WIND
ENERGY
SCIENCE

eawe
european academy of wind energy

Open Access

A control-oriented dynamic wind farm model: WFSim

Sjoerd Boersma[1], **Bart Doekemeijer**[1], **Mehdi Vali**[2], **Johan Meyers**[3], **and Jan-Willem van Wingerden**[1]

[1]Delft University of Technology, Delft Center for Systems and Control,
Mekelweg 2, 2628 CC, Delft, the Netherlands
[2]Wind Energy System Research Group, ForWind, Küpkersweg 70, 26129 Oldenburg, Germany
[3]KU Leuven, Department of Mechanical Engineering, Celestijnenlaan 300A, B3001 Leuven, Belgium

Correspondence: Sjoerd Boersma (s.boersma@tudelft.nl)

Abstract. Wind turbines are often sited together in wind farms as it is economically advantageous. Controlling the flow within wind farms to reduce the fatigue loads, maximize energy production and provide ancillary services is a challenging control problem due to the underlying time-varying non-linear wake dynamics. In this paper, we present a control-oriented dynamical wind farm model called the WindFarmSimulator (WFSim) that can be used in closed-loop wind farm control algorithms. The three-dimensional Navier–Stokes equations were the starting point for deriving the control-oriented dynamic wind farm model. Then, in order to reduce computational complexity, terms involving the vertical dimension were either neglected or estimated in order to partially compensate for neglecting the vertical dimension. Sparsity of and structure in the system matrices make this model relatively computationally inexpensive. We showed that by taking the vertical dimension partially into account, the estimation of flow data generated with a high-fidelity wind farm model is improved relative to when the vertical dimension is completely neglected in WFSim. Moreover, we showed that, for the study cases considered in this work, WFSim is potentially fast enough to be used in an online closed-loop control framework including model parameter updates. Finally we showed that the proposed wind farm model is able to estimate flow and power signals generated by two different 3-D high-fidelity wind farm models.

1 Introduction

Optimizing the control of wind turbines in a farm is challenging due to the aerodynamic interactions among turbines. These interactions come from the fact that downwind turbines are often operating in the wakes of upwind ones (Barthelmie et al., 2009). Two important wake characteristics are (1) a flow velocity deficit and (2) an increase in turbulence intensity. The former reduces power production of the farm while the latter leads to a higher dynamic loading on downstream turbines but also induces wake recovery. Individual turbine control variables can influence the wake's flow velocity, turbulence intensity and also location. Hence, by changing the control variables of individual turbines, power production of and loading on these controlled turbines and the downwind turbines can be manipulated. Wind farm control aims to find control variables under changing

atmospheric conditions such that demanded power production and/or a minimization of the loading can be guaranteed, improving the cost and quality of wind energy. State-of-the-art closed-loop dynamic wind farm controllers are based on computationally expensive wind farm models, which make these methods suitable for analysis though unsuitable for online control. The latter is important because it allows for model adaptation to the time-varying atmospheric conditions using supervisory control and data acquisition (SCADA) measurements. As a consequence, more reliable control settings can be evaluated. A survey on wind farm control can be found in Knudsen et al. (2015) and Boersma et al. (2017), for example. In the latter, a clear distinction is made between model-based and model-free control algorithms. This paper is focussed on the former in which it is assumed that controllers are based on a mathematical model of the system. Consequently, the controller performance depends highly on

the model quality. Modelling is therefore a crucial step towards successful implementation of model-based wind farm control.

Overviews on wind farm models can be found in Crespo et al. (1999), Vermeer et al. (2003), Sanderse (2009), Sanderse et al. (2011), Annoni et al. (2014), Göçmen at al. (2016) and Boersma et al. (2017). The spectrum of these models ranges from low fidelity to high fidelity. The latter tries to capture relatively precise wind farm flow and turbine dynamics, while the former tries to capture only the dominant characteristics (dynamic or static) in a wind farm. Examples of high-fidelity wind farm models are Simulator fOr Wind Farm Applications (SOWFA) (Churchfield et al., 2012), UTD Wind Farm (UTDWF) (Martinez-Tossas et al., 2014), SP-Wind (Meyers and Meneveau, 2010) and PArallelized LES Model (PALM) (Maronga et al., 2015). These three dimensional (3-D) high-fidelity wind farm models can easily have 10^6 or more states. The resulting computation time can be of the order of days or weeks using distributed computation for simulation times less than the computation time. In other words, the computation time needed for large-eddy simulations (LESs) is in general more than the total time that is simulated. Clearly, these types of models are not applicable for online model-based control. Rather, these models serve as analysis or validation tools.

One way to reduce the high complexity of wake modelling is by using two-dimensional (2-D) heuristic models that only capture specific wake and turbine characteristics in a wind farm in the horizontal plane at hub height. These types of models are found on the low-fidelity side of the spectrum. Most of these wake models exclusively estimate a steady state situation for given atmospheric conditions. Examples of static models are the Frandsen model (Frandsen et al., 2006) and the Jensen–Park model (Jensen, 1983; Katic et al., 1986). One extension of the Jensen model resulted in the parametric model called FLOw Redirection and Induction in Steady-state (FLORIS) (Gebraad et al., 2014b). Two examples of low-fidelity dynamic models are SimWindFarm (Grunnet et al., 2010) and the model used in Shapiro et al. (2017a), in which relatively simple approximations of the flow deficit are computed using heuristic expressions.

Medium-fidelity models can be found in the middle of the spectrum as they trade off the accuracy of high-fidelity models with the computational complexity of low-fidelity models. These are in general based on simplified versions of the Navier–Stokes equations. For example, in the 2-D dynamic wake meandering (DWM) model (Larsen et al., 2007), assumptions are made regarding the thin shear layer such that the Navier–Stokes equations can be approximated using less computational effort. The authors in Trabucchi et al. (2016) present a model, which is also based on the thin shear layer approximation, but according to the authors it is applicable for non-axisymmetric wind turbine wakes. Wake-Farm (also referred to as FarmFlow) simulates the wind turbine wakes by solving the steady parabolized Navier–

Stokes equations in three dimensions (Crespo et al., 1988; Özdemir et al., 2013). Other wind farm models based on the 3-D Reynolds-averaged Navier–Stokes (RANS) equations are Avila et al. (2013) and van der Laan et al. (2015). In Annoni and Seiler (2015), time averaging is applied to the Navier–Stokes equations, resulting in the 2-D RANS equations. The number of states is then reduced by employing a state reduction technique.

Also considered as medium-fidelity models are the ones presented in Boersma et al. (2016b) and Soleimanzadeh et al. (2014). These wind farm models are based on the discretized 2-D Navier–Stokes equations. However, these models do not contain a turbulence model that allows for wake recovery. In addition, these 2-D models do not take any neglected 3-D effects into account and no yaw actuation of the individual turbines is included.

In this paper, a model will be presented that can be considered as a building block for the closed-loop control framework as illustrated in Fig. 1.

In current practice, signals such as power can be measured from a wind farm, but current research is also focussing on estimating wake characteristics using a lidar device (Raach et al., 2017). These and other wind farm measurements are called SCADA data and can be used by an estimator that is able to adapt the model parameters to current atmospheric conditions and/or estimate the full state space, e.g. all the flow velocities at hub height in the farm. The work presented in Doekemeijer et al. (2016) illustrates the latter and employs the dynamic wind farm model presented in this paper. Subsequently, the estimation can then be used to compute optimal control variables using a model predictive controller. The work presented in Vali et al. (2016) illustrates the application of such a model predictive wind farm controller using the dynamic model presented in this work.

The online closed-loop control paradigm as depicted in Fig. 1 demands a control-oriented dynamic wind farm model that will be presented in this paper. Characteristics of such control-oriented models are

1. low computational cost such that online model update, state estimation and control signal evaluation is possible;

2. their dynamical nature such that they can deal with varying atmospheric conditions within relatively small timescales.

The dynamic control-oriented wind farm model presented in this paper, referred to as WindFarmSimulator (WFSim), is applicable in the framework discussed above and satisfies the two points above. It is based on corrected 2-D Navier–Stokes equations and contains a heuristic turbulence model. The Navier–Stokes equations are modified in order to partially correct for the neglected vertical dimension. Each turbine is modelled using the actuator disk model (ADM) and features yaw and axial induction actuation. An important model fea-
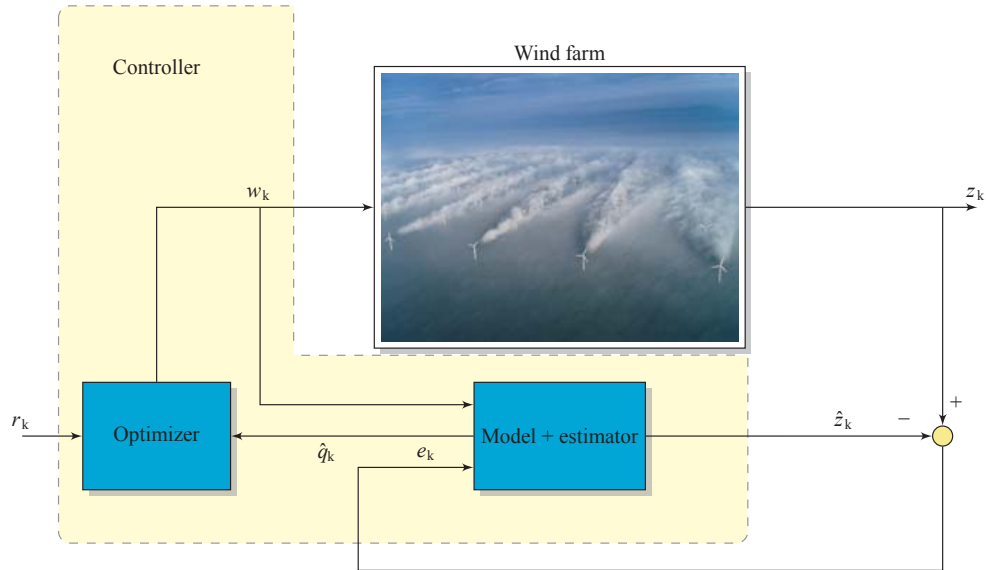
Figure 1. General dynamical closed-loop control framework with measurements z_k and its estimation $\hat{z}_k$ and state estimation $\hat{q}_k$. The signals r_k and w_k are reference and control signals, respectively. In this paper we present a dynamic model that is compatible with this framework.

ture is the exploitation of the sparse system matrices, leading to computational efficiency. WFSim will be compared to high-fidelity flow data and used in a practical control application.

The remainder of this paper is organized as follows. In Sect. 2, the mathematical background of the medium-fidelity wind farm model will be explained including a discussion on the rotor and turbulence model. This section ends with an analysis regarding the wind farm model's computation time. In Sect. 3, WFSim will be validated in two cases using flow velocities in the longitudinal and lateral directions at hub height and turbine power signals computed with two different LES-based wind farm models. The first case considered is a two-turbine wind farm with turbines modelled using the ADM. The second case is a nine-turbine wind farm with turbines modelled employing Fatigue, Aerodynamics, Structures and Turbulence (FAST) (Jonkman and Buhl, 2005). This paper is concluded in Sect. 4.

2 Formulation of a dynamic control-oriented wind farm model

In the current section, a simplified wind farm model is formulated that is sufficiently fast for online control but retains some of the elemental features of three-dimensional turbulent flows. In order for the model to be fast, we envisage a 2-D-like model, but adapted to account for three-dimensional flow relaxation. We will dub the resulting model WFSim (WindFarmSimulator).

As a starting point we use the standard incompressible three-dimensional filtered Navier–Stokes equations, as used in LES, i.e.

$$
\frac{\partial \widetilde{\boldsymbol{v}}}{\partial t} + (\widetilde{\boldsymbol{v}} \cdot \nabla)\widetilde{\boldsymbol{v}} + \nabla \cdot \boldsymbol{\tau}_{\mathrm{M}} + \frac{1}{\rho}\nabla \widetilde{p} - \boldsymbol{f} = 0,
$$

momentum equation,

$$
\nabla \cdot \widetilde{\boldsymbol{v}} = 0,
$$

continuity equation. $\qquad$ (1)

The velocity field $\widetilde{\boldsymbol{v}} = (\widetilde{v}_1, \widetilde{v}_2, \widetilde{v}_3)^T$ and pressure field $\widetilde{p}$ represent filtered variables; $\nabla = (\partial/\partial x, \partial/\partial y, \partial/\partial z)^T$; the air density ρ, which is assumed to be constant; and $\boldsymbol{\tau}_{\mathrm{M}}$ represents the subgrid-scale model, which will be defined in Sect. 2.1. As is common in LESs of high-Reynolds-number atmospheric simulations with grid resolutions in the metre range, direct effects of viscous stresses on the filtered fields are negligible, so that these terms are left out. Finally, the term $\boldsymbol{f}$ represents the effect of turbines on the flow, as further detailed in Sect. 2.2.

Although LES filters are usually implicitly tied to the LES grid and filter length scale in the subgrid-scale model, we presume here that $\widetilde{\boldsymbol{v}}$ corresponds to a top-hat filtered velocity field, with filter width D, where D is the turbine diameter. Thus,

$$
\widetilde{\boldsymbol{v}}(x, y, z) = \frac{1}{D^3} \int\limits_{z-D/2}^{z+D/2} \int\limits_{y-D/2}^{y+D/2} \int\limits_{x-D/2}^{x+D/2} \boldsymbol{v}(x', y', z') \, \mathrm{d}x'\mathrm{d}y'\mathrm{d}z'. \quad (2)
$$

From a wind farm simulation perspective, we are mainly interested in the flow velocity field at hub height z_{h}, i.e. $\widetilde{\boldsymbol{v}}(x, y, z_{\mathrm{h}})$. Moreover, to evaluate turbine forces and power, it suffices to know the velocity at turbine locations $\boldsymbol{t}_n =$

$(x_n, y_n)^T$ (with $n = 1 \cdots \aleph$ and $\aleph$ the number of turbines in the farm) since $\widetilde{\boldsymbol{v}}(x_n, y_n, z_{\mathrm{h}})$ is a reasonable representation of the turbine disk-averaged velocity.

