

Dynamic flow model for real-time application in wind farm control

Rott, Andreas; Boersma, Sjoerd; Van Wingerden, Jan Willem; Kühn, Martin

DOI

[10.1088/1742-6596/854/1/012039](https://doi.org/10.1088/1742-6596/854/1/012039)

Publication date

2017

Document Version

Final published version

Published in

Journal of Physics: Conference Series

Citation (APA)

Rott, A., Boersma, S., Van Wingerden, J. W., & Kühn, M. (2017). Dynamic flow model for real-time application in wind farm control. In A. Barney, S. Ivanell, M. Klemm, & J. N. Sørensen (Eds.), *Journal of Physics: Conference Series: Proceedings of the Wake Conference 2017* (Vol. 854-1). Article 012039 (Journal of Physics: Conference Series; Vol. 854, No. 1). IOP Publishing. <https://doi.org/10.1088/1742-6596/854/1/012039>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Dynamic Flow Model for Real-Time Application in Wind Farm Control

Andreas Rott^{1*}, Sjoerd Boersma², Jan-Willem van Wingerden²,
Martin Kühn¹

¹ Institute of Physics, ForWind, Carl von Ossietzky University of Oldenburg, Oldenburg, Germany

² Delft Center for Systems and Control, Delft University of Technology, Delft, the Netherlands

E-mail: andreas.rott@uni-oldenburg.de

Abstract. For short-term power predictions and estimations of the available power during curtailment of a wind farm, it is necessary to consider the flow dynamics and aerodynamic interactions of the turbines. In this paper, a control-oriented dynamic two-dimensional wind farm model is introduced that aims to incorporate real-time measurements such as flow velocities at turbine locations to estimate the ambient wind farm flow. The model is intended to derive flow predictions for real-time applications. Since fully resolved computational fluid dynamics are too CPU-intensive for such a task, the dynamic model presented in this paper relies on an approximation of the flow equations in a two-dimensional framework. A semi-Lagrangian advection scheme and a step-wise flow solver together offer fast calculation speed, which scales linearly with the number of grid points. In order to emulate effects of realistic three-dimensional wind farm flow, a relaxation of the two-dimensional continuity equation is presented. Furthermore, with little extra computational expense, additional dynamic state variables for various possible applications can be propagated along the wind flow. For instance, a dynamic confidence parameter can provide estimations of the accuracy of flow predictions, while a turbulence parameter adds the possibility to estimate wake induced loads on downstream turbines. In order to demonstrate the performance and validity of the new model it is compared with other models. At first a two turbine reference case is compared with a steady-state model and secondly with results obtained by the dynamic wind farm flow model WFSim. Finally a small wind farm is simulated in order to show the computational scaling of the model.

1. Introduction

Wind energy plays an important role in the increase of the renewable energies to meet the requirements of the Paris Agreement. Consequently, wind farms become larger and wake effects become even more prominent. In particular, offshore wind farms, which often are constructed with a regular mesh-like layout, are subject to aerodynamic interactions that cause significant power losses and wake induced loads [1, 2]. While onshore wind energy is a strong competitor in the energy market and has become at good sites, one of the cheapest power sources [3], the energy generation cost of offshore wind energy still has a lot of potential for improvement. One promising approach to achieve that, is optimized wind farm control with wake steering [4]. Therefore, the consideration of the aerodynamic interactions in the control of the individual turbines within a wind farm control scheme is an active field of research. One key point for those concepts is knowledge about the current ambient wind farm flow. In the past years,



great effort was spent on the development and improvement of steady wake models and flow simulations to estimate and predict the power output of a wind farm, that are computationally fast enough to be used for optimized wind farm control. An extensive overview is provided in [5]. But only in the most recent years research also focuses on dynamic models that are fast enough for online predictions and for control design. These models are required to accurately represent the dynamic wind farm flow in large wind farms because of the high sensitivity of changes of the wind direction on the power output [6]. In [7] a propagation model for LiDAR measurements to correct the temporal distortion and to reconstruct the wind field between the measurements is introduced. In this paper this propagation model is extended to a complete 2D dynamic wind farm flow model that is able to simulate and predict the wind farm flow at hub-height with a relatively high reliability at relatively low computational costs. Additionally the semi-Lagrangian structure of the model allows to propagate any number of other parameters/properties with the flow, which can be utilized for various purposes. The model is designed so that it works for flow estimations from SCADA data or as a basis for wind farm control optimizations. Therefore the objectives of this papers are threefold: To introduce the structure of the dynamic model and describe its application (see section 2), demonstrate simulation results and the performance of the model (see section 3) and to discuss its capabilities (see section 4) and finally a brief conclusion is given in section 5.

