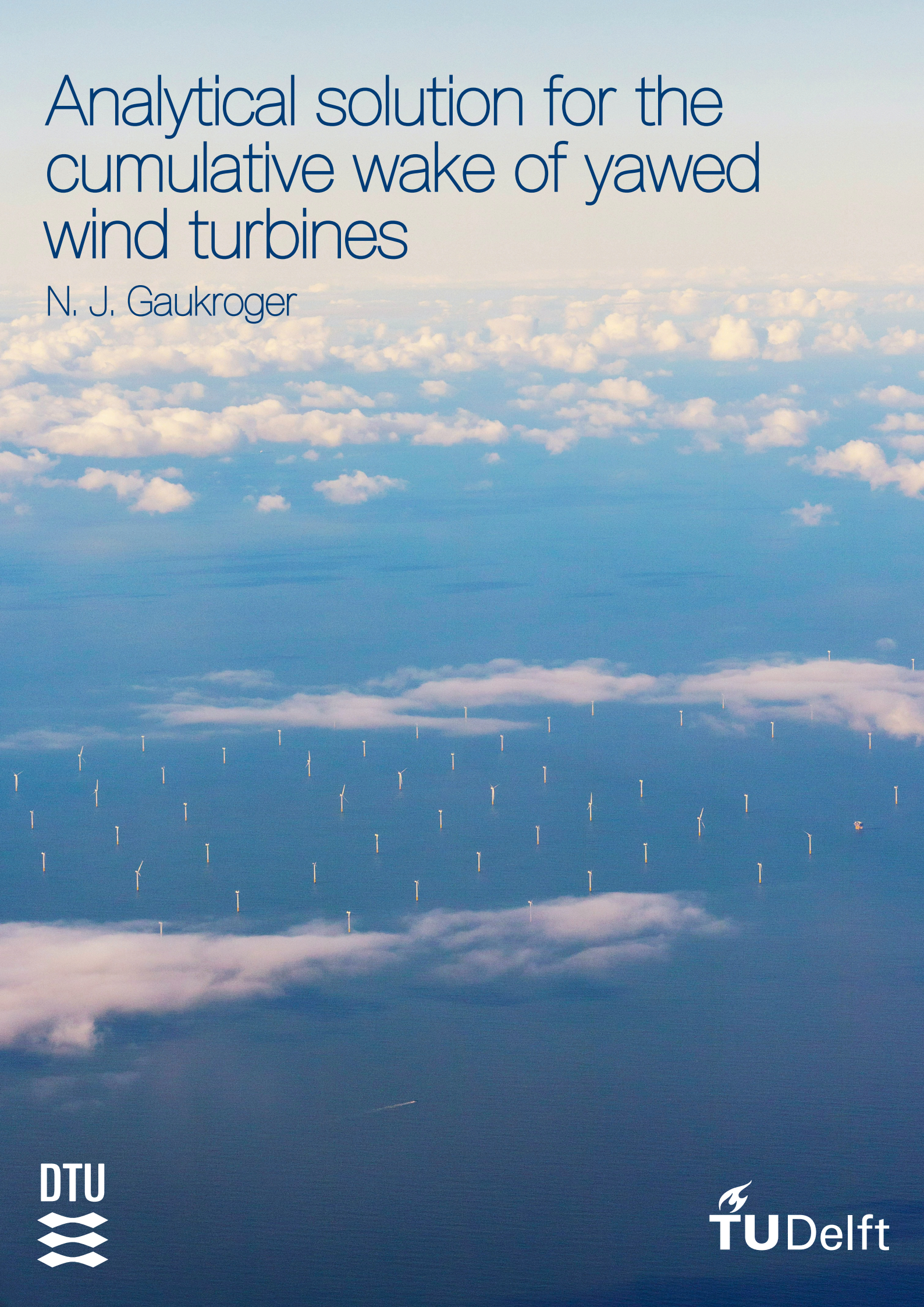
Analytical solution for the cumulative wake of yawed wind turbines

N. J. Gaukroger



DTU

TU Delft

Analytical solution for the cumulative wake of yawed wind turbines

by

Nils Joseph Gaukroger

to obtain the degrees of

Master of Science
in Engineering (European Wind Energy)
at Technical University of Denmark

Master of Science
in Aerospace Engineering
at Delft University of Technology

To be defended on the 5th October 2022

Supervisors:	Maarten Paul van der Laan	DTU
	Dr. Majid Bastankhah	Durham University
	Prof. ir. S.J. Watson	TU Delft
Defence committee:	Dr. D.J.N. Allaerts	TU Delft
	Dr. D. Modesti	TU Delft
Start date:	1st November 2021	
End date:	11th September 2022	
Student numbers:	DTU:	s203209
	TU Delft:	5319188

Cover photo by Joel Arbaje
`https://unsplash.com/photos/Ah4i-kc9B9Y`

An electronic version of this thesis is available at
`https://repository.tudelft.nl` and `https://findit.dtu.dk`

The code written for this thesis is available at
`https://github.com/NilsGaukroger/Analytical-solution-for-the-cumulative-wake-of-yawed-wind-turbines`

Abstract

This thesis sets out to improve the physical grounding and predictive accuracy of cumulative wake effect modelling within wind farms with yawed turbines. It derives an analytical solution for the lateral velocity field within a wind farm and compares its predictions to those of computational fluid dynamics.

A parametric study is performed using a Reynolds-averaged Navier-Stokes (RANS) solver with the k-ε-f_P turbulence model, Joukowsky rotor-based actuator disc, and neutral log-law inflow within the PyWakeEllipSys framework to determine the effects of yaw angle, thrust coefficient, and turbulence intensity on the lateral wake.

The results of this parametric study are used to solve an approximate form of conservation of mass and momentum in the lateral direction for a turbine within a wind farm. The solution is an explicit equation predicting the lateral velocity distribution and lateral wake deflection within a wind farm of arbitrary layout and with arbitrarily yawed turbines. It also provides a first mathematical proof of secondary wake steering.

The solution is implemented in Python and used to predict the velocity distributions in several wind farm cases, including for a single turbine, a two-turbine arrangement, and two wind farm cases with aligned and staggered layouts. These predictions are then compared against those of the RANS setup. The model significantly overestimates wake deflections unless corrected to neglect the near wake, but the corrected version shows promise, particularly in predicting wind farm power of the staggered layout, where the prediction is 19% closer to the RANS result than the prediction that considers lateral velocities equal to zero.

Acknowledgements

‘No man is an island’, and though this thesis bears only my name, it is as much credit to the patience, perseverance, and passion of those around me as it to my own.

I have had the good fortune to be supervised by some brilliant researchers and people. To Paul and Mads at Risø, thank you for your perpetual willingness to help me, I believe the greatest compliment I can pay you is to tell you you have made me want to come back. To Majid, your work continues to inspire myself and others, and should I go on to achieve even half of what you have, I will consider it a success.

To my friends, old and new, thank you for a wonderful last two years. As much as wind energy aerodynamics will continue to fascinate me for many years, it pales in comparison to the joy of spending time with you all.

And of course to my family, who have always done their utmost to support me in everything I do, and to whom I owe more than I could ever expect to repay.

Lastly, to my Dad, I hope I did you proud.

Contents

Nomenclature

Acronyms & initialisms

BEM	Blade element momentum
CVP	Counter-rotating vortex pair
DTU	Danmarks Tekniske Universitet (Technical University of Denmark)
DWM	Dynamic wake meandering
HAWT	Horizontal axis wind turbine
LES	Large-eddy simulation
RANS	Reynolds-averaged Navier-Stokes
WFC	Wind farm control

Greek Symbols

ΔU	Streamwise velocity deficit	$\mathrm{m\,s^{-1}}$
γ	Yaw angle (positive clockwise when viewed from above)	$^\circ$
ν	Kinematic viscosity	$\mathrm{m^2\,s^{-1}}$
ρ	Fluid density	$\mathrm{kg\,m^{-3}}$

Roman Symbols

$\boldsymbol{U}$	Ensemble-averaged total velocity	$\mathrm{m\,s^{-1}}$
a	Axial induction factor	
C_T	Thrust coefficient	
D	Rotor diameter	m
f	Self-similar function	
f_i	External body force vector per unit volume	$\mathrm{N\,m^{-3}}$
I	Turbulence intensity	
k	Turbulent kinetic energy	$\mathrm{m^2\,s^{-2}}$
p	Pressure	Pa

t	Time	s
U	Ensemble-averaged streamwise velocity	$\mathrm{m\,s^{-1}}$
u_i	Flow velocity vector	$\mathrm{m\,s^{-1}}$
V	Ensemble-averaged lateral velocity	$\mathrm{m\,s^{-1}}$
W	Ensemble-averaged vertical velocity	$\mathrm{m\,s^{-1}}$
x	Streamwise coordinate	m
y	Lateral coordinate	m
z	Vertical coordinate	m

Subscripts

$(\cdot)_\infty$	Freestream value
$(\cdot)_h$	Hub-height value

List of Figures

List of Tables

CHAPTER 1

Introduction

Wind and other renewable energy sources are vital means to meeting future global energy needs whilst mitigating against the worst effects of climate change and ensuring energy security. Global energy demand continues to grow at 2% per year, but despite their evident consequences for the climate, fossil fuels still account for 70% of the energy mix [67]. Recent geopolitical events in Europe and concerns over the safety and costs of nuclear power have led to ambitious targets regarding energy provision from renewable sources, which are set to account for almost 95% of the increase in global power capacity before 2026 [67]. The realisation of these targets demands a substantial increase in renewable capacity in the near future, including in wind energy. This itself implies not only the planning and installation of ever larger new wind farms, but also the retrofitting of existing farms, which typically occupy sites of considerable wind resource.

Aerodynamics is pivotal to the planning and operation of these farms. The study of wind turbine aerodynamics, particularly for the optimisation of the horizontal axis wind turbine (HAWT), began with the seminal works of Froude [52] and Rankine [116] on Blade Element and Momentum Theory, respectively. This approach was later taken up independently by Lanchester [90], Betz [16], and Zhukovsky [149] to formulate Betz's Law, which gives the theoretical limit on efficiency of a thin rotor [79]. In 1926, Glauert then combined the locally focused Blade Element Theory with the globally focused Momentum Theory and, accounting also for the rotation in the wake, produced Blade Element Momentum (BEM) theory [54], a cornerstone of modern wind turbine aerodynamics.

From the 1980s until very recently, the field has focused primarily on wind turbine wakes. Initiated by the studies of Lissaman [94] and Vermeulen [140], there is now a wealth of literature on the flow regions surrounding a wind turbine, and the reader is directed to reviews by Crespo et al. [32], Vermeer et al. [139], Göçmen et al. [55], and Porté-Agel et al. [111] for more details on the topic. In contrast with individual wind turbine aerodynamics, their cumulative effects, for example for within wind farms, is a less mature discipline. Wind farm aerodynamics has been a topic of interest since the early 2000s, and several approaches to determining flows within wind farms have been developed. These approaches are typically divided into 'top-down', which approaches the problem from the larger, atmospheric scale; and 'bottom-up', which seeks to combine single turbine flows via a superposition method [111, 136]. Recent studies have highlighted that due to the satisfactory performance of these classical wake superposition methods in combination with simpler wake models, the study of the underlying physics has been somewhat neglected [14, 99]. There is, therefore, a need for a more physics-based understanding of cumulative wake effects. This project aims to contribute to that understanding through the development of an analytical solution for lateral velocities in a wind farm with yawed wind turbines.

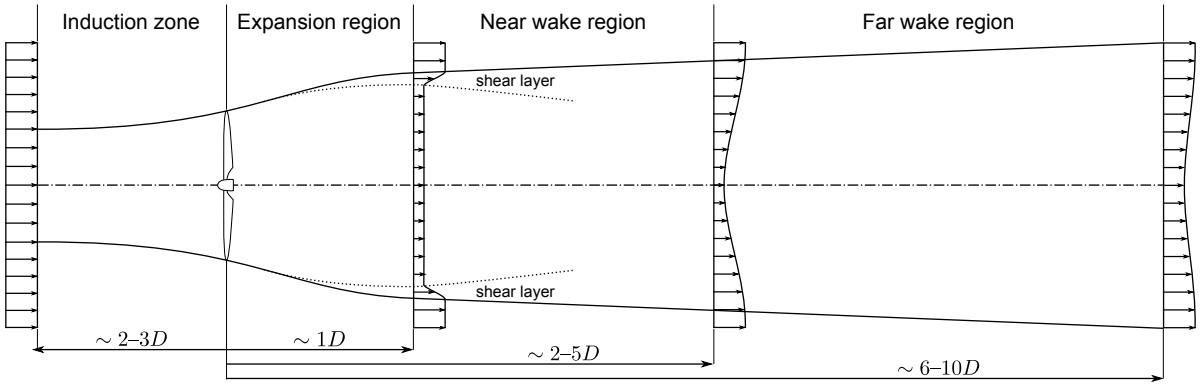


Figure 1.1: A schematic representing the flow regions around a wind turbine. Figure by Dries Allaerts.

1.1 Wind turbine flows

Typically the flow through a wind turbine is divided into regions as shown in Figure 1.1 [32, 111, 139]. The presence of the turbine affects the flow both upstream and downstream [57, 143], starting with the induction region. The flow behaviour here and in the expansion region is well described by momentum theory, which predicts a deceleration and expansion in front of the turbine, accompanied by an increase in pressure. These predictions have been observed in lidar measurements by Simley et al. [131], and Medici et al. [98] has developed an expression based on vortex sheet theory which describes the velocity deficit as a function of the upstream distance x, and the axial induction factor a predicted by momentum theory,

$$\frac{U}{U_\infty} = 1 - a\left(1 + \frac{2x}{D}\left(1 + \left(\frac{2x}{D}\right)^2\right)^{-\frac{1}{2}}\right), \tag{1.1}$$

where D is the rotor diameter, and U_∞ the freestream velocity.

Over the rotor there is then a sudden drop in pressure, followed by a region of highly complex flow with non-uniform deficits in pressure and velocity called the expansion region. Momentum theory predicts that this pressure will recover and the streamtube expand until the pressure reaches ambient, marking the start of the near wake. Throughout these two regions, the expansion and near wake, features of the rotor are clearly distinguishable. Most evident are the helical tip vortices, which form shortly downstream of the rotor from the vortex sheets shed from the blades. These vortices, along with the rest of the flow within the wake, rotate in the opposite direction to the blades, a consequence of conservation of angular momentum which can be explained by Euler's turbine equation [57, App. A]. For sufficiently high tip-speed ratios, the inclination angle of these helices is small enough that they can be interpreted as a cylindrical shear layer separating the wake from the external ambient flow [32, 139].

Due to turbulent diffusion, the thickness of the shear layer encircling the wake grows with downstream distance. As a result, the near-wake is often modelled as a central region of uniform velocity deficit, bordered by shear layers of variable deficit, which expand downstream until the central region is engulfed [12, 140]. The consummation of this shear layer denotes that the wake is now fully-developed, and thus the far wake region has begun.

1.2 Far wake

Unlike the near wake, the flow characteristics of the far wake are almost universal. For example, neglecting ambient shear the perturbation profiles of both velocity and turbulence intensity are axisymmetric with self-similar, Gaussian profiles. The magnitude of this 'velocity deficit' profile (being the difference between the incoming and wake flow velocities), initially a function of the momentum extracted by the rotor, decays as the wake entrains momentum from the external flow and grows radially with downstream distance. The growth of a turbulent, axisymmetric wake is expected to follow a 1/3 power law in the far downstream [110, §5.4.4, 91]; however, this scaling assumes that the turbulent momentum transport is dominated by the shear within the wake, whereas typically it is in fact atmospheric turbulence from roughness and thermal effects that determines wake expansion. Thus, other empirical scalings have been proposed and generally the wake growth rate, as it is known, is closer to linear.

When turbines are grouped into wind farms, it is not uncommon for one to operate in the wake of another. The reduced velocity and increased turbulence intensity symptomatic of wakes lead to reduced power output and increased dynamic loads (and hence reduced fatigue life) at the downstream turbine. Furthermore, given inter-turbine spacings in wind farms are typically between 3 and 10 rotor diameters, hereafter D, turbines most often operate within the far wake of one another [111].

These factors have motivated the development of an array of modelling techniques for the far wake, which can be categorised according to fidelity (and by consequence also computational cost) as per Table 1.1 [29]. Broadly speaking, the models follow one of two methods, which themselves can be divided into categories.

On the one hand, there are engineering wake models, which are attempts to distill the necessary information required to model wakes by various degrees to reduce the computational cost whilst adequately describing their behaviour. The most simple of engineering wake models are the analytical models. These assume a wake shape based often on observations and theory and then derive the development of the wake with downstream distance via conservation of

Table 1.1: Hierarchy of wake models in order of increasing complexity. Adapted from [29].

Category	Examples
Analytical	Park model [72, 76] Frandsen model [48] Gaussian models [11]
Linearised or parabolic RANS	Ainslie model [2] Fuga [108] UPMWAKE [33]
Other models	Dynamic wake meandering (DWM) model [92] Vortex models Stochastic models
Elliptic RANS	Actuator disc Actuator line
LES	Blade resolved

mass and/or momentum. The most famous example is the Park model [72, 76], which was the first wake model, and remains one of the most widely used, particularly in industry [125].

Linearised or parabolic RANS models do not necessarily assume a wake shape, but instead solve simplified versions of the Reynolds-averaged Navier-Stokes equations. The Ainslie model [2], for example, is a finite-difference solution to the thin shear-layer simplification of the RANS equations. Models within this category lend themselves to formulation either as lookup tables, such as Fuga [55, 108], or as simple computer codes, such as UPMWAKE [33], for simplicity of application.

There exist some engineering wake models that do not easily fit into one of the aforementioned categories. The dynamic wake meandering model [92], for example, was borne of the need to describe observations of unsteady wake behaviour, where wakes were observed to deviate from a straight line path downstream of the rotor, due to turbulent structures in the atmosphere with length scales larger than that of the wake width [41]. On the other hand, vortex models represent the wake by means of vorticity, and then predict the wake properties by solving transport equations of vorticity (or some simplified form). Given a vorticity distribution, one can then calculate the corresponding velocities from the Biot-Savart law. The author is directed to Sanderse et al. [124] for further details on vortex models. There are also stochastic models, which approximate some wake properties as stochastic processes with particular distributions, often based on large-eddy simulations (LES) [37].

Finally, there are CFD models. Instead of seeking to simplify the problem using theory or observation like engineering wake models, these aim to solve some filtered version of the Navier-Stokes equations directly. For Reynolds-averaged Navier-Stokes (RANS) approaches, this means solving the RANS equations with the use of a turbulence model to approximate the Reynolds stresses. Even just considering the modelling of wakes, many of these turbulence models exist [85]. For LES, more of the turbulence spectrum is resolved than with RANS, but a sub-grid scale model is employed for the smallest, unresolved scales [110, Ch. 13]. CFD models model the force distributions on the rotor directly, and thus require some kind of rotor model. There are various options for a rotor model [112], these are summarised in order of increasing fidelity in Table 1.1. The actuator disc is similar to the concept used in axial momentum theory, but the uniform thrust distribution used there is just one possible distribution. Actuator discs can also have analytical thrust distributions such as that described by the Joukowsky (Жуко́вский) rotor [133], or that come from aerofoil data, both of which methods also include tangential forces. More details on actuator discs are given in Section 2.6. The next increment in terms of fidelity is the actuator line model, which, instead of using azimuthally integrated distributions, has discrete blades. The forces on the blades are derived from BEM theory, which assumes inviscid flow and therefore does not resolve the viscous sublayer around the blade. Blade resolved models go one step further and do model the viscous sublayer.

Naturally, each of these wake modelling approaches has its advantages and disadvantages. As mentioned previously, with increasing fidelity comes increasing computational cost. For instance, the computational expense of CFD models renders them almost infeasible for use in common applications such as wind farm layout optimisation, which in the early stages can require the computation of thousands or millions of cases [62, 125]. However, RANS (and even sometimes LES) is frequently used within industry for annual energy production (AEP) assessments, and LES in particular is popular within academia. Chapter 2 gives more details of codes in use within a more complete description of CFD methods. The principle of engineering wake models is to reduce the problem to only the most important variables. As such, the derivation of engineering wake models requires a better understanding of the relevant

importance of the physical processes involved than is necessarily required for higher fidelity models. The associated decrease in computational expense enables the investigation of a larger number and range of cases. From a theoretical perspective, the ability to describe a problem in its simplest form, without oversimplifying, can provide the greatest insight into the nature of the problem. For this reason, this project will focus on analytical wake models, in particular the Gaussian model of Bastankhah and Porté-Agel [11, 12].

The Gaussian wake model describes the velocity deficit in the wake as

$$\frac{\Delta U}{U_\infty} = \left(1 - \sqrt{1 - \frac{C_T}{8\left(k^* x/D + 0.2\sqrt{\beta}\right)^2}}\right) \exp\left(-\frac{1}{2\left(k^* x/D + 0.2\sqrt{\beta}\right)^2}\left\{\left(\frac{z - z_h}{D}\right)^2 + \left(\frac{y}{D}\right)^2\right\}\right), \quad (1.2)$$

where β is a function of the thrust coefficient, C_T, and can be expressed as

$$\beta = \frac{1}{2}\frac{1 + \sqrt{1 - C_T}}{\sqrt{1 - C_T}}, \quad (1.3)$$

k^* is the wake growth rate mentioned previously, x, y, and z the streamwise, lateral, and vertical coordinates, respectively, and z_h the turbine hub height. Several studies have established that the wake growth (or recovery) rate, is directly proportional to the incoming turbulence intensity, I_∞ [14, 27, 106, 129, 148]. As a result, the predominant factors in determining the wake deficit profile are the thrust coefficient, C_T, where a larger C_T will increase the deficit at a fixed x, and the incoming turbulence intensity I_∞, where a larger I_∞, and therefore wake recovery rate, will stretch the profile laterally and vertically, thus reducing the peak deficit at a larger x. This is why cases of low/high C_T and I_∞ are used as validation for new wake and turbulence models [84, 88, 89]. It is important to clarify at this stage the distinction between the streamwise incoming turbulence intensity, $I_{\infty,u} = \sigma_u/U$, and the total incoming turbulence intensity,

$$I_\infty = \frac{\sqrt{\frac{1}{3}\left(\sigma_u^2 + \sigma_v^2 + \sigma_w^2\right)}}{\sqrt{U^2 + V^2 + W^2}} = \frac{\sqrt{\frac{2}{3}k}}{|\boldsymbol{U}|} \quad (1.4)$$

where $\sigma_u = \mathrm{std}(u) = \sqrt{\overline{(u')^2}}$, $U = \mathrm{mean}(u) = \bar{u}$, and $k = \frac{1}{2}\left(\sigma_u^2 + \sigma_v^2 + \sigma_w^2\right)$, the turbulent kinetic energy. The streamwise version is very common in the literature, but the RANS model employed by this thesis uses total turbulence intensity, thus this will be the standard.