Therefore, we focus on formulating a 2-D-like set of equations for $\widetilde{\boldsymbol{v}}(x, y, z_{\mathrm{h}})$. To this end, we define

$$\boldsymbol{u} = \begin{pmatrix} \widetilde{v}_1(x, y, z_{\mathrm{h}}) & \widetilde{v}_2(x, y, z_{\mathrm{h}}) \end{pmatrix}^T, \tag{3}$$

$$= \begin{pmatrix} u & v \end{pmatrix}^T, \tag{4}$$

and $w = \widetilde{v}_3(x, y, z_{\mathrm{h}})$ and $p = \widetilde{p}(x, y, z_{\mathrm{h}})/\rho$. Moreover, we assume that $w \approx 0$, so that the LES equations given in Eq. (1) can be reformulated in terms of $\boldsymbol{u}$ as

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla_{\mathrm{H}})\boldsymbol{u} + \nabla_{\mathrm{H}} \cdot \boldsymbol{\tau}_{\mathrm{H}} + \nabla_{\mathrm{H}} p - \boldsymbol{f}$$
$$= -\frac{\partial(uw + \tau_{\mathrm{M},13})}{\partial z}\boldsymbol{e}_1 - \frac{\partial(vw + \tau_{\mathrm{M},23})}{\partial z}\boldsymbol{e}_2, \tag{5}$$

$$\nabla_{\mathrm{H}} \cdot \boldsymbol{u} = -\frac{\partial w}{\partial z}, \tag{6}$$

with $\nabla_{\mathrm{H}} = (\partial/\partial x, \partial/\partial y)^T$, $\boldsymbol{\tau}_{\mathrm{H}}$ a 2-D tensor containing the horizontal components of the subgrid stresses $\boldsymbol{\tau}_{\mathrm{M}}$, and $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ the unit vectors in the x and y directions, respectively.

Finally, we further simplify the equations above using two additional assumptions. First of all, we presume $\partial w/\partial z \approx \partial v/\partial y$. When centred at the turbine axis, this is one of the conditions required for axial symmetry, though axial symmetry also requires further conditions on $\partial v/\partial z$ and $\partial w/\partial y$, which are not imposed. In general, $\partial w/\partial z \approx \partial v/\partial y$ implies equal divergence–convergence of streamlines in y and z directions. Although this is a very simple condition, we presume it to be good enough to resolve the lack of relaxation of purely 2-D models. If necessary, a more general form ($w \approx 0$ and $\partial w/\partial z \approx c \cdot \partial v/\partial y$), with c a tuning parameter (e.g. obtained through state estimation) could be considered, but results in the current work indicate that this may not be necessary. Secondly, we simply neglect the right-hand side of Eq. (5). Though this is a rather crude assumption, the rationale is that the modelling term $\boldsymbol{\tau}_{\mathrm{H}}$ will suffice in the context of a control model, where model coefficients can be updated online based on feedback (see also the discussion in Sect. 2.1). Hence our final 2-D-like model corresponds to

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla_{\mathrm{H}})\boldsymbol{u} + \nabla_{\mathrm{H}} \cdot \boldsymbol{\tau}_{\mathrm{H}} + \nabla_{\mathrm{H}} p - \boldsymbol{f} = 0, \tag{7}$$

$$\nabla_{\mathrm{H}} \cdot \boldsymbol{u} = -\frac{\partial v}{\partial y}. \tag{8}$$

We emphasize here that the model above is not a classical 2-D model due to the difference in formulation of the continuity equation. In contrast to a standard 2-D model, this allows for flow relaxation in the third direction when, for example, encountering slow down by a wind turbine. This can be seen in Fig. 2, in which simulation results obtained with the model above, a standard 2-D dynamic wind farm model and LES are shown. The simulation case itself will
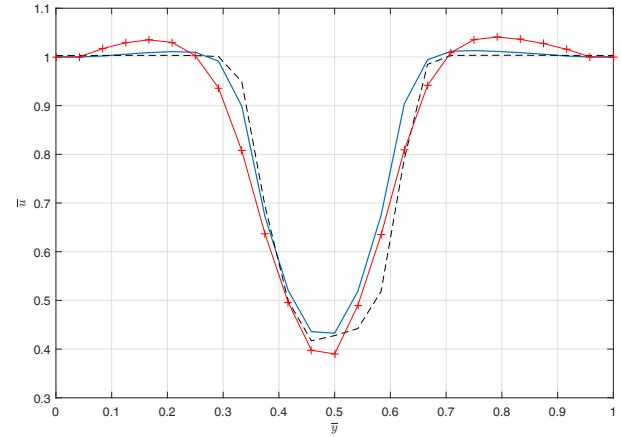
Figure 2. Results of two-turbine simulations. Normalized time-averaged wake deficit at hub height $5D$ downwind from the downwind turbine using standard 2-D Navier–Stokes equations (red crossed), our model with the adapted continuity equation (blue), and LES data (black dashed).

be discussed in more detail in Sect. 3.2.1. Here we depict the normalized flow deficit in the wake at $5D$ downstream of the turbine along the cross-stream axis. The figure illustrates that the standard 2-D Navier–Stokes equations lead to a significant speed up at the wake edges. This is a result from conservation of mass in two dimensions and the flow deceleration in front of the turbine, pushing part of the air around the turbine. In the WFSim model, this speed up is smaller, as mass can also flow around the turbine in the third dimension. In Fig. 2 it can be seen that LES data are better estimated when imposing flow relaxation in the third dimension. Finally, note that partially modelling the missing vertical dimension as proposed above is novel with respect to the work presented in Boersma et al. (2016a).

This section is further organized as follows. First, in Sect. 2.1, the subgrid-scale model will be introduced. Then, in Sect. 2.2, the turbine model will be explained. The discretization of the equations is presented in Sect. 2.4, and boundary and initial conditions are discussed in Sect. 2.5.

2.1 Turbulence model

In the literature, many subgrid-scale models are documented, and to date, model accuracy remains a challenge in LES research (see e.g. Sagaut, 2006). However, in the current paper, an important factor in the selection of a model is simplicity and computational efficiency, rather than accuracy. In fact, in contrast to conventional modelling, in a control-oriented model completeness of the turbulence model is not a major issue since unknown model coefficients can be calibrated online using measurements and feedback (Shapiro et al., 2017b), thus also controlling the overall error. Therefore, in this work we fall back to one of the simplest and first known turbulence models, Prandtl's mixing length model.
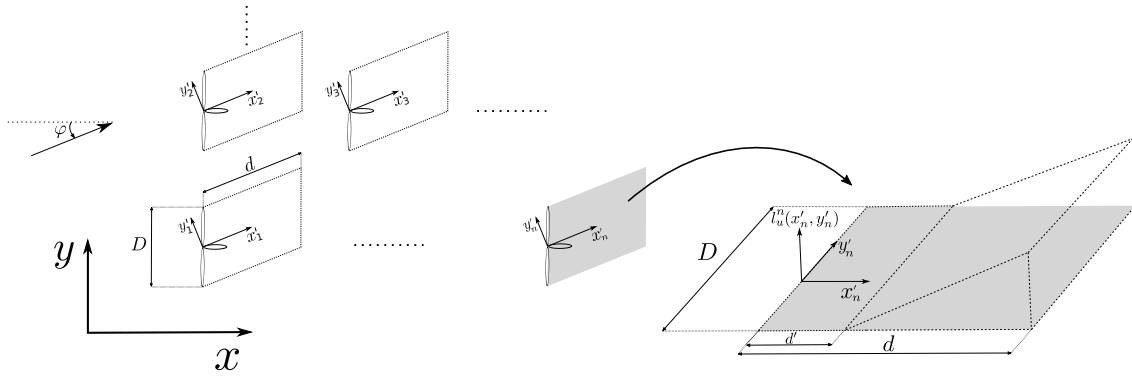
Figure 3. Schematic illustration of the mixing length.

We formulate the stress tensor $\boldsymbol{\tau}_H$ using an eddy-viscosity assumption, i.e.

$$\boldsymbol{\tau}_H = -\nu_t \boldsymbol{S}, \tag{9}$$

with $\boldsymbol{S} = \frac{1}{2}(\nabla_H \boldsymbol{u} + (\nabla_H \boldsymbol{u})^T)$ the 2-D rate-of-strain tensor, and ν_t the eddy viscosity. The latter is further modelled as in Prandtl (1925):

$$\nu_t = l_u(x, y)^2 \left| \frac{\partial u}{\partial y} \right|, \tag{10}$$

where $l_u(x, y)$ is the mixing length. It could be interesting to define the mixing length for each position in the wind farm separately, but this will lead to too many tuning variables. Moreover, in Iungo et al. (2015), the authors illustrate that in a turbine's near wake the mixing length is roughly invariant for different downstream locations, but in the far wake, the mixing length increases linearly with downstream distance. We use this to formulate a simple heuristic parametrization for the mixing length model so that the number of decision variables will be reduced drastically. From now on we assume that the wind is coming from the east, but can have a direction defined by φ. Then, the wind farm will be divided in segments as illustrated in Fig. 3.

Each segments has its own (x'_n, y'_n) coordinate system located in the global (x, y) coordinate system. Now we propose the following mixing length parametrization:

$$l_u(x, y) = \begin{cases} G(x'_n, y'_n) * l_u^n(x'_n, y'_n), & \text{if } x \in \mathcal{X} \text{ and } y \in \mathcal{Y}. \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

with $G(x, y)$ a (smoothing) pillbox filter with radius 3, $*$ the 2-D spatial convolution operator, $\mathcal{X} = \{x : x'_n \leq x \leq x'_n + \cos(\varphi)d\}$, $\mathcal{Y} = \{y : y'_n - \frac{D}{2} + \sin(\varphi)x'_n \leq y \leq y'_n + \frac{D}{2} + \sin(\varphi)x'_n\}$ and φ defined as the mean wind direction (see Fig. 3), which we bound by $|\varphi| \leq 45°$ in this work. In addition we constraint d by $\cos(\varphi)d \leq |x_q - x_n|$, with x_n a turbine x coordinate and x_q its downwind turbines x coordinate. We can see $l_u^n(x'_n, y'_n)$ as the local mixing length that belongs to

turbine n and denote it as

$$l_u^n(x'_n, y'_n) = \begin{cases} (x'_n - d')l_s, & \text{if } x'_n \in \mathcal{X}'_n \text{ and } y'_n \in \mathcal{Y}'_n. \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

with $\mathcal{X}'_n = \{x'_n : d' \leq x'_n \leq d\}$ and $\mathcal{Y}'_n = \{y'_n : |y'_n| \leq D\}$ and tuning parameter l_s that defines the slope of the (linearly increasing) local mixing length parameter. In fact, this parameter could be related to turbulence intensity, i.e., the amount of wake recovery. In this work we will not investigate this relation further. With the formulation above, the number of tuning variables that belong to the turbulence model (l_s, d, d') is reduced to $3\aleph$. Additionally, we assume that l_s, d and d' are equal for each turbine in the farm, which reduces the number of tuning variables that belong to the turbulence model to three, a quantity that could be dealt with by an online estimator. However, in order to have only three tuning variables, the included turbulence model is defined as a simplified mixing length model found heuristically using and adapting information from Iungo et al. (2015).

2.2 Turbine model

Turbines are modelled using a classical non-rotating actuator disk model (ADM). In this method, each wind turbine is represented by a uniformly distributed force acting on the grid points where the rotor disk is located. Figure 4 depicts a schematic top-view representation of a turbine with yaw angle γ.

Using such an approach, the force exerted by the turbines can be expressed as

$$\boldsymbol{f} = \sum_{n=1}^{\aleph} \boldsymbol{f}_n, \quad \text{with}$$

$$\boldsymbol{f}_n = \frac{c_f}{2} C'_{T_n} [U_n \cos(\gamma_n)]^2 \begin{pmatrix} \cos(\gamma_n + \varphi) \\ \sin(\gamma_n + \varphi) \end{pmatrix}$$

$$\times \text{H} \left[\frac{D}{2} - ||\boldsymbol{s} - \boldsymbol{t}_n||_2 \right] \delta \left[(\boldsymbol{s} - \boldsymbol{t}_n) \cdot \boldsymbol{e}_{\perp, n} \right], \tag{13}$$
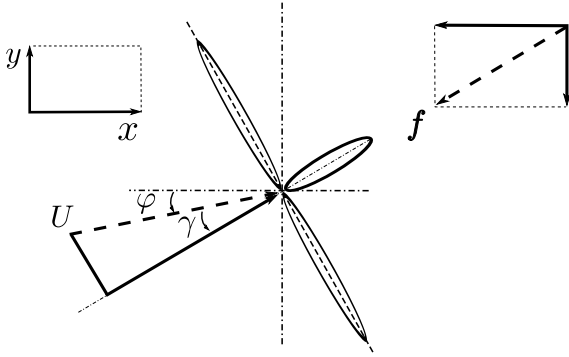
Figure 4. Schematic representation of a turbine with yaw angle γ_n and flow velocity $U = ([u(x_n, y_n)]^2 + [v(x_n, y_n)]^2)^{1/2}$ at the rotor. Figure adapted from Jiménez et al. (2010).

with $\boldsymbol{s} = (x, y)^T$, H[·] the Heaviside function and $\delta[·]$ the Dirac delta function, $\boldsymbol{e}_{\perp,n}$ the unit vector perpendicular to the nth rotor disk with position $\boldsymbol{t}_n$. Furthermore, we have C'_{T_n} the disk-based thrust coefficient following Meyers and Meneveau (2010), which can be expressed in terms of the classical thrust coefficient C_{T_n} using the following relation: $C'_{T_n} = C_{T_n}/(1-a_n)^2$, with a_n the axial induction factor of the nth turbine. Interestingly, the coefficient C'_{T_n} can be directly related to the turbine set point in terms of blade pitch angle and rotational speed (see, e.g. Appendix A in Goit and Meyers, 2015). In the WFSim model, C'_{T_n} and yaw angle γ_n are considered as the control variables and can thus be used to regulate the wakes and hence wind farm performance. Furthermore, the scalar c_f in Eq. (13) can be regarded as a tuning variable and will in this work be set equal for all turbines in the farm.