2. Method

The wind farm flow model presented in this paper is a 2D medium fidelity flow and property propagation scheme. The algorithm solves the discretized Navier-Stokes equation for incompressible flow in a step-wise scheme, that includes a relaxation of the continuity equation to approximate 3D wind farm flow with relatively low computational expenses. Next to the longitudinal and lateral wind velocities (u and v , respectively) of the 2D domain, any number of additional parameters can be propagated with the flow by the presented method as explained in 2.2. These parameters are referred to as additional states in this paper, they can be used to simulate a variety of flow properties.

2.1. General flow equations

The Navier-Stokes equation and the continuity equation (for incompressible flow) are the governing flow equations illustrated in the vector equation (1) and the scalar equation (2)

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla) \vec{u} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{u} + \vec{f} \quad (1)$$

$$\nabla \cdot \vec{u} = 0 \quad (2)$$

Here the vector $\vec{u} := (u, v)^T$ is the short form for the smooth function $\vec{u}(x, y, t) \in C^\infty(\mathbb{R}^2 \times \mathbb{R})$ and it describes the 2D flow velocity, while the superscript T denotes the transpose of a matrix or vector. The variables $t, \rho, p, \nu \in \mathbb{R}$ are the time, density, pressure and the kinematic viscosity of the fluid, respectively. $\vec{f} := \left(\frac{f_x}{\rho}, \frac{f_y}{\rho} \right)$ (short for the vector function $\vec{f}(x, y, t) : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^2$)

is the external force (over density) acting on the flow. The symbol $\nabla := \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)^T$ is a vector operator of the partial spatial derivatives with $x, y \in \mathbb{R}$ as the horizontal and vertical positions.

The widely used Helmholtz-Hodge Decomposition [8] is a useful mathematical tool that lets us define a projection, which allows to reduce (1),(2). Furthermore, this technique is used to build a relaxation of the continuity equation as will be described in section 2.8. The fundamental theorem of vector calculus states that every sufficiently smooth vector field is uniquely composed of a solenoidal vector field and a conservative vector field. In more mathematical terms: Let $\vec{\xi}(x, y) \in \mathbb{R}^2$ be a smooth vector field. Hence there exists a unique vector field $\vec{q}(x, y) \in \mathbb{R}^2$, with

$\nabla \cdot \vec{q} = 0$ (solenoidal, here ‘ \cdot ’ is the dot product) and a unique vector field $\vec{r}(x, y) \in \mathbb{R}^2$, with $\nabla \times \vec{r} = 0$ (rotational free, here ‘ \times ’ is the cross product), with:

$$\vec{\xi} = \vec{q} + \vec{r} \quad (3)$$

According to this decomposition the projection P can be defined, that projects a smooth vector field onto its divergence free component:

$$P(\vec{\xi}) := \vec{\xi} - \vec{r} = \vec{q} \quad (4)$$

The construction of the projection can be derived by taking the divergence of (3) and the linearity of the divergence operator:

$$\nabla \cdot \vec{\xi} = \underbrace{\nabla \cdot \vec{q}}_{=0} + \nabla \cdot \vec{r} = \nabla \cdot \vec{r} \quad (5)$$

by definition the vector field \vec{r} is rotational free, which is equivalent to the existence of a scalar field $p(x, y) \in \mathbb{R}$, with $\vec{r} = \nabla p$. Therefore we can substitute \vec{r} and get:

$$\nabla \cdot \vec{\xi} = \nabla^2 p \quad (6)$$

This is a Poisson-equation, which can be numerically solved to find p , from where \vec{r} can be derived, which fully defines the projection according to (4). The use of the letter p here is on purpose, because this scalar field indeed resembles the pressure field of the Navier-Stokes equation.