1.3 Cumulative wake effects

Typically, turbines do not operate in isolation, but rather in wind farms of tens to hundreds of turbines. Due to logistical and electrical constraints, inter-turbine spacings in wind farms are such that turbines very often operate within the far wake of one another. Moreover, turbines can expect to operate in the wake of not just one, but several upstream turbines. Using engineering wake models, the traditional approach to modelling these cumulative wake effects has been an atomistic one. A two-step method is employed that treats each turbine separately, finding first the value of its wake velocity deficit at a given location using one of the engineering models named above, before combining the contribution of each wake at these locations using a superposition method.

Table 1.2: Characteristics of traditional atomistic approach to estimating cumulative wake effects within wind farms with engineering wake models in the literature. Adapted from Bastankhah et al. [14].

A. Incoming velocity		B. Superposition method	
A.I. Global	A.II. Local	B.I. Linear	B.II. Root sum square
$\Delta U_i = U_0 - U_i$	$\Delta U_i = U_{in,i} - U_i$	$\Delta U = \sum_{i=1}^{n} \Delta U_i$	$\Delta U = \sqrt{\sum_{i=1}^{n} \Delta U_i^2}$

This atomistic cumulative wake method can be characterised by the two facets presented in Table 1.2, namely, A. the reference wind speed used to calculate the individual velocity deficits, ΔU_i for wind turbine i (WT_i); and B. the method used to superpose these to find the total velocity deficit ΔU [7, 14, 150]. Individual wake models are normally expressed as a (normalised) velocity deficit, as in Equation 1.2, which is defined as the difference between the incoming velocity and the wake velocity. Assuming large inter-turbine spacing and thus weak interaction between wakes, one can approximate the incoming velocity at the downstream turbine by the freestream, U_0 [94, 76], denoted in Table 1.2 as A.I. This assumption, despite significantly simplifying the problem, is often too strong an assumption, leading to erroneous results. It can be removed so that the inflow velocity is instead the rotor-averaged wind speed perceived at each turbine, $U_{in,i}$ - A.II. in Table 1.2. This must be determined consecutively from upstream to downstream, but improves model accuracy [106, 142].

There are also several ways to approach the superposition of the individual turbine wakes. Both methods in Table 1.2 involve the summation over WT_i, where i goes from 1 to n, the number of turbines upstream of the position in question, and were derived in an attempt to conserve certain flow properties. Lissaman [94], for instance, used a passive scalar analogy to derive the linear superposition method shown under B.I. in Table 1.2, which claims to conserve momentum deficit. The root-sum-square approach of Katić et al. [76] (B.II.) makes similar conservation claims, this time about kinetic energy. The theoretical justification of these claims is dubious however, and they are known to have problems in certain cases [56, 141]. For example, a combination of A.I. and B.I. is known to result in negative velocities in large wind farms [14].

In summary, methods based on combinations from Table 1.2 should be considered empirical relations at best. This realisation has recently led to the development of a new branch of holistic wake models, which, instead of treating each turbine separately and applying an ad hoc superposition method, derive equations for wake velocities within wind farms from the governing equations for wind farms. Zong and Porté-Agel [150] developed a momentum-conserving wake superposition model based on the concept of mean convection velocity, which is used as a weighting in calculating the combined velocity deficit from a weighted sum of the velocity deficits of each turbine. Their method is relatively expensive computationally, however, as the mean convection velocity and combined velocity deficit equations must be solved iteratively. Bastankhah et al. [14] avoided this issue by developing an explicit cumulative wake model based on a simplified version of the Reynolds-averaged Navier-Stokes equations coined the 'conservation of momentum deficit'. By equating the integral of the change in momentum deficit with the thrust force, and then assuming a self-similar Gaussian wake profile, they provide a single equation which predicts the streamwise velocity at any point within the wind farm. Despite the intrinsic approximations typical of any engineering wake model, the results are promising, and other authors have begun to build upon the principle [7, 97].

1.4 Wake steering

The inter-turbine aerodynamic interaction that arises when turbines are situated in wind farms can be accounted for in coordinating the control actions of individual turbines to improve the overall performance of a wind farm. The field of research concerned with this is called wind farm control (WFC) [145]. The primary aims of WFC include 'increasing power production, reducing turbine loads, and providing electricity grid support services', and it can be categorised according to the three distinct approaches within the literature to achieving these aims [145]:

1. *Axial induction control* (AIC) is an approach where wake interactions are modified by derating upstream or uprating downstream turbines, decreasing loading or increasing power production, respectively. This method can also be used to provide ancillary grid services.
2. *Wake mixing* also uprates and downrates turbines, but on a much shorter timescale than with AIC, to mitigate wake interaction losses.
3. Finally, *wake steering* involves the yawing or tilting of turbines to redirect or 'steer' the wakes they produce. Steering wakes away from downstream turbines can be used to increases their power production or avoid partial wakes, which lead to asymmetric loading.

Despite unconvincing results in wind tunnel studies by Campagnolo et al. [24, 25], a marginal power improvement was demonstrated using AIC in control field trials at a commercial wind farm by Hoek et al. [60]. Wake mixing, or dynamic induction control (DIC), is a much less mature technology, demonstrated first in 2018 in simulations performed by Munters and Meyers [105], with a first attempt at validation in the wind tunnel by Frederik et al. [50] in 2020. Whilst recent focus has been on examining the effect of DIC on the wake and understanding the physics [61], DIC is still developing, evidenced by the introduction of the *helix* approach, which triggers wake mixing using individual pitch control [51]. The potential gains from wake steering have been well documented in the literature, through simulations [53, 73], wind tunnel studies [13, 24, 26], and field campaigns [1, 36, 44, 45, 46, 65, 132]. The reader is referred to Houck [63] for an up-to-date review of studies on all three wake management techniques.

Despite being the most advanced control approach in terms of commercial realisation, with Siemens Gamesa Renewable Energy offering it as part of their 'Wake Adapt' feature [130], wake steering still has its challenges [100, 145]. The issue lies primarily in understanding the complex response of a wake to the yaw misalignment of a turbine. As shown in Figure 1.2, on a basic level yawing a turbine misaligns the thrust force from the incoming wind direction, redistributing momentum from the streamwise to the lateral direction in the wake. As a result, the yawed turbine produces less power, but the deflected wake can be steered away from downstream turbines, which produce more power as a result. The increase in power production at the downstream turbine outweighs the decrease at the upstream turbine, thus increasing the overall power production.

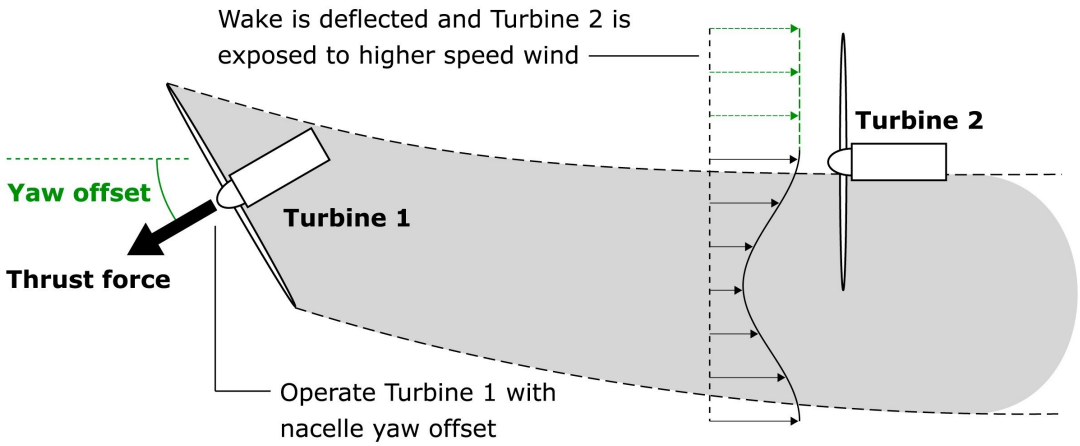


Figure 1.2: Overview of yaw-based wake steering control strategy for mitigating wake effects within wind farms. The schematic shows a top-down view of two wind turbines aligned with the incoming freestream wind, and demonstrates wake deflection by the yaw offset of Turbine 1, highlighted in green, which leads to a higher rotor-averaged wind speed at Turbine 2. Figure taken from Kheirabadi and Nagamune [77].

Besides simply being deflected, the wake of a yawed turbine also has a different cross-sectional shape to that of its unyawed counterpart. The introduction of a counter-rotating vortex pair (CVP) leads the wake to deform into a kidney shape as it progresses downstream [10, 12, 43, 66]. Moreover, as a result of the wake rotation and this wake deformation, wake deflection is asymmetric with yaw offset angle [10, 12, 43]. All of this behaviour is also strongly affected by atmospheric conditions such as stratification and wind veer [100].

The additional lateral component of the thrust force also leads to a 'lateral wake' [5, 6], i.e. a significant lateral velocity component in the wake in the form of a CVP that results in the lateral deflection of the streamwise velocity deficit. The persistence of this lateral wake can create a non-aligned inflow at downstream turbines and, as first observed by Fleming et al. [43], the CVP within a steered wake can deflect the wake of a downstream turbine, even if that turbine is not yawed, in a process termed 'secondary wake steering'.

Field implementation of wake steering is typically implemented by forming a lookup table with pre-optimised values of yaw angle for a set of wind conditions, e.g. wind speed, wind direction, and turbulence intensity. This requires a model of wake steering for the wind farm that potentially includes wake shape and secondary steering. Moreover, the model should be computationally cheap so as not to make the optimisation process to lengthy. Development of such models began with that of Jiménez et al. [73], but have come into focus since the discovery of the CVP. Bastankhah and Porté-Agel [12], for instance, uses wind tunnel observations to develop a means of incorporating the deflection of the streamwise wake into the Gaussian wake model. Shapiro et al. [128] instead models a yawed turbine as a lifting surface and uses Prandtl's lifting line theory to predict the lateral velocity distribution and the magnitude of the CVP. Martínez-Tossas et al. [96] models the CVP directly using Lamb-Oseen vortices and combines this with the rotation due to blade rotation to derive a solution for the streamwise and lateral wakes from the linearised Navier-Stokes momentum equations. These latter two methods have the benefit of not having to assume a wake shape.

Nevertheless, secondary wake steering is a phenomenon of inter-turbine interaction, thus

requires a wind farm scale model. Martínez-Tossas et al. [97] addresses this by developing the model from Martínez-Tossas et al. [96] to a wind farm flow model, and, most recently, Howland et al. [64] used the superposition model of Zong and Porté-Agel [150] described in Section 1.3 to formulate a model for secondary wake steering and evaluated its impact using LES. As was also addressed in Section 1.3 though, the method of Zong and Porté-Agel [150] suffers from not having an explicit expression for the velocity, instead relying on a computationally expensive iterative solution. Bastankhah et al. [14] solved this problem for the streamwise wake velocity, developing an explicit analytical solution for the cumulative wake within wind farms. However, unlike Zong and Porté-Agel [150], Bastankhah et al. [14] does not propose a solution for the lateral wake velocity. There is therefore a need for an explicit solution for the lateral wake velocity that can model wind farm flows including phenomena such as secondary wake steering, which this thesis seeks to address.

There is an important caveat to be made when discussing the applicability of wake steering. The implications for loading have been studied using models and experiment, but only recently has the scope been extended to large wind farms from two- or three-turbine studies. The overall conclusions are that as wake steering is typically applied at wind speeds below rated, it has little effect on ultimate loads, but it does impact fatigue loading [127]. Other studies have investigated the effects on both the upstream and downstream turbines [9, 30, 34, 40, 75, 126, 138, 147], and the partial wakes, which, when wake steering is used for power maximisation, can be made more likely [59]. In practice therefore, the implementation of wake steering as a wind farm control method requires the balancing of power maximisation with the management of loading [46].

1.5 Objectives and research questions

The main research objective of this thesis can be stated as being

> "To improve the physical grounding and prediction accuracy of cumulative wake effect modelling within wind farms with yawed wind turbines, by means of an analytical solution for the lateral velocity field within a wind farm of arbitrary layout and arbitrarily yawed wind turbines that compares well with CFD".

The realisation of this is reliant on the ability to answer the following research questions:

RQ1.1 What is the influence of yaw angle on the lateral wake of a single turbine?

RQ1.2 What is the influence of thrust coefficient on the lateral wake of a single turbine?

RQ1.3 What is the influence of turbulence intensity on the lateral wake of a single turbine?

RQ1.4 Under which conditions is the normalised lateral wake velocity profile self-similar, and which function is most suitable to describe the self-similar profile?

RQ2 How can the method used to derive an analytical solution for the streamwise velocity in Bastankhah et al. [14] be applied for the lateral velocity of yawed turbines?

RQ3 How do the predictions of the resulting analytical model compare to RANS simulations from PyWakeEllipSys?

The work is structured as follows: Chapter 2 aims to explain the relevant aspects of the setup used for the computational fluid dynamics (CFD) simulations examined. Chapter 3 then develops the concept of the lateral wake introduced in Section 1.4 through a parametric study looking at the effects of yaw angle, thrust coefficient, and turbulence intensity. Based on the conclusions from Chapter 3, Chapter 4 derives the analytical solution for the lateral velocity assuming a Gaussian lateral velocity surplus profile. In Chapter 5, there is a brief explanation of how this analytical model is implemented, including the calculation of wake deflections, and subsequently this implementation is used to make predictions of the velocities within several different wind farm layouts in Chapter 6. Chapter 7 presents conclusions from the work in a form intended to answer the research questions above and sets out some items for further research.

CHAPTER 2
CFD Simulation Set-up

One means by which to assess the performance of simpler analytical models, such as that to be developed in this project, is to compare them to the higher fidelity CFD models mentioned in Section 1.2. These CFD models are ubiquitous within academia, with LES being the modus operandi, see for example Kheirabadi and Nagamune [77, Table 1]. RANS models are preferred within industry as they are easier to set up and typically more robust, but their speed relative to LES also makes them popular within some areas of academia. It also makes them very suitable for master's theses, where the time available to acclimatise to the model is limited. As such, this project will utilise the in-house finite-volume solver EllipSys3D of the Technical University of Denmark (Danmarks Tekniske Universitet, DTU), initially developed by Michelsen [101, 102] and Sørensen [134]. The FORTRAN solver code will be interacted with through the Python-based `PyWakeEllipSys` interface.

This section seeks to give an overview of the setup of the CFD simulations in PyWakeEllipSys. Firstly, Section 2.1 introduces the governing equations, this is followed by the turbulence model in Section 2.2 and treatment of the inflow in Section 2.3, before Section 2.4 briefly discusses the flow solver, with details specific to wind energy CFD such as the flow domain and boundary conditions in Section 2.5, and the rotor model in Section 2.6.

2.1 Governing equations

The basic equations of turbulent flows, the incompressible Navier-Stokes equations, consist of the continuity equation (conservation of mass)

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{2.1}$$

where Einstein's summation convention is used and u_i represents in 3D the three velocity components in the three directions x_i: streamwise x, lateral y, and vertical z; and the momentum equations

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = \frac{\partial \sigma_{ij}}{\partial x_j} + \frac{1}{\rho} f_i, \quad \text{where} \quad \sigma_{ij} = -\frac{1}{\rho} p \delta_{ij} + \nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{2.2}$$

ρ is the fluid density, p the pressure, δ_{ij} the Kronecker delta function, $\nu = \mu/\rho$ the kinematic viscosity of the fluid, and f_i are external body forces per unit volume, which for wind farm applications could represent wind turbine, Coriolis, or buoyancy forces. Throughout this thesis the flow is assumed to have constant density and constant viscosity and whilst Equation 2.2 is given in differential, non-conservative form, other forms will be used where appropriate.

Through Reynolds decomposition, pressure and velocities can be separated into mean and fluctuating components and the RANS equations derived from Equation 2.2. The RANS continuity equation is simply

$$\frac{\partial U_i}{\partial x_i} = 0, \tag{2.3}$$

and the RANS momentum equations are

$$\frac{\partial U_i}{\partial t} + \frac{\partial U_i U_j}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\bar{\sigma}_{ij} - \overline{u_i' u_j'}\right) + \frac{1}{\rho} F_i, \quad \text{where} \quad \bar{\sigma}_{ij} = -\frac{1}{\rho} P \delta_{ij} + \nu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \quad (2.4)$$

where F_i are the Reynolds-averaged external body forces per unit volume and the term $-\overline{u_i' u_j'}$ represents the Reynolds stress tensor. These Reynolds stresses pose a problem known as the turbulence closure problem. The RANS equations have a total of ten unknowns (three mean velocity components U_i, mean pressure P, and six independent components of the Reynolds stress tensor $-\overline{u_i' u_j'}$), but only four equations with which to solve for them (the time-averaged continuity equation, Equation 2.3, and the three RANS momentum equations in Equation 2.4). To overcome the closure problem, a turbulence model is often employed that assumes a relationship between the gradient of the mean flows and the Reynolds stresses of the form

$$-\overline{u_i' u_j'} = \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \frac{2}{3} k \delta_{ij}, \quad (2.5)$$

where ν_t is the (kinematic) turbulent eddy viscosity, and k is the turbulent kinetic energy. This relationship is known as the Boussinesq hypothesis for constant density flows.

2.2 The k-ε-f_P turbulence model

Substituting Equation 2.5 into Equation 2.4 and simplifying using Equation 2.3 yields

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left((\nu + \nu_t)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \frac{2}{3} k \delta_{ij}\right) + \frac{1}{\rho} F_i, \quad (2.6)$$

where the six unknowns in the Reynolds stress tensor have now been replaced by two new unknowns, ν_t and k. Lots of turbulence models exist based on this equation, the most well-known being the k-ε eddy-viscosity model (EVM), which uses a constant eddy-viscosity coefficient C_μ to solve for the turbulent eddy viscosity ν_t as

$$\nu_t = C_\mu \frac{k^2}{\varepsilon}. \quad (2.7)$$

The turbulent kinetic energy, k, and turbulent dissipation, ε, are then modelled through two transport equations

$$\begin{aligned} \frac{Dk}{Dt} &= \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\nabla k\right] + \mathcal{P} - \varepsilon, \\ \frac{D\varepsilon}{Dt} &= \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\nabla \varepsilon\right] + (C_{\varepsilon,1}\mathcal{P} - C_{\varepsilon,2}\varepsilon)\frac{\varepsilon}{k}, \end{aligned} \quad (2.8)$$

where ν is the kinematic molecular viscosity, $\mathcal{P}$ is the turbulent production, and $C_{\varepsilon,1}$, $C_{\varepsilon,2}$, σ_k, and σ_ε are empirical constants. The production term can be derived to be

$$\mathcal{P} = -\overline{u_i' u_j'} \frac{\partial U_i}{\partial x_j}, \quad (2.9)$$

where the Reynolds stress tensor is again modelled using Equation 2.5 [81, (12.46)]. The system of equations is now closed, with 6 equations (2.3, 2.6, and 2.8) for 6 unknowns (U, V, W, P, k, and ε).

The standard k-ε EVM, however, is known to be overly diffusive for flows with high shear, such as in the shear layer of the near wake of a wind turbine, when compared to LES. Amongst other things, this leads to a faster wake recovery than expected, and, by extension, an overprediction of power production in wind farms. This trait of the k-ε EVM has been attributed to its use of a constant eddy-viscosity coefficient, C_μ, and can therefore be addressed by modifying C_μ, via the introduction of a function f_P, such that it instead becomes the flow-dependent parameter, C_μ^*. This is the essence of the k-ε-f_P EVM [84, 3].

In the k-ε-f_P EVM, the eddy-viscosity coefficient is defined as

$$C_\mu^* = C_\mu f_P, \tag{2.10}$$

where f_P is a scalar function of local variables that models the effect of non-equilibrium flow conditions and is written

$$f_P\left(\frac{\sigma}{\tilde{\sigma}}\right) = \frac{2f_0}{1+\sqrt{1+4f_0(f_0-1)\left(\frac{\sigma}{\tilde{\sigma}}\right)^2}}, \quad \text{where} \quad f_0 = \frac{C_R}{C_R-1}, \tag{2.11}$$

with C_R as the Rotta constant, which can be calibrated, as will be explained in Section 2.3. This definition also includes the dimensionless shear parameter,

$$\sigma \equiv \frac{k}{\varepsilon}\sqrt{(U_{i,j})^2}, \tag{2.12}$$

which is used to quantify how much the local flow deviates from the log-law regime of a simple shear flow, for which the k-ε-f_P is calibrated. In the calibration flow, the shear parameter σ is equal to $\tilde{\sigma}$, i.e. $\tilde{\sigma} = \frac{k}{\varepsilon}\left\|\frac{\partial U}{\partial z}\right\| = 1/\sqrt{C_\mu}$, using the log-law solution of the k-ε EVM [120]. Hence, f_P is also a function of C_μ, i.e., $f_P(C_\mu)$ [84]. Note that when $\sigma = \tilde{\sigma}$, $f_P = 1$, that is to say, the flow is in equilibrium. As per Equation 2.10, when $f_P < 1$, $C_\mu^* < C_\mu$, and so the model is less dissipative than the standard k-ε EVM. Physically therefore, f_P acts as a limiter on the turbulence length scale [82].

2.3 Inflow modelling

Inflow modelling is very closely tied to turbulence modelling, as in practice the turbulence modelling determines the inflow profile. This section will explain that relationship in the context of the implementation of an adiabatic log-law inflow in Section 2.3.1, before reducing the number of relevant variables via Reynolds number similarity in Section 2.3.2 to just the turbulence intensity, which is then addressed in Section 2.3.3.