2.3 Power

From the resolved flow velocity components, the power generated by the farm is computed as

$$P = \sum_{n=1}^{\aleph} \frac{1}{2} \rho A \, C_{P_n} [U_n \cos(\gamma_n)]^3. \tag{14}$$

It is stated in Goit and Meyers (2015) (Appendix A) that when there is no drag and swirl is added to the wake, $C'_{T_n} = C_{P_n}$. Since this is an idealized situation, a loss factor will be introduced such that $C_{P_n} = c_p C'_{T_n}$. The scalar c_p can be seen as a tuning variable and will be set equal for all turbines in the farm. In the power expression above, we have the factor $\cos(\gamma_n)^3$ with exponent 3. In literature such as Gebraad et al. (2014a) and Medici (2005, p. 37), for example, numerical values for the exponent were given according to LES and wind tunnel data, respectively. However, to date, the exact value for it is still under research and since this is outside the scope of this study, the value of the exponent will be 3.

This concludes the formulation of the WFSim model. In order to resolve for flow velocity components and wind farm
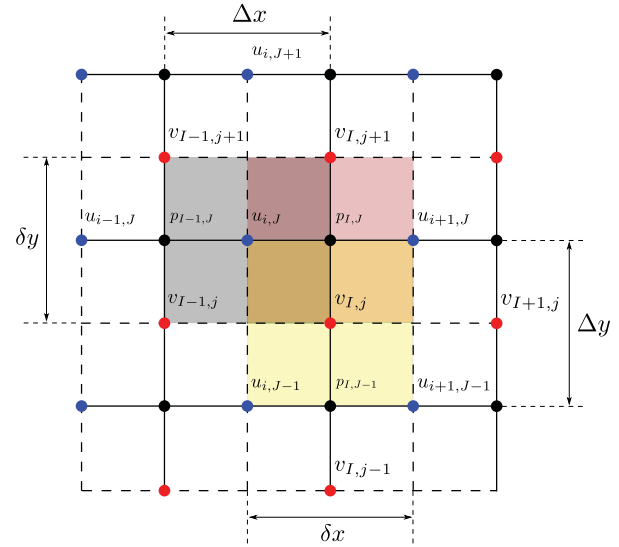

Figure 5. One cell for the x momentum equation (grey, $u_{i,J}$), one for the y momentum equation (yellow, $v_{I,j}$) and one for the continuity equation (pink, $p_{I,J}$). All three cells have equal dimensions and overlap.

power, the governing equations given in Eqs. (7) and (8) need to be discretized, a topic that will be discussed in the following section.

2.4 Discretization

The set of equations are spatially discretized over a staggered grid following Versteeg and Malalasekera (2007). This is carried out by employing the finite volume method and the hybrid differencing scheme. Temporal discretization is performed using the implicit method that is unconditionally stable (Versteeg and Malalasekera, 2007). This comes down to deriving the integrals:

$$\int\limits_{\Delta t} \int\limits_{\Delta V} \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla_H)\boldsymbol{u} + \nabla_H \cdot \boldsymbol{\tau}_H + \nabla_H p - \boldsymbol{f} \right] dV \, dt = 0,$$

$$\int\limits_{\Delta t} \int\limits_{\Delta V} \left[\nabla_H \cdot \boldsymbol{u} + \frac{\partial v}{\partial y} \right] dV \, dt = 0, \tag{15}$$

with ΔV the volume of one cell (see Fig. 5) and Δt the sample period. One obtains, for each cell, the following fully discretized Navier–Stokes equations (for detailed derivation we refer to Appendix A):

– x momentum equation for the (i, J)th cell (black in Fig. 5):

$$a_{i,J}^{px} u_{i,J} = \begin{pmatrix} a_{i,J}^{nx} & a_{i,J}^{sx} & a_{i,J}^{wx} & a_{i,J}^{ex} \end{pmatrix}$$
$$\times \begin{pmatrix} u_{i,J+1} & u_{i,J-1} & u_{i-1,J} & u_{i+1,J} \end{pmatrix}^T$$
$$- \delta y_{j,j+1} \left(p_{I,J} - p_{I-1,J} \right) + f_{i,J}^x + \dots$$

$$\ldots + \begin{pmatrix} a_{i,J}^{nwx} & a_{i,J}^{swx} & a_{i,J}^{nex} & a_{i,J}^{sex} \end{pmatrix}$$
$$\times \begin{pmatrix} v_{I-1,j+1} & v_{I-1,j} & v_{I,j+1} & v_{I,J} \end{pmatrix}^T \tag{16}$$

– y momentum equation for the (I, j)th cell (yellow in Fig. 5):

$$a_{I,j}^{py} v_{I,j} = \begin{pmatrix} a_{I,j}^{ny} & a_{I,j}^{sy} & a_{I,j}^{wy} & a_{I,j}^{ey} \end{pmatrix}$$
$$\times \begin{pmatrix} v_{I,j+1} & v_{I,j-1} & v_{I-1,j} & v_{I+1,j} \end{pmatrix}^T$$
$$- \delta x_{i,i+1} (p_{I,J} - p_{I,J-1}) + f_{I,j}^y + \ldots$$
$$\ldots + \begin{pmatrix} a_{i,J}^{nwy} & a_{i,J}^{swy} & a_{i,J}^{ney} & a_{i,J}^{sey} \end{pmatrix}$$
$$\times \begin{pmatrix} u_{i,J} & u_{i,J-1} & u_{i+1,J} & u_{i+1,J-1} \end{pmatrix}^T \tag{17}$$

– continuity equation for the (I, J)th cell (pink in Fig. 5):

$$0 = \delta y_{j,j+1} (u_{i+1,J} - u_{i,J}) + 2\delta x_{i,i+1} (v_{I,j+1} - v_{I,j}). \tag{18}$$

The states $u_{\bullet,\bullet}$, $v_{\bullet,\bullet}$, $p_{\bullet,\bullet}$ are defined for the time $k+1$ while the coefficients $a_{\bullet,\bullet}^{\bullet}$ and the forcing terms $f_{\bullet,\bullet}^{\bullet}$ depend on the state at time k. Detailed definitions of these coefficients are given in Appendix A, Table A1. Note the appearance of the previously explained factor 2 (see Eq. 8) in Eq. (18).

Next, the state vectors u_k, v_k, and p_k and control variable vectors v_k and γ_k at time step k will be defined:

$$u_k = \begin{pmatrix} u_{3,2} \\ \vdots \\ u_{3,N_y-1} \\ u_{4,2} \\ \vdots \\ u_{4,N_y-1} \\ \vdots \\ u_{N_x-1,2} \\ \vdots \\ u_{N_x-1,N_y-1} \end{pmatrix}, \quad v_k = \begin{pmatrix} v_{2,3} \\ \vdots \\ v_{2,N_y-1} \\ v_{3,3} \\ \vdots \\ v_{3,N_y-1} \\ \vdots \\ v_{N_x-1,3} \\ \vdots \\ v_{N_x-1,N_y-1} \end{pmatrix},$$

$$p_k = \begin{pmatrix} p_{2,2} \\ \vdots \\ p_{2,N_y-1} \\ p_{3,2} \\ \vdots \\ p_{3,N_y-1} \\ \vdots \\ p_{N_x-1,3} \\ \vdots \\ p_{N_x-1,N_y-2} \end{pmatrix}, \quad v_k = \begin{pmatrix} C_{T_1}' \\ C_{T_2}' \\ \vdots \\ C_{T_\aleph}' \end{pmatrix}, \quad \gamma_k = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_\aleph \end{pmatrix}, \tag{19}$$

with N_x and N_y the number of cells in the x and y directions, respectively, and $\aleph$ the number of turbines in the wind farm.

Each component in u_k, v_k and p_k represents a flow velocity and pressure at a point in the field defined by the subscript. For clarity reasons, an example of a staggered grid is depicted in Fig. 6.

2.5 Boundary and initial conditions

All the components that are not contained in the vector, u_k, v_k and p_k, but do appear in the staggered grid need to be defined. For the flow velocity components, first-order conditions on the west side of the grid are prescribed assuming the wind is coming from the east. These Dirichlet inflow boundary conditions are related to the ambient inflow defined as u_b and v_b and can vary over time. Zero stress (also referred to as Neumann) boundary conditions are prescribed on the other boundaries. Therefore, for the flow velocity components on the boundaries we define

$$u_{2,J} = u_b \qquad \text{for } J = 1, 2, \ldots, N_y,$$
$$u_{i,N_y} = u_{i,N_y-1} \qquad \text{for } i = 3, 4, \ldots, N_x,$$
$$u_{i,1} = u_{i,2} \qquad \text{for } i = 3, 4, \ldots, N_x,$$
$$u_{N_x,J} = u_{N_x-1,J} \qquad \text{for } J = 2, 3, \ldots, N_x - 1,$$

$$v_{1,j} = v_b \qquad \text{for } j = 2, 3, \ldots, N_y,$$
$$v_{I,N_y} = v_{I,N_y-1} \qquad \text{for } I = 2, 3, \ldots, N_x,$$
$$v_{I,2} = v_{I,3} \qquad \text{for } I = 2, 3, \ldots, N_x,$$
$$v_{N_x,j} = v_{N_x-1,j} \qquad \text{for } j = 3, 4, \ldots, N_y - 1.$$

For the initial conditions, we define all longitudinal and latitudinal flow velocity components in the field as u_b and v_b, respectively, which are the boundary velocity values. The initial pressure field is set to zero. Note that by defining the boundary conditions as given above, the assumption is that the wind is coming from the east in Fig. 6, which coincides with the definition of the mixing length (see Sect. 2.1). Finally, the equations given in Eqs. (7) and (8) can be transformed to the difference algebraic equation[1]:

$$\underbrace{\begin{pmatrix} A_x(u_k, v_k) & A_{xy}(u_k) & B_1 \\ A_{yx}(u_k) & A_y(u_k, v_k) & B_2 \\ B_1^T & 2B_2^T & 0 \end{pmatrix}}_{E(q_k)} \underbrace{\begin{pmatrix} u_{k+1} \\ v_{k+1} \\ p_{k+1} \end{pmatrix}}_{q_{k+1}}$$
$$= \underbrace{\begin{pmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u_k \\ v_k \\ p_k \end{pmatrix}}_{q_k} + \underbrace{\begin{pmatrix} b_1(u_k, v_k, v_k, \gamma_k) \\ b_2(u_k, v_k, v_k, \gamma_k) \\ b_3 \end{pmatrix}}_{b(q_k, w_k)}, \tag{20}$$

with $n_q = n_u + n_v + n_p$ and $u_k \in \mathbb{R}^{n_u}$, $v_k \in \mathbb{R}^{n_v}$, $p_k \in \mathbb{R}^{n_p}$ containing all flow velocities in the longitudinal and lateral

---

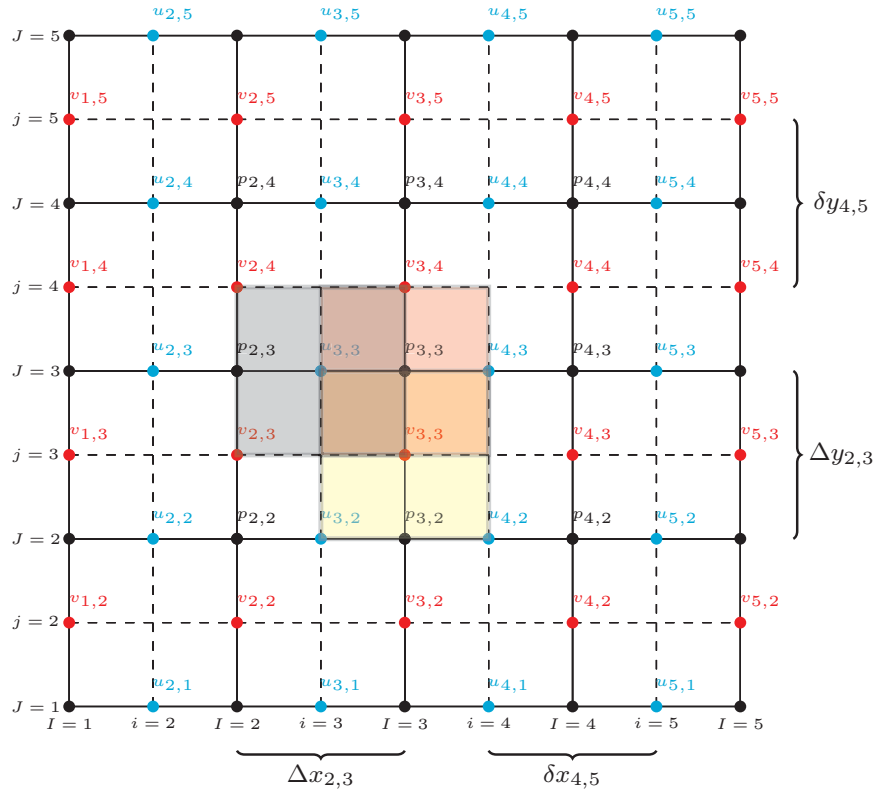[1]This type of system can also be referred to as a quasi-linear parameter varying model or descriptor model.

Figure 6. Example of a staggered grid with cells each having volume ΔV. In WFSim, the grid is quadrilateral.

Table 1. Mean computation time per simulation time step Δt^{cpu} vs. number of states n_q for the WFSim representation as given in Eq. (21). Computations are performed on a regular notebook with one core.

n_q	Δt^{cpu} (s)	n_q	Δt^{cpu} (s)	n_q	Δt^{cpu} (s)	n_q	Δt^{cpu} (s)
3×10^3	0.02	27×10^3	0.22	115×10^3	1.19	239×10^3	3.1
6×10^3	0.04	43×10^3	0.37	147×10^3	1.66	258×10^3	3.5
9×10^3	0.06	64×10^3	0.60	182×10^3	2.12	268×10^3	3.7
14×10^3	0.10	88×10^3	0.88	221×10^3	2.50	276×10^3	3.8

direction and the pressure vector at time k and control variable $w_k = \left(v_k^T \ \gamma_k^T \right)^T \in \mathbb{R}^{2\aleph}$. The non-singular square descriptor matrix $E(q_k)$ contains the coefficients $a_{\bullet,\bullet}^\bullet$, appearing in Eqs. (16) and (17), that depend on the state at time k. The square constant matrix A solely depends on grid spacing and sample period Δt. Note that the state vector contains three states for every cell; hence an increase in grid resolution results in an increase in matrix dimensions. However, the system matrices that occur in Eq. (20) are sparse and efficient numerical solvers are available for these types of problems. This will be demonstrated in Sect. 2.6. The vector $b(q_k, w_k)$ contains the forcing terms (turbines) and boundary conditions.