Overall we have shown that the solution of the projection P fulfills the continuity equation. Therefore we can use this projection to combine (1) and (2) in one equation as it was done in [9]:

$$P\left(\frac{\partial \vec{u}}{\partial t}\right) = P\left(-(\vec{u} \cdot \nabla)\vec{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\vec{u} + \vec{f}\right) \quad (7)$$

Due to the fact that $P(\vec{u}) = \vec{u}$, the linearity of the projection and $P(\nabla p) = 0$, since ∇p is rotational free, the final equation describing the flow field is:

$$\frac{\partial \vec{u}}{\partial t} = P\left(-(\vec{u} \cdot \nabla)\vec{u} + \nu\nabla^2\vec{u} + \vec{f}\right) \quad (8)$$

2.2. General advection equation

The structure of the flow-solver allows to simulate additional states, which can represent flow properties, as long as they can be substituted by a scalar value. The propagation of an additional state is modeled by the following equation [9]:

$$\frac{\partial s}{\partial t} = -(\vec{u} \cdot \nabla)s + \kappa\nabla^2 s - \beta s + S \quad (9)$$

The function $s := s(x, y, t) : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ describes the value of the state at every grid point, the variables $\kappa, \beta \in \mathbb{R}$ are parameters that model the diffusion and the dissipation, respectively. The function $S := S(x, y, t) : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ is the source term of the respective state.

This formula is strongly related to the Navier-Stokes equation, aside from the dissipation term. In the following the algorithm is presented that is used to solve a discretized version of (8) and (9) resulting in the flow velocities and additional parameters.

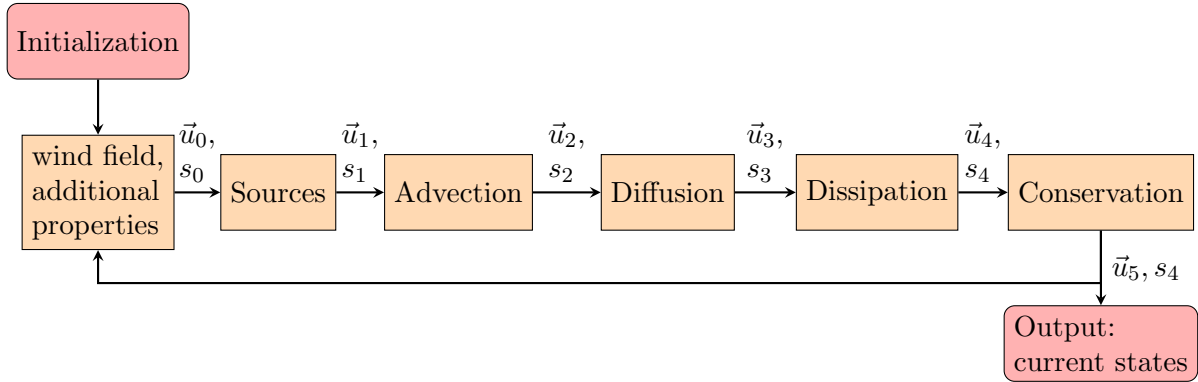


Figure 1: Structure of the flow-solver.

2.3. Structure of the solver

In order to solve the flow velocities at hub-height and additional flow properties (states), a fast, modular, step-wise approximation of the flow-equations has been implemented similar to the solver presented in [9]. The separate terms of the flow equations are solved successively as illustrated in Fig. 1.

In the initialization, the initial windfield \vec{u}_0 and the initial additional states s_0 are constructed. This can be done by assuming a uniform flow for the flow velocities and zero for the additional states. This initial wind field is passed to the iteration-phase of the algorithm, the individual steps of which are described in the following sections.

2.4. Source - Step

In the first step of the algorithm the sources are applied to the simulation domain. This is represented by the external forces \vec{f} in terms of the flow velocities and by the source S for additional parameters. The step description of the algorithm yields:

$$\vec{u}_1 = \vec{u}_0 + \vec{f} \quad (10)$$

$$s_1 = s_0 + S \quad (11)$$

The index indicates the steps of the algorithm as illustrated in Fig. 1. The commonly used Actuator Disk model is used to calculate the thrust of a wind turbine. The rotor area is represented by a porous disk, that reduces the wind velocity. The magnitude of deceleration is calculated by the thrust T of a turbine:

$$T = \frac{1}{2} \rho A u_\infty^2 c_T = \frac{1}{2} \rho A u_\perp^2 \frac{1}{(1-a)^2} 4a(1-a) \quad (12)$$

$$= 2 \rho A u_\perp^2 \frac{a}{1-a} \quad (13)$$

Where A is the swept rotor area, u_\perp is the wind speed at the rotor that is perpendicular to the rotor area. According to momentum theory it holds $u_\perp = u_\infty(1-a)$, where u_∞ is the free-stream velocity perpendicular to the rotor area. $c_T := 4a(1-a)$ is the thrust coefficient, which is brought into relation with the axial induction factor $a \in [0, \frac{1}{2})$ of the turbine. Finally the discretized volume force $\vec{f} = \Delta t \frac{T}{\Delta x \Delta y} \vec{n}$ is applied to the wind field. $\Delta x, \Delta y$ are the lateral and longitudinal grid size, Δt is the time-step size and \vec{n} is the normal vector of the rotor area.