2.3.1 Logarithmic wind profile

Wind turbines typically operate within what is called the atmospheric surface layer (ASL), which represents the bottom 10% of the atmospheric boundary layer (ABL). The presence of the ground and the no-slip condition imply that the wind velocity must be zero at $z = 0$. The variation of the wind velocity with height within the ASL, known as the vertical wind shear, can be derived using dimensional analysis assuming constant shear stress, τ, which yields an analytical formula for the velocity profile within the ASL,

$$U(z) = \frac{u_*}{\kappa}\ln\left(\frac{z+z_0}{z_0}\right) \tag{2.13}$$

where $u_* = \sqrt{\tau/\rho}$, with $\kappa = 0.4$ the von Kármán constant and z_0 the roughness length, a function of the terrain. Note that in some applications - such as PyWakeEllipSys - z is in the numerator of the argument of the logarithm, which implies $U(z = z_0) = 0$, instead of $U(z = 0) = 0$.

Equation 2.13 is known as the adiabatic log law, as it does not account for thermal effects. It is therefore only suitable for the case of a 'neutrally stable' ABL, where mechanical and shear effects dominate over thermal effects in the production / suppression of turbulence. The stability of the ABL, which describes the relative magnitudes of these effects can, and does, vary significantly between stable, neutral, and unstable due to variations in temperature gradients, and this affects not only the height of the ABL, but also the velocity profile. Surface layer stability effects can be accounted for within the analytical description of the velocity profile via the introduction of a stability parameter, such as that described by Monin-Obukhov Similarity Theory (MOST) [104], which then gives the diabatic log law as

$$U(z) = \frac{u_*}{\kappa}\left[\ln\left(\frac{z + z_0}{z_0}\right) - \Psi_m\right] \tag{2.14}$$

where Ψ_m is the stability parameter [83]. The evaluation of a new analytical model in this thesis will be limited to neutral conditions and thus Equation 2.13, but could easily be extended to non-neutral stabilities using Equation 2.14.

For CFD simulations it is important that the inflow profile at the inlet boundary is in equilibrium with the turbulence model, i.e. the inflow profile is a solution to the governing equations. In the case of CFD for wind energy, this ensures that there is no streamwise development of the profile before it encounters the turbine, and that characteristics of the flow such as the hub height velocity, $U_{\infty,h}$, and turbulence intensity, $I_{\infty,h}$, are the same at the inflow and the turbine, regardless of where the turbine is located.

The neutral log law is in equilibrium with the k-ε EVM provided

$$k = \frac{u_*^2}{\sqrt{C_\mu}}, \tag{2.15}$$

$$\varepsilon = \frac{C_\mu^{\frac{3}{4}} k^{\frac{3}{2}}}{\kappa(z + z_0)} = \frac{u_*^3}{\kappa(z + z_0)}, \tag{2.16}$$

and

$$\sqrt{C_\mu}\sigma_\varepsilon(C_{\varepsilon,1} - C_{\varepsilon,2}) + \kappa^2 = 0, \tag{2.17}$$

resulting in the well known expression for the eddy viscosity of the neutral atmospheric boundary layer

$$\nu_t = \kappa u_*(z + z_0). \tag{2.18}$$

Moreover, for steady, incompressible, two-dimensional flow modelling of the ASL with the k-ε EVM it can be shown, by inserting Equation 2.13 and Equations 2.15-2.17 into Equations 2.3, 2.6, and 2.8, that $\frac{Dk}{Dt} = \frac{D\varepsilon}{Dt} = \frac{Du}{Dt} = \frac{Dw}{Dt} = 0$, i.e. the flow is equilibrium.

2.3.2 Reynolds number similarity

As has been mentioned in Section 1.2, the two most important factors for determining the wake profile are the thrust coefficient, C_T, and the turbulence intensity, I_∞. The notable absence

here is the freestream velocity, U_∞. This is because, due to Reynolds number similarity, the Navier-Stokes momentum equations in normalised form are independent of the characteristics scales of length and velocity [86].

Expressed in normalised form, the incompressible RANS momentum equations are

$$\frac{\mathcal{L}}{\mathcal{U}\mathcal{T}}\frac{\partial U_i'}{\partial t'} + U_j'\frac{\partial U_i'}{\partial x_j'} = -\frac{\partial P^{*\prime}}{\partial x_i'} + \frac{\partial}{\partial x_j'}\left(\left(\frac{1}{\mathrm{Re}} + \nu_t'\right)\left(\frac{\partial U_i'}{\partial x_j'} + \frac{\partial U_j'}{\partial x_i'}\right)\right) + F_i', \tag{2.19}$$

where the non-dimensional terms are defined as

$$U_i' \equiv \frac{U_i}{\mathcal{U}}, \quad P^{*\prime} \equiv \frac{P^*}{\rho\mathcal{U}^2}, \quad \nu_t' \equiv \frac{\nu_t}{\mathcal{U}\mathcal{L}}, \quad F_i' \equiv \frac{F_i\mathcal{L}}{\rho\mathcal{U}^2}, \quad x_i' \equiv \frac{x_i}{\mathcal{L}}, \quad t' \equiv \frac{t}{\mathcal{T}}. \tag{2.20}$$

The interested reader is directed to Appendix A for the complete derivation.

Under the following four conditions, Equation 2.19 is independent of both the characteristic velocity, $\mathcal{U}$, and length, $\mathcal{L}$, scales:

1. Re is sufficiently large,
$$\mathrm{Re} \to \infty, \tag{2.21}$$
2. all external forces scale with $\rho\mathcal{U}^2/\mathcal{L}$
$$F_i \propto \frac{\rho\mathcal{U}^2}{\mathcal{L}}, \tag{2.22}$$
3. the characteristic time scale, $\mathcal{T}$, of the flow is equal to $\mathcal{L}/\mathcal{U}$
$$\mathcal{T} = \frac{\mathcal{L}}{\mathcal{U}}, \tag{2.23}$$
4. and the eddy viscosity, ν_t, scales with $\mathcal{U}\mathcal{L}$
$$\nu_t \propto \mathcal{U}\mathcal{L}. \tag{2.24}$$

For utility-scale wind turbines operating in atmospheric conditions, the flows are characterised by high Reynolds numbers, $\mathrm{Re} \sim \mathcal{O}(10^6)$ up to $\mathcal{O}(10^8)$ - based on the turbine diameter. Moreover, modelled wind turbine forces, for example those of the actuator disc as described in Section 2.6, do not typically include a viscous component, and, for the RANS simulations in this thesis, the unsteady term can be removed, thus the characteristic time scale is irrelevant. Finally, as is shown in Appendix A, the k-ε-f_P is consistent with the last condition.

As a result, all simulations in this thesis are performed independent of the inflow velocity, U_h, and the rotor diameter, D. This is because all results are expressed in normalised form, and, as has been shown, are therefore only dependent on parameters such as C_T and turbulence intensity.

2.3.3 Turbulence intensity

Given the non-dimensional results have been shown to be independent of the freestream velocity, and the thrust coefficient is a feature of the turbine, the variable of interest for inflow modelling is the turbulence intensity, I_∞. When using the k-ε or k-ε-f_P EVM, the (total) turbulence

intensity can be determined by first substituting Equation 2.15 into the definition of I_∞, Equation 1.4, to give

$$I_\infty = \frac{\sqrt{\frac{2}{3}k}}{|\boldsymbol{U}(z)|} = \frac{\sqrt{\frac{2}{3}}u_*/C_\mu^{1/4}}{|\boldsymbol{U}(z)|}. \tag{2.25}$$

Rearranging Equation 2.13 for

$$u_* = \frac{\kappa U(z)}{\ln\left(\frac{z+z_0}{z_0}\right)}, \tag{2.26}$$

and substituting it into Equation 2.25 then yields

$$I_\infty = \frac{\sqrt{\frac{2}{3}}\kappa U(z)}{|\boldsymbol{U}(z)|C_\mu^{1/4}\ln\left(\frac{z+z_0}{z_0}\right)}. \tag{2.27}$$

One can then assume that $|\boldsymbol{U}(z)| = U(z)$, i.e. $V = W = 0$, which is true for much of the work in this thesis, to find an expression,

$$I_\infty = \frac{\sqrt{\frac{2}{3}}\kappa}{C_\mu^{1/4}\ln\left(\frac{z+z_0}{z_0}\right)}, \tag{2.28}$$

or, at hub height,

$$I_{\infty,h} = \frac{\sqrt{\frac{2}{3}}\kappa}{C_\mu^{1/4}\ln\left(\frac{z_h+z_0}{z_0}\right)}, \tag{2.29}$$

where for a fixed z_h, $I_{\infty,h}$ is only a function of the eddy viscosity coefficient, C_μ, and the roughness length, z_0. To obtain a desired $I_{\infty,h}$, there is therefore a choice between varying C_μ or z_0. For the standard k-ε EVM, it makes little difference, but the behaviour of the k-ε-f_P EVM changes when the constant C_μ is modified because of its presence in f_P, i.e. $f_P(C_\mu)$. Lower values of C_μ correspond, via Equation 2.29, to higher turbulence intensities, which are associated with greater mixing and thus increased wake recovery. Laan et al. [87] showed, however, that f_P reduces wake recovery for lower values of C_μ, which is unphysical. Therefore, for the k-ε-f_P EVM the preferred choice for determining $I_{\infty,h}$ is to vary z_0, not C_μ.

It is also worth noting that whilst great care has been taken to ensure that there is streamwise development of the inflow profile, due to numerical errors near the rough wall boundary condition this can still occur when using an analytical log-law inflow. Typically these errors are on the order of 0.5% in wind speed at hub height over 50 km when using a large first cell height of 0.5 m, however, this can be important for studies examining small effects, for example wind farm blockage, where errors can be magnified in power production due to its cubic dependence on wind speed. In PyWakeEllipSys the proposed solution to this issue is the use of a numerical solution from a one-dimensional precursor simulation (EllipSys1D), based on the vertical wind farm grid.

The k-ε-f_P EVM includes seven constants as summarised in Table 2.1, five from the standard k-ε EVM, namely $C_{\varepsilon,1}$, $C_{\varepsilon,2}$, σ_k, σ_ε, and κ; and additionally a constant $C_\mu = 0.03$, chosen to describe a neutral atmospheric boundary layer (ABL), and the Rotta constant, C_R, calibrated against eight LES cases of a single wind turbine [84].

Baungaard [15, §2.4.7] has verified profiles of all the flow variables defined in this section on inflow modelling and addressed outliers such as the deviation of k near the rough wall.

Table 2.1: k-ε-f_P eddy-viscosity model constants [84].

C_R	C_μ	$C_{\varepsilon,1}$	$C_{\varepsilon,2}$	σ_k	σ_ε	κ
4.5	0.03	1.21	1.92	1.00	1.30	0.40

2.4 Solver

The RANS model in PyWakeEllipSys uses the general purpose flow solver EllipSys3D, developed by Sørensen [134] based upon the Basis2D/3D platform of Michelsen [101, 102]. The following section aims to highlight some of the most relevant features of the EllipSys3D solver, such as the finite-volume discretisation of the RANS equations and its use of the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm for their solution; its interpolation schemes, including Rhie-Chow interpolation; and finally the wind-energy CFD features specific to the PyWakeEllipSys RANS-based wind farm model, such as the flow domain and boundary conditions and rotor model. There is of course much to be said about both EllipSys and PyWakeEllipSys, and the reader is referred to the original PhD thesis of Sørensen [134] and the PyWakeEllipSys documentation [39], respectively, for more details.

2.4.1 Discretisation of the governing equations

In order to solve the continuous RANS equations presented in Equation 2.1 and Equation 2.2, numerical flow solvers such as EllipSys3D must discretise the equations. EllipSys3D uses a finite-volume discretisation approach, which divides the flow domain up into a mesh of non-overlapping cells as in Figure 2.2 and then applies the discretised form of the governing equations to each cell.

This discretised form can be derived starting with Equation 2.2. Equation 2.2, however, expresses the RANS momentum equations in non-conservative form, and whilst for theoretical aerodynamics the form of the equations makes little difference, this does not hold for CFD applications. The principal advantage of the conservative form here is that its use allows flux terms in the discretised solution to "telescope", i.e. summing over all the terms, the intercell fluxes cancel and so the solution depends only on the boundary conditions.

In conservative form then, the RANS momentum equations for incompressible flows are

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial(\rho U_i)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[(\mu+\mu_t)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] - \frac{2}{3}\frac{\partial(\rho k)}{\partial x_i} + F_i. \tag{2.30}$$

This differential form must then be integrated over the control (cell) volume, V, such that

$$\iiint_V \frac{\partial U_i}{\partial t}\,\mathrm{d}V + \iiint_V U_j \frac{\partial(\rho U_i)}{\partial x_j}\,\mathrm{d}V = \iiint_V \left(-\frac{\partial P}{\partial x_i} - \frac{2}{3}\frac{\partial(\rho k)}{\partial x_i} + F_i\right)\mathrm{d}V + \iiint_V \frac{\partial}{\partial x_j}\left[(\mu+\mu_t)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right]\mathrm{d}V. \tag{2.31}$$

The conservative form is also sometimes known as the divergence form due to the expression of the advective and viscous terms as a divergence of a quantity. This feature can be exploited through the use of the divergence theorem, which states that the flux of a vector field, for

example (ρU_i), through a closed surface of area, A, and with outward-pointing normal vector, n_i, must be equal to the divergence of the field within an enclosed volume, i.e.

$$\iiint_V \frac{\partial(\rho U_i)}{\partial x_j}\,\mathrm{d}V = \oiint_A (\rho U_i) n_i\,\mathrm{d}A, \tag{2.32}$$

Applying this to the advective and viscous terms in Equation 2.31 yields

$$\iiint_V \frac{\partial U_i}{\partial t}\,\mathrm{d}V + \oiint_A U_j(\rho U_i) n_i\,\mathrm{d}A = \iiint_V \left(-\frac{\partial P}{\partial x_i} - \frac{2}{3}\frac{\partial(\rho k)}{\partial x_i} + F_i\right)\mathrm{d}V + \oiint_A \left[(\mu+\mu_t)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] n_i\,\mathrm{d}A. \tag{2.33}$$

For a cell with volume V and Ψ faces, each with area A^ψ and normal vector n_i^ψ, the surface integrals can be expressed as finite sums and the momentum equations can therefore be discretised as

$$\frac{\partial U_i^\psi}{\partial t}V + \sum_{\psi=1}^{\Psi} U_j^\psi (\rho U_i)^\psi n_i^\psi A^\psi = \left(-\frac{\partial P}{\partial x_i} - \frac{2}{3}\frac{\partial(\rho k)}{\partial x_i} + F_i\right)V + \sum_{\psi=1}^{\Psi}(\mu+\mu_t^\psi)\left(\frac{\partial U_i^\psi}{\partial x_j} + \frac{\partial U_j^\psi}{\partial x_i}\right) n_i^\psi A^\psi, \tag{2.34}$$

where $(\cdot)^\psi$ indicates a value specific to face ψ.

2.4.2 The SIMPLE algorithm

The incompressible Reynolds-averaged Navier-Stokes equations as shown in Equations 2.3 and 2.4 provide an equation for U, V, and W, but not explicitly for P. Moreover, the continuity equation, rather than being a means of solving for the velocity fields, functions as a constraint on the solution. The velocity fields must together satisfy the continuity equation. An equation for pressure is therefore required if one is to solve the complete RANS equations.

One might propose the use of an equation of state such as the ideal gas law to compute the pressure, but for constant-density flows such as in this thesis this is not applicable. However, by combining the momentum and continuity equations, it is possible to derive a Poisson equation for pressure, and this forms the first of the two tenets of the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm. The second then being a corrector for the velocity field, that ensures the solution satisfies the continuity equation.

The solution process is then as follows:

1. Solve the momentum equations (Equation 2.4) for the velocity field, U, V, and W. The resultant velocity field does not satisfy the continuity equation.
2. Solve the Poisson equation for the pressure field, P.
3. Use the pressure field to correct the velocity field such that it now satisfies the continuity equation.

4. Solve other equations such as the energy, turbulence scalar (e.g. k, ε), and species transport equations.
5. The resultant velocity field now does not satisfy the momentum equations, thus the process is repeated to convergence.

2.4.3 Rhie-Chow interpolation

In a segregated solver such as EllipSys3D, where each variable is solved for across the domain sequentially, the pressure field can exhibit odd/even pressure decoupling. These so-called 'chequerboard oscillations' are a consequence of collocated storage of velocity and pressure fields and linear interpolation of the velocity (momentum flux) in the formulation of the pressure equation. As can be seen in Equation 2.34, the finite-volume discretised RANS equations require the value at the face centres, and in a collocated system these must be found by interpolating between the cell centroids. Many interpolation schemes exist, Jasak [71] gives a good overview, however, careful consideration must be given to the choice of scheme, as some can lead to chequerboard oscillations, where two independent solutions for pressure can exist within the same domain and present, in 2D, as a chequerboard pattern.

One way to mitigate this is to use a staggered grid approach, where velocity is stored and calculated at the cell faces, rather than the centroids, and thus the pressure gradient is calculated from the adjacent centroids, which avoids the pressure-decoupling [58]. However, the use of staggered grids restricts the user to the use of only Cartesian or polar grids, without any skewed cells, and thus limits the choice of geometries [42].

Rhie & Chow interpolation, originally formulated by Rhie and Chow [119] and again by Bartholomew et al. [8] - with clearer notation - circumvents the issue of pressure-decoupling for a collocated grid. It does so by separating the pressure term from the rest of the momentum terms in the derivation of face velocities. A correction is introduced for the pressure term to account for the error introduced by interpolating from the already interpolated pressure gradients at the adjacent cell centroids. The YouTube series of Wimshurst [144] provides an excellent explanation of Rhie-Chow interpolation.

The original formulation by Rhie and Chow [119] does not include discrete body forces, such as those from an actuator disc for example. Therefore, these required the introduction of additional terms. The 'pressure jump' approach developed by Réthoré and Sørensen [117] and Réthoré et al. [118] treats the body forces in a similar way to the pressure term, separating them from the rest of the momentum terms and transforming them to pressure jumps located at each cell face. In practice then, the pressure jump corresponding to the body force is 'smeared' over the faces of the nearest neighbouring cells [117]. The pressure jumps allocated to each face also depend on the position within the cell where the force acts, so as to avoid issues where the original body force does not act at the cell centroid [118].

2.4.4 Convergence and the grid sequence approach

As mentioned in Section 2.4.2, the SIMPLE algorithm is an iterative process. With each loop of the algorithm, the solution approaches a converged solution, defined as one where successive iterations produce no change. However, the path to convergence is asymptotic, and thus without bounds would run forever. The typical method in CFD is to compromise between speed and accuracy of the final solution by setting a limit on the solution, characterised by residuals.

Following the finite-volume method, each of: (i) Equation 2.34, (ii) the Poisson equation for pressure, and the transport equations for (iii) k and (iv) ε, ends up as a linear, decoupled, and algebraic equation with the following generic form for each cell centroid P,

$$A_P \phi_P + \sum_{\psi=1}^{\Psi} A_{nb} \phi_{nb} = S, \tag{2.35}$$

where $(\cdot)_{nb}$ indicates the neighbour cell and S represents the collected source terms. These equations can be expressed as a linear system $\underline{\underline{A}}\boldsymbol{\phi} = \boldsymbol{S}$, which can be solved iteratively using linear algebra. The residual at iteration n is defined as

$$\boldsymbol{\rho}^n \equiv \boldsymbol{S} - \underline{\underline{A}}\boldsymbol{\phi}^n. \tag{2.36}$$

In EllipSys3D, these residuals are expressed as a single scalar for each flow variable, normalised by the initial solution. Convergence is determined by the value of the residual, for which a limit is set based on the desired speed and accuracy of the solution. When this convergence criterion is reached, the simulation is stopped.

Figure 2.1 shows how the residuals for each variable develop over the course of a typical simulation. Evidently however, the first time that the residuals reach the limit here of $\rho^n_{\{U,V,W,P,k,\varepsilon\}} = 1 \times 10^{-5}$, the simulation continues. This is due to EllipSys3D's use of a grid sequence approach to accelerate convergence. In this method the initially defined 'finest' mesh resolution is downsampled to produce a coarser mesh which is solved first using the same boundary and initial conditions. This is the behaviour seen in roughly the first 200 iterations of Figure 2.1. The converged solution from the coarser mesh is then interpolated on to the finer mesh and solved again. As the solution from the coarser mesh already qualitatively includes much of the flow physics expected in the final solution but requires significantly less computational time, the overall method can significantly reduce total computational time - up to 50% in 3D using a three-level grid sequence [134].

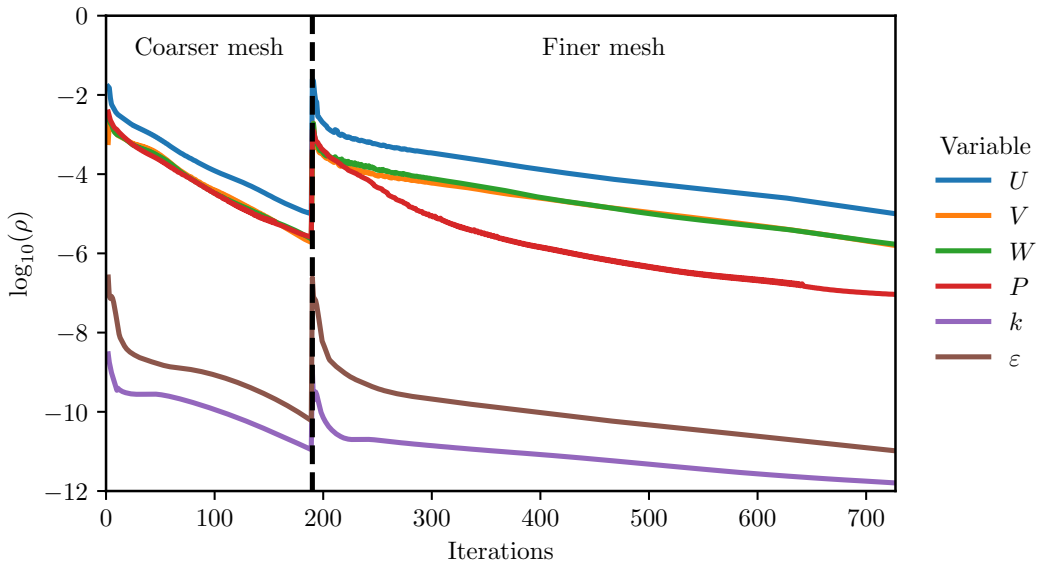


Figure 2.1: Example of variation in residuals of each flow variable with iteration number over the course of a PyWakeEllipSys simulation.