By defining $N_x, N_y, \Delta x_{I,I+1}, \Delta y_{J,J+1}$, and the turbine positions, a wind farm topology is determined. Next, ambient flow conditions u_b and v_b, tuning parameters c_f, c_p, d, d', l_s

and the control variable w_k need to be specified. The system given in Eq. (20) is then fully defined and can be solved.

2.6 Computation time

When discretizing partial differential equations, a trade-off has to be made between the computation time and grid resolution. Typically, a higher resolution results in more precise computation of the variables but also increasing computation time. In WFSim, computational cost is reduced by exploiting sparsity and by applying the reverse Cuthill–McKee algorithm (George and Liu, 1981).[2] The latter is applicable due to the fact that the matrix structure is fixed. The inter-

---

[2] The sparse toolbox and reverse Cuthill–McKee algorithm are both utilized in MATLAB.

Table 2. Summary of the WFSim simulation set-up.

Domain size $L_x \times L_y$	$2 \times 0.63 \,(\text{km}^2)$	Turbine rotor diameter D	$126.4 \,(\text{m})$
Grid size $N_x \times N_y$	50×25	Turbine arrangement	2×1
Cell size $\Delta x \times \Delta y$	$40 \times 23 \,(\text{m}^2)$	Turbine spacing	$5D$
Times	$\Delta t = 1, \Delta t^{\text{cpu}} = 0.02 \,(\text{s})$	Atmospheric conditions	$u_\text{b} = 8, v_\text{b} = 0 \,(\text{m s}^{-1}), \rho = 1.2 \,(\text{kg m}^{-3})$
Force and power factor	$c_f = 1.7, c_p = 0.95$	Turbulence model	$d = 530, d' = 122 \,(\text{m}) \; l_s = 0.06$
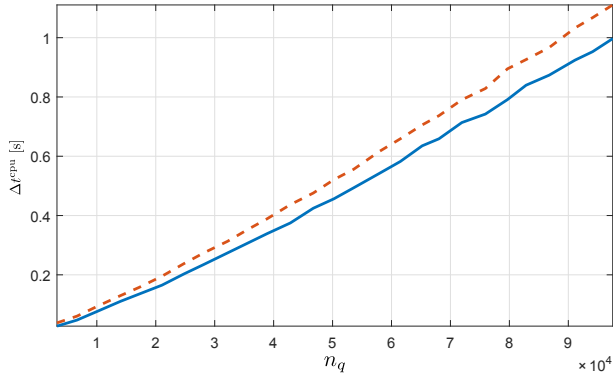
Figure 7. Mean computation time per simulation time step Δt^{cpu} vs. number of states n_q. The red dashed line is WFSim as presented in Eq. (20) and blue is WFSim as presented in Eq. (21). Note that the number of cells is approximately $n_q/3$ with n_q the number of states.

ested reader is referred to Doekemeijer et al. (2016) for more information on the Cuthill–McKee algorithm in WFSim.

In this section, the mean computation time needed for one time step Δt^{cpu} will be analysed. The presented results are obtained on a regular notebook with an Intel Core i7-4600U 2.7 GHz processor employing one core and MATLAB. Since the objective is to do online control, i.e., it is desired to reduce computational complexity, this section introduces a second WFSim representation. The first representation was given in Eq. (20) while the second is defined as

$$
\underbrace{\begin{pmatrix} A_x(u_k, v_k) & 0 & B_1 \\ 0 & A_y(u_k, v_k) & B_2 \\ B_1^T & 2B_2^T & 0 \end{pmatrix}}_{E(q_k)} \underbrace{\begin{pmatrix} u_{k+1} \\ v_{k+1} \\ p_{k+1} \end{pmatrix}}_{q_{k+1}}
$$
$$
= \underbrace{\begin{pmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u_k \\ v_k \\ p_k \end{pmatrix}}_{q_k} + \underbrace{\begin{pmatrix} b_1(u_k, v_k, v_k, \gamma_k) \\ b_2(u_k, v_k, v_k, \gamma_k) \\ b_3 \end{pmatrix}}_{b(q_k, w_k)}. \quad (21)
$$

The difference can be found in the descriptor matrix. In the representation above, the elements $A_{xy}(u_k), A_{yx}(u_k)$ that occur in Eq. (20) are set to zero. This can be justified by the fact that their contribution is negligible since these matrices contain elements that, for our case studies, are of the order of $\mathcal{O}(1)$ while the elements in $A_x(u_k, v_k)$ and $A_y(u_k, v_k)$ are

of the order of $\mathcal{O}(3)$. Consequently, no significant change in the computed flow field has been observed, but a decrease in Δt^{cpu} has (see Fig. 7). Therefore, the remainder of this paper will continue with the WFSim representation given in Eq. (21). Table 1 depicts more numerical values of Δt^{cpu} for this WFSim representation.

From Table 1 we can conclude that Δt^{cpu} increases between quadratic and linear with respect to the number of states n_q for $n_q < 221 \times 10^3$. It depends on the computer properties how much you can increase the number of states until the CPU is out of memory.

3 Simulation results

In this section, WFSim flow and power data will be compared with LES data and it is organized as follows. In Sect. 3.1, quality measures are introduced. In Sect. 3.2.1, WFSim data are compared with PALM data, and in Sect. 3.2.2 WFSim is validated against SOWFA data. In both simulation cases, the thrust coefficients C_T' are varied while the yaw angles are set to zero.

3.1 Quality measures

Suppose we have at time k a measurement of one quantity $z_k \in \mathbb{R}^N$ and its estimation $\hat{z}_k \in \mathbb{R}^N$. Define the prediction error $e_k = \hat{z}_k - z_k$. The quality measure RMSE is, for time step k, defined as

$$
\text{RMSE}(z_k, \hat{z}_k) = \sqrt{\frac{1}{N} e_k^T e_k}. \quad (22)
$$

This measure is used to compare the flow centreline velocity $U_k^c(x)$ and power signals from LES and WFSim data for different model parameters. The flow centreline is, for one time step, defined as the laterally averaged longitudinal flow velocity throughout the simulation domain across the rotor diameter. Mathematically this can, for LES data at time step k at longitudinal position x_i, be defined as

$$
U_k^c(x_i) = \frac{1}{N_{\bar{y}}} \sum_{s=1}^{N_{\bar{y}}} u_k(x_i, y_s), \quad (23)
$$

with y_s the y coordinate of one cell across the line $\bar{y} \subset y$, which contains $N_{\bar{y}}$ number of cells and has a length equal to the rotor diameter. From WFSim data, the flow velocity component at the rotor centre will be taken across the position x.
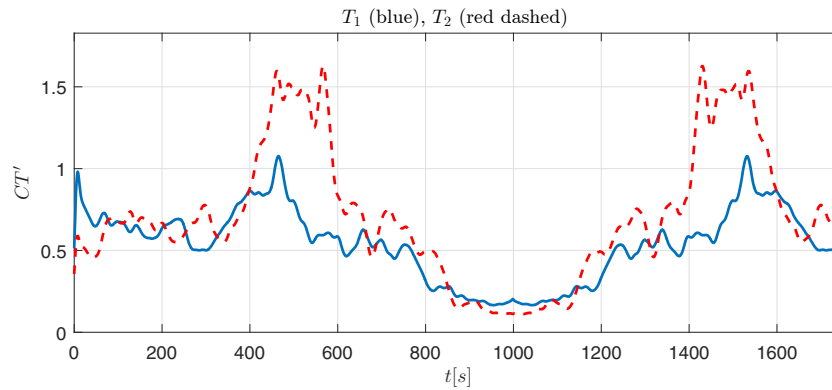
Figure 8. Excitation signals for the two-turbine simulation case. The yaw angles are set to zero.
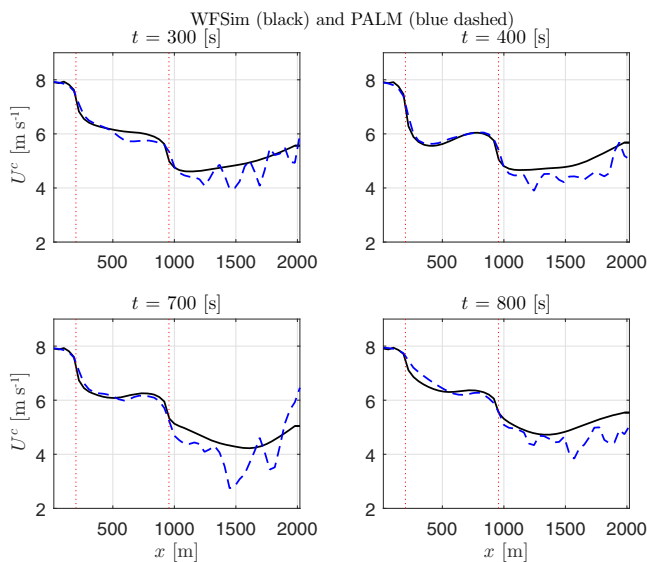

Figure 9. Mean flow centreline at four time instances through the farm. The vertical red dashed lines indicate the positions of the turbines.

In this work we compare lateral and longitudinal flow velocity components at hub height and power signals calculated with LES with lateral and longitudinal flow velocity components and power signals calculated with WFSim.[3]

3.2 Axial induction actuation

Studies such as Shapiro et al. (2017a), Munters and Meyers (2017), Vali et al. (2017) and van Wingerden et al. (2017) illustrate that axial induction actuation can be used in active power control where the objective is to provide grid facilities. In order to utilize the WFSim model in active power control, it is important to first validate it when exciting the thrust coefficient.

---

[3]The LES flow data are mapped onto the grid of WFSim using bilinear interpolation techniques.

In the following, WFSim is compared with simulation data from PALM (Maronga et al., 2015) and SOWFA (Church-field et al., 2012), both high-fidelity wind farm models that were briefly discussed in Sect. 1. The latter includes the actuator line model (ALM) while the former employs the ADM.[4]

3.2.1 PArallelized LES Model (PALM) and WFSim

PALM predicts the 3-D flow velocity vectors and turbine power signals in a wind farm using LES and is based on the 3-D incompressible Navier–Stokes equations.[5] Table 2 gives a summary of the two-turbine wind farm simulated in WFSim. A summary of the PALM simulation set-up can be found in Appendix B. The applied control signals are depicted in Fig. 8 and are chosen such that different system dynamics are excited. The final values for the tuning parameters are obtained using a grid search. Figures 9 and 10 show a comparison of the mean flow centreline and the wind farm power, respectively. A flow field evaluated with both the WFSim model and PALM can be found in Appendix B.

In Fig. 9, the mean flow centrelines through the farms of WFSim and PALM are relatively similar. The PALM data exhibit more turbulent fluctuations due to the presence of a more sophisticated turbulence model, which allows for better capturing small-scale dynamics such as turbine-induced turbulence. However, the WFSim model is capable of estimating wake recovery similar to the PALM model. The recovery in the WFSim model is due to the turbulence model as presented in Sect. 2.1. It is in fact the slope of the local mixing length parameters that can determine the amount of wake recovery, or more precisely, the larger this slope, the higher the wake recovery. It is therefore interesting to link this tuning variable to the turbulence intensity in the farm. Furthermore, it can be seen in Fig. 10 that the WFSim model

---

[4]PALM also includes the rotating ADM, but in our case study the ADM is employed.

[5]In this work we consider PALM as a wind farm model since PALM is simulated with turbine models. However, PALM is also applicable for simulating oceanic behaviour.
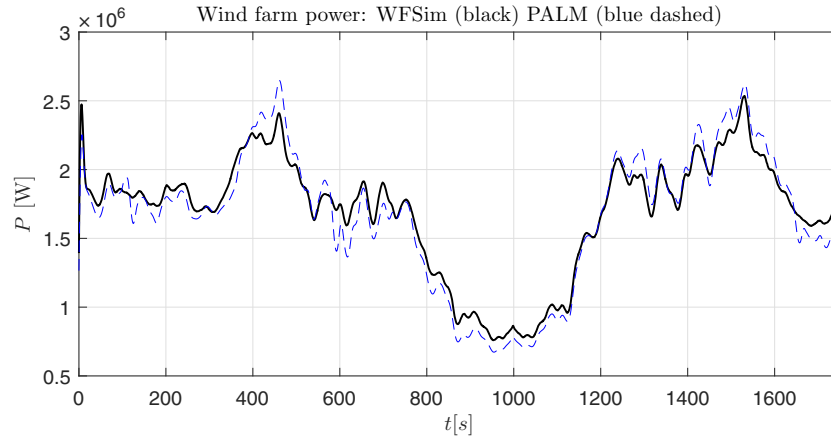
Figure 10. Wind farm power from PALM (blue dashed) and WFSim (black).
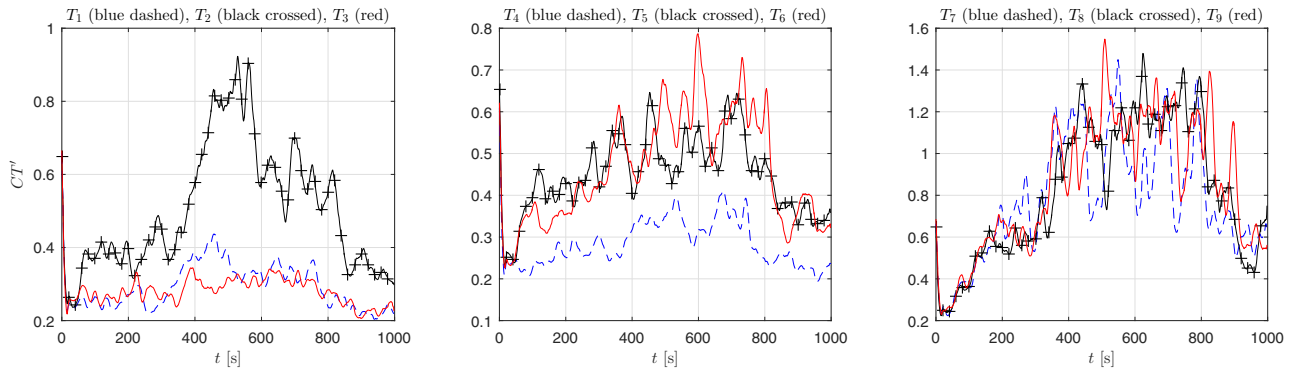

Figure 11. Excitation signals for the nine-turbine simulation case. The yaw angles are set to zero.

is capable of estimating the wind farm power. Since both the WFSim model and PALM employ the ADM, fast fluctuations in the power signal can be observed. This is due to the lack of rotor inertia in both simulation cases. The simulation case presented in this section illustrates that the WFSim model, in which the third dimension is partially neglected, is able to estimate wind farm flow and power signals computed with a 3-D LES wind farm model. In Sect. 3.2.2, a SOWFA case study will be presented, a LES model that includes turbine dynamics.