Modeling the source term of an additional state cannot be generalized. As an example, in order to model a turbine induced turbulence intensity a direct relation between the force, that

is generated by the turbine, can be formulated. Here the relation is modeled as a proportional correlation:

$$S = c_{TI} \|\vec{f}\| \quad (14)$$

The coefficient $c_{TI} \in \mathbb{R}$ can be parameterized to scale the induced turbulence intensity properly. The source of confidence coefficient on the other hand can be modeled as a constant that overwrites the current confidence state at the locations close to the measurement point, whenever a new measurement is available, as described in section 2.9.

2.5. Advection step

The advection step involves the non-linear term of the Navier-Stokes equation and it is also the most computational intensive step of the algorithm. In this step, the following is solved:

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla) \vec{u} \quad (15)$$

$$\frac{\partial s}{\partial t} = -(\vec{u} \cdot \nabla) s \quad (16)$$

The algorithm calculates the current values of the states u, v, s from the prior time-step according to the following scheme, which was derived from the method of characteristics:

Let $\vec{\lambda}(\tau) := (x(\tau), y(\tau))^T = (x_c, y_c)^T + \vec{u}_c \tau$ be a path in the domain, that passes the point $(x_c, y_c)^T$ at $\tau = 0$, where $\vec{u}_c = (u_c, v_c)^T$ is the velocity at $(x_c, y_c)^T$ and $\tau \in \mathbb{R}$. This is a characteristic of the wind field and it allows us to parameterize the spatial and temporal components with $t(\tau) = \tau$ such that $s(x(\tau), y(\tau), t(\tau)) = s(\lambda(\tau), t(\tau))$ is only depending on one variable τ . According to the chain rule it holds:

$$\begin{aligned} \frac{ds}{d\tau} &= \frac{\partial s}{\partial \vec{\lambda}} \frac{d\vec{\lambda}}{d\tau} + \frac{\partial s}{\partial t} \frac{dt}{d\tau} \\ &= \frac{\partial s}{\partial \vec{\lambda}} \vec{u}_c + \frac{\partial s}{\partial t} \end{aligned} \quad (17)$$

With (16) we can show, that this expression vanishes, since:

$$\begin{aligned} \frac{\partial s}{\partial \vec{\lambda}} \vec{u}_c + \frac{\partial s}{\partial t} &= \left(\frac{\partial s}{\partial x}, \frac{\partial s}{\partial y} \right) \cdot \vec{u}_c + \frac{\partial s}{\partial t} \\ &= (\vec{u}_c \cdot \nabla) s + \frac{\partial s}{\partial t} \\ &= 0 \end{aligned} \quad (18)$$

Hence s is constant with respect to τ on the path λ . The temporarily discretized statement we derive from this is:

$$\begin{aligned} s(\vec{\lambda}(\tau), t(\tau)) &= s(\vec{\lambda}(\tau - \Delta\tau), \tau - \Delta\tau) \\ &= s(\vec{\lambda}(\tau) - \vec{u}_c \Delta\tau, \tau - \Delta\tau) \\ \Rightarrow s(x_c, y_c, t) &= s(x_c - u_c \Delta t, y_c - v_c \Delta t, t - \Delta t) \end{aligned} \quad (19)$$

where $\Delta\tau := \Delta t$ is chosen to resemble one time step. Since $\vec{\lambda}$ can be chosen arbitrarily, this is valid in the spatial discretization at every grid point. The proof for the velocity vector is analogous.

