RESEARCH ARTICLE

WInc3D: A novel framework for turbulence-resolving simulations of wind farm wake interactions

Georgios Deskos | Sylvain Laizet | Rafael Palacios

1Department of Aeronautics, Imperial College London, London, UK

Correspondence
G. Deskos, S. Laizet, R. Palacios, Department of Aeronautics, Imperial College London, London, UK
E-mail : g.deskos14@imperial.ac.uk
E-mail : s.laizet@imperial.ac.uk
E-mail : r.palacios@imperial.ac.uk

Funding Information
This research was supported by the Engineering and Physical Sciences Research Council (EPSRC). Grant Numbers: EP/R007470/1, EP/L000261/1 and the EU PRACE project (Number:2010PA5024).

Abstract

A fast and efficient turbulence-resolving computational framework, dubbed as WInc3D (Wind Incompressible 3-Dimensional solver), is presented and validated in this paper. WInc3D offers a unified, highly-scalable, high-fidelity framework for the study of the flow structures and turbulence of wind farm wakes and their impact on the individual turbines’ power and loads. Its unique properties lie on the use of higher-order numerical schemes with "spectral-like" accuracy, a highly efficient parallelisation strategy which allows the code to scale up to $O(10^4)$ computing processors and software compactness (use of only native solvers/models) with virtually no dependence to external libraries. The work presents an overview of the current modelling capabilities along with model validation. The presented applications demonstrate the ability of WInc3D to be used for testing farm-level optimal control strategies using turbine wakes under yawed conditions. Examples are provided for two turbines operating in-line as well as a small array of 16 turbines operating under “Greedy” and “Co-operative” yaw angle settings. These large-scale simulations were performed with up to 8192 computational cores for under 24 hours, for a computational domain discretised with $O(10^9)$ mesh nodes.

KEYWORDS:
high-order wind farm simulator; wind farm wakes; optimal farm control

1 | INTRODUCTION

Wind turbines operating within large-scale wind plants interact with each other through their wakes. Surveys over a number of utility-scale wind farms (e.g. Horns Rev I, Nysted, Anholt, London array etc.) have shown that wakes are responsible for annual energy losses of up to 25% [1,2,3,4]. Additionally, wake-generated turbulence can significantly increase fatigue loading and therefore the lifetime of wind turbine blades [5]. Numerical predictions of wind farm wakes are often based on simple, analytical models such as the well-known Park model [6,7] or the most recent integrated framework FLORIS [8] which also allows for layout or control optimisation (e.g. turbines under yawed conditions). While such models can be used for many design or optimisation purposes they cannot provide insight into the complex interactions between the individual wakes and atmospheric turbulence. To study the turbulent structure of turbine wakes and their impact on farm-level operation numerically, high-performance numerical codes have been devised which are often based on large-eddy simulation (LES) and turbine parametrisations (e.g. actuator line). Such frameworks often referred to as wind farms simulators (WFS) can provide integrated solutions by resolving Atmospheric Boundary Layer (ABL) dynamics to a desired spatial and temporal scale, while accounting for the aero-servo-elastic behaviour of the individual wind turbines.

To this day, a number of WFSs exists, offering a multitude of modelling options, including modelling the ambient atmospheric flow conditions, various turbine parametrisations (e.g. actuator disc (AD), actuator line (AL) or actuator surface (AS) models) as well as active control either on a turbine or plant level. A review of state-of-the-art LES codes for wind farm simulations was recently presented by Breton et al. [9]. Examples of WFS are the EllipSys3D [10,11,12], the EPFL model [13], the JHU model [14], NREL’s SOWFA [15], the SP-Wind model [16], WITTS [17], WIS [18], PALM.
The various models employ different discretisation schemes (e.g., finite differences/volumes or pseudo-spectral) and often use second- to fourth-order accurate schemes. The ability of a WFS to accurately capture complex flow dynamics of utility-scale wind farms has been demonstrated by [22,10] for the Lillgrund offshore wind farm, and [13] for the Horns Rev wind farm, while a number of other wind farm simulations have been undertaken for theoretical wind farm layouts [23,24] to e.g., assess the impact of micro-siting. More recently, optimal control strategies have been either tested [25] or obtained via high-fidelity WFS [26].

With Exascale computing systems being expected to launch at some point during the next decade, high-fidelity LES models with increased resolution can be used to better understand the complex phenomena involved in the interaction of atmospheric turbulence with arrays of wind turbines [27]. To this end, the development and validation of Win3D – a high-fidelity, highly-scalable, open-source WFS – is aligned with the prospects of utilising large computational resources to tackle engineering problems of interest. Win3D is a Fortran 90 code based on higher-order compact finite-difference discretisation schemes [28] with "spectral-like" accuracy, a 2D domain decomposition strategy [29] and native turbine parametrisation models [30,31]. Here we present an overview of all of its current modelling capabilities as well as its ability to resolve real-size wind farm wakes dynamics. To this end, the remainder of this paper starts with a short description of the fluid flow solver including the governing equations and the numerical implementation. Particular focus is given of the code’s excellent scalability properties across a number of HPC platforms with different hardware architectures (Sunway TaihuLight, ARCHER etc.). Validation for the wind turbine models as well the accurate generation of the atmospheric background turbulence are presented in sections 4 and 5. Applications of the developed models are finally presented in Section 6 for two-turbines operating in a row, by varying their relative offset distance and yaw angle setting, and a full array of 16 turbines operating with and without optimal yaw control. Results are discussed together with the proposed future directions in Section 7.

2 FLUID SOLVER

2.1 Governing equations

Win3D uses an explicit LES formulation to solve the unsteady, incompressible, filtered Navier-Stokes equations,

\[
\frac{\partial \tilde{u}_i}{\partial t} + \frac{1}{\rho} \left( \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_i}{\partial x_j} \tilde{u}_j \right) = -\frac{1}{\rho} \frac{\partial \tilde{p}}{\partial x_i} - \frac{1}{\rho} \frac{\partial p_\infty}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + \frac{F_{iT}}{\rho}, \quad i=1,2,3 \tag{1}
\]

\[
\frac{\partial \tilde{u}_i}{\partial x_i} = 0, \tag{2}
\]

where \( \tilde{p} = \tilde{p} + 1/3 \tilde{u}_i \tilde{u}_i \), and \( \tilde{u}_i \) are the filtered components of the modified pressure and velocity fields, \( \rho \) the fluid density, \( \tau_{ij} \) the sub-filter stresses and \( F_{iT} \) the turbine forcing term. The calculation of the turbine forcing term is described in more detail in section 3, while the divergence of the sub-filter stresses \( \partial_j \tau_{ij} \) is calculated using the standard Smagorinsky model [32]

\[
\tau_{ij} = -2(C_S \Delta)^2 \tilde{S}_i \tilde{S}_j, \quad \tilde{S}_i = \frac{1}{2} \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \tag{3}
\]

where \( C_S \) is the Smagorinsky constant corrected near the wall using the Mason-Thomson [33] formula,