2.5 Flow domain and boundary conditions

As has been alluded to, EllipSys3D solves the RANS equations upon a numerical grid. Of the four available grid types [39], the configuration used in this thesis is the so-called 'flatbox', a Cartesian grid with a uniform roughness and elevation suited to flat terrain wind farm simulations such as offshore wind farms.

Figure 2.2 shows the flow domain and boundary conditions used for the PyWakeEllipSys simulations in this thesis. The wind farm sits within an internal, refined domain, shown as a blue rectangle in Figure 2.2(c), where the wind turbine wakes are resolved. This is then surrounded by a buffer domain, which can be used to resolve wind farm scale effects such as wind farm wakes and is represented by the magenta rectangle in Figure 2.2(c). The entire domain must, however, be large enough as to ensure a fully-developed inflow and avoid numerical blockage effects. It is therefore significantly larger than the internal 'wake domain' and buffer region, but with coarser resolution to reflect the relative lack of flow features.

When a wind farm is simulated in flat terrain and homogeneous roughness, such as is the case in this thesis, it is possible to simulate different wind directions in a single simulation by rotating the wind farm layout, whilst keeping the inflow wind direction constant [39]. As per PyWakeEllipSys convention, the inflow direction is always 270° (i.e. from left to right in Figure 2.2(a). The coordinate system is defined to match this arrangement, with the domain origin always the wind farm centre, calculated as the mean of the wind turbine positions.

The coordinate system is right-handed with x as the streamwise, y the lateral, and z the vertical direction. The dimensions of the outer domain are set by a 'grid radius' parameter,

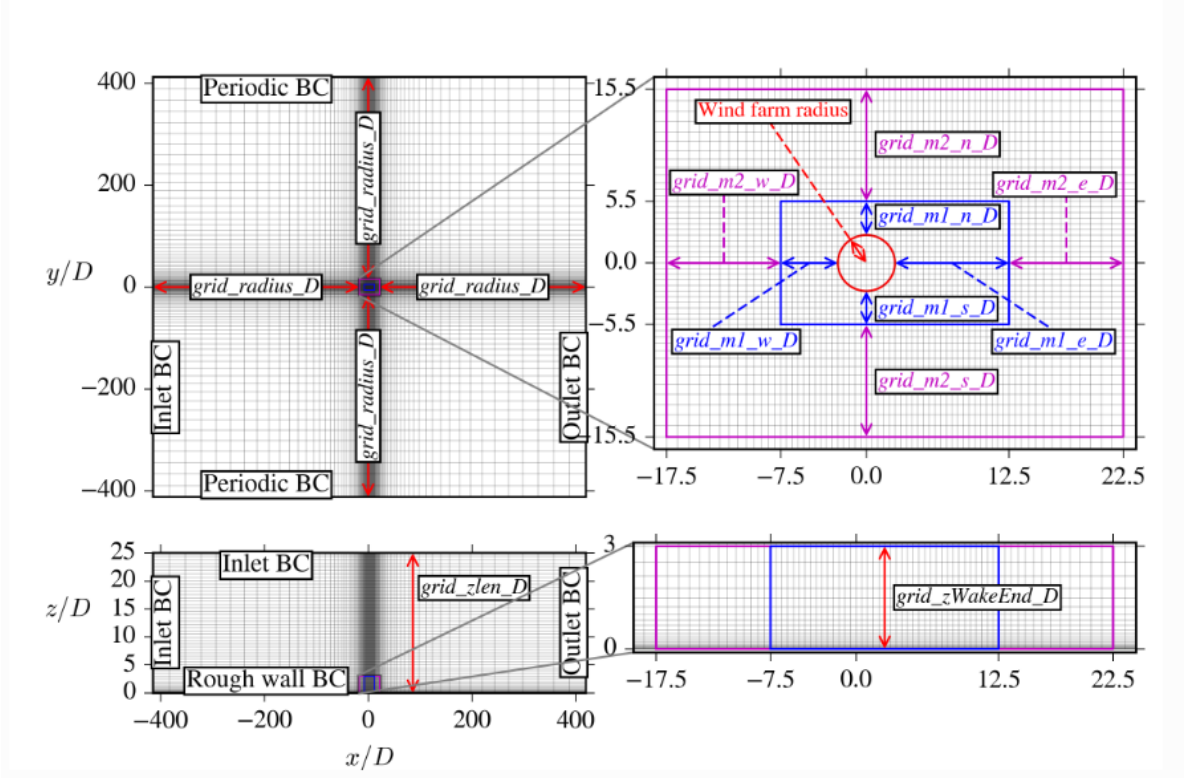


Figure 2.2: Flow domain and boundary conditions for PyWakeEllipSys simulations.

which for this thesis is set at 50 km and the height of the domain is $25D$. This results in a box-shaped domain (x, y, z) of 50 km $\times$ 50 km $\times$ $25D$, containing a wake domain of dimensions $25D \times 3D \times 3D$. Section 2.7 presents a grid convergence study in which the mesh resolution is chosen to be 8 cells per diameter within the wake domain.

Boundary conditions for EllipSys3D are set using an 'attribute' approach, where the user specifies in advance the values of flow variables at the cell faces based on the physical conditions [134]. The boundary conditions used for the simulations in this thesis are shown in Figure 2.2 and summarised in Table 2.2. They are also explained in detail in Sørensen [134, §6] and Sørensen et al. [135].

Each boundary requires the specification of six variables - U, V, W, P, k, ε - to satisfy the RANS equations. Generally speaking, the inflow, top, and bottom BCs are based on the logarithmic law-of-the-wall and the lateral and outflow BCs are designed to be physically consistent with this. The inflow boundary, the left-hand side of Figure 2.2(a), is assumed to have conditions determined by Equations 2.13, 2.15, and 2.16, with zero lateral and vertical velocities. The top boundary, the lid of the computational domain, is also assumed to be satisfied by the same equations, but where z is constant. At the outlet, the flow is assumed to be fully developed, which is implemented here by setting the streamwise derivatives of all flow variables except P equal to zero. This method does not, however, guarantee global mass conservation, and so requires that the velocities are scaled by the ratio of inlet to outlet mass flux. The two lateral boundaries can use either symmetry or periodic conditions. Symmetry implies $V = 0$ and $\frac{\partial U}{\partial y} = \frac{\partial W}{\partial y} = \frac{\partial k}{\partial y} = \frac{\partial \varepsilon}{\partial y} = 0$, but as a result cannot be applied for flows without y-symmetry, e.g. veered inflow or yawed turbines. In these cases periodic boundaries are more suitable, which are defined as

$$\phi(x, y_l, z) = \phi(x, y_r, z) \tag{2.37}$$

where ϕ is any of the six flow variables excluding P, and y_l and y_r are y-coordinates of the left and right lateral boundaries, respectively. The absence of pressure, P from the discussion thus far is due to the absence of an equation explicitly for it. In the general case, pressure at the boundaries is extrapolated from values in interior cells as per Sørensen [134, Eq. 101].

The bottom boundary condition is slightly more complicated than the others. It is based upon the wall shear stress, evaluated at the cell centroid as

$$\tau_w = \frac{\rho \kappa C_\mu^{\frac{1}{4}} k^{\frac{1}{4}} U(z_P)}{\ln(\Delta z + z_0)}, \tag{2.38}$$

where Δz is the distance from the bottom face of the cell to the cell centroid, z_P. The boundary condition for turbulent kinetic energy reduces to a balance between turbulent production and dissipation. This is implemented in EllipSys3D by setting a von Neumann boundary condition on k ($\frac{dk}{dz} = 0$) and replacing the production term in the first cell by the equilibrium value [135]. The turbulent production can be determined by averaging over the wall cell as

$$\mathcal{P} = \frac{\tau_w}{2\Delta z} \frac{\ln(2\Delta z + z_0)}{\ln(\Delta z + z_0)} U(z_P). \tag{2.39}$$

The velocities are then determined as per Sørensen [134, §7.3]. Finally, the ε transport equation given in Equation 2.8 is known to be in error near the wall due to its assumption of high Re. Thus, typically Equation 2.8 is abandoned in the wall cell and instead the dissipation is specified according to the balance between production and dissipation obtained for the fully-developed flow given in Equation 2.16 [135].

Table 2.2: Boundary conditions for PyWakeEllipSys simulations.

Boundary	BC type	U	V	W	P	k	ε
Inflow	Inlet	Eq. 2.13	0	0	[134, Eq. 101]	Eq. 2.15	Eq. 2.16
Top	Inlet	Eq. 2.13	0	0	[134, Eq. 101]	Eq. 2.15	Eq. 2.16
Outflow	Outlet	$\frac{\partial U}{\partial x} = 0$	$\frac{\partial V}{\partial x} = 0$	$\frac{\partial W}{\partial x} = 0$	[134, Eq. 101]	$\frac{\partial k}{\partial x} = 0$	$\frac{\partial \varepsilon}{\partial x} = 0$
Lateral	Periodic	Eq. 2.37	Eq. 2.37	Eq. 2.37	[134, Eq. 101]	Eq. 2.37	Eq. 2.37
Bottom	Rough wall	[134, §7.3]	0	0	[134, Eq. 101]	$\frac{dk}{dz} = 0$	Eq. 2.16

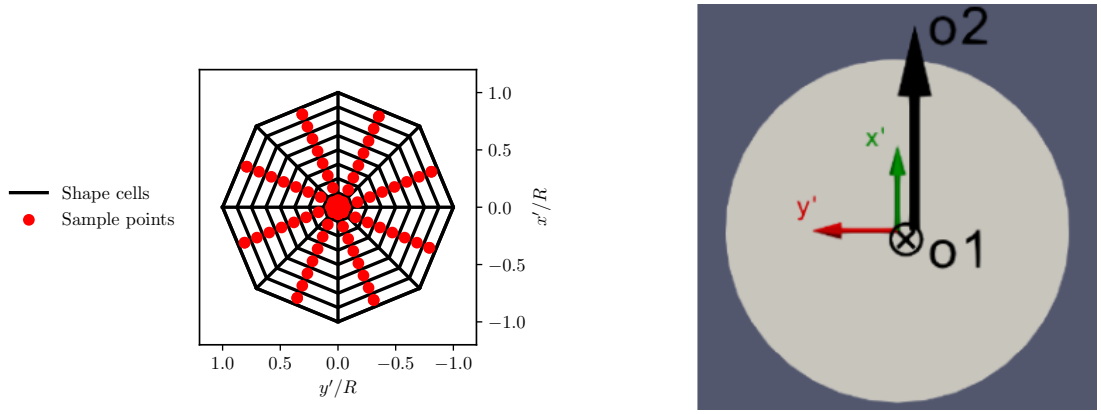
It should be noted that the boundary conditions are slightly different for the precursor simulation. As the final profile is not known a priori, it is not possible to allocate the top boundary in the same fashion as for the full domain. A symmetry condition is therefore used instead, with the height of the domain set on the order of 200 km to diminish the influence of the non-physical boundary condition.

Finally, as far as initial conditions are concerned, the practice is to initialise all interior cells with log-law values from Equations 2.13, 2.15, and 2.16.

2.6 Rotor model

Within the Navier-Stokes equations, the influence of the rotor is implemented as a body force, i.e. the term f_i in Equation 2.2, the idea being that the forces imposed by the rotor impact upon the wind in the same way that the wind impacts upon the rotor, as per Newton's third law of motion. Under the umbrella of this 'permeable body force' implementation, and as discussed in Section 1.2, there are several different ways to model a wind turbine rotor in CFD. Table 1.1 for example mentions the actuator disc (AD), actuator line (AL), and blade resolved approaches. PyWakeEllipSys uses the actuator disc, and therefore this will be the focus hereafter.

2.6.1 PyWakeEllipSys actuator disc definition



(a) Polar AD grid for $n_r = n_\theta = 8$

(b) Orientation vectors $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$, from [39].

Figure 2.3: Polar AD grid and orientation vectors $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$

The actuator disc, as it is implemented in PyWakeEllipSys, is a 2D disc on a polar grid. It has its own independent, polar coordinate system (θ, r), but for consistency with the full domain is also often defined in a Cartesian system (x', y', z') as in Figure 2.3(a). The disc is oriented with the two vectors shown in Figure 2.3(b), $\boldsymbol{O}_1$ indicating the normal direction to the disc surface, and $\boldsymbol{O}_2$ the vertical direction. The z'-axis follows the $\boldsymbol{O}_1$ vector and thus all z'-coordinates are zero for an AD. The polar grid is discretised into a number of 'shape cells', which is determined by the discretisation in the two axes, n_θ and n_r. The optimal values for n_θ and n_r are investigated in Baungaard [15, §2.4.3] and found to be when $n_r = n_\theta = 32$ or 64. This thesis uses either 32 or 64 depending on the flow case, but this is specified at the time.

2.6.2 Disc loading

All permeable body force methods, including the AD, follow the same three-step approach [118]:

1. Blade forces prescribed or calculated from the local flow information.
2. Forces redistributed in the computational domain.
3. Forces applied in the flow solver.

The final step was mentioned towards the end of Section 2.4.3 and is discussed in further detail in Réthoré and Sørensen [117]. The first two steps are addressed by Réthoré et al. [118] through the introduction of the 'actuator shape' (AS) model, which will be elaborated upon in this section.

The AS model begins with determining the blade forces, for which there are three methods within PyWakeEllipSys: fixed force distributions, analytical force distributions based on the Joukowsky rotor, or force distributions from aerofoil data. The fixed force distributions allow the user to specify normalised normal and tangential blade loadings, which are then further normalised before being applied to the AD grid. To use the aerofoil data method, one needs to supply EllipSys3D's BEM model with the aerofoil polars, blade geometry, pitch and rpm curves. Whilst this is an accurate method, this data is not always readily available, and results

are specific to the chosen blade. For this study, a more generalised approach is preferred, and so the force distributions are based on the Joukowsky rotor [80].

The analytical body force specification of Sørensen et al. [133] is an attempt to represent the axial and azimuthal rotor loadings of the typical turbine in a simple way. Assuming constant circulation, and including Prandtl's classical tip correction [17] and a tip correction based on the work of Delery [35], simple analytical expressions are derived for the axial and tangential force distributions which depend only on rotor radius, tip speed ratio, and thrust coefficient, and compare well with results using aerofoil data. The force distributions at each node point, for example the sample points in Figure 2.3(a), are written as

$$\begin{aligned}\Delta F^0_{n,\alpha\beta} &= 4\rho q_0 \frac{gF}{\chi_\alpha}\left(\lambda\chi_\alpha \frac{1}{2} q_0 \frac{g_{\text{tip}} g_{\text{root}}}{\chi_\alpha}\right)\frac{U^2_{\text{AD},\alpha\beta}\Delta A_{\alpha\beta}}{(1+\sqrt{1-C_T})^2},\\ \Delta F^0_{\theta,\alpha\beta} &= 2\rho q_0 \frac{g_{\text{tip}} g_{\text{root}}}{\chi_\alpha}\frac{U_{\text{AD},\alpha\beta}\Delta A_{\alpha\beta}}{1+\sqrt{1-C_T}},\end{aligned} \tag{2.40}$$

Here, the α and β indices represent the radial and azimuthal directions, respectively, χ_α is the normalised radius r_α/R, λ the tip-speed ratio, $\Delta A_{\alpha\beta}$ is the area of the AD element, and $U_{\text{AD},\alpha\beta}$ is the local streamwise velocity. Furthermore, g_{tip} and g_{root} are the tip and root corrections mentioned above, and q_0 is a dimensionless reference circulation defined in Sørensen et al. [133, Eq. 17]. During later implementation of this model in Laan et al. [86], it was found that the total thrust, power, and torque outputs were only consistent with the input C_T and C_P if the normal force distribution, $\Delta F^0_{n,\alpha\beta}$, was scaled by C_T, and the tangential force distribution, $\Delta F^0_{\theta,\alpha\beta}$, by C_P and λ, such that the applied force distributions are actually

$$\begin{aligned}\Delta F_{n,\alpha\beta} &= \frac{\frac{1}{2}\rho C_T A U^2_{\infty,h}\Delta F^0_{n,\alpha\beta}}{\sum_\alpha\left(\sum_\beta\left[\Delta F^0_{n,\alpha\beta}\Delta A_{\alpha\beta}\right]\right)},\\ \Delta F_{\theta,\alpha\beta} &= \frac{\frac{1}{2}\rho C_P A U^2_{\infty,h}\Delta F^0_{\theta,\alpha\beta}}{\lambda\sum_\alpha\left(\sum_\beta\left[\Delta F^0_{\theta,\alpha\beta}\Delta A_{\alpha\beta}\right]\right)},\end{aligned} \tag{2.41}$$

where A is the total AD area.

2.6.3 Force redistribution

The forces distributions specified by Equation 2.41 are implemented on the polar AD grid. In order to be resolved as part of the RANS equations, however, they must be implemented on the Cartesian flow domain grid. There must therefore be a method for redistributing the forces from one grid to the other. The approach taken to this redistribution can have a significant effect on the computational efficiency of the solver, as a poorly implemented redistribution algorithm requires a denser mesh to yield the same accuracy of solution. This redistribution is not a trivial task however. At first glance, there are several potential methods to take the forces from the shape cells to the domain cells in Figure 2.4. This is further complicated considering the disc can be yawed or tilted within the 3D domain and is therefore not always aligned with grid as shown. For the sake of simplicity, we shall only consider here the 2D case shown in the figure.

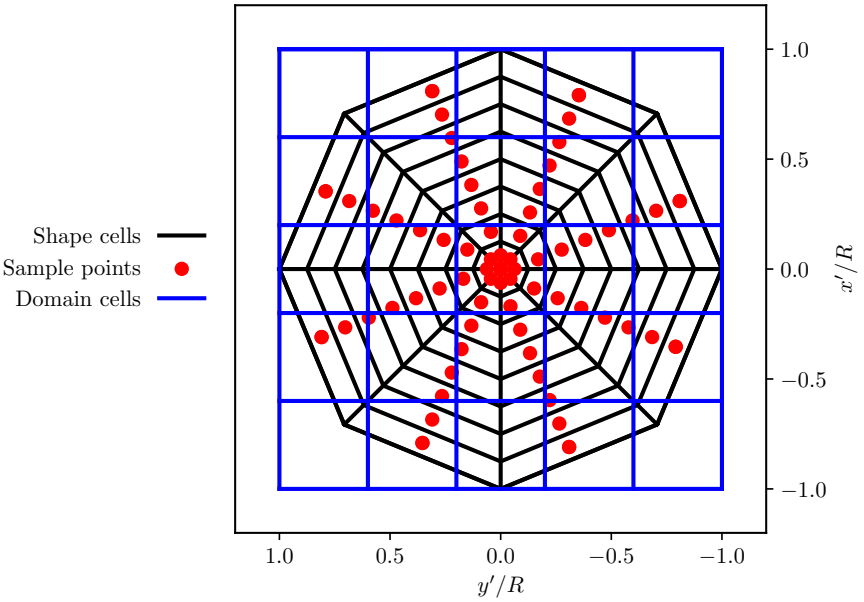


Figure 2.4: Example of a polar AD grid with $n_r = n_\theta = 8$ within a Cartesian flow domain grid with $n_y = n_z = 5$. Note that the domain is in fact 3D, and the disc can take any orientation within it.

Assuming each shape cell (SC), j, has a generic force acting from its cell centroid, cc_j^{SC}, given by $\boldsymbol{F}_j^{\mathrm{SC}}$, the contribution of this cell to the forcing in the domain cell (DC), i, denoted $\boldsymbol{F}_{j,i}^{\mathrm{DC}}$, can be determined by various methods [15, §2.2], for example:

1. the SC can be assigned to a single DC based on the location of its cell centroid, whereby the force contribution is given by

$$\boldsymbol{F}_{j,i}^{\mathrm{DC}} = \begin{cases} \boldsymbol{F}_j^{\mathrm{SC}}, & \text{if } cc_j^{\mathrm{SC}} \in V_i^{\mathrm{DC}}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.42}$$

   where V_i^{DC} is the volume of domain cell i,

2. or, it can be assigned based on the overlap area between the shape and domain cell, such that

$$\boldsymbol{F}_{j,i}^{\mathrm{DC}} = \begin{cases} \boldsymbol{F}_j^{\mathrm{SC}}, & \text{if } \frac{A_{i,j}}{A_j} = \max_i \left(\frac{A_{i,j}}{A_j} \right), \\ 0, & \text{otherwise}, \end{cases} \tag{2.43}$$

   where A_j is the area of shape cell j, and $A_{i,j}$ is the overlap, or intersectional, area, between domain cell i and shape cell j.

3. Lastly, the force from a SC can be distributed over several domain cells, with the contributions weighted by the intersectional area, in this case the force contribution is given by

$$\boldsymbol{F}_{j,i}^{\mathrm{DC}} = \frac{A_{i,j}}{A_j} \boldsymbol{F}_j^{\mathrm{SC}}. \tag{2.44}$$

In each case the total force at domain cell, i, is given by the sum of the contributions from all shape cells as

$$\boldsymbol{F}_i = \sum_{j=1}^{n_{\mathrm{SC}}} \boldsymbol{F}_{j,i}^{\mathrm{DC}}, \tag{2.45}$$

where n_{SC} is the total number of shape cells. The reader is referred to Figures 1 and 3 of Réthoré et al. [118] for better understanding of the concepts of the 'intersectional polygon', the intersectional area between SC and DC, and the 'intersectional grid', the collection of all intersectional polygons.

The final method is the one used by PyWakeEllipSys. Despite the method being presented here only for the case of a 2D disc, it is in fact applicable to any arbitrary surface. Moreover, it has been demonstrated to simulate wind turbine flows accurately with as few as 5 cells per diameter (cD) [118], with no tuning of parameters required, which was an issue with previous methods based on the use of a convolution smearing function pioneered by Mikkelsen [103] [146, 95].

2.6.4 Force control methods

In order to calculate the force distributions given in Equation 2.41, one requires knowledge of the thrust and power coefficients, C_T and C_P, as well as the freestream hub height velocity, $U_{\infty,h}$. C_T and C_P are usually expressed in lookup tables as functions of $U_{\infty,h}$. However, whilst $U_{\infty,h}$ is well-defined for a standalone turbine, for one in the wake of another, $U_{\infty,h}$ is unknown. When multiple ADs are then used to simulate wake interaction in wind farms, the force on each AD is not known a priori. A method must be found to determine the force distributions which allows them to vary as the simulation progresses to convergence. The following section sets out some details of the methods available in PyWakeEllipSys, their advantages and disadvantages, and consequently the method employed in this thesis.