Tantamount to the above we can say that the current state at the location (x, y) can be derived from the state of the previous time-step at the location $(x, y) - (u, v)\Delta t$. The step description for the algorithm results in:

$$\vec{u}_2(x, y) = \vec{u}_1(x - u\Delta t, y - v\Delta t) \quad (20)$$

$$s_2(x, y) = s_1(x - u\Delta t, y - v\Delta t) \quad (21)$$

The algorithm approximates this by linear interpolation of the states to obtain the value at the position $(x_+, y_+) - (u_+, v_+)\Delta t$ for each grid point (x_+, y_+) , where (u_+, v_+) is the respective velocity.

2.6. Diffusion step

The diffusion step simulates viscous stress and the blending of each state in one time step. In terms of flow velocity this term is defined by the viscosity term. According to [9] the straight forward explicit representation for the step description of the discrete algorithm is:

$$\vec{u}_3 = \vec{u}_2 + \Delta t \nu \nabla^2 \vec{u}_2 \quad (22)$$

$$s_3 = s_2 + \Delta t \kappa \nabla^2 s_2 \quad (23)$$

Here the parameter ν for the viscosity is a synthetic parameter, which needs to be adjusted to measurement or simulation data, in order to correctly represent 3D flow diffusion.

2.7. Dissipation step

The dissipation term reduces the value of a state in each iteration by a defined percentage $\beta \in [0, 1]$ per second. This simulates the disappearance of the respective state from the simulation domain. The step-wise term description for the dissipation is:

$$s_4 = s_3(1 - \beta)^{\Delta t} \quad (24)$$

The exponent Δt is necessary, since the dissipation parameter β is the dissipation rate per second. For the wind speeds u, v an adjusted dissipation term is implemented in the algorithm, although the Navier-Stokes equation does not include a dissipation term. The dissipation term for the wind speeds is set up not to reduce the wind speeds, but rather to reduce wake deficits to better fit to measurement data. The drawback of the 2D model is that the mixing with the free flow is underrepresented due to the missing wake recovery above and below the wake. To emulate the behaviour the following term is implemented in the model:

$$\vec{u}_4 = (1 - \gamma)^{\Delta t} \vec{u}_3 + (1 - (1 - \gamma)^{\Delta t}) \vec{u}_\infty \quad (25)$$

The parameter $\gamma \in [0, 1]$ defines the magnitude of the wake dissipation, the vector $\vec{u}_\infty \in \mathbb{R}^2$ represents the free-flow velocity. This term lets the entire flow field converge to the free-stream wind velocity, while the change is proportional to the difference to the free-stream flow.

2.8. Conservation step

The conservation step is the last step of the flow solver and it is only used for the flow velocities u and v . It represents the continuity equation (2) for incompressible fluids, which ensures that in every time-step the amount of mass entering a finite volume equals the amount of mass leaving it. As already explained before, it is possible to combine the continuity equation for incompressible fluids with the Navier-Stokes equation into one equation by using the Helmholtz-Hodge decomposition. The presented method utilizes a relaxation of the continuity equation with the help of the described projection as explained in the following.

Regarding the conservation of mass a 2D model does not accurately represent a 3D domain, since the application of the continuity equation ensures that all the mass is conserved within the 2D horizontal plane. A turbine in such a model is comparable to a wind turbine model in a wind tunnel that occupies the entire height of the wind tunnel, since there is no possibility for the wind to flow above or below the turbine. Therefore, the jet effect on the sides of the turbine and its wake is higher than expected. Additionally the effect of lateral forces (e.g. caused by a yawed turbine) is damped by the 2D mass conservation, because a shift in the flow direction has to transfer directly to the boundaries of the domain.

To mitigate these "2D effects" a relaxation parameter is introduced into the model. With this the implementation of the relaxed continuity Equation is as follows:

$$\vec{u}_5 = P_\delta(\vec{u}_4) = \vec{u}_4 - \delta \vec{r} \quad (26)$$

Where P_δ is the adjusted version of the projection P explained above. $\delta \in [0, 1]$ is the relaxation parameter. $\vec{r} := \nabla p$, where p is the solution of the Poisson-Equation $\nabla \cdot \vec{u}_4 = \nabla^2 p$.

In the case of $\delta = 1$ the full continuity equation is applied in each time-step. For smaller $\delta \in [0, 1)$ the wind field is not completely projected onto its solenoidal component, but "pushed" into this direction i.e not all the pressure is resolved to fulfill the continuity equation.