\[
C_S = \left( C_0^n + \left\{ \kappa \left( \frac{y}{\Delta} + \frac{y_0}{\Delta} \right) \right\}^{-n} \right)^{-1/n} \tag{4}
\]

where \( y_0 \) is the roughness length scale, \( y = x_2 \) is the vertical distance measured from the bottom of the domain, and \( \Delta = \sqrt{\Delta x_1 \Delta x_2 \Delta x_3} \) the geometrically-averaged grid size. It is worthwhile noting that in the above described formulation, the viscous term of the Navier-Stokes equation has
been neglected due to the high-Reynolds number and the fact that the mesh is coarse enough so that the near-ground viscous layer will not have any impact on the calculations. \( WInc3D \) can handle a number of initial and boundary conditions (periodic, inflow-outflow) however to accurately model the atmospheric boundary layer, we apply two types of boundary conditions in the vertical direction \((y\text{-direction})\). These are a stress-free condition \( \left( \partial_y \tilde{u}_i = \tilde{v}_i = 0 \right) \) on the top boundary \((y = H)\) combined with a free-slip/wall-stress model on the bottom of the domain \((y = 0)\) (see figure 1). The stress model uses the rough wall model of Moeng [34] and in particular its local formulation modification proposed by Bou-Zeid et al. [35].

\[
\tau_{\text{wall}}(x, z) = \tau_v(x, y) = \frac{\tilde{u}_i(x, \Delta y/2, z)}{\sqrt{\tilde{u}_i^2(x, \Delta y/2, z) + \tilde{u}_j^2(x, \Delta y/2, z)}} \tag{5}
\]

and

\[
\tau_v(x, z) = -\left[\frac{\kappa}{\ln\left(\frac{\Delta y/2}{y_0}\right)}\right]^2[\tilde{u}_i^2(x, \Delta y/2, z) + \tilde{u}_j^2(x, \Delta y/2, z)] \tag{6}
\]

where \( \tilde{u} \) is a twice-filtered velocity field. The \( \langle \rangle \) corresponds to a grid-size cut-off filter whereas \( \langle \rangle \) to a test Gaussian filter at twice the grid size.

### 2.2 Numerical solver

To solve the unsteady, filtered Navier-Stokes equations a compact finite difference scheme framework is used for differentiating, interpolating and filtering. High-order compact finite differences are used to guarantee “spectral-like” accuracy for the numerical solver [36][28] while keeping some flexibility when choosing the boundary conditions. The main advantage of high-order compact schemes by comparison to more conventional low-order schemes is their ability to capture a wider range of scales (especially the small ones) for a given spatial resolution. This feature is crucial for Direct and Large Eddy Simulations of turbulent flows for which the correct representation of the relevant scales is essential. To the best of our knowledge, only a handful of wind farm simulators use high-order methods in their implementation e.g. high-order spectral element methods (SEM) in Chatterjee and Peet [24]. Most other WFSs are based on low-order numerical schemes, for which the required resolution can be substantially more demanding. Nonetheless, it should be noting that, overall, the mesh size/resolution depends primarily on the actuator line method and more precisely on the chord size/smearing parameter as previous studies have shown [37]. For the present work, we have chosen the grid size to be equal to half the average chord size and the smearing parameter to be twice the mesh size. The final mesh is similar to that of previous studies [15][25].

Assuming a uniform distribution of \( n_x \) nodes \( x_i = (i - 1)\Delta x, 1 \leq i \leq n_x \) within a domain \([0, L_x]\) the operators can be computed as shown in table 1. It is well known that the treatment of incompressibility is a real difficulty to obtain solutions of incompressible Navier-Stokes equations. In \( WInc3D \) the Poisson equation for the incompressibility of the velocity field is fully solved in spectral space via the use of relevant 3D Fast Fourier transforms (FFTs). With the help of the concept of modified wavenumber [36], the divergence free condition is ensured up to machine accuracy. The pressure mesh is staggered from the velocity one by half a mesh to avoid spurious pressure oscillations observed in a fully collocated approach [28], hence the need for interpolation schemes. Note finally, that conventional an explicit third-order Adams-Bashforth time advancement scheme is used for time marching.

### 2.3 Parallelisation strategy and code scalability

The simplicity of a Cartesian mesh allows for an easy implementation of a 2D domain decomposition based on the standardized and portable Message Passing Interface (MPI) standard [27]. The computational domain is split into a number of equally sized sub-domains (pencils) which are each assigned to an MPI-process. The implicit derivative, interpolation and filtering operations in the x-direction (y-direction, z-direction) can easily be performed in X-pencils (Y-pencils, Z-pencils) as seen in the left-hand side of figure 2 where 9 sub-domains are used as an example. The 3D FFTs required by the Poisson solver are also broken down as series of one-dimensional FFTs computed in one direction at a time. Up to 70 global transfers to switch from one pencil to another are performed at each time step with the MPI command \( \text{MPI1\_ALLTOALL(} \text{V}) \), a global communication command which sends data from all to all processes. This high level of parallelisation is achieved with the highly scalable 2DECOMP&FFT library, openly available at [http://www.2decomp.org](http://www.2decomp.org). It also provides a parallel I/O module to help \( WInc3D \) handles large data. It takes advantage of the decomposition information available in the library kernel and uses MPI-1 - 10 to implement some most frequently used I/O functions for applications based on 3D Cartesian data structures. Within \( WInc3D \) it is possible to read/write more than 2 GB of data per second with up to \( O(10^4) \) cores, depending on the supercomputers. The scalability of \( WInc3D \) has been tested on various Tier-1/0 supercomputers ARCHER, MareNostrum (MN) and Sunway TaihuLight (ST) [www.archer.ac.uk](http://www.archer.ac.uk), [https://www.bsc.es/marenostrum](https://www.bsc.es/marenostrum) and [http://www.nscwx.cn/wxcyw/](http://www.nscwx.cn/wxcyw/) for different simulation sizes. The scalability plots are shown in the right-hand side of figure 2. The scaling plots show that \( WInc3D \) exhibits very good “strong scaling” properties across a number of different HPC platforms and for up to 65 536 computing processing cores. Despite the large number of global communications, the scalability is excellent, except maybe when the number of mesh nodes per core is too small and the number of small communications is too important, a trend which can be observed on Sunway TaihuLight (in red in the right-hand side of 2) when the reported scaling is moving away from the ideal one.
TABLE 1 Table of discrete operators (derivation, interpolation and filtering) together with the selected coefficients for 6th order accuracy.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Compact scheme</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Derivative</td>
<td>( \alpha f'<em>{i-1} + f'<em>i + \alpha f'</em>{i+1} = a \frac{f</em>{i+1} - f_{i-1}}{2\Delta x} + b \frac{f_{i+2} - f_{i-2}}{4\Delta x} )</td>
<td>( \alpha = \frac{1}{3}, \quad a = \frac{14}{9}, \quad b = \frac{1}{9} )</td>
</tr>
<tr>
<td>Second Derivative</td>
<td>( \alpha f''<em>{i-1} + f''<em>i + \alpha f''</em>{i+1} = a \frac{f</em>{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + b \frac{f_{i+2} - 2f_i + f_{i-2}}{4\Delta x^2} )</td>
<td>( \alpha = \frac{2}{11}, \quad a = \frac{12}{11}, \quad b = \frac{3}{11} )</td>
</tr>
<tr>
<td>Interpolation</td>
<td>( \alpha f'<em>{i-1/2} + f'</em>{i+1/2} + \alpha f'<em>{i+3/2} = a \frac{f</em>{i+1} - f_{i-1}}{\Delta x} + b \frac{f_{i+2} - f_{i-2}}{3\Delta x} )</td>
<td>( \alpha = \frac{9}{62}, \quad a = \frac{63}{62}, \quad b = \frac{17}{62} )</td>
</tr>
<tr>
<td>Filtering</td>
<td>( \alpha \hat{f}<em>{i-1} + \hat{f}<em>i + \alpha \hat{f}</em>{i+1} = a f_i + b \frac{f</em>{i+1} + f_{i-1}}{2} + c \frac{f_{i+2} + f_{i-2}}{2} + d \frac{f_{i+3} + f_{i-3}}{2} )</td>
<td>( \alpha = \frac{1}{16} (11 + 10\alpha), \quad a = \frac{1}{32} (15 + 34\alpha), \quad b = \frac{1}{32} (15 + 34\alpha), \quad c = \frac{1}{16} (-3 + 6\alpha), \quad d = \frac{1}{32} (1 - 2\alpha) )</td>
</tr>
</tbody>
</table>