Starting with the assumption that the force distributions are known a priori, one must assume that the incoming velocity is equal to the freestream velocity throughout the simulation. This is to assume that the incoming velocity is affected neither by the wake of the upstream turbine, nor the induction of the turbine in question, both of which reduce the incoming velocity. Employing these assumptions, however, the thrust and power coefficients corresponding to the freestream velocity can be read from the lookup table and used to calculate the force distributions to be used throughout the simulation. This is a fast, simple method, but overpredicts incoming velocity.

More accurate estimations of the freestream velocity as defined for a waked AD can be obtained by one of two methods in PyWakeEllipSys, as set out in Laan et al. [88] and summarised in Baungaard [15, App. A]. The first, and most simple, method is based on 1D momentum theory, where the freestream velocity of a waked AD, $U_{\infty,h}$, can be estimated via the axial induction factor, a, as

$$\begin{aligned} a &= \frac{1}{2}\left(1 - \sqrt{1 - C_T}\right), \\ U_{\infty,h} &= \frac{\langle U \rangle_{\mathrm{AD}}}{1 - a}, \end{aligned} \tag{2.46}$$

where $\langle U \rangle_{\mathrm{AD}}$ is the 'disc velocity', i.e. the velocity averaged over the AD area, which can be extracted from the flow field. Employing this method, an estimate must be made of $U_{\infty,h}$

prior to the first iteration, this then informs a first estimate of the force distributions, before $\langle U \rangle_{\mathrm{AD}}$ is probed from the flow field and used to correct $U_{\infty,h}$ based on Equation 2.46. Despite being more accurate than not applying any force control, whilst not being significantly slower, this method's roots in 1D momentum theory means it is not appropriate for yawed or tilted rotors. Moreover, it generally overpredicts power production and can be numerically unstable for certain flow conditions [88].

An alternative was introduced by Laan et al. [88], known as the AD variable scaling method, to treat the shortcomings of other methods. In a calibration procedure run before the main simulation, new variables, C_T^* and C_P^*, are created to represent C_T and C_P as a function of the disc velocity, instead of the freestream. Lookup tables populated by these variables can then be used to dynamically calculate the force distributions during the simulation from probed values of $\langle U \rangle_{\mathrm{AD}}$. The calibration routine is performed by cycling through wind speeds from cut-in to cut-out and probing $\langle U \rangle_{\mathrm{AD}}$ to fill the lookup table with

$$C_T^* = C_T \left(\frac{U_{\infty,h}}{\langle U \rangle_{\mathrm{AD}}} \right)^2, \quad \text{and} \quad C_P^* = C_P \left(\frac{U_{\infty,h}}{\langle U \rangle_{\mathrm{AD}}} \right)^3. \tag{2.47}$$

This method does not suffer from the numerical instabilities mentioned above, nor does it overpredict power production in the same way [88], thus it is the chosen method for this thesis.

2.7 Mesh convergence study

Inherent to finite element/volume methods is the tradeoff between accuracy and computational effort. This balance is most evident in the choice of cell size. Smaller cells, equivalent to more cells per unit volume, lead to a more accurate solution, but come at greater computational cost. To demonstrate the suitability of the proposed solver and provide context regarding the baseline level of error it is necessary to perform a mesh convergence study.

Given this thesis is concerned with both the streamwise and lateral wake velocities, the mesh convergence study examines the effect of cell size on both with downstream distance. It uses a turbine with a fixed thrust coefficient, $C_T = 0.8$, and an inflow with a fixed turbulence intensity of $I_{\infty,h} = 4\%$. The specific values were chosen to represent the 'worst case' to be examined later on, for instance in Chapter 3. In order to capture the error associated with the other directions, i.e. y and z, the independent variables are disc-averaged, that is to say

$$U_{AD}(x) = \int_A U(x, y, z) \, \mathrm{d}y \, \mathrm{d}z, \tag{2.48}$$

where A here is the area of the rotor disc. As will be shown in Chapter 3, for an unyawed turbine the lateral wake velocity profile has an integral equal to zero, because the turbine does not add any lateral momentum to the flow. This means that V_{AD} is not a useful measure of convergence. For this reason, the absolute value of lateral velocity is used instead, such that

$$|V|_{AD}(x) = \int_A |V(x, y, z)| \, \mathrm{d}y \, \mathrm{d}z. \tag{2.49}$$

In order to quantitatively assess the relative error associated with the different cell sizes, one requires a reference error value. One could for example use the values from the finest grid, however, this is of course not the converged value. Whilst finding the true converged value would require a lot of time and computation, one can make a reasonable estimation by

extrapolating from the computed values. The traditional extrapolation method is Richardson extrapolation using 3 levels of grid refinement [28, 121]. Réthoré et al. [118] highlights the ability of the mixed-order analysis of Roy [123] to give better predictions and insights into the behaviour of numerical errors than the traditional approach, although it does require more grid levels. Using the mixed-order analysis, the discretisation error for each grid, $\epsilon_{RE,n}$, can be written as

$$\epsilon_{RE,n} = f_n - f_{h\to\infty} = g_1 h_n + g_2 h_n^2 + g_3 h_n^3 + \mathcal{O}\left(h_n^4\right), \tag{2.50}$$

where n denotes the grid level, f_n the grid solution, h_n the grid size and g_1, g_2 and g_3 are constants. In calculating the four unknowns, namely $f_{h\to\infty}$, g_1, g_2, and g_3, four grids are necessary. The complete system of equations is then

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 4 & 16 & 64 \\ 1 & 8 & 64 & 128 \end{bmatrix} \begin{Bmatrix} f_{h\to\infty} \\ g_1 \\ g_2 \\ g_3 \end{Bmatrix}, \tag{2.51}$$

where $h_1 = 1$ and a refinement ratio $r = 2$ is used, and hence, $h_{n+1} = 2h_n$. Fourth-order and higher-order terms are neglected.

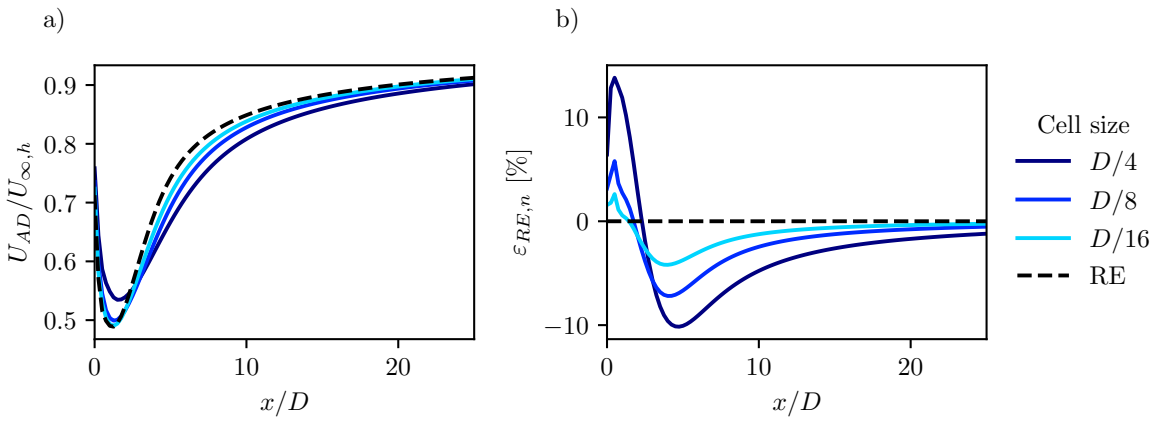


Figure 2.5: a) Normalised disc-averaged streamwise velocity and b) its discretisation error for an unyawed turbine ($\gamma = 0°$) with downstream distance for various wind farm domain cell sizes. Black dashed line indicates the value obtained from Richardson extrapolation [123]. $I_{\infty,h} = 4\%$ and $C_T = 0.8$.

The variation in $U_{AD}/U_{\infty,h}$ in Figure 2.5 is as would be expected, with a peak deficit in the near wake that slowly recovers downstream. Regarding convergence, Figure 2.5 shows firstly that the solution of U_{AD} is convergent. Moreover, it shows that for this worst-case scenario, the expected error in U_{AD} from the extrapolated value for a cell size of $D/8$ peaks at just under a 6% underprediction at a downstream position of $x \approx 4D$.

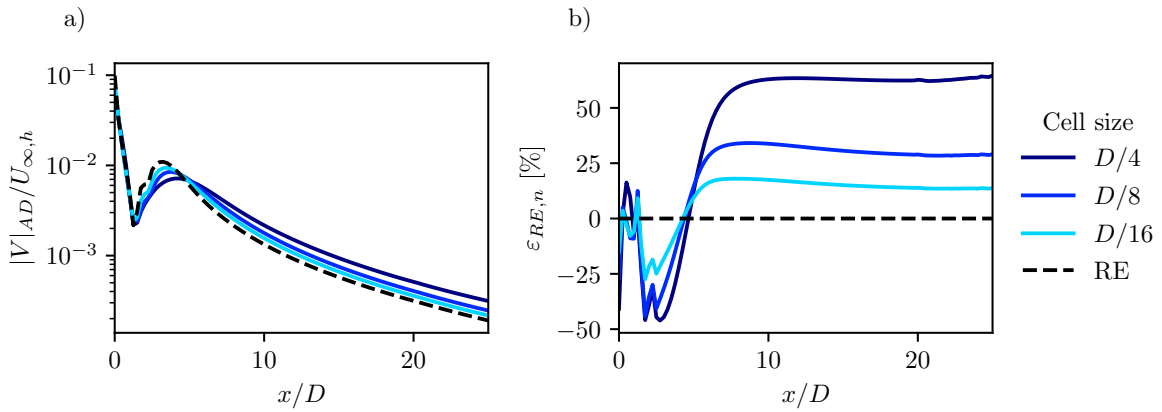


Figure 2.6: a) Normalised disc-averaged absolute lateral velocity on a log scale and b) its discretisation error for an unyawed turbine ($\gamma = 0°$) with downstream distance for various wind farm domain cell sizes. Black dashed line indicates the value obtained from Richardson extrapolation [123]. $I_{\infty,h} = 4\%$ and $C_T = 0.8$.

The variation in $|V|_{AD}/U_{\infty,h}$ is much smaller than for the streamwise velocity, as it is driven by the expansion and subsequent contraction of the streamtube around the turbine, for which the forces are much smaller than the thrust force that drives the streamwise velocity deficit. For the sake of clarity in view of these small variations, Figure 2.6(a) is plotted on a logarithmic scale. Despite the small magnitudes, there is a clear convergence of the solution with cell size, most clearly shown by the errors in Figure 2.6(b). Whilst the percentage error is larger than for the streamwise value, roughly 30% for a cell size of $D/8$, given the relative magnitude of the velocities this is not expected to be significant.

Figure 2.7 aims to put these errors into the context of figures that are common throughout this thesis by showing the effect of cell size on the the streamwise velocity deficit and lateral velocity (surplus) profiles. The profiles are taken at $x = 6D$ to provide clear comparison to the profiles in Section 3.1. For both velocity components, Figure 2.7 highlights that the effects of

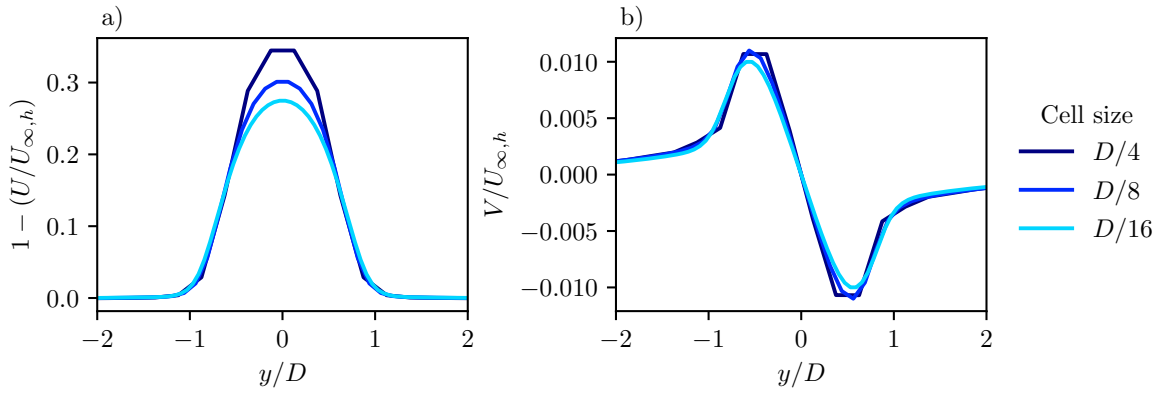


Figure 2.7: Normalised a) streamwise velocity deficit and b) lateral velocity profiles of an unyawed turbine ($\gamma = 0°$) at at $x/D = 6$ for various wind farm domain cell sizes. $I_{\infty,h} = 4\%$ and $C_T = 0.8$.

discretisation are most keenly felt for $0.5 < y/D < 0.5$. In the case of the streamwise velocity, the decrease in cell size not only leads to a less smooth profile, but also an overprediction of the peak velocity deficit. Figure 2.5 would suggest this is a trend throughout the far wake ($x \gtrapprox 2D$). Figure 2.7(b) demonstrates the point made about the large errors seen in Figure 2.6(b) being relatively inconsequential in absolute terms given the magnitude of the lateral velocities in the wake of an unyawed turbine.

As the title would suggest, this thesis is primarily concerned with yawed turbines. Therefore it is important also to make observations regarding convergence and discretisation errors for a yawed turbine. To this end, the same simulations were performed for a turbine yawed at $\gamma = 25°$.

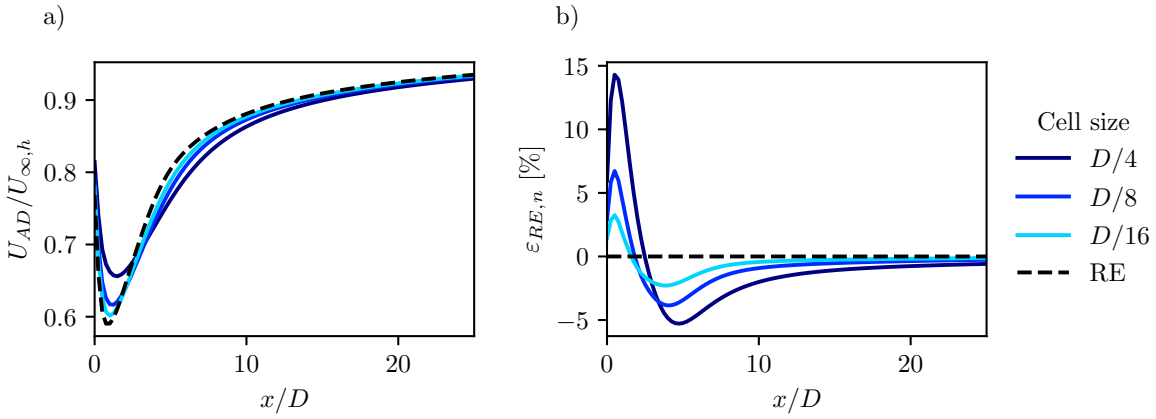


Figure 2.8: a) Normalised disc-averaged streamwise velocity and b) its discretisation error for a turbine yawed at $\gamma = 25°$ with downstream distance for various wind farm domain cell sizes. Black dashed line indicates the value obtained from Richardson extrapolation [123]. $I_{\infty,h} = 4\%$ and $C_T = 0.8$.

The variation of $U_{AD}/U_{\infty,h}$ is similar to that of the unyawed turbine - Figure 2.5(a). The streamwise thrust force component has been scaled by the cosine of the yaw angle, and thus the deficit is slightly smaller in Figure 2.8(a), but the shape is much the same. The same can be said of the discretisation error shown in Figure 2.8(b). In the yawed case, however, the peak overshoot of error in the near wake is larger than in the unyawed case, whilst the peak undershoot is smaller. This could be attributed to the more linear behaviour of the deficit associated with the smaller thrust force. In terms of convergence, the introduction of a yaw angle has not affected the convergence to the extrapolated solution with decreasing cell size.

For the lateral velocities, there is a significant increase in magnitude for $|V|_{AD}/U_{\infty,h}$ with the increase in yaw angle. This is to be expected given the lateral component of the thrust force varies with the sine of the yaw angle, and thus is zero for an unyawed turbine and non-zero for a yawed turbine. Despite the increase in magnitude, and the use of logarithmic axes in Figure 2.6(a), the shape of the $|V|_{AD}/U_{\infty,h}$ curve is very similar in Figure 2.9(a), with a point of inflection around $x = D/2$. This is because, despite the difference in forcing at the rotor disc, the wake recovery process for the unyawed and yawed cases is the same. There are notable differences, however, in the discretisation errors between Figure 2.9(b) and Figure 2.6(b) after $x \approx 4D$. Whilst for the unyawed case the error is roughly constant, for the yawed case the error peaks and then decays with increasing x. Moreover, the error in the yawed case is generally

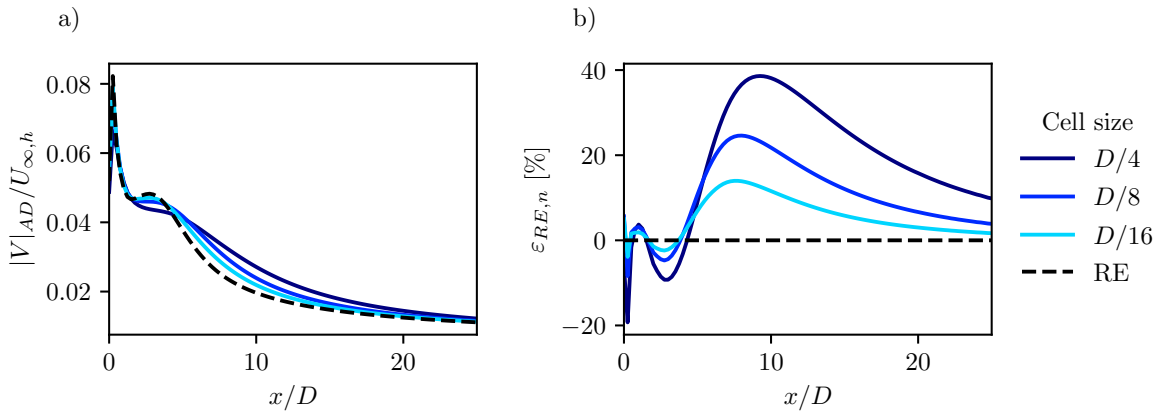


Figure 2.9: a) Normalised disc-averaged absolute lateral velocity and b) its discretisation error for a turbine yawed at $\gamma = 25°$ with downstream distance for various wind farm domain cell sizes. Black dashed line indicates the value obtained from Richardson extrapolation [123]. $I_{\infty,h} = 4\%$ and $C_T = 0.8$.

smaller, with for instance the peak for a cell size of $D/8$ at just over 20%, compared to the roughly constant error of over 25% in Figure 2.6(b). Whilst this peak error is still relatively large, the decrease of error with x is a useful feature in terms of simulation accuracy.

Finally, Figure 2.10 again demonstrates the effect of discretisation error on the two velocity profiles. A cell size of $D/4$ is clearly inadequate to predict the small depression in the lateral velocity profile at $x \approx 0.8D$, which is an important feature of the discussion in Section 3.1. Both $D/8$ and $D/16$ are capable of capturing this behaviour, though $D/16$ better predicts the peak velocities of both U and V.

The most important results of this mesh convergence study are that: (i) for both unyawed and yawed cases, the CFD setup exhibits a convergence of the relevant integral quantities towards the extrapolated value with increasing mesh density, (ii) the peak discretisation error

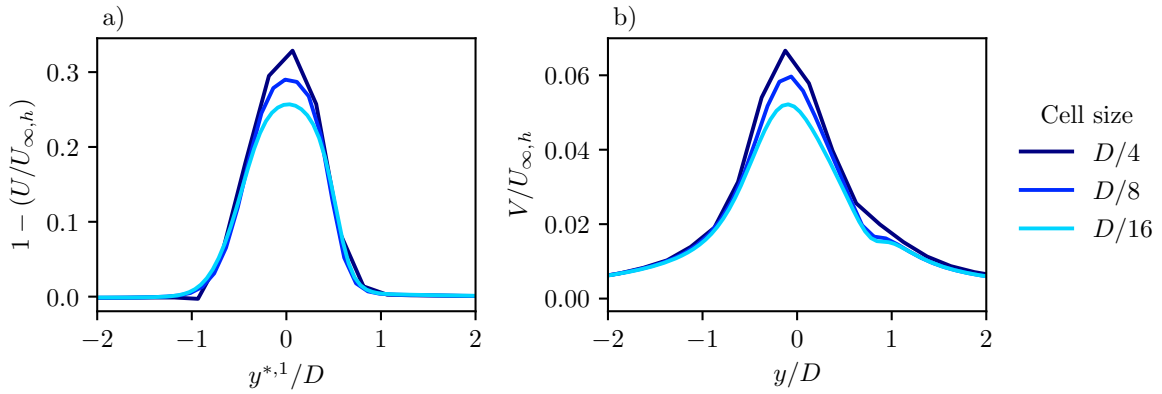


Figure 2.10: Normalised a) streamwise velocity deficit and b) lateral velocity profiles of a turbine yawed at $\gamma = 25°$ at $x/D = 7$ for various wind farm domain cell sizes. $I_{\infty,h} = 4\%$ and $C_T = 0.8$. Note the use of $y^{*,1}$ in (a) is to remove the effects of wake steering, see the discussion at the beginning of Section 3.2.

for the streamwise velocity is of the order of 5% for both the unyawed and yawed cases for a cell size of $D/8$, (iii) the peak discretisation error for the lateral velocities in the wake of an unyawed turbine are significant in percentage terms, over 25% for a cell size of $D/8$, but small in absolute terms, and (iv) the peak discretisation error for $|V_{AD}|$ for a yawed turbine is also significant, over 20% for a cell size of $D/8$, but decreases with downstream distance. These conclusions would suggest that whilst a cell size of $D/8$ is sufficient to model the streamwise velocities and trends in the lateral velocities, the error is large enough that a cell size $D/16$ or finer is more appropriate. This conclusion was reached late on in the project however, and thus the subsequent simulations are performed with a cell size of $D/8$. Finally, the effects of the actuator disc discretisation are not presented here, they are assumed to be similar to those in Baungaard [15, §2.4.3], however, this is a matter for further research.

