA Applied Aerodynamics Conference; 2010 July; Chicago, Illinois.
15. Creech A, Fröh W-G, Eoghan MA. Simulations of an offshore wind farm using large-eddy simulation and a torque-controlled actuator disc model. *Surveys in Geophysics.* 2015;36(3):427-481.
16. Kirby AC, Brazell M, Yang Z, Roy R, Reza Ahrabi B, Mavriplis D, Sitaraman J, Stoellinger MK. Wind farms simulations using an overset hp-adaptive approach with blade-resolved turbine models. *23rd AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics.* 2017.
17. Piggott MD, Pain C, Gorman G, Power P, Goddard A. h, r, and hr adaptivity with applications in numerical ocean modelling. *OCEAN MODELLING.* 2005;10:95-113.
18. Piggott MD, Gorman GJ, Pain CC, et al. A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes. *Int J Numer Methods Fluids.* 2008;56(8):1003-1015.
19. Piggott MD, Farrell PE, Wilson CR, Gorman GJ, Pain CC. Anisotropic mesh adaptivity for multi-scale ocean modelling. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences.* 2009;367(1907):4591-4611.
20. Fluidity. Fluidity manual. Applied Modelling and Computation Group (AMCG). 2005. <https://doi.org/10.6084/m9.figshare.1387713>.
21. Menter FR. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA J.* 1994;32(8):1598-1605.
22. Krogstad Per-Åge, Eriksen P. E. "Blind test" calculations of the performance and wake development for a model wind turbine. *Renewable Energy.* 2013;50:325-333.
23. Pierella F, Krogstad P-Å, Sætran L. Blind test 2 calculations for two in-line model wind turbines where the downstream turbine operates at various rotational speeds. *Renewable Energy.* 2014;70:62-77. Special issue on aerodynamics of offshore wind energy systems and wakes.
24. Dahlberg J-Å. . Assessment of the Lillgrund wind farm: power performance, wake effects. LG Pilot Rep., Vattenfall Vindkraft AB, Stockholm. 2009. Available at <https://corporate.vattenfall.se/globalassets/sverige/om-vattenfall/om-oss/var-verksamhet/vindkraft/lillgrund/assessment.pdf>.

25. Menter FR, Kuntz M, Langtry R. Ten years of industrial experience with the sst turbulence model. In: Internal symposium, turbulence, heat and mass transfer, Vol. 4; 2003; Antalya, Turkey. 625-632.
26. Pain C, Piggott MD, Goddard AJH, Fang F, Gorman G, Marshall DP, Eaton MD, Power P, de Oliveira CRE. Three-dimensional unstructured mesh ocean modelling. *Ocean Modelling*. 2005;10(1-2):5-33.
27. Donea J, Huerta A. *Finite Element Methods for Flow Problems*. John Wiley & Sons, Ltd; 2005:284-287.
28. Cotter CJ, Ham DA, Pain CC, Reich S. Stability of a mixed galerkin finite element pair for fluid flow simulations. *J Comput Phys*. 2009;228(2):336-348.
29. Kuzmin D. A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *J Comput Appl Math*. 2010;233(12):3077-3085. Finite Element Methods in Engineering and Science (FEMTEC 2009).
30. Pain C, Umpheley AP, de Oliveira CRE, Goddard AJH. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Comput Methods Appl Mech Eng*. 2001;190(29):3771-3796.
31. Farrell PE, Piggott MD, Pain CC, Gorman GJ, Wilson CR. Conservative interpolation between unstructured meshes via supermesh construction. *Comput Meth Appl Mech Eng*. 2009;198(33):2632-2642.
32. Farrell PE, Maddison JR. Conservative interpolation between volume meshes by local Galerkin projection. *Comput Meth Appl Mech Eng*. 2011;200(1-4):89-100.
33. Piggott MD, Pain CC, Gorman GJ, Marshall DP, Killworth PD. Unstructured adaptive meshes for ocean modeling, *Ocean Modeling in an Eddy Regime*; 2008:383-408.
34. Sørensen J, Shen WZ. Numerical modeling of wind turbine wakes. *J Fluids Eng*. 2002;124(2):393-399. 2002.
35. Trolborg N, Sørensen JN, Mikkelsen RF. Actuator line modeling of wind turbine wakes. *Ph.D. Thesis*; 2009.
36. Shen WZ, Mikkelsen R, Sørensen JN, Bak C. Tip loss corrections for wind turbine computations. *Wind Energy*. 2005;8(4):457-475.
37. Sarlak H, Meneveau C, Sørensen J. N. Role of subgrid-scale modeling in large eddy simulation of wind turbine wake interactions. *Renewable Energy*. 2015;77:386-399.
38. Martínez-Tossas LA, Churchfield MJ, Meneveau C. Optimal smoothing length scale for actuator line models of wind turbine blades based on gaussian body force distribution. *Wind Energy*. 2017;20(6):1083-1096.
39. Sarlak H. Large Eddy Simulation of Turbulent Flows in Wind Energy. PhD thesis Technical University of Denmark 2014.
40. Pope SB. *Turbulent Flows*. New York: Cambridge University Press; 2000;463-557.
41. Eriksson O, Lindvall J, Breton S-P, Ivanell S. Wake downstream of the Lillgrund wind farm—a comparison between LES using the actuator disc method and a wind farm parametrization in WRF. *J Phys Conf*. 2015;625:012-028.
42. Jonkman J, Butterfield S, Musial W, Scott G. Definition of a 5-MW reference wind turbine for offshore system development. *National Renewable Energy Laboratory*. 2009.
43. Deskos G, Abolghasemi MA, Piggott MD. Wake predictions from two turbine models using mesh-optimisation techniques. In: Proceedings of the Twelfth European Wave and Tidal Energy Conference EWTEC; 2017; University College Cork, Ireland.
44. Hansen KS, Réthoré P-E, Palma J, Hevia BG, Prospathopoulos J, Peña A, Ott S, Schepers G, Palomares A, van der Laan MP, Volker P. Simulation of wake effects between two wind farms. *Journal of Physics: Conference Series*. 2015;625(1):012008.
45. King RN, Dykes K, Graf P, Hamlington PE. Optimization of wind plant layouts using an adjoint approach. *Wind Energy Science*. 2017;2(1):115-131.

How to cite this article: Deskos G, Piggott MD. Mesh-adaptive simulations of horizontal-axis turbine arrays using the actuator line method. *Wind Energy*. 2018;21:1266–1281. <https://doi.org/10.1002/we.2253>