From the scalability plots we may infer that optimal (ideal) scaling is achieved when the number of mesh nodes per processor is around 125,000-175,000 (except maybe for Sunway TaihuLight). Note that this estimate is dependant on the supercomputer characteristics. For the interest of the reader, ARCHER is a Cray XC30 based on 2 x 12 core Intel® Xeon® processors running at 2.7 GHz using the Aries interconnect (dragonfly topology [38]). MARENOSTRUM is a Lenovo SD530 based on 2 x 24 core Intel® Xeon® processors running at 2.1 GHz using the Intel® Omnipath Full-Fat Tree interconnect. SUNWAY TAIHULIGHT is based on Chinese-designed SW26010 manycore 64-bit Reduced instruction set computer (RISC) processors (in-house customised architecture).

3 | TURBINE PARAMETRISATION

3.1 | Aerodynamic modelling

Wind turbines are parametrised using an enhanced actuator line approach. The key concept underlying ALM is based on representing the rotor’s blades through rotating lines discretised into blade elements as shown in figure 3. Calculations are made on a two-dimensional basis (each blade element is considered separately) while three-dimensional corrections can be used to compute the final blade element forces. Winac30 offers the possibility to calculate aerodynamic loads either with a static or a dynamic model. The first model, which is the standard approach followed by most actuator line implementations (see, e.g. [39]) calculates the instantaneous local relative velocity and angle of attack via,

\[
U_{rn} = (U - U_b) \cdot n, \tag{7a}
\]

\[
U_{rt} = (U - U_b) \cdot t, \tag{7b}
\]

where \( U \) is the incident velocity extracted from the fluid solver by evaluating the flow velocity at the blade element midpoint, \( U_b \) is the solid body velocity of the AL which can either be assumed to be known \textit{a priori} or computed at run-time (e.g. by means of a standard solver), and \( n \) and \( t \) are the element’s normal and tangential unit vectors. Subsequently, the relative velocity magnitude is equal to

\[
U_r = \sqrt{U_{rn}^2 + U_{rt}^2}. \tag{8}
\]
and the effective angle of attack

\[ \alpha = \tan^{-1} \left( \frac{U_{rn}}{U_{rt}} \right) \]  

(9)

Once the relative velocity and angle of attack have been calculated, lift, drag and pitching moment coefficients \((C_L, C_D, C_{M25})\) are evaluated for each blade element from look-up tables using the local Reynolds number \(Re_c = U_c c / \nu\) (where \(\nu\) is the actual kinematic viscosity of the fluid) and angle of attack \(\alpha\). Finally, the coefficients are translated into normal, tangential and pitching moment aerodynamic coefficients via,

\[ C_N = C_L \cos \alpha + C_D \sin \alpha, \]  

(10a)

\[ C_T = -C_L \sin \alpha + C_D \cos \alpha, \]  

(10b)

\[ C_M = C_{M25}, \]  

(10c)

and used to calculate the final forces,

\[ F_N = \frac{1}{2} \rho C_N A U_r^2, \]  

(11a)

\[ F_T = \frac{1}{2} \rho C_T A U_r^2, \]  

(11b)

\[ M_S = \frac{1}{2} \rho C_{M25} c A U_r^2. \]  

(11c)

The second (dynamic) model involves the use of a dynamic stall model. The dynamic model which can be equally used for both pre- and post-stall conditions comprises of three sub-modules: Firstly, the attached model which computes a circulatory force component to be proportional to the flow-solver calculated angle of attack \(\alpha\),

\[ C_N = C_{N\alpha} \alpha, \]  

(12)

and an impulsive force component with the aid of indicial functions as

\[ \Delta C_{NI} = \Delta \lambda N \frac{4}{M} \exp(-\Delta s/T_I), \]  

(13a)

\[ \Delta C_{MI} = \Delta \lambda M \frac{4}{M} \exp(-\Delta s/T_I), \]  

(13b)
where $\Delta s$ is a time variable normalised by the chord size $c$ and the relative velocity $U_r$ and $T_f$ is an airfoil-dependent inertial time scale. Secondly, the separation model where the non-linear normal, tangential forces and pitching moment are computed using the Kirchhoff flow equation

\[
\begin{align*}
C_N &= C_{N_0} (\alpha - \alpha_0) \left( \frac{1 + \sqrt{f}}{2} \right)^2, \\
C_T &= \eta C_{N_0} (\alpha - \alpha_0)^2 (\sqrt{f} - E_0), \\
\frac{C_M - C_{M_0}}{C_N} &= k_0 + k_1 (1 - f) + k_2 \sin(\pi f^2),
\end{align*}
\]

(14a) (14b) (14c)

where $f$ is a separation location along the chord-wise direction of the airfoil and $\eta, k_0, k_1$ and $k_2$ are all empirical parameters. For the separation location the piecewise exponential function of Beddoes [40] is used which here takes the form

\[
f = \begin{cases} 
1 - 0.4 \exp \left( \frac{\alpha - \alpha_1}{S_1} \right) & \alpha \leq \alpha_1 \\
0.02 + 0.58 \exp \left( \frac{\alpha - \alpha_2}{S_2} \right) & \alpha > \alpha_1,
\end{cases}
\]