CHAPTER 3
The Lateral Wake

The wake of a wind turbine is most commonly defined in terms of the streamwise velocity, see for example the discussions of wind turbine flow regions and the far wake in Sections 1.1 and 1.2. The idea of the 'lateral wake' [5, 6] was introduced in Section 1.4, but, given the small velocity magnitudes relative to the streamwise wake, particularly for unyawed turbines and power production, lateral wakes have received appreciably less attention in literature. Whilst there has been investigation of the lateral wake velocity profile in the context of yawed turbines, particularly for determining the wake trajectory and skew angle [12], and as a result of vortex models [22], there is no literature specifically covering the physics of lateral wakes.

The self-similarity of the streamwise wake velocity deficit profile of a standalone wind turbine is well established based upon the theory of axisymmetric bluff body wakes [110, §5.4.4, 137]. Furthermore, it is a fundamental tenet of most wake models, including the streamwise solution of Bastankhah et al. [14]. On the other hand, the self-similarity of the lateral wake velocity profile is less well researched, to the extent that there is not a definitive experimental or computational study of the self-similarity of the profile. To apply the methods of Bastankhah et al. [14] to the lateral wake, it is fundamental to first establish the self-similarity of the lateral wake velocity profile, and then to determine a suitable self-similar function with which to describe it. This chapter therefore covers a parametric study in PyWakEllipSys that was undertaken as part of this thesis to determine whether or not the lateral wake velocity profile of a yawed turbine is self-similar, and, subsequently, what distribution best fits the profile.

The study covers the effects of yaw angle, γ, thrust coefficient, C_T, total incoming turbulence intensity at hub height, $I_{\infty,h}$, and wake rotation, on the magnitude, shape, and self-similarity of the lateral wake velocity profile with downstream distance of a single NREL 5 MW turbine [74]. The aim of the study is to produce generally applicably results, and as such the turbine uses an analytical force distribution based on the Joukowksky rotor [133]. The yaw offset is limited to 40°, as this is the largest value investigated in the literature [77], the thrust coefficient varies between 0.2 and 0.8, to cover the range of values typical of offshore turbines such as the NREL 5 MW, and the turbulence intensities follow the same logic, varying between 4 and 10% to reflect typical offshore values. The yaw angle is defined as being positive for a clockwise rotation when the rotor is viewed from above, in line with PyWakeEllipSys convention. All simulations use the adiabatic log law with an inflow precursor simulation to ensure there is no downstream development of the profile. The turbulence model, boundary conditions, and rotor model are all as detailed in Chapter 2. Given this study is only concerned with standalone turbines, it is possible to prescribe the thrust coefficient in every case. The TSR was kept constant throughout, set at the value for the NREL 5 MW at $8\,\mathrm{m\,s^{-1}}$, 7.5.

The effect of the aforementioned parameters on the lateral wake is investigated via four features in this study. Firstly, there is the effect of each on the velocity profile itself, both the shape and magnitude, in Section 3.1. There is then an examination of their effect on the self-similarity of the profile with downstream distance in Section 3.2, which is followed by a brief investigation of suitable distributions to fit the profile, considering both the fit and the

utility of the distributions, in Section 3.3. Finally, there is an investigation of the effects of each parameter on the wake width and centre location, including a comparison with the streamwise wake in Section 3.4 and Section 3.5, respectively.

3.1 Velocity profile

The lateral velocity around a wind turbine is determined by three main driving forces: (i) the tendency of the flow to go around the turbine, i.e. the expansion of the streamtube, (ii) the addition of lateral momentum by any lateral component of the thrust force, and (iii) the subsequent entrainment of momentum from the freestream in the wake recovery process.

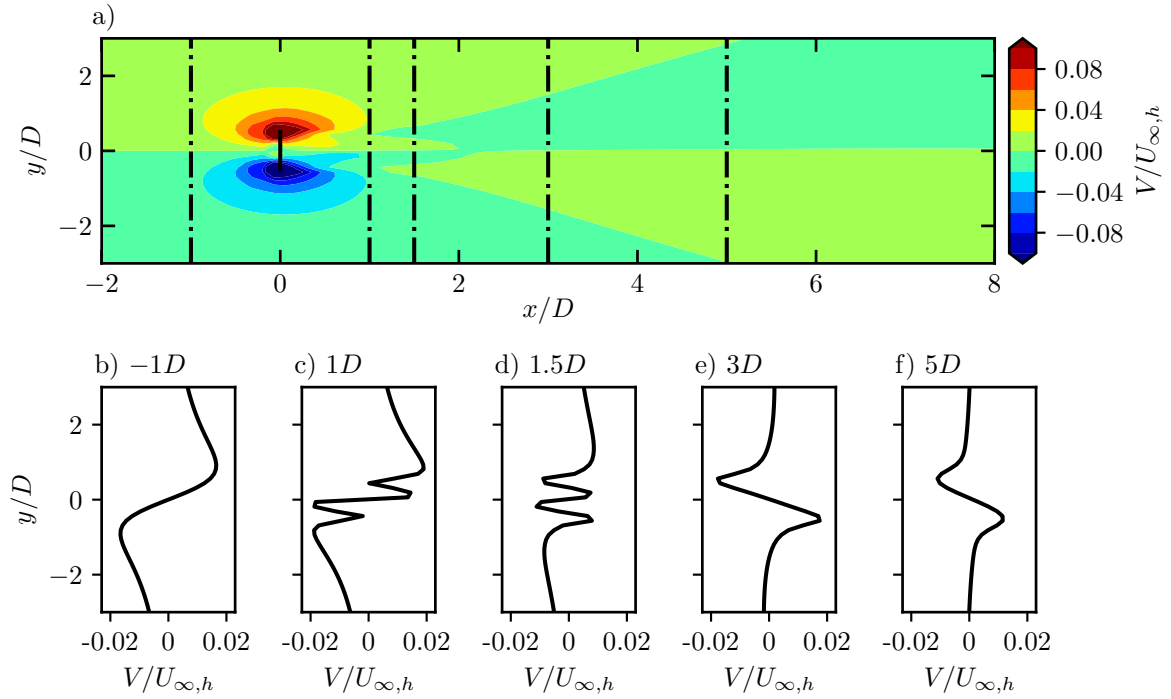


Figure 3.1: (a) Contour plot, and lateral profiles of normalised hub-height lateral velocity for an unyawed turbine at $x/D =$ (b) -1, (c) 1, (d) 1.5, (e) 3, and (f) 5. $C_T = 0.8$, $I_{\infty,h} = 6\%$, and $z_h = 90\,\mathrm{m}$. Turbine located at $(0, 0)$.

The relative magnitude of these effects determine the shape of the lateral velocity profile within each of the wind turbine flow regions described in Figure 1.1. For an unyawed turbine, as shown in Figure 3.1, the induction region is dominated by the divergence of flow around the turbine, which manifests in the profile as a sinusoidal shape with $V = 0$ at the rotor centre. In the near wake this shape is disrupted by the strong pressure fluctuations within $-0.5 \leq y/D \leq 0.5$, but relatively unaffected outside. Further downstream however, the process of wake recovery, wherein momentum is drawn into the momentum sink caused by the extraction of momentum by the rotor, begins to change the overall direction of lateral velocity. By the start of the far wake, at approximately $x/D = 2$ in Figure 3.1, the lateral velocity profile has essentially flipped so that flow is towards the centreline.

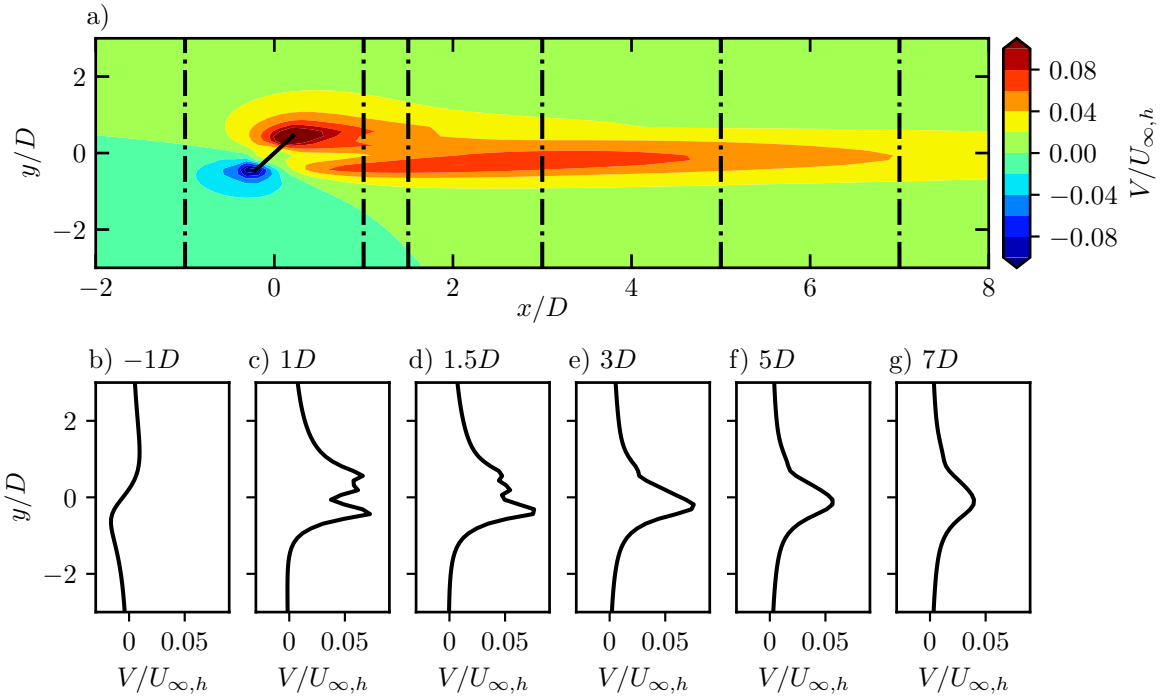


Figure 3.2: (a) Contour plot, and lateral profiles of normalised hub-height lateral velocity for a turbine with yaw angle $\gamma = 25°$, at $x/D =$ (b) -1, (c) 1, (d) 1.5, (e) 3, (f) 5, and (g) 7. $C_T = 0.8$, $I_{\infty,h} = 6\%$, and $z_h = 90\,\text{m}$. Turbine located at $(0, 0)$.

The interplay of these driving forces changes when the turbine is yawed. As per Figure 3.2, whilst the behaviour is similar in the induction zone, the effect of the lateral component of the thrust force now affects the flow in the wake. For example, at $x/D = 1$, the velocity profile is overwhelmingly positive, i.e. the lateral forcing from the turbine has cancelled out the effect of flow diverging around it. This continues into the far wake, where although the divergence has been replaced by the effects of wake recovery, the profile is still dominated by the lateral forcing. Moreover, unlike with the streamwise wake profile, which would be steered towards positive values of y for this yaw angle, the peak of the lateral profile appears to tend asymptotically towards $y = 0$ in the far wake from its origins at negative values of y. This will be discussed further in Section 3.5.

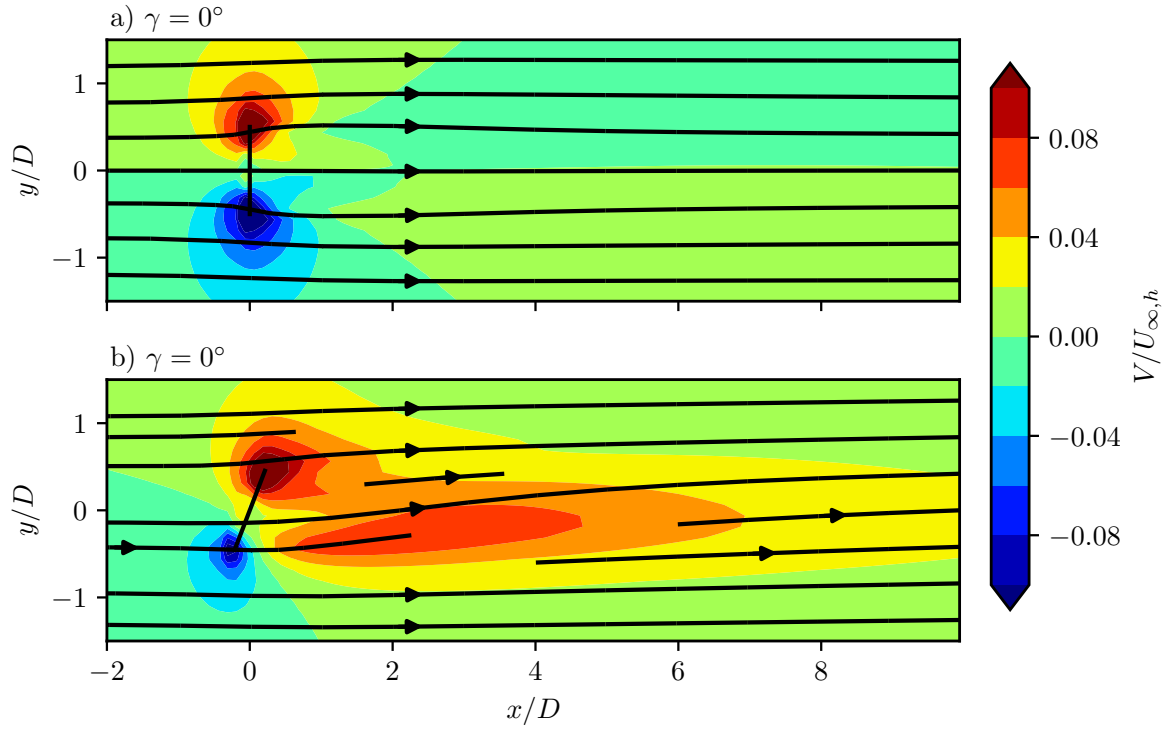


Figure 3.3: Contour plots of normalised hub-height lateral velocity with streamlines for a turbine with yaw angle (a) $\gamma = 0°$ and (b) $\gamma = 25°$. $C_T = 0.8$, $I_{\infty,h} = 6\%$, and $z_h = 90\,\mathrm{m}$. Turbine located at $(0, 0)$.

Streamlines provide a useful way of visualising the differences in the flow behaviour between yawed and unyawed turbines. Although the lateral velocities are small compared to the streamwise, and thus the lateral deviation of the streamlines is minimal, the divergence of the streamlines in the induction zone and near wake is evident in Figure 3.3(a), as is the contraction in the far wake, where the velocity profile is flipped. On the other hand, Figure 3.3(b) shows the dominance of the lateral forcing in the wake.

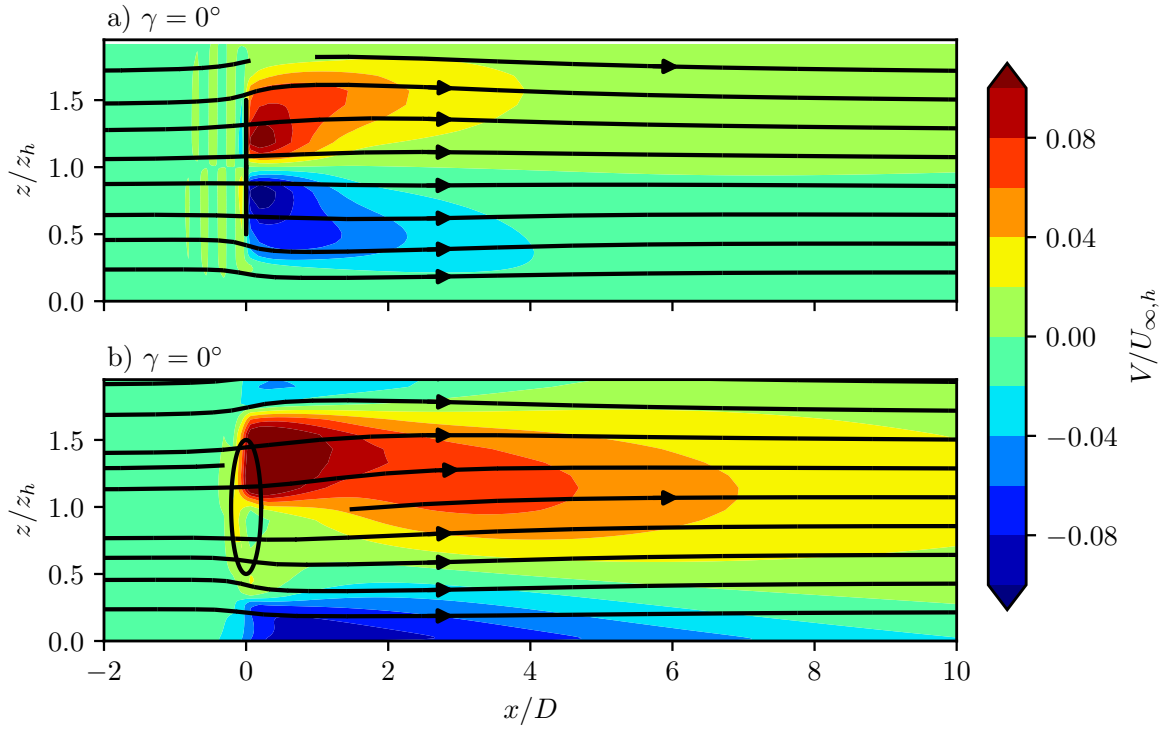


Figure 3.4: Contour plots of normalised lateral velocity in a x-z plane at $y = 0$ for a turbine with yaw angle (a) $\gamma = 0°$ and (b) $\gamma = 25°$. $C_T = 0.8$, $I_{\infty,h} = 6\%$, and $z_h = 90\,\mathrm{m}$. Turbine located at $(0, 1)$.

Naturally, it should be pointed out that the behaviour described so far is a 2D snapshot of a 3D phenomenon. Outside of the hub-height x-y plane, the rotation of the rotor introduces lateral velocity which dominates the behaviour in the x-z plane for unyawed turbines, as can be seen in Figure 3.4(a). From Figure 3.4(b), however, it is evident that for a yaw angle of 25° the lateral forcing also dominates the behaviour in this plane, particularly in the far wake. Note that these simulations do not include any wind veer or Coriolis forces, which otherwise would also affect the flow behaviour.

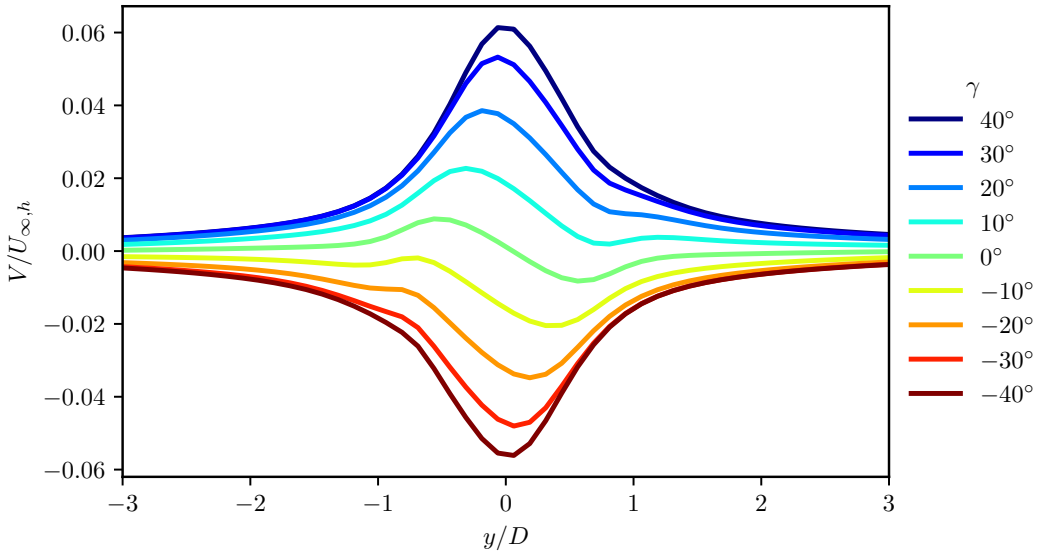


Figure 3.5: Lateral profile of normalised hub-height lateral velocity profile at $x/D = 6$ for various yaw angles. $C_T = 0.8$ and $I_{\infty,h} = 6\%$.

The previous examples showed an unyawed and a yawed case ($\gamma = 25°$), but as the lateral forcing is dependent on the magnitude of the yaw angle, one could imagine that this also has an effect on the balance of the driving forces. This is demonstrated in Figure 3.5 for a far-wake downstream position of $x/D = 6$.

Most evidently, increasing the yaw angle, in either direction, increases the lateral velocity for this downstream position. This is because of the three driving forces, the lateral forcing from the turbine is mostly strongly influenced by the yaw angle. That is not to say, however, that the wake recovery process is unaffected by the yaw angle, this will be addressed in Section 3.4. Nevertheless, the lateral component of the thrust force is related to the sine of the yaw angle, which causes this increase in lateral velocity. Increasing the yaw angle also changes the shape of the profile. This is also due to the balance of these driving forces, as a yaw angle of zero means no contribution of turbine lateral forcing, and so at this far-wake location the profile is solely determined by the lateral entrainment of momentum in the wake recovery process. The presentation of a transition in yaw angles as shown in Figure 3.5 also highlights the migration of the peak (positive or negative) velocity towards $y = 0$ with an increase in yaw angle magnitude. This is likely because the thrust forcing has a peak at the rotor centre, unlike the wake recovery or flow divergence, which have a point of inflection.