3.2.2 Simulator fOr Wind Farm Applications (SOWFA) and WFSim

SOWFA predicts the 3-D flow velocity vectors in a wind farm using LES and is based on the 3-D incompressible Navier–Stokes equations. For turbine modelling it employs the ALM, which is a more sophisticated model than the ADM (Sanderse et al., 2011). In addition, the FAST model from the National Renewable Energy Laboratory (NREL) is implemented (Jonkman and Buhl, 2005). This model calculates turbine power production, blade forces on the flow and structural loading on the turbine. In the SOWFA sim-

ulation presented, the NREL 5 MW wind turbine is simulated (Jonkman et al., 2009).

The SOWFA data set used in this work for validation is equivalent to the set used in van Wingerden et al. (2017). The thrust coefficient C_T' is not a control variable in SOWFA due to the employment of the ALM and therefore has to be estimated. This will be discussed in the following paragraph.

Turbine operating settings

For estimating the control signals C_{T_n}', the turbine's fore-aft bending moment M_k^{sowfa} calculated with FAST is exploited. Using the relation $M_k^{\mathrm{sowfa}} = F_k^{\mathrm{sowfa}} z_{\mathrm{h}}$ with z_{h} the hub height, the (indirect) measured thrust force F_k^{sowfa} can be derived. An estimation from SOWFA data of the rotor flow velocity U_k^{sowfa} is obtained by averaging the flow velocity components across the rotor. Using the standard ADM yields for each turbine

$$F_k^{\mathrm{sowfa}} = \frac{1}{2} A \rho C_T' \left[U_k^{\mathrm{sowfa}} \right]^2 \begin{pmatrix} \cos(\gamma_k + \varphi_k) \\ \sin(\gamma_k + \varphi_k) \end{pmatrix}. \quad (24)$$

Since F_k^{sowfa}, U_k^{sowfa} and ρ can be obtained from SOWFA data and the yaw angles are given, all the variables in Eq. (24)

Table 3. Summary of the WFSim simulation set-up.

Domain size $L_x \times L_y$	$2.5 \times 1.5 \,(\text{km}^2)$	Turbine rotor diameter D	$126.4 \,(\text{m})$
Grid size $N_x \times N_y$	100×42	Turbine arrangement	3×3
Cell size $\Delta x \times \Delta y$	$25 \times 15 \,(\text{m}^2)$	Turbine spacing	$5D \times 3D$
Times	$\Delta t = 1, \Delta t^{\text{cpu}} = 0.1 \,(\text{s})$	Atmospheric conditions	$u_b = 12, v_b = 0 \,(\text{m s}^{-1}), \rho = 1.2 \,(\text{kg m}^{-3})$
Force and power factor	$c_f = \frac{5}{2}, c_p = 1.1$	Turbulence model	$d = 635, d' = 76.2 \,(\text{m}) \, l_s = 0.17$
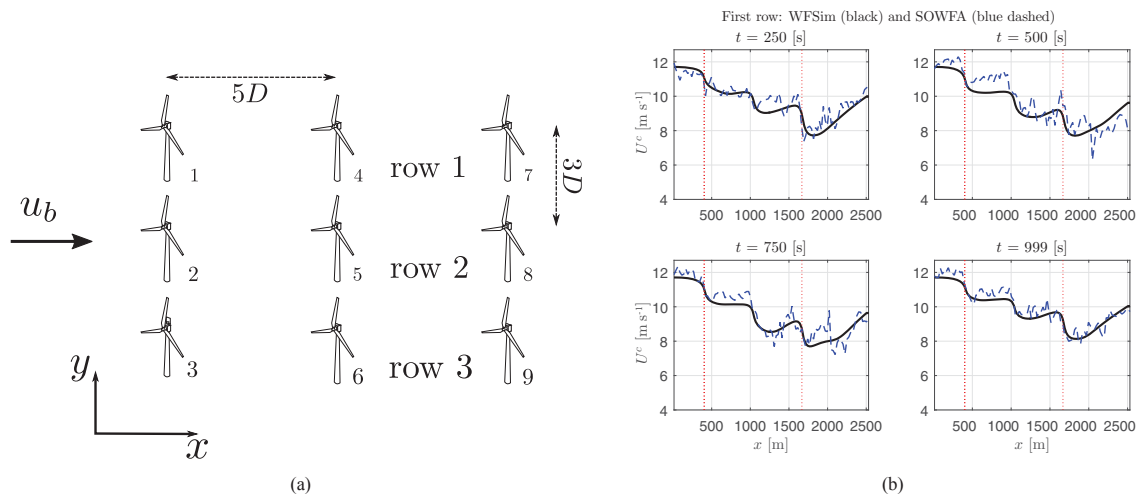
Figure 12. Topology simulated wind farm **(a)** and mean flow centreline at four time instances through the first row **(b)**.

are known; hence the control variable C'_T can be estimated from SOWFA data for each turbine.[6] It will be used, together with the yaw angle, as an input to the WFSim model.

In the following, flow data at hub height from a nine-turbine SOWFA simulation case will be compared with WFSim data. See Fig. 12a for the simulated wind farm topology. The turbines are excited with thrust coefficients as depicted in Fig. 11. These excitation signals are estimated from SOWFA data using the relation defined in Eq. (24). Table 3 presents the WFSim parameters used during simulations. The tuning variables of the WFSim model are found using a grid search and the inflow conditions u_b, v_b are estimated from SOWFA data.

Figures 12b and 13 depict a mean flow centreline (see Eq. 23) comparison for each row at four time instances. It can be concluded that the mean flow centreline derived from WFSim data approximates the mean flow centreline derived from SOWFA data. In Fig. 14, time series of the power signals from SOWFA and WFSim are depicted. The signals from the latter are more oscillating than the power signals from SOWFA. This is due to the fact that the power expression in WFSim is a non-linear static map depending on the C'_T. Thus, no turbine dynamics are taken into account, which is contrary to SOWFA in which the FAST turbine model is simulated. However, important characteristics can be cap-

tured with WFSim. A flow field evaluated with both the WFSim model and SOWFA can be found in Appendix C.

WFSim is capable of estimating dominant wake dynamics, the objective of the control-oriented model WFSim. Smaller-scale and stochastic effects can be measured by sensors and incorporated using an estimator based on WFSim, as has been shown in Doekemeijer et al. (2016, 2017).

4 Conclusions

Current literature on wind farm control can be categorized into model-free and model-based methods. This paper focused on the latter category. Here, a distinction can be made between type of model employed, a steady-state or dynamic wind farm model. In order to use the closed-loop control paradigm, and account for model uncertainties, we think it is important to utilize a dynamic wind farm model for controller design and possible online wind farm control. In this paper, such a control-oriented dynamic wind farm model, referred to as WFSim, has been presented.[7] It is a wind farm model that can predict flow fields and power production and includes turbines that are modelled using actuator disk theory and is based on modified two-dimensional Navier–Stokes equations. Completely neglecting the third (vertical) dimension is a too crude assumption to describe the flow in

---

[6]The estimated C'_T from SOWFA data is relatively noisy and hence filtered.

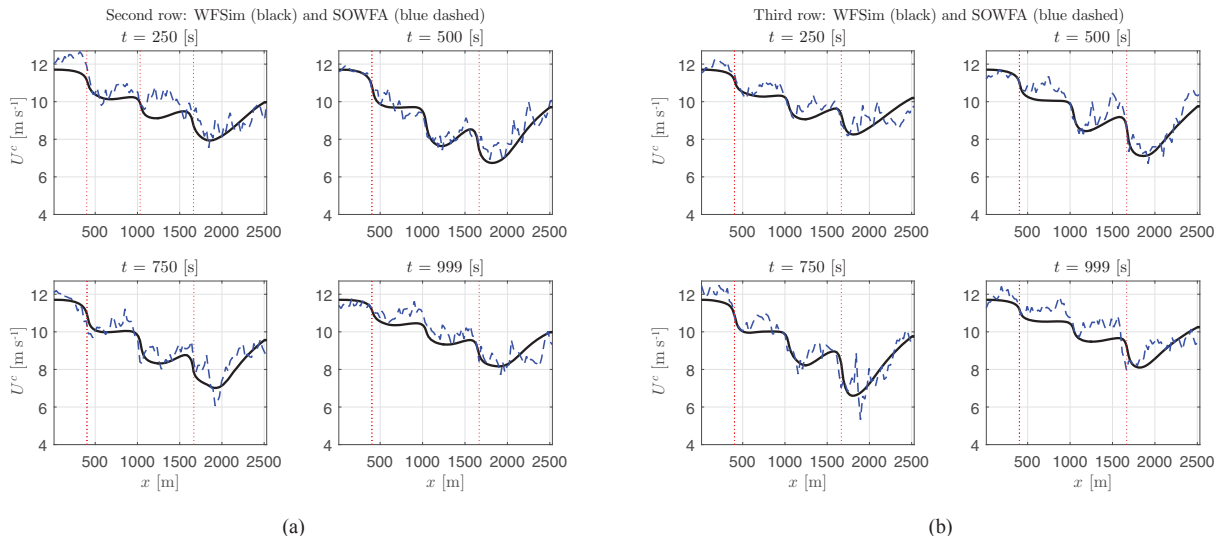[7]The WFSim repository can be found in WFSim (2015).

Figure 13. Mean flow centreline at four time instances through the second row **(a)** and third row **(b)** of turbines. The vertical red dashed lines indicate the positions of the turbines.
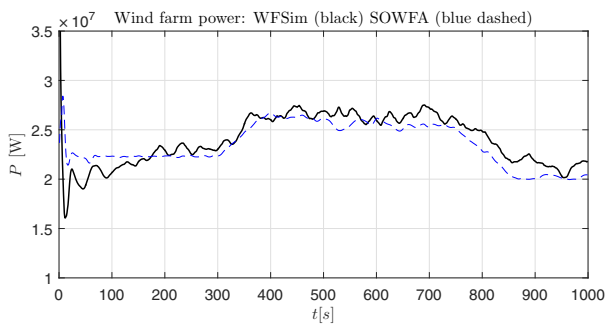

Figure 14. Wind farm power from SOWFA (blue dashed) and WF-Sim (black).

a wind farm accurately enough for control purposes. In this paper, we included a correction term in the continuity equation. It has been illustrated that the inclusion of this factor reduces the effect of neglecting the third (vertical) dimension. More precisely, it has been shown that the speed-up effect of the flow on the right and left downwind of a turbine will be reduced when solving for the corrected Navier–Stokes equations compared to the standard two-dimensional Navier–Stokes equations. It has been shown that this resulted in a better approximation of LES data.

In addition, a turbulence model was included, taking into account the desired wake recovery. The heuristically found turbulence model is based on Prandtl's mixing length hypotheses, where the mixing length parameter is made dependent on the downstream distance from the turbine rotors and also dependent on the mean wind direction. After theoretically formulating the WFSim model, this paper followed by illustrating that the computed flow velocities and power signals from the 2-D-like WFSim model can estimate flow velocity data and power signals from the 3-D high-fidelity wind farm models PALM and SOWFA. The necessary computation time of the WFSim model is a fraction of what is needed to perform LES, making it suitable for online control. This work focussed on axial induction actuation, but future work will also include the validation of yaw actuation and wind direction changes. For the simulation cases presented, no grid convergence studies have been performed, but future work should entail this. In addition, future work will entail the online update of the tuning variables c_f, c_p, d, d', l_s by an observer and the employment of the presented dynamic wind farm model in an online closed-loop control scheme.

Appendix A: Discretizing the Navier–Stokes equations

This section will present the necessary derivations to go from Eqs. (15) to (20), i.e. it will elaborate on the discretization of the NS equations. In the following subsections, all terms in the NS equations will subsequently be dealt with.