(15)

where $\alpha_1$ is a limiting angle of attack defined by the dynamic stall onset criterion and $S_1, S_2$ are curve fitting parameters of the static airfoil data. The model in Sheng et al. [41] assumes the same piecewise function for both the dynamic data. However, to take into account hysteresis in the change of the effective angle of attack a deficiency function $D_n$

\[
D_n = D_{n-1} \left[ 1 - \exp \left( - \frac{\Delta s}{T} \right) \right],
\]

(16)

is applied to the angle of attack so that

\[
\alpha' = \alpha - \Delta \alpha_n,
\]

(17)

where $\Delta \alpha_n$ is computed through a deficiency function with a time parameter $T_n$. To decide whether dynamic stall has occurred or not, a critical angle of attack $\alpha_{cr}$ is considered as

\[
\alpha_{cr} = \begin{cases} 
\alpha_{cr} = \alpha_{d0}, & r \geq r_0 \\
\alpha_{cr} = \alpha_{ss} + (\alpha_{d0} - \alpha_{ss}) \frac{r}{r_0}, & r < r_0,
\end{cases}
\]

(18)

where, $\alpha_{d0}$ is a critical stall-onset of the angle of attack (AOA), $\alpha_{ss}$ is the static stall-onset AOA, $r = \dot{\alpha}/(2U_{rel})$ is the reduced pitch rate, and $r_0$ is a limiting pitching rate after which dynamic stall is delayed. Inherently, to achieve dynamic stall, the delayed angle of attack absolute value of $\alpha'$ should be greater than the critical one. Last but not least, the lift force induced by the convecting leading edge vortex provides yet another overshoot to the normal force after dynamic stall occurs and is represented by a further delay in the separation location computed from

\[
f'' = f - \Delta f.
\]

(19)
Again the separation location is computed via a deficiency function. Finally, a vortex model which computes the normal force and pitching moment coefficients due to the leading-edge vortex propagation as
\[ C_{NV} = B_1(f'' - f)V_x, \]  
\[ C_{MV} = B_2 \left[ 1 - \cos \left( \frac{\pi \tau}{T_v} \right) \right] C_{NV}, \]  
where
\[ V_x = \begin{cases} \sin^{3/2} \left( \frac{\pi \tau}{2T_v} \right) & \text{for } 0 < \tau < T_v, \\ \cos^2 \left( \frac{\pi (\tau - T_v)}{4T_v} \right) & \text{for } \tau > T_v, \end{cases} \]  
and \( \tau \) is a non-dimensional tracking time for the leading edge vortex and \( B_1, B_2 \) are coefficients dependent on the airfoil’s shape. The final, aerodynamic coefficients of the dynamic model are computed by adding all of the above described components:
\[ C_N = C_N + C_{NI} + C_{NV}, \]  
\[ C_T = C_T, \]  
\[ C_M = C_M + C_{MI} + C_{MV}. \]  
Independently of whether a static or a dynamic model is used, the elements near the tip of the blade, a tip loss correction factor \( F_{\text{tip}} \) equal to
\[ F_{\text{tip}} = \frac{2}{\pi} \cos^{-1} \left[ \exp \left( - \frac{0.31 N_i (R - r)}{2R \sin \phi} \right) \right], \]  
is applied as suggested by Shen et al. [42], where the parameter \( g \) is given by
\[ g = \exp \left[ -c_1 (N_i \lambda - c_2) \right] + 1. \]  
\( N_i \) is the number of rotor blades, \( \lambda \) the rotor’s tip speed ratio, \( R \) the rotor radius, \( r \) the distance of the element from the tip, \( \phi \) is the angle between the axis of rotation and the element’s local relative velocity, and \( c_1 = 0.125, c_2 = 21 \) are two empirically obtained coefficients. Once the final AL point forces have been computed, they are projected to the fluid mesh by applying a smoothing interpolation function \( \eta \) such that,
\[ F_{ij} = -\sum_{i=0}^{N} F_{AL} \eta(|x_j - r_{ij}|), \]  
where index \( j \) runs over all the mesh nodes, \( i \) over the actuator line nodes, \( N \) is the total number of AL nodes. It should be noted here that the actuator line forces need to be expressed in the fluid solver frame of reference \((x'y'z')\). A rotational matrix \( \mathbf{R}_{xyz} \) is used so that \( \mathbf{F}_{xyz} = \mathbf{R}_{xyz} \cdot \mathbf{F}_{nts} \). This operation takes place before the projection. The standard smoothing interpolation function used within \( \text{WInc3D} \) is the 3D Gaussian kernel
\[ \eta_{ij} = \frac{1}{c^3 \pi^{3/2}} \exp \left[ - \left( \frac{r}{c \alpha L} \right)^2 \right], \]  
where \( c \) is a smoothing parameter taken equal to \( 2\Delta \), a value which has shown to maintain stability in many other actuator line simulations [43] [44] [45] [37].

3.2 Turbine-level controller

The turbine-level controller is implemented through a “five-regions” approach [46] [47]. The five regions correspond to region 1 which considers the turbine operation at before cut-in speed where the generator torque is considered is zero and no power is generated, region 2 in which the controller is considered to be in the optimal state (optimal tip speed ratio) and region 3 (above-rated) in which pitch control is enabled. The other two-regions correspond to linear transitions between region 1 and 2 and region 2 and 3. Within region 2, the generator torque is proportional to the square of the filtered generator speed \( T_{\text{Gen}} = K_2 \Omega_{\text{Gen}}^2 \), where \( K_2 \) is a constant calculated as
\[ K_2 = \frac{1}{2} \rho A_r R^3 \frac{C_{P_{\text{max}}}}{N_{\text{GearBox}}^2}, \]  
where \( C_{P_{\text{max}}} \) is the maximum power coefficient obtained at the optimum tip-speed ratio \( \lambda_* = \frac{\Omega_{\text{Gen}}}{\Omega_{\text{LB}}} \) and \( N_{\text{GearBox}} \) is the high-speed to the low-speed gear box ratio. On the other hand, in region 3 the generator power is held constant and therefore the generator torque is inversely proportional to the filtered generator speed, \( T_{\text{Gen}} = K_3 / \Omega_{\text{Gen}} \). To relate the generated low-speed (shaft) torque to the generator torque a number of calculations need to be undertaken, starting with the consideration that the produced shaft torque is equal to
\[ T_{\text{shaft}} = N_{\text{GearBox}} T_{\text{Gen}} = (I_{\text{Rotor}} + N_{\text{GearBox}}^2 I_{\text{Gen}}) \frac{d\Omega_{\text{shaft}}}{dt}, \]  
where
where $I_{rotor}$ and $I_{generator}$ are the moments of inertia of the rotor and the generator respectively, and $\Omega_{shaft}$ the desired optimal low-speed rotational speed of the wind turbine. The procedure for calculating $\Omega_{shaft}$ starts with an initial guess, and subsequently the shaft and generator torques are calculated to update it.

## 4 ACTUATOR LINE MODEL VERIFICATION

The developed actuator line model is first verified using analytical solutions and reference data from experiments. These reference data include the circulation distribution along the span of an elliptic wing as well as the dynamic stall response of an airfoil to a pitching motion. Investigation of the near-wake field for the present actuator line model can be found in Deskos et al. [31].