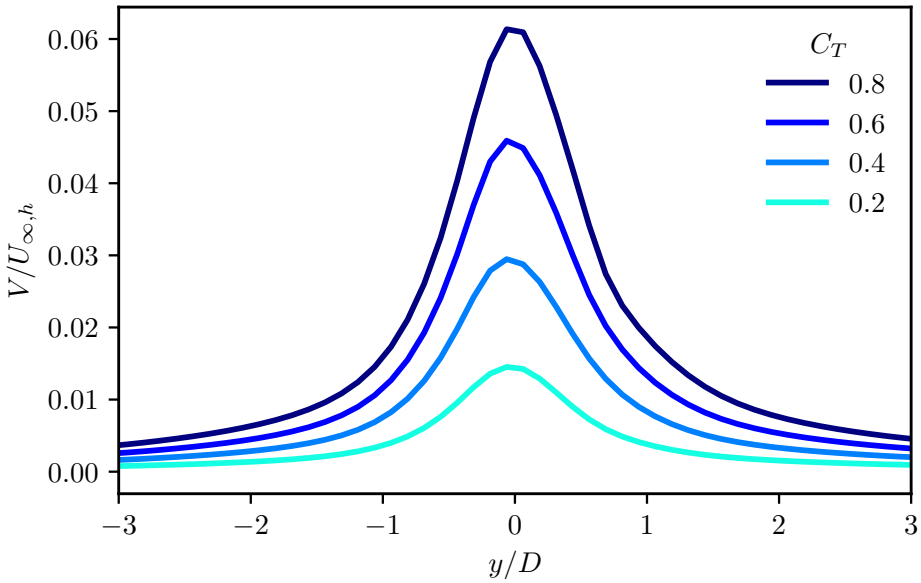


Figure 3.6: Hub-height lateral velocity profile at $x/D = 6$ for various thrust coefficients. $\gamma = 25°$ and $I_{\infty,h} = 6\%$.

As discussed in Section 2.3.2, the two determining parameters for the flow behaviour in a non-dimensionalised system such as this are the thrust coefficient and the incoming turbulence intensity. The thrust coefficient is incorporated as a scaling factor for the force distribution presented in Equation 2.41, and thus also scales the lateral forcing for a yawed turbine. This scaling can be seen for a turbine yawed by 25° at $x/D = 6$ in Figure 3.6.

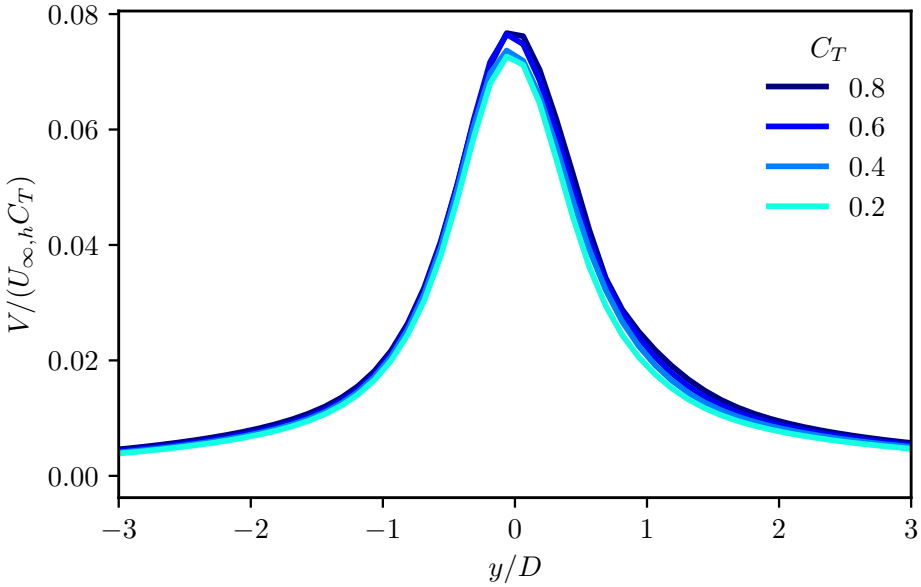


Figure 3.7: Hub-height lateral velocity profile normalised by thrust coefficient at $x/D = 6$ for various thrust coefficients. $\gamma = 25°$ and $I_{\infty,h} = 6\%$.

Normalising the profiles by thrust coefficient as in Figure 3.7 shows that in fact the scaling is linear. This has also been shown for streamwise wake profiles, but only for small values of thrust coefficient [122, §2.2.2]. Given the lateral velocities are so small, the scaling is linear over a wider range of thrust coefficients for the lateral profile.

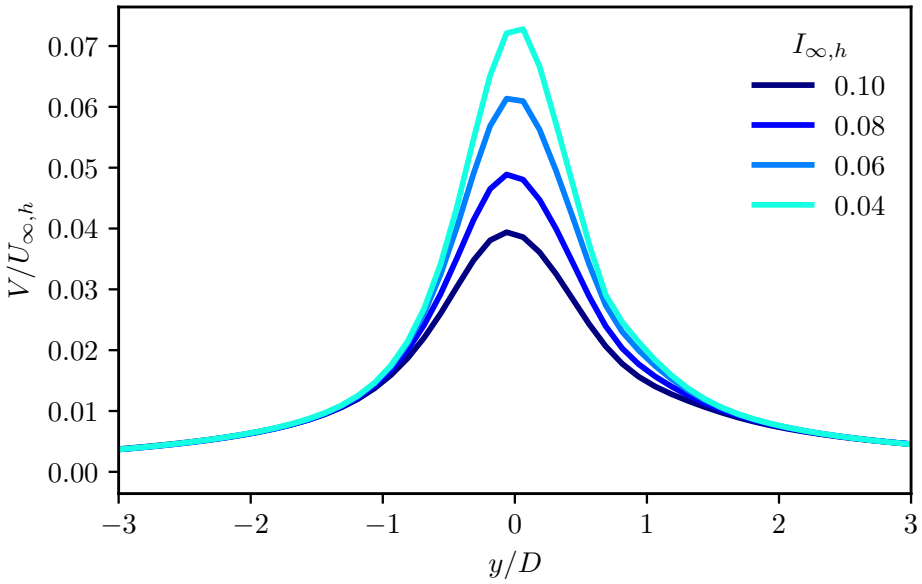


Figure 3.8: Hub-height lateral velocity profile at $x/D = 6$ for various hub-height incoming turbulence intensities. $\gamma = 25°$ and $C_T = 0.8$.

The effect of incoming turbulence intensity is less well defined. Turbulence intensity tends to affect the interaction between the flow and forcing elements such as the rotor. For example, in Figure 3.8 one can observe that the profile is only affected within the bounds of around $1D$ either side of the rotor centre. Within this range, there is an inverse relationship between $I_{\infty,h}$ and the magnitude of the lateral velocity. The relationship is not reciprocal however, nor is it logarithmic, and determining the exact relationship and its causes requires more research.

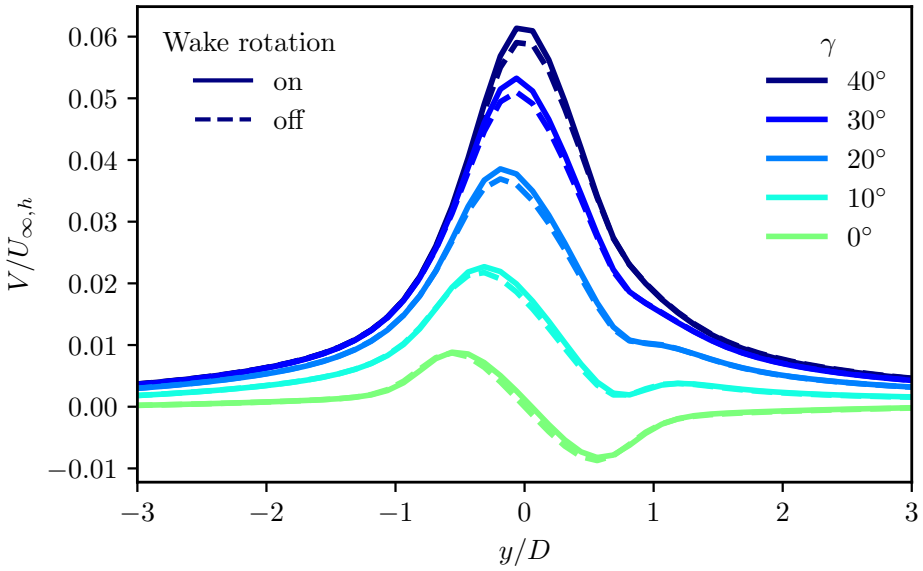


Figure 3.9: Hub-height lateral velocity profile at $x/D = 6$ for various yaw angles with wake rotation on (filled markers) and off (unfilled markers). $C_T = 0.8$ and $I_{\infty,h} = 6\%$.

Finally, in order to rule out the effect of wake rotation, Figure 3.9 shows that the difference between the profile with tangential forces on and off is minimal, and diminishes for smaller yaw angles. This would suggest that the lateral forcing is more strongly affected by wake rotation than wake recovery, but also that the effect on either is negligible.

3.2 Self-similarity

Self-similarity as a concept arises within several areas in the study of turbulent flows, and a general definition can be found in Pope [110, pg. 99]. For the hub-height lateral wake velocity profile, $V_h(x, y)$, one can define two characteristic scales as functions of downstream distance, $V_c(x)$, the maximum lateral velocity, and $\sigma_{y,2}(x)$, the width of the profile. This leads to the scaled variables

$$\zeta \equiv y/\sigma_{y,2}(x), \quad \text{and} \quad \tilde{V}(\zeta, x) \equiv \frac{V_h(x, y)}{V_c(x)}. \tag{3.1}$$

If $\tilde{V}$ is independent of x, i.e. there is a function $f(\zeta)$ such that

$$\tilde{V}(\zeta, x) = f(\zeta), \tag{3.2}$$

and so

$$V_h(x, y) = V_c(x) f(\sigma_{y,2}(x), y), \tag{3.3}$$

then $V_h(x, y)$ is self-similar.

In practice, this means that normalising the magnitude of the velocity profile by a characteristic velocity and profile width by a characteristic width should cause the normalised profiles from several downstream positions to collapse on to a single self-similar function, f. Both the characteristic scales given in Equation 3.1 can be determined in several ways [12, §4], but for establishing the degree of self-similarity the consistency of the approach is more important the

approach itself. For this thesis, the characteristic velocity is simply the maximum velocity in the profile, and the characteristic width is defined as [12, Eq. 4.1]

$$\sigma_{y,2}(x) = \frac{1}{\sqrt{2\pi}|V|} \lim_{y\to\infty} \int_{-y}^{y} |V| \, \mathrm{d}\hat{y}, \tag{3.4}$$

where $\hat{y}$ is the integration variable and which reduces to the standard deviation for a purely Gaussian profile. Finally, for yawed turbines, the wake centre is defined as $y^{*,2}$, which can deviate from y for yawed turbines. As a result, instead of the independent scaled variable, ζ is instead defined as

$$\zeta^* = y^*/\sigma_{y,2}(x) \tag{3.5}$$

where $y^{*,2}$ is the y-position of the centre of the lateral wake profile. This can also be defined in several ways, but this thesis uses the y-location of the maximum velocity. Note also that $y^{*,2}$ is not necessarily equal to $y^{*,1}$, the centre of the streamwise wake velocity deficit profile, as will be discussed in Section 3.5.

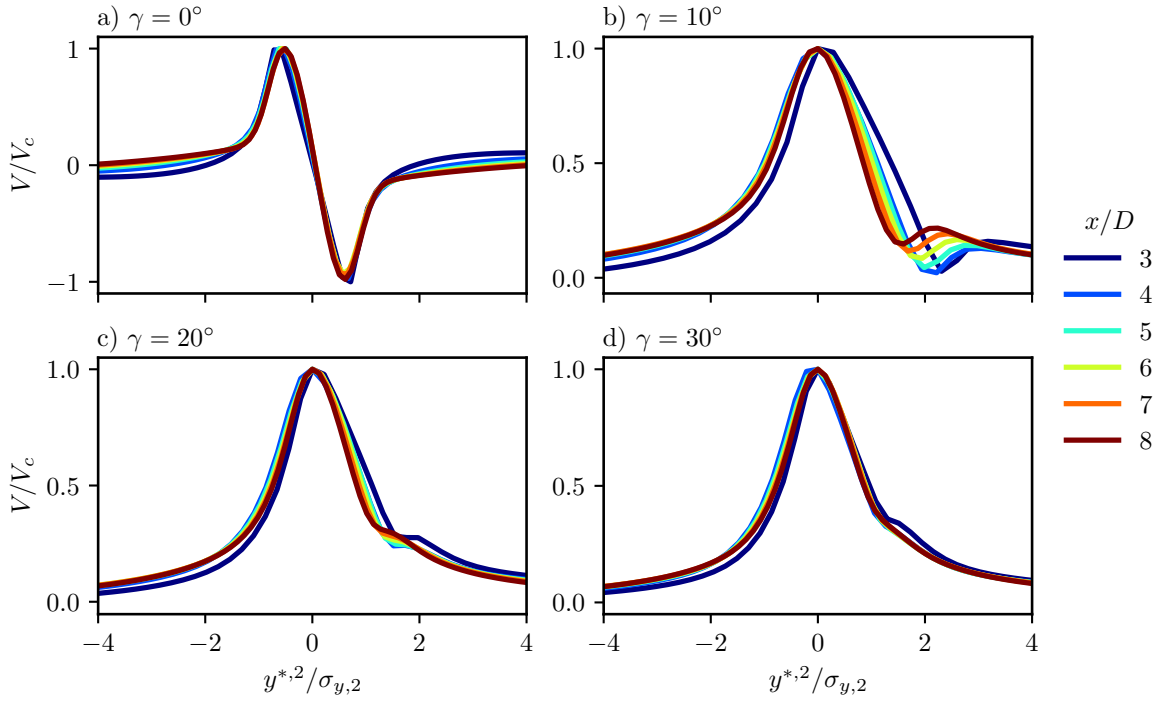


Figure 3.10: Normalised profiles of lateral velocity at various downstream positions for yaw angles of (a) 0°, (b) 10°, (c) 20°, and (d) 30°. $C_T = 0.8$ and $I_{\infty,h} = 6\%$.

Bastankhah and Porté-Agel [12] used wind tunnel experiments to demonstrate the self-similarity of the streamwise velocity deficit profile for yaw angles of 0°, 10°, 20°, and 30°. However, there appears to be very limited literature investigating the same for the lateral velocity profile. Simulating the cases using PyWakeEllipSys and then normalising according to the method above, Figure 3.10 highlights the importance of yaw angle in the degree of self-similarity exhibited by the lateral velocity profile. For an unyawed turbine, where the velocity is dominated by the lateral entrainment of momentum, there is a reasonable degree of self-similarity, as seen in Figure 3.10(a). Within the region $-1 < y^{*,2}/\sigma_{y,2} < 1$ in particular,

there is a clear collapse of the various downstream positions. However, outside of this there is a development of the normalised profile with x. This can be attributed to the balance between the diminishing effect of the flow divergence and the increasing effect of wake recovery, as the flow transitions from moving away from the wake laterally, to moving towards it. For non-zero yaw angles, the degree of self-similarity increases with increasing yaw angle. Again, the digression from self-similarity can be attributed to the balance of driving forces. Where a single physical phenomenon dominates the flow behaviour, the profile is self-similar, but where there is the contribution of more than one effect, such as around $y^{*,2}/\sigma_{y,2} = 1$ in Figure 3.10(b) where the lateral forcing competes with the wake recovery, the self-similarity is affected by how the balance of these agents changes with downstream distance.

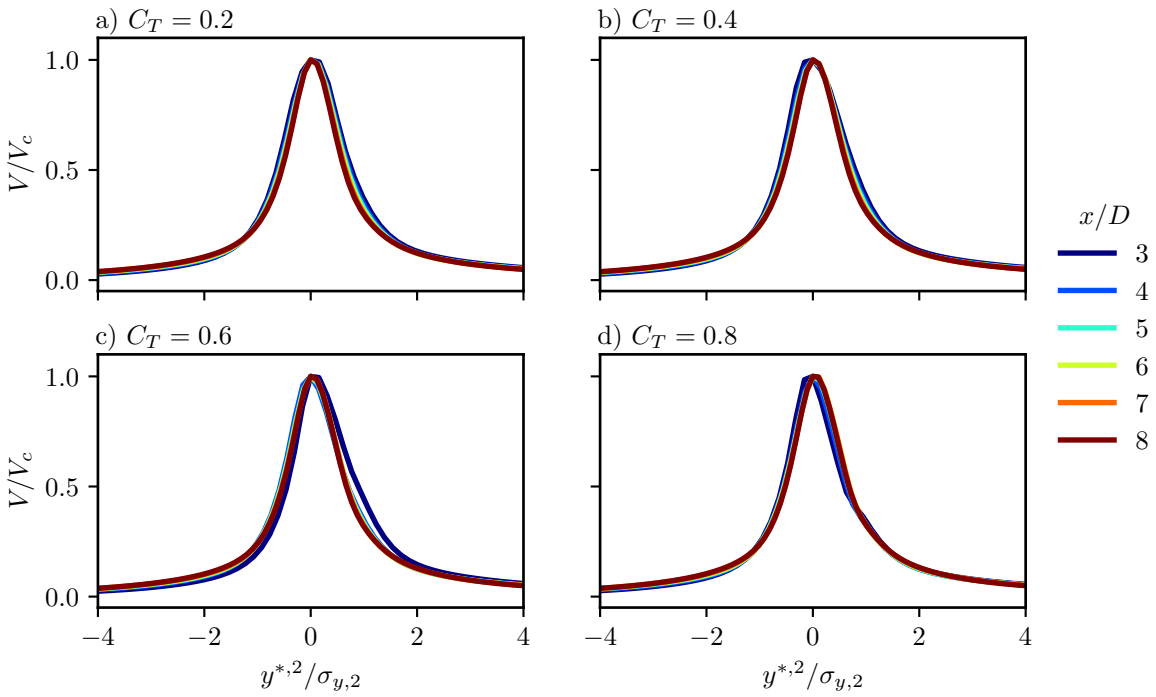


Figure 3.11: Normalised profiles of lateral velocity at various downstream positions for thrust coefficients of (a) 0.2, (b) 0.4, (c) 0.6, and (d) 0.8. $\gamma = 25°$ and $I_{\infty,h} = 6\%$.

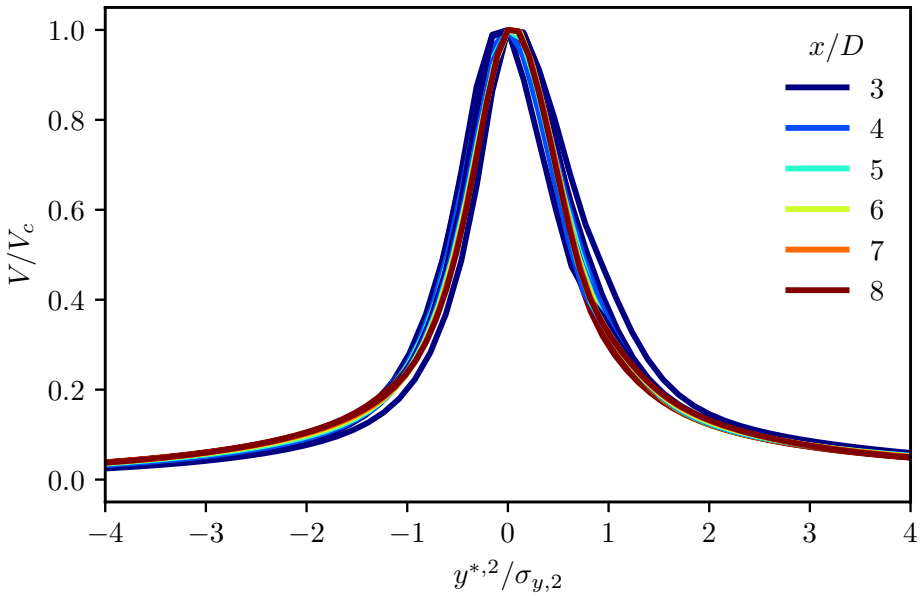


Figure 3.12: Normalised profiles of lateral velocity at various downstream positions for various thrust coefficients. $\gamma = 25°$ and $I_{\infty,h} = 6\%$.

It is also possible to isolate the influence of the thrust coefficient on self-similarity. Figure 3.11 demonstrates that the self-similarity is relatively unaffected by C_T, and in fact not only is the profile self-similar for a given C_T value, it is self-similar across all four values of C_T investigated, as can be seen in Figure 3.12, where all four subfigures of Figure 3.11 are plotted atop one another. This can be explained by the conclusions drawn in Section 3.1, where it was shown that V scales linearly with C_T. This implies that V/V_c is independent of C_T. Moreover, it was shown that C_T has no effect on the width of the profile, $\sigma_{y,2}$.

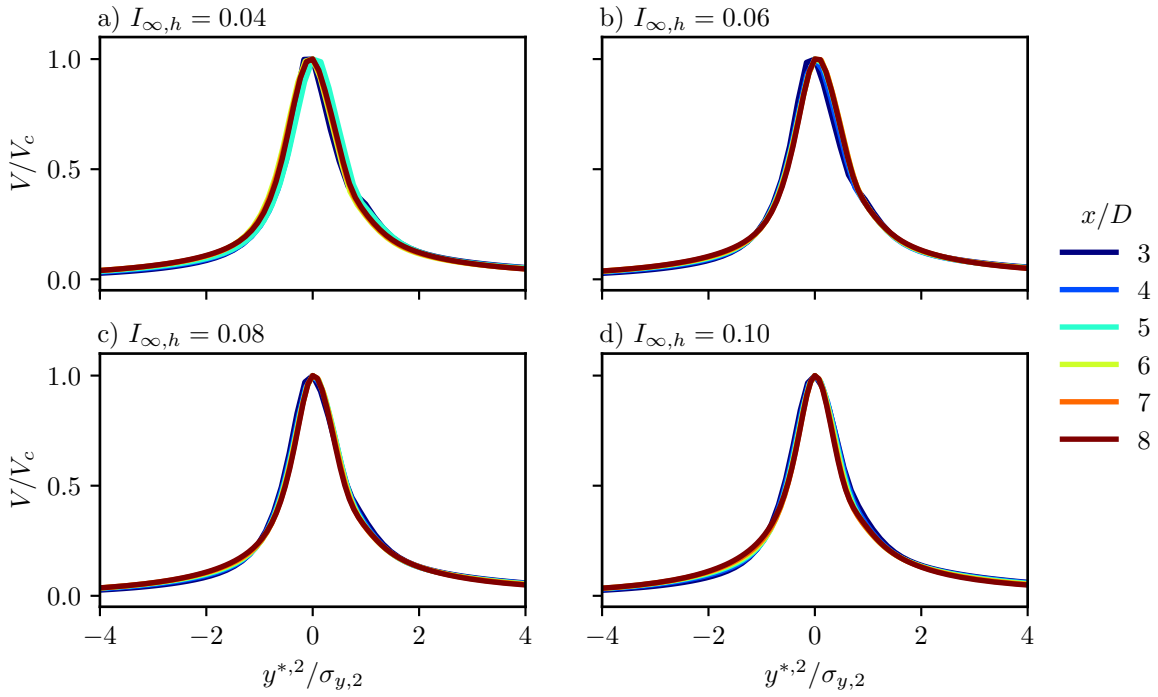


Figure 3.13: Normalised profiles of lateral velocity at various downstream positions for incoming hub-height turbulence intensities of (a) 4%, (b) 6%, (c) 8%, and (d) 10%. $\gamma = 25°$ and $C_T = 0.8$.