A1 Discretizing the convection (non-linear) terms

The non-linear term that occurs in the momentum equations can be spatially discretized by deriving

$$\int_{\Delta V} \rho (\mathbf{u} \cdot \nabla) \mathbf{u} \, dV = \int_{\Delta V} \rho \begin{pmatrix} \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \\ \frac{\partial vu}{\partial x} + \frac{\partial v^2}{\partial y} \end{pmatrix} dV.$$

x momentum equation

Deriving the term in the x momentum equation (first element in the vector above) yields

$$\int_{\Delta V} \rho \left[\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \right] dV$$
$$= \rho \left[\left(u^2 \delta y \right)_e - \left(u^2 \delta y \right)_w + (uv \Delta x)_n - (uv \Delta x)_s \right],$$

where $(u^2 \delta y)_e, (u^2 \delta y)_w$ are the quantities u^2 at the east and west sides of the cell with surface $\delta y_e, \delta y_w$, respectively. Similarly, $(uv \Delta x)_n, (uv \Delta x)_s$ are the quantities uv at the north and south sides of the cell with surface $\Delta x_n, \Delta x_s$, respectively. Assuming $\delta y = \delta y_e = \delta y_w$ and $\Delta x = \Delta x_n = \Delta x_s$, the above can be written as

$$\int_{\Delta V} \rho \left[\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \right] dV$$
$$= \rho \left[\left(u^2 \right)_e \delta y - \left(u^2 \right)_w \delta y + (uv)_n \Delta x - (uv)_s \Delta x \right].$$

$F^{ex} = \rho u_e \delta y, F^{wx} = \rho u_w \delta y, F^{nx} = \rho v_n \Delta x,$ and $F^{sx} = \rho v_s \Delta x$. In Versteeg and Malalasekera (2007) this is referred to as a convective mass flux approximation. The above can then be written as

$$\int_{\Delta V} \rho \left[\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \right] dV$$
$$= F^{ex} u_e - F^{wx} u_w + F^{nx} u_n - F^{sx} u_s.$$

In Fig. 5 we observe that $u_e, u_w, u_n, u_s, v_n, v_s$ are not defined for the black cell. Applying central differencing approximates the terms as follows:

$$u_e = \frac{u_{i+1,J} + u_{i,J}}{2}, \quad u_w = \frac{u_{i-1,J} + u_{i,J}}{2},$$

$$u_n = \frac{u_{i,J+1} + u_{i,J}}{2}, \quad u_s = \frac{u_{i,J-1} + u_{i,J}}{2},$$
$$v_n = \frac{v_{I-1,j+1} + v_{I,j+1}}{2}, \quad v_s = \frac{v_{I-1,j} + v_{I,j}}{2}. \tag{A1}$$

We can now write

$$\int_{\Delta V} \rho \left[\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \right] dV$$
$$= F_{i,J}^{ex} u_{i+1,J} - F_{i,J}^{wx} u_{i-1,J} + F_{i,J}^{nx} u_{i,J+1} - F_{i,J}^{sx} u_{i,J-1}$$
$$+ \left(F_{i,J}^{ex} - F_{i,J}^{wx} + F_{i,J}^{nx} - F_{i,J}^{sx} \right) u_{i,J}.$$

In Eq. (A1), central differencing is applied. A disadvantage of this method is that it does not use prior knowledge on the flow direction. The upwind differencing scheme, however, employs this prior knowledge as explained in Versteeg and Malalasekera (2007). A combination of the central and upwind differencing scheme is the hybrid differencing scheme. When applying this, the above can be written as

$$\int_{\Delta V} \rho \left[\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \right] dV$$
$$= c_{i,J}^{ex} u_{i+1,J} - c_{i,J}^{wx} u_{i-1,J} + c_{i,J}^{nx} u_{i,J+1}$$
$$- c_{i,J}^{sx} u_{i,J-1} + c_{i,J}^{px} u_{i,J}, \tag{A2}$$

with $c_{i,J}^{ex} = \max[-F_{i,J}^{ex}, 0], c_{i,J}^{wx} = \max[F_{i,J}^{wx}, 0], c_{i,J}^{nx} = \max[-F_{i,J}^{nx}, 0], c_{i,J}^{sx} = \max[F_{i,J}^{sx}, 0]$ and $c_{i,J}^{px} = c_{i,J}^{ex} + c_{i,J}^{wx} + c_{i,J}^{nx} + c_{i,J}^{sx} + F_{i,J}^{ex} - F_{i,J}^{wx} + F_{i,J}^{nx} - F_{i,J}^{sx}$. In WFSim, the coefficients $c_{i,J}^{\bullet}$ and $F_{i,J}^{\bullet}$ are evaluated for time k while the other flow velocity components are computed for time $k+1$.

y momentum equation

Deriving the non-linear term in the y momentum equation yields

$$\int_{\Delta V} \rho \left[\frac{\partial v^2}{\partial y} + \frac{\partial vu}{\partial x} \right] dV$$
$$= F_{I,j}^{ey} v_{I+1,j} - F_{I,j}^{wy} v_{I-1,j} + F_{I,j}^{ny} v_{I,j+1} - F_{I,j}^{sy} v_{I,j-1}$$
$$+ \left(F_{I,j}^{ey} - F_{I,j}^{wy} + F_{I,j}^{ny} - F_{I,j}^{sy} \right) v_{I,j},$$

with $F_{I,j}^{ey} = \rho u_e \Delta y, F_{I,j}^{wy} = \rho u_w \Delta y, F_{I,j}^{ny} = \rho v_n \delta x, F_{I,j}^{sy} = \rho v_s \delta x$ and

$$v_e = \frac{v_{I+1,j} + v_{I,j}}{2}, \quad v_w = \frac{v_{I-1,j} + v_{I,j}}{2},$$
$$v_n = \frac{v_{I,j+1} + v_{I,j}}{2}, \quad v_s = \frac{v_{I,j-1} + v_{I,j}}{2},$$
$$u_e = \frac{u_{i+1,J} + u_{i+1,J-1}}{2}, \quad u_w = \frac{u_{i,J} + u_{i,J-1}}{2}.$$

The intermediate steps are omitted here since they are similar to the steps presented when handling the non-linear term

in the x momentum equation. Note, however, that the discretization is evaluated using the yellow cell (see Fig. 5). When applying the hybrid differencing scheme, the above can be written as

$$\int_{\Delta V} \rho \left[\frac{\partial v^2}{\partial y} + \frac{\partial vu}{\partial x} \right] dV$$

$$= c^{ey}_{I,j} v_{I+1,j} - c^{wy}_{I,j} v_{I-1,j} + c^{ny}_{I,j} v_{I,j+1}$$
$$- c^{sy}_{I,j} v_{I,j-1} + c^{py}_{I,j} v_{I,j}, \tag{A3}$$

with $c^{ey}_{I,j} = \max[-F^{ey}_{I,j}, 0], c^{wy}_{I,j} = \max[F^{wy}_{I,j}, 0], c^{ny}_{I,j} = \max[-F^{ny}_{I,j}, 0], c^{sy}_{I,j} = \max[F^{sy}_{I,j}, 0]$ and $c^{py}_{I,j} = c^{ey}_{I,j} + c^{wy}_{I,j} + c^{ny}_{I,j} + c^{sy}_{I,j} + F^{ey}_{I,j} - F^{wy}_{I,j} + F^{ny}_{I,j} - F^{sy}_{I,j}$. Similar as before, the coefficients $c^{\bullet}_{I,j}$ and $F^{\bullet}_{I,j}$ are evaluated for time k while the other flow velocity components are computed for time $k+1$.

A2 Discretizing the pressure gradient

For the pressure gradient we evaluate

$$\int_{\Delta V} \left(\begin{matrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{matrix} \right) dV = \left(\begin{matrix} (p_{I,J} - p_{I-1,J}) \delta y \\ (p_{I,J} - p_{I,J-1}) \delta x \end{matrix} \right).$$

The pressure components are evaluated for time $k+1$.

A3 Discretizing the stress term

Evaluate

$$\int_{\Delta V} \boldsymbol{\tau} \nabla \, dV$$

$$= \int_{\Delta V} \left(\begin{matrix} \frac{\partial}{\partial x} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \frac{\partial u}{\partial x} \right] \\ + \frac{\partial}{\partial y} \frac{1}{2} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] \\ \frac{\partial}{\partial y} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \frac{\partial v}{\partial y} \right] \\ + \frac{\partial}{\partial x} \frac{1}{2} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] \end{matrix} \right) dV. \tag{A4}$$

x momentum equation

Considering the x momentum equation we have to evaluate multiple terms. The first term evaluates as

$$\int_{\Delta V} \frac{\partial}{\partial x} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \frac{\partial u}{\partial x} \right] dV$$

$$= \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \frac{\partial u}{\partial x} \right]_e \delta y - \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \frac{\partial u}{\partial x} \right]_w \delta y.$$

Here we have

$$\left. \frac{\partial u}{\partial y} \right|_e = \frac{u_{i,J+1} - u_{i,J}}{\Delta y_{J,J+1}}, \quad \left. \frac{\partial u}{\partial x} \right|_e = \frac{u_{i+1,J} - u_{i,J}}{\delta x_{i,i+1}},$$

$$\left. \frac{\partial u}{\partial y} \right|_w = \frac{u_{i,J} - u_{i,J-1}}{\Delta y_{J-1,J}}, \quad \left. \frac{\partial u}{\partial x} \right|_w = \frac{u_{i,J} - u_{i-1,J}}{\delta x_{i-1,i}},$$

and $\delta y = \delta y_{j,j+1}$. Substituting these expressions yields

$$\int_{\Delta V} \frac{\partial}{\partial x} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \frac{\partial u}{\partial x} \right] dV$$

$$= \underbrace{l_u(x_{I-1}, y_J)^2 \left| \frac{(u_{i,J+1} - u_{i,J}) \delta y_{j,j+1}}{\Delta y_{J,J+1} \delta x_{i,i+1}} \right| (u_{i+1,J} - u_{i,J}) \dots}_{T^{ex}_{i,j}}$$

$$- \underbrace{l_u(x_I, y_J)^2 \left| \frac{(u_{i,J} - u_{i,J-1}) \delta y_{j,j+1}}{\Delta y_{J-1,J} \delta x_{i-1,i}} \right| (u_{i,J} - u_{i-1,J})}_{T^{wx}_{i,j}}. \tag{A5}$$

The second term evaluates as

$$\int_{\Delta V} \frac{\partial}{\partial y} \frac{1}{2} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] dV$$

$$= \frac{1}{2} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right]_n \Delta x$$

$$- \frac{1}{2} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right]_s \Delta x.$$

Here we have

$$\left. \frac{\partial u}{\partial y} \right|_n = \frac{u_{i,J+1} - u_{i,J}}{\Delta y_{J,J+1}}, \quad \left. \frac{\partial v}{\partial x} \right|_n = \frac{v_{I,j+1} - v_{I-1,j+1}}{\Delta x_{I-1,I}},$$

$$\left. \frac{\partial u}{\partial y} \right|_s = \frac{u_{i,J} - u_{i,J-1}}{\Delta y_{J-1,J}}, \quad \left. \frac{\partial v}{\partial x} \right|_s = \frac{v_{I,j} - v_{I-1,j}}{\Delta x_{I-1,I}},$$

and $\Delta x = \Delta x_{I-1,I}$. Substituting yields

$$\int_{\Delta V} \frac{\partial}{\partial y} \frac{1}{2} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] dV$$

$$= \frac{1}{2} \left[l_u(x_i, y_{j+1})^2 \left| \frac{u_{i,J+1} - u_{i,J}}{\Delta y_{J,J+1}} \right| \right.$$

$$\left. \times \left(\frac{u_{i,J+1} - u_{i,J}}{\Delta y_{J,J+1}} + \frac{v_{I,j+1} - v_{I-1,j+1}}{\Delta x_{I-1,I}} \right) \right] \Delta x_{I-1,I} \dots$$

$$- \frac{1}{2} \left[l_u(x_i, y_j)^2 \left| \frac{u_{i,J} - u_{i,J-1}}{\Delta y_{J-1,J}} \right| \right.$$

$$\left. \times \left(\frac{u_{i,J} - u_{i,J-1}}{\Delta y_{J-1,J}} + \frac{v_{I,j} - v_{I-1,j}}{\Delta x_{I-1,I}} \right) \right] \Delta x_{I-1,I},$$

which can be rearranged to

$$\int_{\Delta V} \frac{\partial}{\partial y} \frac{1}{2} \left[l_u(x,y)^2 \left| \frac{\partial u}{\partial y} \right| \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] dV$$

$$= \underbrace{\frac{1}{2} l_u(x_i, y_{j+1})^2 \left| \frac{(u_{i,J+1} - u_{i,J}) \Delta x_{I-1,I}}{\Delta y^2_{J,J+1}} \right| (u_{i,J+1} - u_{i,J}) \dots}_{T^{nx}_{i,j}}$$

$$+\underbrace{\frac{1}{2}l_u(x_i,y_{j+1})^2\left|\frac{(u_{i,J+1}-u_{i,J})}{\Delta y_{J,J+1}}\right|}_{T_{i,J}^{newx}}(v_{I,j+1}-v_{I-1,j+1})\dots$$

$$\underbrace{-\frac{1}{2}l_u(x_i,y_j)^2\left|\frac{(u_{i,J}-u_{i,J-1})\Delta x_{I-1,I}}{\Delta y_{J-1,J}^2}\right|(u_{i,J}-u_{i,J-1})\dots}_{T_{i,J}^{sx}}$$

$$\underbrace{-\frac{1}{2}l_u(x_i,y_j)^2\left|\frac{(u_{i,J}-u_{i,J-1})}{\Delta y_{J-1,J}}\right|(v_{I,j}-v_{I-1,j}).}_{T_{i,J}^{sewx}}\qquad\text{(A6)}$$

Summarizing the above,

$$\frac{\partial}{\partial x}\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\frac{\partial u}{\partial x}\right]+\frac{\partial}{\partial y}\frac{1}{2}\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]$$
$$=T_{i,J}^{ex}u_{i+1,J}+T_{i,J}^{wx}u_{i-1,J}+T_{i,J}^{nx}u_{i,J+1}+T_{i,J}^{sx}u_{i,J-1}$$
$$+T_{i,J}^{px}u_{i,J}+T_{i,J}^{newx}(v_{I,j+1}-v_{I-1,j+1})$$
$$+T_{i,J}^{sewx}(v_{I-1,j}-v_{I,j}),$$

with $T_{i,J}^{px}=T_{i,J}^{ex}+T_{i,J}^{wx}+T_{i,J}^{nx}+T_{i,J}^{sx}$. The coefficients $T_{i,J}^{\bullet}$ will be computed for time k while the flow components will be evaluated for time $k+1$.

y momentum equation

Considering the y momentum equation, the first term evaluates as

$$\int_{\Delta V}\frac{\partial}{\partial y}\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\frac{\partial v}{\partial y}\right]\mathrm{d}V$$
$$=\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\frac{\partial v}{\partial y}\right]_n\Delta x-\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\frac{\partial v}{\partial y}\right]_s\Delta x.$$