### 4.1 Elliptical Wing

Classic potential theory [48] can be used to predict the circulation distribution along the span of the wing. Considering a span length $b$ and a root chord $c_0$, the analytical circulation solution is obtained using Prandtl’s lifting line theory [48] to obtain,

$$\Gamma(r) = \Gamma_{\text{max}} \left[ 1 - \left( \frac{r}{b/2} \right)^2 \right]^{1/2}, \quad \Gamma_{\text{max}} = \frac{2bU_{\infty}(\alpha - \alpha_0)}{1 + AR/2}$$

for $-b/2 < r < b/2$, where $\Gamma_{\text{max}}$ is the maximum circulation at $r = 0$, $U_{\infty}$ is the uniform upstream velocity, $\alpha$ and $\alpha_0$ are the simulated and zero-lift angles of attack and $AR = 4b/\pi c_0$ the wing’s aspect. The corresponding actuator line model for the elliptic wing uses a cosine distribution for the blade elements which provides more resolution towards the tip of the wing.

The computational domain is considered to be of dimensions $10b \times 10b \times 12b$ imposing negligible blockage effects on the wing. The computational mesh is considered to be given a uniform mesh equal to $c_0/4$. This results in a computational domain of 401 × 401 × 481 uniformly distributed mesh nodes. At the inlet boundary a uniform velocity $U_{\infty} = 1$ m s$^{-1}$ while 1D advective boundary conditions are applied at the outlet. On all other boundaries (lateral and top/bottom walls) free-slip conditions are applied. Regarding the aerodynamics of the wing, a thin cambered airfoil is used to compute the sectional lift coefficient using:

$$C_l = C_{l_{0}} + 2\pi\alpha_e$$

where $\alpha_e$ is the effective angle of attack and $C_{l_{0}} = 0.4$ is the lift of zero incidence determined by the actuator line model at each collocation node. By using the standard static version of actuator line, the velocity is sampled at the collocation point of each element and used to compute the circulation and therefore the respective “bound” circulation $\Gamma$ of each element is computed using a combination of the Joukowski theorem ($L = \rho U_{\infty}^2 \Gamma r$, $L$ being the lift force and $U_{\infty}$ the sectional relative velocity) and the definition of the lift coefficient ($C_L = L/(0.5 \rho c U_{\infty}^2)$) leading to,

$$\Gamma = \frac{C_L \rho U_{\infty}^2}{2}$$

For our test we consider different angles of attack $0 \leq \alpha \leq 10$ and more specifically by looking at six (6) cases in which $\alpha = \{0, 2, 4, 6, 8, 10\}$. The normalised circulation distribution $\Gamma(r)$ obtained by the actuator line model is shown for the case $\alpha = 10^\circ$ in figure 4 as a function of the normalised chord distance $r/c_0$. It should be noted that due to the linear relationship between $\Gamma(r)$ and $U_{\infty}$ the results are independent of the upstream velocity. Looking at the error between the analytical and the actuator line solutions, we may observe that the error over most of the wing span remains well below 3%. Greater percentages of error reaching ~20% are found for the actuator line element mid-points at $r/c_0 = 4.8$. Nonetheless, this large absolute error at the tips can be corrected using a tip-loss correction model – not used in the present analysis – although it may be argued that the contribution of the tips to the overall integral lift on the wing is negligible.

### 4.2 Pitching wing

Dynamic stall is a crucial component of the dynamic blade response under turbulent conditions. Here we present a verification study for the developed actuator line model/dynamic stall coupling, using experimental and numerical data obtained for the NREL S825 Airfoil by Sheng et al [49]. For the analysis we adopt the same dynamic stall parameters used by [49] which are shown for completeness in table 2 below. The ALM simulations use the same set-up (inlet and boundary conditions, domain size etc.) as in the elliptic wing simulations, however, the wing is extended to the entire span of the domain and its chord size is considered to be constant and equal to $c_0 = 0.1$ m. To match the oscillatory motion of the simulations with those
of the experiments, the pitching motion frequency is taken to be \( \omega = 2U_\infty k/c_1 = 1.66 \text{ s}^{-1} \), where \( k = 0.083 \) is the reduced frequency used in the experiments and the calculations of Sheng et al. \cite{49}. The simulations were carried out for 100 oscillation periods and data were averaged for the last 50 time units, over the blade element located at the middle of the oscillating wing. The model’s results are shown in figure \( \text{Figure 5} \). The good match between the experiments and the two force reconstructions (present model and the \cite{49}) suggest that an effective coupling between the actuator line model and the dynamic stall one is achieved. Moreover, it can be seen from the lift force that both the stall hysteresis and deep stall onset are correctly predicted by both models. Larger discrepancies are only found for the pitching moment, however the solution again remains within the experimental data range. These discrepancies may be attributed to the fact in the case of the actuator line/dynamic stall coupling the dynamic inflow effect is calculated by the fluid solver (see eq. \cite{12}) and not an indicial function as in the original Sheng et al. \cite{49} model.

5 | ATMOSPHERIC BOUNDARY LAYER

In this section, we present a sensitivity analysis of ABL simulations. The ability to predict the mean velocity \( U \) and its vertical gradient \( \partial U/\partial y \) (\( y \) being the wall normal direction) with high accuracy, is essential in the large-eddy simulation (LES) of the atmospheric boundary layer. In particular, the mean velocity gradient not only affects the streamwise velocity variance, it is also responsible for the turbulent transport of the quantities of interest (e.g. turbine wakes). In the inertial surface-layer of the ABL, the mean velocity follows the law-of-the-wall (LOTW) scaling which implies that its gradient \( \partial U/\partial y \) scales with \( u_*/y \), where \( u_* \) is the wall friction velocity. Mason & Thomson \cite{33} first pointed out that in conventional LES
DAMPING FUNCTION IS USED, BY ADOPTING \((\text{RESEARCH HAS ALREADY BEEN PREVIOUSLY PUBLISHED ON THE SUBJECT})^{(53,35,17)}\). INSTEAD, THE STANDARD 3 MAG OR INS MODEL WITH THE -ATION & 4HOMSON PREDICTION OF THE AB, THE MEAN VELOCITY GRADIENT, SCALING ACCORDING TO THE INERTIAL, 47:

\[ \text{ACTUATOR LINE MODEL, THE FORCE RECONSTRUCTIONS AND EXPERIMENTAL DATA AS PRESENTED BY}^{(49)}. \]