2.9. Online measurements

The embedding of online measurements into the dynamic flow model is an extension of the flow solver and will be introduced here briefly. It is a crucial part for the application of the model in order to predict the wind farm flow derived from measurements. The challenge is to mix online information about the current flow, gathered by measurements at distinct point wise locations, into the model, so that a coherent flow field can be established. To achieve this, the confidence coefficient $c(x, y, t) : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ is introduced and is treated by the solver as an additional state. It is initialized as $c(x, y, 0) \equiv 0$. When new measurements are available at the locations $(x_{m,i}, y_{m,i})$ at the time instance t_m a continuous flow field $\vec{m}(x, y, t_m)$ can be derived from these measurements by the shepard-interpolation [10]. Now the prevailing flow field \vec{u} is mixed with \vec{m} according to:

$$\vec{u}_{\text{new}} = c \cdot \vec{u} + (1 - c) \cdot \vec{m} \quad \forall x, y \text{ and } t = t_m \quad (27)$$

This results in the new flow field \vec{u}_{new} . The confidence coefficient gets updated to c_{new} , when a measurement is embedded according to :

$$c_{\text{new}}(x, y, t_m) = \begin{cases} 1 & \text{if there exists an } i \text{ with } (x - x_{m,i})^2 + (y - y_{m,i})^2 \leq d^2 \\ c(x, y, t_m) & \text{otherwise} \end{cases} \quad (28)$$

Where d is a configurable distance, that influences the confidence reach of a measurement.

Since c is a state, its values are propagated downstream with the flow by the flow solver, which means, that the influence of new measurements is predominantly upstream.

3. Results

To demonstrate the performance of the model, comparisons with two other models are presented in this section. In the first part, a comparison is made with a steady-state model, which is based on the well-known Jensen Park model [11]. The second part validates the model presented here with the dynamic model WFSim [12]. For these comparisons an area of $1400 \text{ m} \times 2400 \text{ m}$ is used on which two turbines of the type NREL 5MW [13] are placed at a distance of approximate 6 rotor diameters. The grid spacing is $50 \text{ m} \times 50 \text{ m}$ and the time step is 1 s. The wind speed is set to 8 ms^{-1} , coming from the south with uniform boundary conditions, so that the turbines build a full wake situation. The parameters are set to minimize the RMS error of the wind fields in a domain around the wakes, as illustrated in by the red box in Fig. 2 (right image).

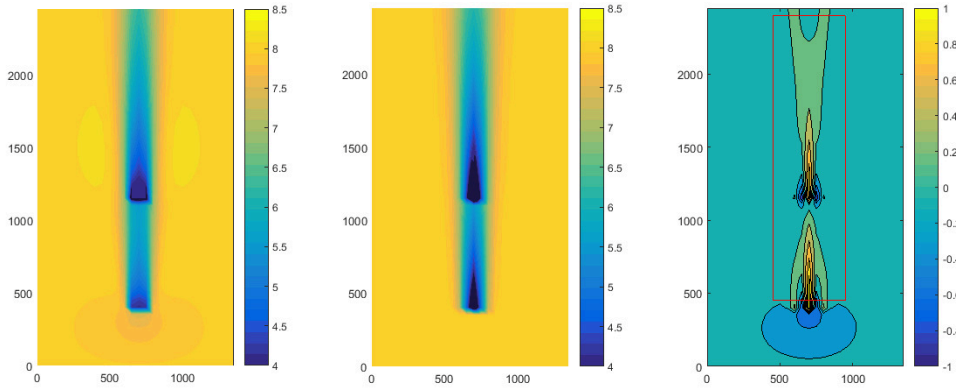


Figure 2: Test Case 1a: The plots show the longitudinal flow velocities in $[\text{ms}^{-1}]$ over the simulation domain computed by the presented dynamic model (left) and the steady-state model (center). The contour plot (right) shows the difference

Table 1: Dynamic model parameters

Parameters	
ν	$11.5 \text{ m}^2 \text{ s}^{-1}$
γ	0.0029
δ	0.03

Table 2: Difference in the wake area

Results	Test case 1a	Test case 2a
RMS	0.244 ms^{-1}	0.230 ms^{-1}
avg. abs. err	0.149 ms^{-1}	0.144 ms^{-1}

3.1. Steady-State Model comparison

For the comparison with a steady-state model the dynamic model has to run for a short time under stationary inflow conditions so that the flow converges to an equilibrium state. To make sure that this equilibrium is reached and that there is no influence of the propagation the simulation time for the convergence was set to 625 s, which equals the time the free-stream flow needs to pass the whole domain twice. The steady-state model, which was used for comparison, calculates the wake deficit based on the Jensen-Park model [14], where wake interactions are modeled according to [15]. Furthermore a Gaussian overlay was used according to [16].