Here we have

$$\left.\frac{\partial u}{\partial y}\right|_n=\frac{u_{i+1,J}-u_{i+1,J-1}}{\Delta y_{J-1,J}},\quad\left.\frac{\partial v}{\partial y}\right|_n=\frac{v_{I,j+1}-v_{I,j}}{\delta y_{j,j+1}},$$
$$\left.\frac{\partial u}{\partial y}\right|_s=\frac{u_{i,J}-u_{i,J-1}}{\Delta y_{J-1,J}},\quad\left.\frac{\partial v}{\partial y}\right|_s=\frac{v_{I,j}-v_{I,j-1}}{\delta y_{j-1,j}},$$

and $\Delta x=\delta x_{i,i+1}$. Substituting these expressions yields

$$\int_{\Delta V}\frac{\partial}{\partial y}\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\frac{\partial v}{\partial y}\right]\mathrm{d}V$$
$$=\underbrace{l_u(x_I,y_J)^2\left|\frac{(u_{i+1,J}-u_{i+1,J-1})\delta x_{i,i+1}}{\Delta y_{J-1,J}\delta y_{j,j+1}}\right|}_{T_{I,j}^{ny}}$$
$$\times(v_{I,j+1}-v_{I,j})\dots$$
$$\underbrace{-l_u(x_I,y_{J-1})^2\left|\frac{(u_{i,J}-u_{i,J-1})\delta x_{i,i+1}}{\Delta y_{J-1,J}\delta y_{j-1,j}}\right|(v_{I,j}-v_{I,j-1}).}_{T_{I,j}^{sy}}\qquad\text{(A7)}$$

The second term evaluates as

$$\int_{\Delta V}\frac{\partial}{\partial x}\frac{1}{2}\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]\mathrm{d}V$$
$$=\frac{1}{2}\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]_e\Delta y$$
$$-\frac{1}{2}\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]_w\Delta y.$$

Here we have

$$\left.\frac{\partial u}{\partial y}\right|_e=\frac{u_{i+1,J}-u_{i+1,J-1}}{\Delta y_{J-1,J}},\quad\left.\frac{\partial v}{\partial x}\right|_e=\frac{v_{I+1,j}-v_{I,j}}{\Delta x_{I,I+1}},$$
$$\left.\frac{\partial u}{\partial y}\right|_w=\frac{u_{i,J}-u_{i,J-1}}{\Delta y_{J-1,J}},\quad\left.\frac{\partial v}{\partial x}\right|_w=\frac{v_{I,j}-v_{I-1,j}}{\Delta x_{I-1,I}},$$

and $\Delta y=\Delta y_{J-1,J}$. Substituting these expressions yields

$$\int_{\Delta V}\frac{\partial}{\partial x}\frac{1}{2}\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]\mathrm{d}V$$
$$=\underbrace{\frac{1}{2}l_u(x_i,y_j)^2\left|\frac{u_{i+1,J}-u_{i+1,J-1}}{\Delta y_{J-1,J}}\right|(u_{i+1,J}-u_{i+1,J-1})}_{T_{I,J}^{ensy}}\dots$$
$$+\underbrace{\frac{1}{2}l_u(x_i,y_j)^2\left|\frac{u_{i+1,J}-u_{i+1,J-1}}{\Delta x_{I,I+1}}\right|(v_{I+1,j}-v_{I,j})}_{T_{I,J}^{ey}}\dots$$
$$\underbrace{-\frac{1}{2}l_u(x_{i+1},y_j)^2\left|\frac{u_{i,J}-u_{i,J-1}}{\Delta y_{J-1,J}}\right|(u_{i,J}-u_{i,J-1})}_{T_{I,J}^{wnsy}}\dots$$
$$\underbrace{-\frac{1}{2}l_u(x_{i+1},y_j)^2\left|\frac{u_{i,J}-u_{i,J-1}}{\Delta x_{I-1,I}}\right|(v_{I,j}-v_{I-1,j}).}_{T_{I,J}^{wy}}\qquad\text{(A8)}$$

Summarizing the above,

$$\int_{\Delta V}\frac{\partial}{\partial y}\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\frac{\partial v}{\partial y}\right]\mathrm{d}V$$
$$=\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\frac{\partial v}{\partial y}\right]_n\Delta x-\left[l_u(x,y)^2\left|\frac{\partial u}{\partial y}\right|\frac{\partial v}{\partial y}\right]_s\Delta x$$
$$=T_{I,j}^{ey}v_{I+1,j}+T_{I,j}^{wy}v_{I-1,j}+T_{I,j}^{ny}v_{I,j+1}+T_{I,j}^{sy}v_{I,j-1}$$
$$+T_{I,j}^{py}v_{I,j}+T_{I,j}^{ensy}(u_{i+1,J}-u_{i+1,J-1})$$
$$+T_{I,j}^{wnsy}(u_{i,J}-u_{i,J-1}),$$

with $T_{I,j}^{py}=T_{I,j}^{ey}+T_{I,j}^{wy}+T_{I,j}^{ny}+T_{I,j}^{sy}$. The coefficients $T_{I,j}^{\bullet}$ will be computed for time k while the flow components will be evaluated for time $k+1$.

A4 Discretizing the forcing term

$$\int_{\Delta V} \frac{1}{2}\rho C_T' \left[U\cos(\gamma)\right]^2 \begin{pmatrix} \cos(\gamma+\varphi) \\ \sin(\gamma+\varphi) \end{pmatrix} \mathrm{d}V$$

$$= \frac{1}{2}\rho C_T' \left[U\cos(\gamma)\right]^2 \begin{pmatrix} \cos(\gamma+\varphi) \\ \sin(\gamma+\varphi) \end{pmatrix} \Delta V$$

A5 Discretizing the unsteady term

Evaluate

$$\int_{\Delta V} \begin{pmatrix} \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial t} \end{pmatrix} \mathrm{d}V = \begin{pmatrix} \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial t} \end{pmatrix} \Delta V.$$

Temporal discretization yields $\frac{u_{k+1}-u_k}{\Delta t}$ and $\frac{v_{k+1}-v_k}{\Delta t}$ and we define

$$a_0^{px} = \frac{\Delta V}{\Delta t} \quad \text{and} \quad a_0^{py} = \frac{\Delta V}{\Delta t}$$

A6 Discretizing the Continuity equation

$$0 = \int_{\Delta V} \frac{\partial u}{\partial x} + 2\frac{\partial v}{\partial y} \, \mathrm{d}V$$

$$= \left(u_{i+1,J} - u_{i,J}\right)\delta y_{j,j+1} + 2\left(v_{I,j+1} - v_{I,j}\right)\delta x_{i,i+1}.$$

All the coefficients derived above are given in Table A1.

Table A1. Fully discretized Navier–Stokes equations and all their coefficients.

---

x momentum equation:
$$a_{i,J}^{px}u_{i,J} = \begin{pmatrix} a_{i,J}^{nx} & a_{i,J}^{sx} & a_{i,J}^{wx} & a_{i,J}^{ex} \end{pmatrix}\begin{pmatrix} u_{i,J+1} & u_{i,J-1} & u_{i-1,J} & u_{i+1,J} \end{pmatrix}^T - \delta y_{j,j+1}\left(p_{I,J}-p_{I-1,J}\right)+f_{i,J}^x + \dots$$
$$+ \begin{pmatrix} a_{i,J}^{nwx} & a_{i,J}^{swx} & a_{i,J}^{nex} & a_{i,J}^{sex} \end{pmatrix}\begin{pmatrix} v_{I-1,j+1} & v_{I-1,j} & v_{I,j+1} & v_{I,J} \end{pmatrix}^T$$
y momentum equation:
$$a_{I,j}^{py}v_{I,j} = \begin{pmatrix} a_{I,j}^{ny} & a_{I,j}^{sy} & a_{I,j}^{wy} & a_{I,j}^{ey} \end{pmatrix}\begin{pmatrix} v_{I,j+1} & v_{I,j-1} & v_{I-1,j} & v_{I+1,j} \end{pmatrix}^T - \delta x_{i,i+1}\left(p_{I,J}-p_{I,J-1}\right)+f_{I,j}^y + \dots$$
$$+ \begin{pmatrix} a_{i,J}^{nwy} & a_{i,J}^{swy} & a_{i,J}^{ney} & a_{i,J}^{sey} \end{pmatrix}\begin{pmatrix} u_{i,J} & u_{i,J-1} & u_{i+1,J} & u_{i+1,J-1} \end{pmatrix}^T$$
Continuity equation:
$$0 = \delta y_{j,j+1}\left(u_{i+1,J}-u_{i,J}\right) + 2\delta x_{i,i+1}\left(v_{I,j+1}-v_{I,j}\right),$$

$$a_{i,J}^{ex} = \max\left[-F_{i,J}^{ex},0\right]+T_{i,J}^{ex}, \quad a_{i,J}^{wx} = \max\left[F_{i,J}^{wx},0\right]+T_{i,J}^{wx}, \quad a_{i,J}^{nx} = \max\left[-F_{i,J}^{nx},0\right]+T_{i,J}^{nx}, \quad a_{i,J}^{sx} = \max\left[F_{i,J}^{sx},0\right]+T_{i,J}^{sx},$$

$$a_{I,j}^{ey} = \max\left[-F_{I,j}^{ey},0\right]+T_{I,j}^{ey}, \quad a_{I,j}^{wy} = \max\left[F_{I,j}^{wy},0\right]+T_{I,j}^{wy}, \quad a_{I,j}^{ny} = \max\left[-F_{I,j}^{ny},0\right]+T_{I,j}^{ny}, \quad a_{I,j}^{sy} = \max\left[F_{I,j}^{sy},0\right]+T_{I,j}^{sy},$$

$$a_{i,J}^{nex} = T_{i,J}^{newx}, \quad a_{i,J}^{nwx} = T_{i,J}^{newx}, \quad a_{i,J}^{sex} = T_{i,J}^{sewx}, \quad a_{i,J}^{swx} = T_{i,J}^{sewx},$$
$$a_{I,j}^{ney} = T_{I,j}^{ensy}, \quad a_{I,j}^{nwy} = T_{I,j}^{wnsy}, \quad a_{I,j}^{sey} = T_{I,j}^{ensy}, \quad a_{I,j}^{swy} = T_{I,j}^{wnsy},$$

$$a_{i,J}^{px} = a_{i,J}^{nx} + a_{i,J}^{ex} + a_{i,J}^{sx} + a_{i,J}^{wx} + F_{i,J}^{nx} + F_{i,J}^{ex} - F_{i,J}^{sx} - F_{i,J}^{wx} + T_{i,J}^{px} + a_0^{px},$$
$$a_{I,j}^{py} = a_{I,j}^{ny} + a_{I,j}^{ey} + a_{I,j}^{sy} + a_{I,j}^{wy} + F_{I,j}^{ny} + F_{I,j}^{ey} - F_{I,j}^{sy} - F_{I,j}^{wy} + T_{I,j}^{py} + a_0^{py},$$

in which
$$F_{i,J}^{ex} = \tfrac{1}{2}\rho\left(u_{i+1,J}+u_{i,J}\right)\delta y_{j,j+1}, \qquad F_{i,J}^{wx} = \tfrac{1}{2}\rho\left(u_{i,J}+u_{i-1,J}\right)\delta y_{j,j+1},$$
$$F_{i,J}^{nx} = \tfrac{1}{2}\rho\left(v_{I,j+1}+v_{I-1,j+1}\right)\Delta x_{I-1,I}, \quad F_{i,J}^{sx} = \tfrac{1}{2}\rho\left(v_{I,j}+v_{I-1,j}\right)\Delta x_{I-1,I},$$
$$F_{I,j}^{ey} = \tfrac{1}{2}\rho\left(u_{i+1,J}+u_{i+1,J-1}\right)\Delta y_{J-1,J}, \quad F_{I,j}^{wy} = \tfrac{1}{2}\rho\left(u_{i,J}+u_{i,J-1}\right)\Delta y_{J-1,J},$$
$$F_{i,J}^{ny} = \tfrac{1}{2}\rho\left(v_{I,j+1}+v_{I-1,j+1}\right)\Delta x_{I-1,I}, \quad F_{i,J}^{sy} = \tfrac{1}{2}\rho\left(v_{I,j}+v_{I-1,j}\right)\Delta x_{I-1,I},$$

$$a_0^{px} = \frac{\Delta x_{I-1,I}\delta y_{j,j+1}}{\Delta t}, \quad a_0^{py} = \frac{\Delta y_{J-1,J}\delta x_{i,i+1}}{\Delta t},$$
$$\Delta x_{I-1,I} = x_I - x_{I-1}, \quad \Delta y_{J-1,J} = y_J - y_{J-1},$$

$$T_{i,J}^{px} = T_{i,J}^{ex} + T_{i,J}^{wx} + T_{i,J}^{sx} + T_{i,J}^{nx}, \quad \text{with } T_{i,J}^{\bullet} \text{ given in (A5) and (A6)},$$
$$T_{I,J}^{py} = T_{I,J}^{ey} + T_{I,J}^{wy} + T_{I,J}^{sy} + T_{I,J}^{ny}, \quad \text{with } T_{I,j}^{\bullet} \text{ given (A7) and (A8)},$$
and
$$f_{i,J}^x = \tfrac{1}{2}\delta y_{j,j+1}\rho C_T'\left[U_k\cos(\gamma_k)\right]^2\cos(\gamma_k+\varphi_k), \quad f_{I,j}^y = \tfrac{1}{2}\delta y_{J-1,J}\rho C_T'\left[U_k\cos(\gamma_k)\right]^2\sin(\gamma_k+\varphi_k), \quad U_k = \sqrt{u_{i,J}^2+v_{I,j}^2}\cos(\gamma_k)$$

Appendix B: PALM case study

In this appendix, a resolved flow field for an arbitrarily chosen time step is depicted for the PALM case study presented in Sect. 3.2.1. Table B1 gives a summary of the PALM simulation set-up.

Table B1. Summary of the simulation set-up.