FUNCTION OF THE NORMALISED VERTICAL DISTANCE \(y/H\):

\[ \text{NUMERICAL EXPERIMENTS WE HAVE SELECTED AN NUMBER OF CASES TO INVESTIGATE THE SOLUTION DEPENDENCE ON THE MESH. THIS COVER} \]

\[ \text{KEEP ASPECT RATIO}^{(52)} \text{ WITH A FOCUS ON THE IMPACT OF MESH RESOLUTION AND ASPECT RATIO TO OBTAIN "HIGH-ACCURACY" SOLUTIONS. THE BEST OF OUR KNOWL} \]

\[ \text{EDGE, THIS IS THE} \]

\[ \text{ST INVESTIGATION WHICH IS CONSIDERING HIGH-ORDER COMPACT} \]

\[ \text{EDOMAIN IS DISCRETISED USING A UNIFORM MESH THE DETAILS OF WHICH ARE PRESENTED IN TABLE 3 BELOW. ALL SIMULATIONS ARE CARRIED} \]

\[ \text{OUT FOR UP TO} \]

\[ \text{AND THE VERTICAL RESOLUTION} \]

\[ \text{SOLUTION AS ANY NEAR-SURFACE ERRORS WILL QUICKLY PROPAGATE UPWARDS IN THE VERTICAL DIRECTION.) IT IS THEREFORE IMPORTANT TO TEST THE ABILITY} \]

\[ \text{ESTHETHE MESH RESOLUTION IS INCREASED} \]

\[ \text{D RESULTS ARE SHOWN FROM THE DYNAMIC} \]

\[ \text{SIMILAR PROBLEMS HAVE BEEN OBSERVED} \]

\[ \text{WEFOLLOW THE ANALYSIS OF} \]

\[ \text{IN A SEMI-LOGARITHMIC PLOT AS A} \]

\[ \text{LOG-LAW} \]

\[ \text{MISMATCH BETWEEN THE NUMERICAL SOLUTION AND THE LOTW SCALING WILL HAVE A DIRECT EFFECT IN THE ENTIRE ABL SOLUTION AS ANY NEAR-SURFACE ERRORS WILL QUICKLY PROPAGATE UPWARDS IN THE VERTICAL DIRECTION. IT IS THEREFORE IMPORTANT TO TEST THE ABILITY} \]

\[ \text{OF THE MESH ASPECT RATIO} A_R = \Delta y/\Delta x = \Delta y/\Delta z \text{ AND THE VERTICAL RESOLUTION} \Delta y \].

The computational domain is taken to be \(L_x \times L_y \times L_z = \pi H \times H \times \pi H\), where \(H = 1000 \text{ m}\) is the domain height. For all simulations presented hereafter we assume a wall roughness length scale of \(y_0 = 0.1 \text{ m}\) and a friction velocity \(u_* = 0.45 \text{ m s}^{-1}\). The domain is discretised using a uniform mesh the details of which are presented in table 3 below. All simulations are carried out for up to \(T_{\text{tot}} = 45L_y/u_* = 100 \text{ 000 s}\) and statistics are collected over the last \(T_{\text{stat}} = 50 \text{ 000 s}\). Finally, as the mesh resolution is increased the time step is decreased accordingly by allowing a CFL number of about 0.1. We start by presenting the effect of the mesh aspect ratio \(A_R\) and vertical resolution on the obtained flow statistics. Figure 6 shows the normalised mean streamwise velocity \(U/u_*\) in a semi-logarithmic plot as a function of the normalised vertical distance \(y/H\) for different mesh resolutions. The log-law \(U = u_*/(\kappa \ln(y/y_0))\) is also plotted for reference. It

**TABLE 3** Table of simulation setup cases for the investigation of aspect ratio and vertical resolution.

<table>
<thead>
<tr>
<th>Case</th>
<th>(N_x \times N_y \times N_z)</th>
<th>Aspect ratio (A_R)</th>
<th>(\Delta y/y_0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>64 \times 65 \times 64</td>
<td>\pi</td>
<td>156.25</td>
</tr>
<tr>
<td>C2</td>
<td>128 \times 65 \times 128</td>
<td>\pi/2</td>
<td>156.25</td>
</tr>
<tr>
<td>C3</td>
<td>192 \times 65 \times 192</td>
<td>\pi/3</td>
<td>156.25</td>
</tr>
<tr>
<td>C4</td>
<td>96 \times 97 \times 96</td>
<td>\pi</td>
<td>117.187</td>
</tr>
<tr>
<td>C5</td>
<td>192 \times 97 \times 192</td>
<td>\pi/2</td>
<td>117.187</td>
</tr>
<tr>
<td>C6</td>
<td>288 \times 97 \times 288</td>
<td>\pi/3</td>
<td>117.187</td>
</tr>
<tr>
<td>C7</td>
<td>128 \times 129 \times 128</td>
<td>\pi</td>
<td>78.125</td>
</tr>
<tr>
<td>C8</td>
<td>256 \times 129 \times 256</td>
<td>\pi/2</td>
<td>78.125</td>
</tr>
<tr>
<td>C9</td>
<td>384 \times 129 \times 384</td>
<td>\pi/3</td>
<td>78.125</td>
</tr>
</tbody>
</table>
can be observed that all cases are quantitatively close to the theoretical log-law solution, however important discrepancies arise for all cases in the region around $0.01 L_y$ and $0.1 L_y$. More specifically, the high aspect ratio solutions appear to overestimate the mean velocity at the lower surface layer region. As we decrease the aspect ratio, (by reducing the horizontal distance while keeping the vertical one constant), the solutions converge to the LOTW scaling irrespective of the vertical resolution. The effect of the mesh aspect ratio is further demonstrated in the normalised mean velocity gradient $\Phi(y) = \kappa y/u_\tau \partial \langle u_x \rangle / \partial y$ profiles shown in figure 7. The vertical dashed line represents the theoretical prediction $\Phi(y) = 1$. Again, we observe a significant overshoot for all cases that use the largest aspect ratio $A_R = \pi$. From the three solutions presented in figure 7, the solution that uses the highest vertical resolution appears to have suppressed the overshoot. The latter result is crucial for an accurate description of ABL, particularly when wind farm simulations for which wind turbines operate where the overshoot is located.

**FIGURE 6** Vertical profiles of the mean streamwise velocity normalised by the friction velocity $u_\tau$. Results are presented in a semi-logarithmic plot for different mesh aspect ratios $A_R = \pi, \pi/2$ and $\pi/3$ and vertical resolutions $N_y = 65, 97$ and 129.