The parameters of the dynamic model are adjusted accordingly in order to minimize the differences in the wake area.

As a performance indicator the average absolute error and the RMS error are calculated for a domain that includes the wakes as depicted by the red box in Fig 2. The first case (Test Case 1a) we look at is the full wake situation, where both turbines run at their local optimum, which means an axial induction factor of $a_{1,2} = \frac{1}{3}$. The determined parameters and the results are shown in Table 1 and Table 2.

In the second case (Test Case 2a) the results of the two models are compared, where the axial induction of the turbines is reduced to $a_{1,2} = \frac{1}{6}$.

3.2. Dynamic model - WFSim

Next the dynamic model will be compared to WFSim [12]. For the comparison we look at the equilibrium state of both models. But this time the relaxation factor δ for the continuity equation was set to 1, since WFSim solves the 2D continuity equation completely. Again in the first test case (Test Case 1b) the models are compared for a standard situation, where the axial induction factors are $a_{1,2} = \frac{1}{3}$.

In the second test case (Test Case 2b) the axial inductions of the turbines are reduced again to $a_{1,2} = \frac{1}{6}$.

Table 3: Parameters for comparison with WFSim

Parameters	
ν	$5.2 \text{ m}^2\text{s}^{-1}$
ε	0.0068
δ	1

Table 4: Difference between dynamic model and WFSim

Results	Test Case 1b	Test Case 2b
RMS	0.345 ms^{-1}	0.187 ms^{-1}
avg. abs. error	0.204 ms^{-1}	0.124 ms^{-1}

Table 5: Performance Test: The CPU-Time is computed for a flow-simulation with two states per grid point (u, v), the values in the brackets represent a flow-simulation with one additional state i.e. three states per grid point (u, v, s).

Resolution	grid points	avg. CPU Time \pm std	avg. CPU Time per iter.
50 m \times 50 m	3136	4.48 s \pm 0.14 s (5.08 s \pm 0.16 s)	0.007 s (0.008 s)
25 m \times 25 m	12319	13.41 s \pm 0.09 s (14.38 s \pm 0.13 s)	0.022 s (0.024 s)
20 m \times 20 m	19118	19.55 s \pm 0.21 s (21.55 s \pm 0.19 s)	0.033 s (0.036 s)
10 m \times 10 m	76156	75.21 s \pm 0.40 s (84.25 s \pm 1.09 s)	0.125 s (0.140 s)

3.3. Performance test

In order to examine the scalability of the algorithm, 600 s of a virtual wind farm consisting of 9 turbines are simulated on a standard PC¹. The size of the domain is fixed to 2400 m \times 3150 m, while the resolution is increased according to Table 5. The experiment is performed for a standard flow-simulation with two states per grid point (u, v) and for three states at each grid point (u, v, s). Each resolution is simulated 10 times and the average calculation time plus standard deviation is shown in Table 5. The results of the CPU time show a clear linear dependence to the number of grid points with a coefficient of determination of $R^2 = 0.9988$. The simulation of an additional state increases the CPU-Time by approximately 12%. In the most complex test case 228468 states have been simulated, while the average calculation time of one time-step took 0.14 s.

4. Discussion

Although the comparisons of the presented model with the steady-state model and WFSim show a relatively good agreement for the equilibrium state, the propagation of the model still needs to be verified. In order to do this, validations with high-fidelity simulations will be carried out in the future.

The configuration variables (ν, ε, δ), allow the model to be parameterized for varying atmospheric conditions. However it needs to be validated how accurately the two-dimensional flow simulation can describe three dimensional flow. As long as the vertical mixing of the flow is sufficiently strong the differences over the height of the rotor area are expected to be neglectable. But especially in stable atmospheric conditions the vertical mixing is very limited, which can lead to significant differences of wind speed and wind direction (wind shear and wind veer) over the height of the rotor area. It will be investigated if separate parallel instances of the flow-solver, that represent different heights over the rotor area, can be used to improve the power estimation. Also for this, validations with high-fidelity simulations will be done in the future.