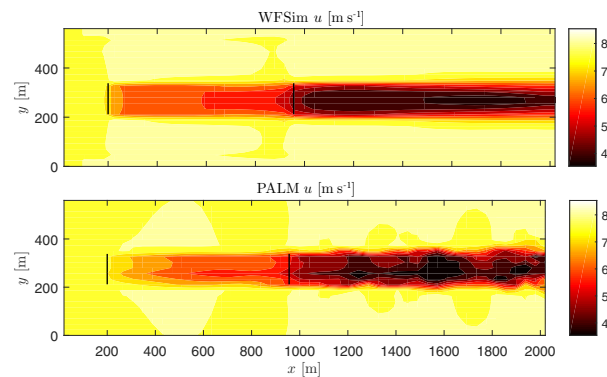| Domain size $L_x \times L_y \times L_z$	$19.2 \times 2.56 \times 1.28\,(\mathrm{km}^2)$	Turbine dimensions	$D = 126\,(\mathrm{m})$, $z_\mathrm{h} = 90\,(\mathrm{m})$
Grid size $N_x \times N_y \times N_z$	$1920 \times 256 \times 1280$	Turbine arrangement	2×1
Cell size $\Delta x \times \Delta y$	$10 \times 10 \times 15\,(\mathrm{m}^2)$	Turbine spacing	$6D$
Sample period Δt	$1\,(\mathrm{s})$	Atmospheric conditions	$u_\mathrm{b} = 8, v_\mathrm{b} = 0, w_\mathrm{b} = 0\,(\mathrm{m\,s}^{-1})$, $\rho = 1.2\,(\mathrm{kg\,m}^{-3})$
Simulation time t	$1750\,(\mathrm{s})$	Inflow	Uniform

Figure B1. Flow field obtained with PALM (below) and WFSim at $t = 750\,(\mathrm{s})$. The black lines indicate the turbines.

Appendix C: SOWFA case study

In this appendix, a resolved flow field for an arbitrarily chosen time step is depicted for the SOWFA case study presented in Sect. 3.2.2. The SOWFA data set presented in van Wingerden et al. (2017) is utilized.
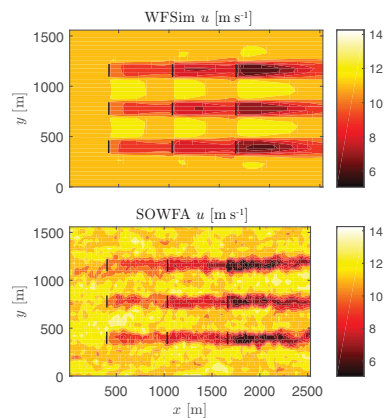

Figure C1. Flow field obtained with SOWFA (below) and WFSim at $t = 250\,(\mathrm{s})$. The black lines indicate the turbines.

Appendix D: Nomenclature

$L_x \times L_y$,	domain size wind farm
$N_x \times N_y$,	number of cells
T_n,	turbine n
U_n,	hub-height flow velocity at the rotor
U^c,	flow centreline velocity
C_T, C_P,	thrust force and power coefficient
l_u,	turbulence model parameter
Δt,	sample period
$q_k = \left(u_k^T \quad v_k^T \quad p_k^T\right)^T$,	state vector with longitudinal and lateral flow velocities and pressure
$w_k = \left(v_k^T \quad \gamma_k^T\right)^T$,	control variables
D,	turbine rotor diameter
$\Delta x \times \Delta y$,	cell size
$\aleph$,	number of turbines
u_b, v_b,	inflow conditions
U_∞,	upstream flow velocity
f,	wind turbine force
τ_H,	2-D stress tensor
k,	time index
n_q,	number of states
z_k,	measurement vector

References

Annoni, J. and Seiler, P.: A low-order model for wind farm control, P. Amer. Contr. Conf., https://doi.org/10.1109/ACC.2015.7170981, 2015.

Annoni, J., Seiler, P., Johnson, K., Fleming, P. A., and Gebraad, P. M. O.: Evaluating wake models for wind farm control, P. Amer. Contr. Conf., https://doi.org/10.1109/ACC.2014.6858970, 2014.

Avila, M., Folch, A., Houzeaux, G., Eguzkitza, B., Prieto, L., and Cabezón, D.: A Parallel CFD Model for Wind Farms, Procedia Comput. Sci., 18, 2157–2166, 2013.

Barthelmie, R., Frandsen, S., Hansen, K., Schepers, J., Rados, K., Schlez, W., Neubert, A., Jensen, L., and Neckelmann, S.: Modelling the impact of wakes on power output at Nysted and Horns rev, European Wind Energy Conference, 2009.

Boersma, S., Gebraad, P. M. O., Vali, M., Doekemeijer, B. M., and van Wingerden, J. W.: A control-oriented dynamic wind farm flow model: "WFSim", J. Phys. Conf. Ser., https://doi.org/10.1088/1742-6596/753/3/032005, 2016a.

Boersma, S., van Wingerden, J. W., Vali, M., and Kühn, M.: Quasi linear parameter varying modeling for wind farm control using the 2D Navier–Stokes equations, P. Amer. Contr. Conf., https://doi.org/10.1109/ACC.2016.7525616, 2016b.

Boersma, S., Doekemeijer, B. M., Gebraad, P. M. O., Fleming, P. A., Annoni, J., Scholbrock, A. K., Frederik, J. A., and van Wingerden, J. W.: A tutorial on control-oriented modelling and control of wind farms, P. Amer. Contr. Conf., https://doi.org/10.23919/ACC.2017.7962923, 2017.

Churchfield, M., Lee, S., Michalakes, J., and Moriarty, P.: A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics, J. Turbul., 13, N14, 2012.

Crespo, A., Hernandez, J., Fraga, E., and Andreu, C.: Experimental validation of the UPM computer code to calculate wind turbine wakes and comparison with other models, J. Wind Eng. Ind. Aerod., 27, 77–88, 1988.

Crespo, A., Hernandez, J., and Frandsen, S.: Survey of modelling methods for wind turbine wakes and wind farms, Wind Energy, 2, 1–24, 1999.

Doekemeijer, B. M., van Wingerden, J. W., Boersma, S., and Pao, L. Y.: Enhanced Kalman filtering for a 2D CFD NS wind farm flow model, J. Phys. Conf. Ser., https://doi.org/10.1088/1742-6596/753/5/052015, 2016.

Doekemeijer, B. M., Boersma, S., van Wingerden, J. W., and Pao, L. Y.: Ensemble Kalman filtering for wind field estimation in wind farms, P. Amer. Contr. Conf., https://doi.org/10.23919/ACC.2017.7962924, 2017.

Frandsen, S., Barthelmie, R., Pryor, S., Rathmann, O., Larsen, S., Højstrup, J., and Thøgersen, M.: Analytical modelling of wind speed deficit in large offshore wind farms, Wind Energy, 9, 39–53, 2006.

Gebraad, P. M. O., Teeuwisse, F. W., van Wingerden, J. W., Fleming, P. A., Ruben, S. D., Marden, J. R., and Pao, L. Y.: A data-driven model for wind plant power optimization by yaw control, P. Amer. Contr. Conf., https://doi.org/10.1109/ACC.2014.6859118, 2014a.

Gebraad, P. M. O., Teeuwisse, F. W., van Wingerden, J. W., Fleming, P. A., Ruben, S. D., Marden, J. R., and Pao, L. Y.: Wind plant power optimization through yaw control using a parametric model for wake effects – a CFD simulation study, Wind Energy, 19, 95–114, 2014b.

George, A. and Liu, J.: Computer Solution of Large Sparse Positive Definite, Prentice Hall, 1981.

Göçmen, T., van der Laan, P., Réthoré, P.-E., Diaz, A. P., and Larsen, G. C.: Wind turbine wake models developed at the technical university of Denmark: a review, Renew. Sust. Energ. Rev., 60, 752–769, 2016.

Goit, J. and Meyers, J.: Optimal control of energy extraction in wind-farm boundary layers, J. Fluid Mech., 768, 5–50, 2015.

Grunnet, J. D., Soltani, M., Knudsen, T., Kragelund, M. N., and Bak, T.: Aeolus toolbox for dynamics wind farm model, simulation and control, The European Wind Energy Conference and Exhibition, 2010.

Iungo, G. V., Viola, F., Ciri, U., Rotea, M. A., and Leonardi, S.: Data-driven RANS for simulations of large wind farms, J. Phys. Conf. Ser., https://doi.org/10.1088/1742-6596/625/1/012025, 2015.

Jensen, N. O.: A note on wind generator interaction, Tech. rep., Risø National Laboratory, 1983.

Jiménez, Á., Crespo, A., and Migoya, E.: Application of a LES technique to characterize the wake deflection of a wind turbine in yaw, Wind Energy, 13, 559–572, 2010.

Jonkman, J. M. and Buhl, M. L.: FAST v6.0 user guide, technical report, Tech. rep, National Renewable Energy Laboratory (NREL), 2005.

Jonkman, J. M., Butterfield, S., Musial, W., and Scott, G.: Definition of a 5-MW reference wind turbine for offshore system development, Tech. rep., National Renewable Energy Laboratory (NREL), 2009.

Katic, I., Hojstrup, J., and Jensen, N. O.: A Simple Model for Cluster efficiency, EWEC, 1986.

Knudsen, T., Bak, T., and Svenstrup, M.: Survey of wind farm control: power and fatigue optimization, Wind Energy, 18, 1333–1351, 2015.

Larsen, G. C., Madsen, H. A., Bingöl, F. Mann, J., Ott, S., Sørensen, J. N., Okulov, V., Troldborg, N., Nielsen, M., Thomsen, K., Larsen, T. J., and Mikkelsen, R.: Dynamic wake meandering modelling, Tech. rep., Risø National Laboratory, 2007.

Maronga, B., Gryschka, M., Heinze, R., Hoffmann, F., Kanani-Sühring, F., Keck, M., Ketelsen, K., Letzel, M. O., Sühring, M., and Raasch, S.: The Parallelized Large-Eddy Simulation Model (PALM) version 4.0 for atmospheric and oceanic flows: model

formulation, recent developments, and future perspectives, Geosci. Model Dev., 8, 2515–2551, https://doi.org/10.5194/gmd-8-2515-2015, 2015.

Martinez-Tossas, L. A., Churchfield, M. J., and Leonardi, S.: Large Eddy Simulations of the flow past wind turbines: actuator line and disk modeling, Wind Energy, 18, 1047–1060, 2014.

Medici, D.: Experimental Studies of Wind Turbine Wakes – Power Optimisation and Meandering, PhD dissertation, Delft University of Technology, 2005.

Meyers, J. and Meneveau, C.: Large Eddy Simulations of large wind-turbine arrays in the atmospheric boundary layer, Aerospace Sciences Meeting, 2010.

Munters, W. and Meyers, J.: An optimal control framework for dynamic induction control of wind farms and their interaction with the atmospheric boundary layer, Philos. T. Roy. Soc. A, 375, 20160100, 2017.

Özdemir, H., Versteeg, M. C., and Brand, A. J.: Improvements in ECN wake model, ICOWES conference, 2013.

Prandtl, L.: Bericht uber Untersuchungen zur ausgebildeten Turbulenz, Z. Angew. Math. Me., https://doi.org/10.1007/978-3-662-11836-8_57, 1925.

Raach, S., Schlipf, D., and Cheng, P. W.: Lidar-based wake tracking for closed-loop wind farm control, Wind Energ. Sci., 2, 257–267, https://doi.org/10.5194/wes-2-257-2017, 2017.

Sagaut, P.: Large Eddy Simulation for Incompressible Flows, Springer-Verlag Berlin Heidelberg ISBN 978-3-540-26403-3, 2006.

Sanderse, B.: Aerodynamics of wind turbine wakes, Literature Review, Report 16, ECN, 2009.

Sanderse, B., van der Pijl, S. P., and B., K.: Review of computational fluid dynamics for wind turbine wake aerodynamics, Wind Energy, 14, 799–819, 2011.

Shapiro, C. R., Bauweraerts, P., Meyers, J., Meneveau, C., and Gayme, D. F.: Model-based receding horizon control of wind farms for secondary frequency regulation, Wind Energy, 20, 1261–1275, 2017a.

Shapiro, C. R., Meyers, J., Meneveau, C., and Gayme, D. F.: Dynamic wake modeling and state estimation for improved model-based receding horizon control of wind farms, P. Amer. Contr. Conf., https://doi.org//10.23919/ACC.2017.7963036, 2017b.

Soleimanzadeh, M., Wisniewski, R., and Brand, A.: State-space representation of the wind flow model in wind farms, Wind Energy, 17, 627–639, 2014.

Trabucchi, D., Vollmer, L., and Kühn, M.: Shear layer approximation of Navier-Stokes steady equations for non-axisymmetric wind turbine wakes: description, verification and first application, J. Phys. Conf. Ser., https://doi.org/10.1088/1742-6596/753/3/032030, 2016.

Vali, M., van Wingerden, J. W., Boersma, S., Petrovic, V., and Kühn, M.: A predictive control framework for optimal energy extraction of wind farms, J. Phys. Conf. Ser., https://doi.org/10.1088/1742-6596/753/5/052013, 2016.

Vali, M., Petrović, V., Boersma, S., van Wingerden, J. W., and Kühn, M.: Adjoint-based model predictive control of wind farms: beyond the quasi steady-state power maximization, International Federation of Automatic Control, https://doi.org/10.1016/j.ifacol.2017.08.382, 2017.

van der Laan, M. P., Sørensen, N. N., Réthoré, P.-E., Mann, J., Kelly, M. C., and Troldborg, N.: The k-ϵ-fP model applied to double wind turbine wakes using different actuator disk force methods, Wind Energy, 18, 2223–2240, 2015.

van Wingerden, J. W., Pao, L. Y., Aho, J., and Fleming, P. A.: Active power control of waked wind farms, International Federation of Automatic Control, https://doi.org/10.1016/j.ifacol.2017.08.378, 2017.

Vermeer, L. J., Sorensen, J. N., and Crespo, A.: Wind turbine wake aerodynamics, Prog. Aerosp. Sci., 39, 467–510, 2003.

Versteeg, H. and Malalasekera, W.: An Introduction to Computational Fluid Dynamics: The Finite Volume Method, Pearson, Prentice Hall, 2007.

WFSim: available at: https://github.com/TUDelft-DataDrivenControl/WFSim (last access: 1 March 2018), 2015.