**FIGURE 7** Normalised mean velocity gradient $\Phi$ in the lower part of the boundary layer ($y/H < 0.3$). Results are presented for different mesh aspect ratios $A_R = \pi, \pi/2$ and $\pi/3$ and vertical resolutions $N_y = 65, 97$ and 129.
6 | APPLICATIONS

6.1 | Two-turbines operating in-line

Turbines operating in-line either perfectly aligned or tilted is a common feature in wind farms and have been studied in many numerical and experimental studies [11, 15, 13]. Fleming et al. [54] simulated two turbines in row with a spacing of 7 rotor diameters (7D) and explored a parameter space of yaw angle re-direction for the front turbine and spanwise offset for the rear one. For their simulations they used the wind farm simulator SOWFA [22] with the turbines being parametrised based on the NREL 5-megawatt (MW) baseline turbine [47]. An identical set-up is used here for WInc3D to allow for a cross-comparison between the two models. To this end, for the present simulations only the static actuator line approach is considered, as no details on the dynamic stall characteristics (other than the stall AOA or the zero lift slope) are provided in the original report [47]. Next, the atmospheric turbulence inflow is generated via precursor simulations following the guidelines described in Section 3 and assuming neutral boundary layer stability. The precursor simulations were run on a 3 km × 1 km × 3 km computational domain using aerodynamic roughness similar to the one of [25] (friction velocity $u_* = 0.281 \text{ m s}^{-1}$ and a roughness lengthscale $y_0 = 0.001 \text{ m}$) so that we achieve a mean hub-height wind speed equal to $U_{\infty} = 8 \text{ m s}^{-1}$ and a turbulence intensity equal to $I = 6\%$. For the turbine simulations, we use a domain of dimensions $L_x = 1512 \text{ m}$ (streamwise length), $L_y = 504 \text{ m}$ (height) and $L_z = 504 \text{ m}$ (transverse length) corresponding to a domain of $12D \times 4D \times 4D$, where $D = 126 \text{ m}$ is the SMW NREL rotor diameter. Inflow/outflow boundary conditions are considered in the streamwise direction and periodic boundary conditions in the transverse. For the domain discretisation we use a computational mesh equal to $N_x \times N_y \times N_z = 768 \times 257 \times 256$ which corresponds to a uniform mesh size of $\Delta x = 1.95 \text{ m}$. The numerical simulations simulations are carried out for up to 1000 s and statistics (e.g. velocity, power, loads) are collected over the last 600 s. A time step equal to $\Delta t = 0.02 \text{ s}$ is used in order to maintain a CFL number value below 0.1. Each of the "two-turbine" simulations was run on 1024 computational cores and results were obtained in under than 24 hours. The mean power output from the 19 "two-turbine" simulations are shown in 8, together with the respective results obtained by [25] using SOWFA. More specifically, in the left-hand side, the power output of the two turbines (and their sum) is plotted against the yaw angle of the front one while in the right-hand side power output is plotted as a function of the rear-turbine spanwise offset distance. In both cases, the two models (WInc3D and SOWFA) agree well with each other as both the front and rear turbines attain similar mean power values. For the yaw angle plots, some difference appear between the two solutions particularly for the rear turbine. To investigate this further, it is possible to compare the front turbine power output with the theoretical scaling obtained by [55]: $P_f = P_0 \cos^2(\gamma_1)$ where $P_2 = 2$ as shown in 7. Gebraad et al. [25] reported a value $P = 1.88$. With our simulations we were able to compute a value closer to $P = 2$ however differences were found between the positive and the negative yaw angle settings. These differences are believed to mainly affect the rear turbine as the rear turbine's power output is proportional to the degree of wake exposure. Another important result to notice is the significant asymmetry present in both solutions which can be more clearly observed in the total power output as a function of the yaw angle. Recent experimental investigations by Schottler et al. [56] have shown that the existence of such asymmetry is attributed to a combination
of shear, veer as well as the velocity’s gradient orientation. The obtained results by WInc3D estimate a milder power output asymmetry compared to the SOWFA solution. This can be attributed to a small misalignment between the mean velocity and the assumed front turbine yaw angle (in the order of $1^\circ$). This effect which was consistent throughout all simulations may also explain the differences in the effect of offset. An $1^\circ$ difference can be translated into an effective rear turbine offset distance difference of $\sin(1^\circ) \cdot 7D \approx 15.4$ m which is consistent with the observed shift in figure 8. Such a small misalignment difference is less important in the yaw case. Nonetheless, for completeness it is also important to present some snapshots from the instantaneous and mean velocity fields. These are shown for yaw cases $\gamma_1 = -30^\circ$, $0^\circ$ and $30^\circ$ in figure 10. Both the mean and the instantaneous velocity fields confirm that by “yawning” the front turbine, the desired wake re-direction is achieved. Vertical profiles also show that in the case of a positive yaw misalignment, the front turbine wake is veered well away from the location of the rear turbine, denoted as a black circle.

![Graph showing scaling of power with yaw angle](image)

**Figure 9** Scaling of the power of yawed turbine with the yaw angle $\gamma_1$. Results are shown for WInc3D, SOWFA and the [55] model.

![Instantaneous and time-average velocity profiles](image)

**Figure 10** Instantaneous and time-average horizontal and vertical streamwise velocity $u_x$ profiles for the two-turbine case. The horizontal profiles depict the streamwise velocity field at hub height while the vertical profiles show the streamwise velocity field at a vertical plane $6D$ downstream the front turbine. In both cases profiles from the yaw cases with $\gamma_1 = -30^\circ$, $0^\circ$ and $30^\circ$ are shown.