Overall the dynamic model shows relatively good scalability and the possibility of simulating additional flow properties can be utilized for a range of applications. E.g. turbine induced turbulence intensity (cf. equation (14)) can be used to estimate wake induced loads at downstream turbines or the confidence coefficient (cf. equation (28)) is necessary to integrate

¹ Intel(R) Core(TM) i5-3340M CPU @2.70 GHz

online measurements into the simulation as explained in section 2.9. But the parameterization and validation of these additional states is a future task.

5. Conclusion

In this work a dynamic flow model for real-time application is introduced. The model has relatively low computational complexity and scales linearly with domain size (i.e. number of grid points). The advantage of the model is, that next to the flow velocities, additional states can be added to the simulation for a variety of applications, without increasing the computational complexity significantly. In future work validations with high-fidelity simulations will be made to further verify the model and its dynamic properties.

Acknowledgements

This work was partially funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) in the scope of the OWP Control project (FKZ 0324131A). The research is supported by the German Academic Exchange Service (DAAD) with funds from the Federal Ministry of Education and Research (BMBF).

References

- [1] M. Méchali, R. Barthelmie, S. Frandsen, L. Jensen, and P.-E. Réthoré. Wake effects at Horns Rev and their influence on energy production. In *Proceedings of the European Wind Energy Conference and Exhibition, Athens, Greece, 2006*.
- [2] P. Frohboese and C. Schmuck. Thrust coefficients used for estimation of wake effects for fatigue load calculation. In *European Wind Energy Conference*, pages 1–10, 2010.
- [3] C. Kost, J. N. Mayer, J. Thomsen, N. Hartmann, C. Senkpiel, S. Philipps, S. Nold, S. Lude, N. Saad, and T. Schlegl. Levelized cost of electricity renewable energy technologies. *Fraunhofer Institute for Solar Energy Systems ISE*, 2013.
- [4] D. Medici and J. Dahlberg. Potential improvement of wind turbine array efficiency by active wake control (awc). In *European Wind Energy Conference, Madrid*.
- [5] S. Boersma, B. M. Doekemeijer, P. M. O. Gebraad, P. A. Fleming, J. Annoni, A. K. Scholbrock, J. A. Frederik, and J. W. van Wingerden. A tutorial on control-oriented modelling and control of wind farms. *ACC*, 2017.
- [6] R. J. Barthelmie, S.C. Pryor, S. T. Frandsen, K. S. Hansen, J.G. Schepers, K. Rados, W. Schlez, A. Neubert, L.E. Jensen, and S. Neckelmann. Quantifying the impact of wind turbine wakes on power output at offshore wind farms. *Journal of Atmospheric and Oceanic Technology*, 27(8):1302–1317, 2010.
- [7] H. Beck, A. Rott, D. Trabucchi, and M. Kühn. Volumetric wind field measurements of wind turbine wake with long-range lidar. In *EWEA2015, 17-20 November 2015, Paris, France, 2015*.
- [8] H. Bhatia, G. Norgard, V. Pascucci, and P.-T. Bremer. The Helmholtz-Hodge decomposition survey. *IEEE Transactions on visualization and computer graphics*, 19(8):1386–1404, 2013.
- [9] J. Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH 1999*. Association for Computing Machinery (ACM), 1999.
- [10] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.
- [11] A. Rott and M. Kühn. Combining Model-Based and Data-Driven Optimization of Wind Farm Operation in a Learning Database. In *EAWWE 10th PhD Seminar on Wind Energy in Europe, Orléans, France, 2014*.
- [12] S. Boersma, P.M.O. Gebraad, M. Vali, B.M. Doekemeijer, and J.W. van Wingerden. A control-oriented dynamic wind farm flow model: WFSim. *Journal of Physics: Conference Series*, 753:032005, 2016.
- [13] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system development. *National Renewable Energy Laboratory, Golden, CO, Technical Report No. NREL/TP-500-38060*, 2009.
- [14] N. O. Jensen. A note on wind generator interaction. *Risø-M-2411 Risø National Laboratory Roskilde*, pages 1–16, 1983.
- [15] I Katic, J Højstrup, and NO Jensen. A simple model for cluster efficiency. *European wind energy association conference and exhibition*, (October):407–410, 1986.
- [16] B. F. Lund, P. N. Oma, and T. Jakobsen. Wind farm production maximization through joint optimization of turbine efficiency and yaw. simulated case studies. *Poster EWEA, 2014*.