### 6.2 Full scale wind farm simulations: A $4 \times 4$ array of wind turbines

After the previous verification exercises, the ability of WInc3D to accurately simulate full-scale wind turbine arrays will be demonstrated. To this end, we have selected an aligned $4 \times 4$ array with $7D$ (streamwise) and $5D$ (spanwise) spacing. Such case is representative of many offshore wind farm layouts and can be considered as a natural extension of the two-turbine problem described in the previous section (The NREL-5MW wind turbine is also used here). Two farm level yaw-settings are considered here, under below-rated conditions ($U_{\infty} = 8$ m/s, $I = 6\%$). The first setting,
namely the “Greedy” setting, represents a case where each turbine is aligned with the mean flow trying to extract as much energy as it can by ignoring the potential favourable effects of co-operating with neighbouring turbines. The second setting, namely the “Cooperative” setting, considers an optimal yaw setting which allows for individual turbines to be misaligned with the wind in order to re-direct their wake away from the downstream turbines and therefore trying to achieve an overall power maximisation of the array. Optimisation of the individual turbines’ yaw settings is considered following the study of Gebrad et al. [25]. By using the produced “Two-turbine” data, we calibrate the engineering wake model FLORIS [8] using a combination of the standard Jensen and wake-redirection submodels. The calibrated analytical wake model is then used in conjunction with a Sequential Least Squares Programming (SLSP) optimiser to compute the optimal yaw settings that maximise the power output of the array. The optimisation also considers certain bounds in the accepted individual-turbines’ yaw angles (−30, 30) and a relative tolerance equal to \( \varepsilon_{\text{tol}} = 1 \times 10^{-5} \) in the calculation of the cost function (total power). Under these settings, FLORIS estimated a 21.4% increase in power using the following yaw angle settings identical across all columns of turbines as shown in Table 4. The two operational modes “Greedy” and “Cooperative” are subsequently run within Win3D to test the validity of the optimiser as well as to further explore the underlying wake dynamics of the array. For the simulations, we consider a domain equal to \( 26D \times 4D \times 20D \) which again utilises a uniform resolution of \( \Delta x = 1.95 \text{ m} \) by using a \( 1665 \times 257 \times 1280 \) mesh nodes (over half a billion degrees of freedom). For the generation of the inflow atmospheric turbulence a similar procedure as in subsection 6.1 is followed while the wind turbine simulations use exactly the same time step, and data are averaged over 600 s. Figure 11 shows the time/row-averages of the normalised power (by the first row turbine), the normalised root-mean-square (r.m.s.) (by the row-averaged mean power) for both the “Greedy” and “Cooperative” cases. It is worth noting that significant differences exist in both the mean and r.m.s. of power between the two cases, however similar trends can be observed. Firstly, the “Greedy” operational mode yields a power row distribution similar to the reported power of many utility-scale offshore wind farms under an aligned operational scenario \([28,57]\). More specifically, the first row of turbines achieves its full power output potential like a stand-alone turbine, while the consecutive rows (2,3 and 4) achieve only 50% of the power output of the first row. On the other hand, under a “Cooperative” operational scenario, the first row of turbines under-performs (the computed normalised power is approximately equal to 86% of the stand-alone one) but also results in a very large “re-bound” of the downstream row turbines. In fact, the computed percentage of overall power increase is found to be equal to 20.26% a value very close to the original FLORIS estimation. It is also important to mention that other experimental and numerical studies \([58,59]\) have also found power efficiency increase in the orders of 20-30% for similar aligned layouts and spacing. As far as the r.m.s. of the two cases is concerned, it is striking that under the “Cooperative” operational scenario, the wind turbines exhibit less variability, although the difference is rather small, of the order of 5%. Integrating all power fluctuations over the whole array can however lead to
significant power fluctuations which may need to be taken into account when grid-integration is sought. Figure 12 shows the total power output of the array for both operational modes. The significance of wake-steering can be appreciated first by recognising the large difference in the magnitude of the power produced as well as the different events occurring throughout this window of operation for the array. However, striking differences can also be found in the farm-level temporal evolution of the total power. These differences are marked by the shaded areas in figure 12. For example, farm-scale temporal effects due to wake re-direction appear to have a different impact on the total produced power around \( t = 500 \) s when the "Cooperative" case exhibits a sudden drop in the produced power. To better appreciate the dynamic behaviour of the power output it is also worth plotting the spectra of the row-averaged power and total turbine power, power spectral density (PSD) functions. These are shown in figure 13 for both the "Greedy" and "Cooperative" cases confirming the observed but small differences in the power fluctuations. It is worth noticing that all row PSDs exhibit a similar behaviour (e.g. peak around three times the rotational frequency in the high-frequency regime). Interestingly, by averaging over the whole array and analysing the PSD of the total power fluctuations, the high-frequency regime appears to be weaker and the PSD to follow well the Kolmogorov scaling law \((-5/3)\). This finding confirms previous studies by Apt and Milan et al. [60,61]. Finally, it is worth visualising the two solutions via their respective instantaneous and time-averaged streamwise velocity fields (figures 14 as well as iso-vorticity volume rendering visualisations in figure 15). For the streamwise velocity snapshots, the adverse effect that "Greedy" operational mode in conjunction with the array's layout has on the power generation is clearly visualised via the developed velocity deficit behind the downstream turbines and
their respective impact on their rear ones. Conversely, once the front turbines’ wakes are steered away wind turbines are shown to enjoy the benefit of operating under an undisturbed incoming wind flow field. In both cases, the instantaneous velocity fields also provide a full picture of the flow unsteadiness occurring throughout the wind turbine array with enhanced turbulence and large wake meandering flow phenomena taking place, particular towards the last row of turbines. These phenomena and many other detailed flow structures (e.g. near-wake tip vortices) can also very well captured by a volume rendered visualisation of the iso-vorticity field as shown in figure 15.

**FIGURE 14** Instantaneous (top) and time-averaged (bottom) profiles of the streamwise velocity at hub height for the $4 \times 4$ turbine array. Both the “Greedy” (left) and “Cooperative” (right) yaw-setting operational modes are shown.

7 | DISCUSSION & FUTURE DIRECTIONS

In this work we have presented v1nc3d a fast and efficient computational fluid dynamics (CFD) framework for turbulence-resolving simulations of wind farm wakes. v1nc3d has a great potential by comparison to other WFS thanks to the use of high-order finite-difference schemes, with the ability to capture a wide range of turbulent scales at a given spatial resolution. Furthermore, it can scale with thousands of computational cores thanks to a powerful 2D Domain Decomposition strategy. In terms of computational efficiency, the use of high-order finite-difference schemes on a Cartesian mesh combined with a direct spectral solver for the Poisson equation means that the cost of a time step (per number of degree of freedom and per computational core) is potentially much lower than for other WFS based on finite-volume approaches. This is justified by the low-cost of finite-difference method simulations on a Cartesian mesh compared to similar resolution finite volume/finite element simulations conducted on unstructured meshes. It is therefore anticipated that v1nc3d would be faster (and more accurate for a given number of degrees of freedom)
than more conventional low-order wind farm simulators based on unstructured meshes. Quantifying the difference would require to run several simulations with different open-source wind farm simulators and obtain a case-by-case comparison of the computational cost and accuracy, and therefore it lies outside the scope of the present study. Nonetheless, we should emphasize that it is possible to perform large-scale simulations with Winc3D within few hours, only using few thousands computing processing cores. As it only relies on a Fortran 90 compiler and the MPI library, the portability of Winc3D is another of its strengths. It has been tested on different supercomputers and similar performance were reported for simulations with $O(10^5)$ mesh nodes. Finally, enhanced turbine parametrisations are available in Winc3D for an accurate representation of the main features of the wind turbines. The performance of the solver was demonstrated throughout this paper by presenting results from validation studies as well as from real-scale wind farm wake problems and data were successfully compare with reference values and other numerical studies. Regarding future directions in software development, the versatility of Winc3D suggests that several different paths for further research can be taken. Future studies will consider different ABL stabilities by incorporating thermal effects through the Boussinesq approximation. Another development may involve additional actuator line model enhancements (e.g. introducing aero-elasticity to the virtual turbine blades) and the introduction of a free-surface solver, based on the well-established level-set method to track the evolution of the free surface, so that it can be possible to deal with wind/wave interactions in combined load situations on floating or bottom-fixed turbines. Winc3D source code is readily available at

https://github.com/ImperialCollegeLondon/Winc3D

ACKNOWLEDGEMENT

The authors would like to acknowledge funding from the FENGBO-WIND EPSRC project (grant numbers EP/R007470/1). The high-fidelity simulations were carried out in the Sunway TaihuLight System, with the support by the National Supercomputer Centre in Wuxi, China. The authors would also like to thank the EPSRC for the computational time made available on the UK supercomputing facility ARCHER via the UK Turbulence Consortium (EP/L000261/1 and EP/R029326/1). Computational time on MARENOSTRUM was obtained via PRACE project 2010PA5024.

References


**How cite this article:** Deskos G., Laizet S., R. Palacios (2019), WiN-C3D: A novel framework for turbulence-resolving simulations of wind farm wake interactions, *Wind Energy*, 2019;00:1–6.