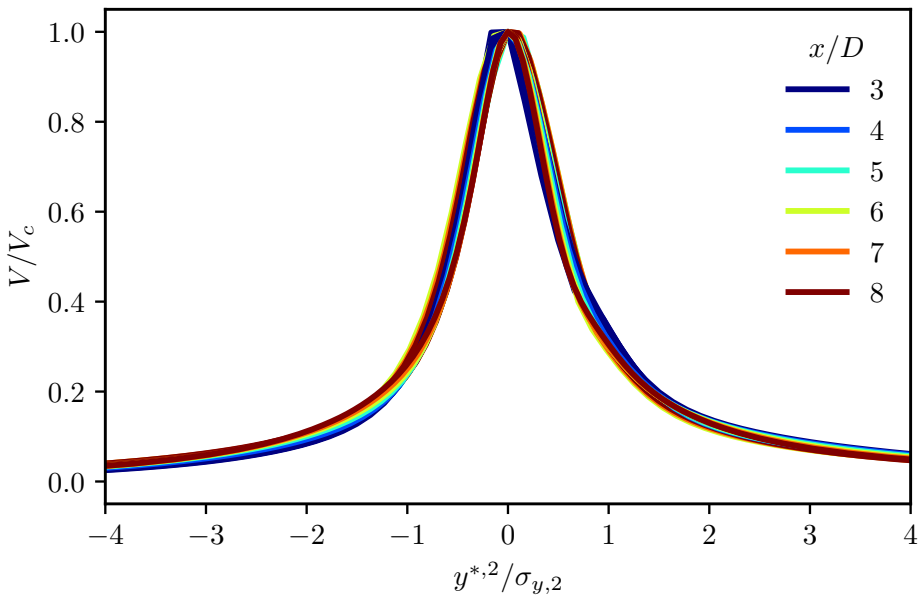


Figure 3.14: Normalised profiles of lateral velocity at various downstream positions for various incoming hub-height turbulence intensities. $\gamma = 25°$ and $C_T = 0.8$.

Finally, Figure 3.13 shows that incoming turbulence intensity does not have a significant effect on the self-similarity of the profile. Whilst Figure 3.8 illustrated that increasing $I_{\infty,h}$

reduced the velocity in around the centre of the profile, the effect appears to have been balanced by a change in width at the centre of the profile, such that the normalised profile is relatively unchanged. Moreover, Figure 3.14 demonstrates the same as for C_T, that the profile is self-similar across all investigated values of $I_{\infty,h}$.

3.3 Distribution fitting

The utility of the profile being self-similar with downstream distance is that one can then use a single function to describe the velocity profile at any downstream position simply by scaling the magnitude and width.

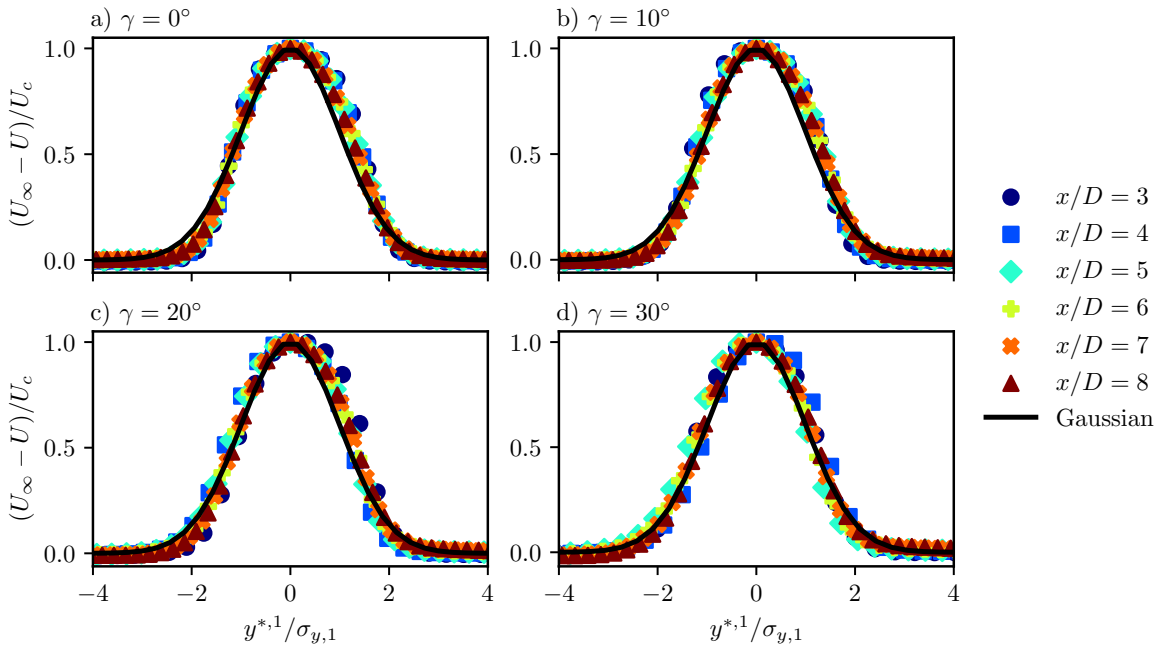


Figure 3.15: Normalised profiles of streamwise velocity deficit at various downstream positions with fitted Gaussian profile for yaw angles of (a) 0°, (b) 10°, (c) 20°, and (d) 30°. $C_T = 0.8$ and $I_{\infty,h} = 6\%$.

The choice of function depends on the flow physics. For example, a Gaussian distribution, derived from a thin-shear simplification of the RANS equations [137], fits the streamwise velocity deficit profile of both a yawed and unyawed turbine very well, as can be seen in Figure 3.15.

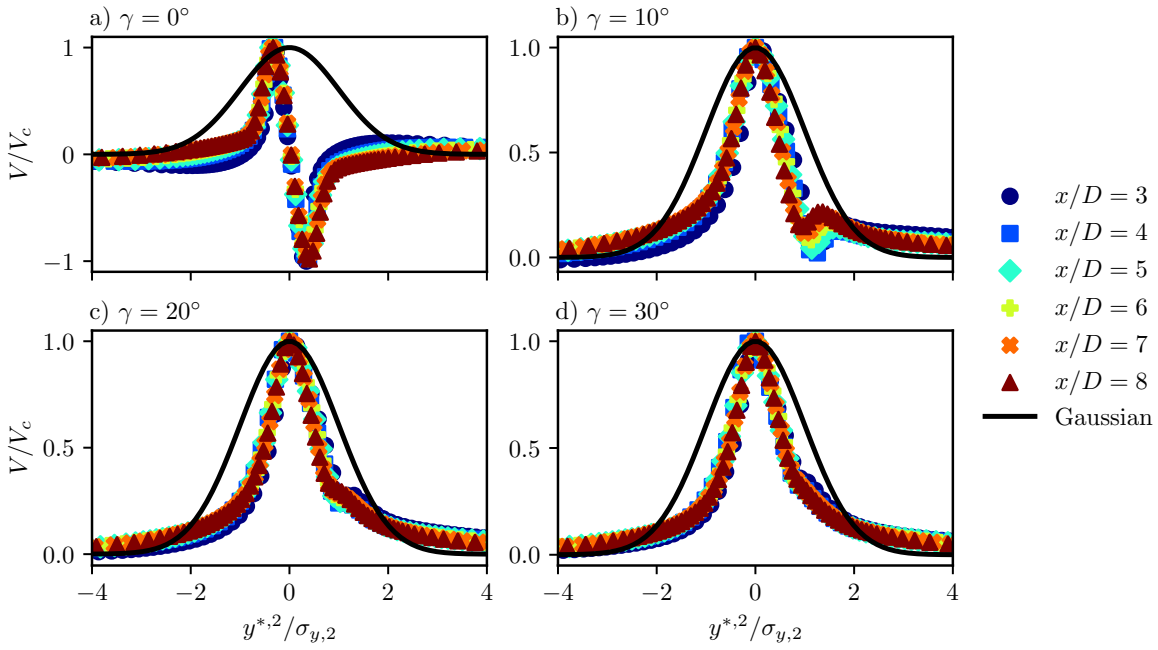


Figure 3.16: Normalised profiles of lateral velocity at various downstream positions with fitted Gaussian profile for yaw angles of (a) 0°, (b) 10°, (c) 20°, and (d) 30°. $C_T = 0.8$ and $I_{\infty,h} = 6\%$.

The lateral velocity distribution, however, is governed by different flow physics. For an unyawed turbine as in Figure 3.16(a), where wake recovery primarily determines the profile, the Gaussian is not appropriate. The fit is better for yawed turbines, likely because the lateral forcing results in a profile which is more Gaussian. Nevertheless, it is clear from Figure 3.16(b)-(d) that the fit is noticeably worse than in Figure 3.15(b)-(d). That is to say, a Gaussian profile is not as suitable for describing the normalised lateral velocity profile produced by lateral forcing as it is for the streamwise velocity deficit profile. Specifically, the Gaussian overpredicts velocities within roughly $-2 \leq y^{*,2}/\sigma_{y,2} \leq 2$, and underpredicts outside of this range. This can be attributed mostly to the tails of the distribution being 'heavier' than is predicted by a Gaussian.

One alternative is to use a vortex model to derive a suitable hub-height velocity distribution. Branlard and Gaunaa [22], building upon Branlard et al. [20] and Branlard and Gaunaa [21, 23], presents such a vortex model, from which a centreline velocity, V_s, as a function of the distance from the rotor hub, s, of the form

$$V_s(s) = U_{\infty,h}\frac{8}{\pi^2}\left(1-\sqrt{1-c_t}\right)\sin\gamma\frac{1}{1+\left(\frac{s}{H/2}\right)^2} \tag{3.6}$$

can be derived, where H is the vertical distance between the two vortices in the counter-rotating vortex pair. By incorporating information regarding the physics of the lateral wake, i.e. the CVP, the vortex model can more accurately model the velocities in the near and far wake. However, as is often the case, this increased fidelity comes at the cost of increased complexity. This increased complexity makes the vortex model solution difficult to apply within the analytical solution framework, which requires the wake velocity function to be integrated. Compared to the well-known Gaussian integral solution, the integral of Equation 3.6 is by no means

trivial, and may not even have an analytical solution, which would defeat the object of the proposed analytical model. For this reason, the Gaussian is chosen as the distribution to fit the normalised lateral velocity distribution, despite its shortcomings under some circumstances. The implications of this choice are discussed in Chapter 6.

3.4 Wake width

Reducing the description of the wake to two characteristic variables, namely, the wake centre velocity, V_c, and the lateral wake width, $\sigma_{y,2}$, raises the question of how the parameters under investigation affect these characteristic features. Most interestingly for the case of yawed turbines and wake steering is how these parameters might affect the location and size of the wake. The subsequent two sections will therefore investigate the effect of yaw angle, thrust coefficient, and turbulence intensity on first the wake width, followed by the wake centre location, making comparison to the same features in the streamwise wake, for which there is a significant body of previous research. For the following discussion, the wake width, $\sigma_{y,2}$, is defined as in Equation 3.4.

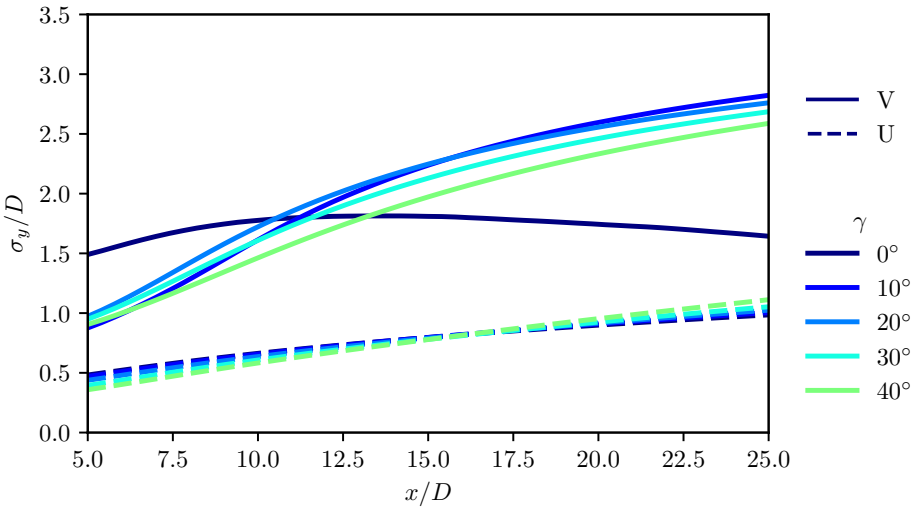


Figure 3.17: Variation of normalised streamwise (dashed line) and lateral (solid line) wake half-width with downstream distance for various yaw angles. $C_T = 0.8$ and $I_{\infty,h} = 6\%$.

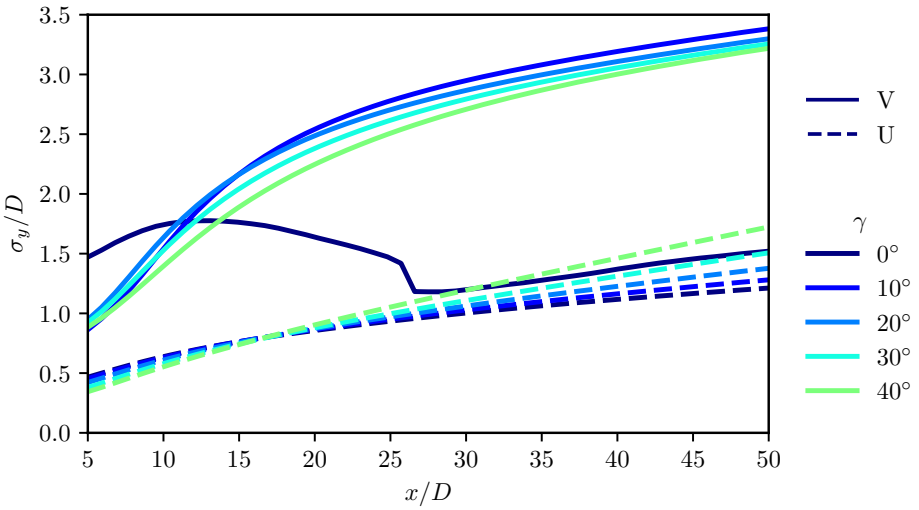


Figure 3.18: Variation of normalised streamwise (dashed line) and lateral (solid line) wake half-width with downstream distance for various yaw angles over an extended domain. $C_T = 0.8$ and $I_{\infty,h} = 6\%$.

Some general observations are immediately clear from the comparison between the streamwise and lateral wakes in Figure 3.17. Firstly, for all yaw angles, the width of the lateral wake is larger than the streamwise. This can be explained as being a product of the flow physics which defines these two wakes. This can be explained by interrogating the physics of the two components wakes. Where the streamwise wake is defined by vortex structures aligned with an x-z plane, the lateral wake is defined by structures in the x-y plane. This means that despite their relatively small vorticity, the lateral wake structures induce velocities which persist much further in the lateral direction. Secondly, whilst the streamwise wake width is affine with downstream distance, the lateral wake width is non-linear. The shear layer that bounds the streamwise wake continues to expand ad infinitum, but the lateral wake exhibits some slowing of the wake recovery rate. This again is due to the difference in the flow physics governing each, but the exact reasons require further examination.

There are also several trends to be discussed regarding the lateral wake width with yaw angle. For example, as has been a recurring theme throughout this chapter, there is a clear transition in behaviour between $\gamma = 0°$ and 10°, after which there is relatively little change in the lateral wake width, $\sigma_{y,2}$, with yaw angle. This supports the discussion from the previous sections regarding the different driving forces for the lateral wake and their dependence on yaw angle. In this example, for $\gamma = 0°$, where the wake is governed mostly by the divergence and then wake recovery process, the wake initially expands, and then begins to contract again. In fact, from Figure 3.18, one can observe that it contracts until roughly its initial width just beyond $x = 25D$ before matching the streamwise wake. The very small magnitudes involved would suggest this tracking of the streamwise wake widths beyond $x = 25D$ is an artefact introduced in the post-processing. On the other hand, for the yawed cases, the wake is initially narrower, but expands more rapidly, and whilst this expansion slows with downstream distance, from Figure 3.18 it is evident that the wake does not begin to contract even within $50D$ downstream. The major difference between the unyawed and yawed cases here is again the presence, or otherwise, of a lateral forcing component. Whilst both the wakes of unyawed and

yawed turbines undergo the opposing processes of wake expansion and recovery which cause the growth and contraction, respectively, the yawed wake also experiences the effects of the lateral forcing. There appears not to be an force which opposes this lateral thrust forcing, and hence the continued expansion in both the lateral and streamwise cases. As a final note, the relative independence from yaw angle for $\gamma > 10°$ simplifies any empirical equation for $\sigma_{y,2}$, such as will be required for the analytical model in Section 5.2.1, where the relationship between wake width and downstream distance will be investigated further.

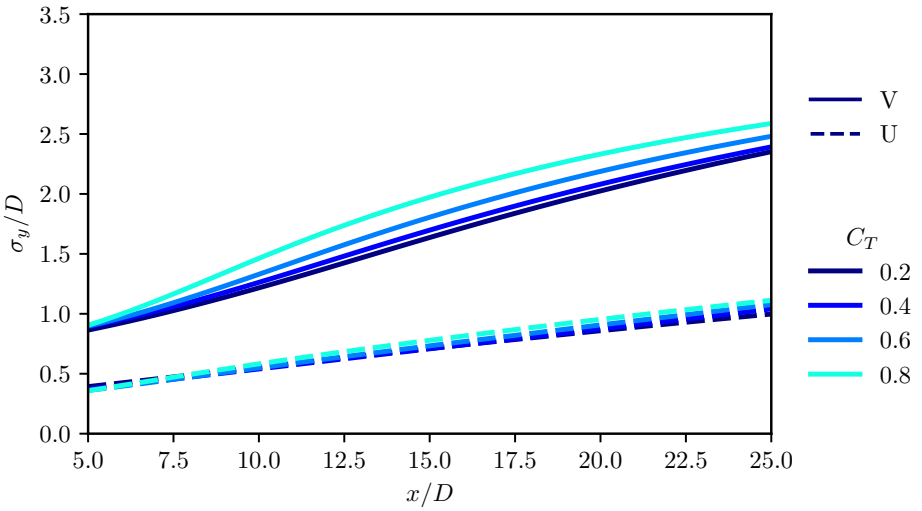


Figure 3.19: Variation of normalised streamwise (dashed line) and lateral (solid line) wake half-width with downstream distance for various thrust coefficients. $\gamma = 25°$ and $I_{\infty,h} = 6\%$.

Thrust coefficient appears to have some weak proportionality to both the lateral and streamwise wake widths, but from Figure 3.19, the effect is negligible. Notably, however, the lower the value of thrust coefficient, the more linear the development of the lateral wake width with downstream distance. This trend towards linearity is also one seen in the streamwise wake [122].

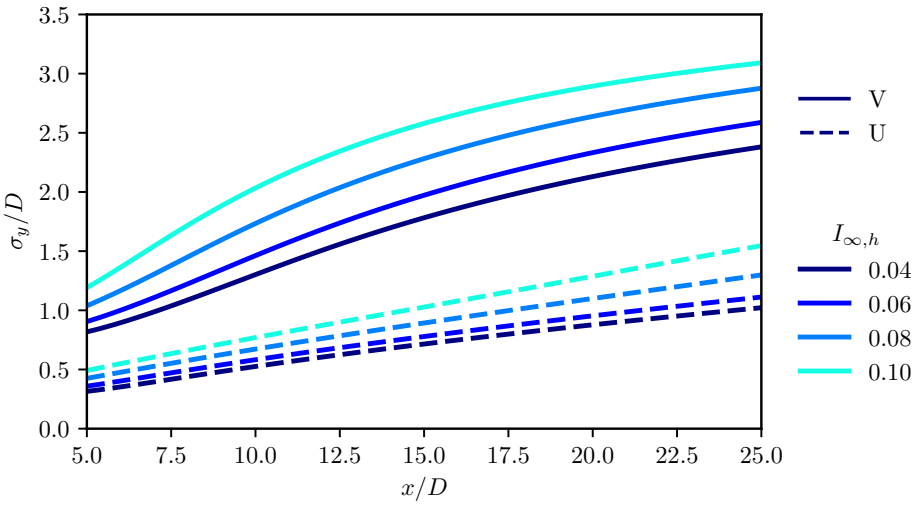


Figure 3.20: Variation of normalised streamwise (dashed line) and lateral (solid line) wake half-width with downstream distance for various hub-height total incoming turbulence intensities. $\gamma = 25°$ and $C_T = 0.8$.

Lastly, Figure 3.20 shows that turbulence intensity has a more significant effect on both the lateral and streamwise wake widths. The effect on the streamwise wake width is well known, and forms an integral part of analytical wake modelling, including the model this thesis is based upon [14]. Numerical [14, 27, 106] and lidar studies [148] have suggested a linear relationship between streamwise wake width and turbulence intensity, which is reflected in Figure 3.20, at least up to $x/D = 15$, after which it appears to diverge slightly from linear. The results presented in this section suggest that for yawed turbines (i.e. $\gamma \geq 10°$), the defining parameter for the determination of the lateral wake width is the turbulence intensity.

3.5 Wake centre

Besides the wake width, the other variable that defines the location of wake is the wake centre location. Unlike the wake width, however, for which there is a clear definition for all conditions, the definition of the wake centre location depends on the shape of the velocity profile. For yawed turbines, the velocity profile, which is shown in Figure 3.5, has a clear peak, which is taken as the wake centre location. For the unyawed turbine, however, the profile has a positive and a negative peak, with a zero-crossing in between. In this thesis, the wake centre of the unyawed profile is assumed to be at this zero-crossing point. The author envisions, however, that this could cause difficulty when attempting to define the wake centre location for profiles in the transition between unyawed ($\gamma = 0°$) and yawed ($\gamma = 10°$), and thus further investigation is required to conduce a more robust definition.

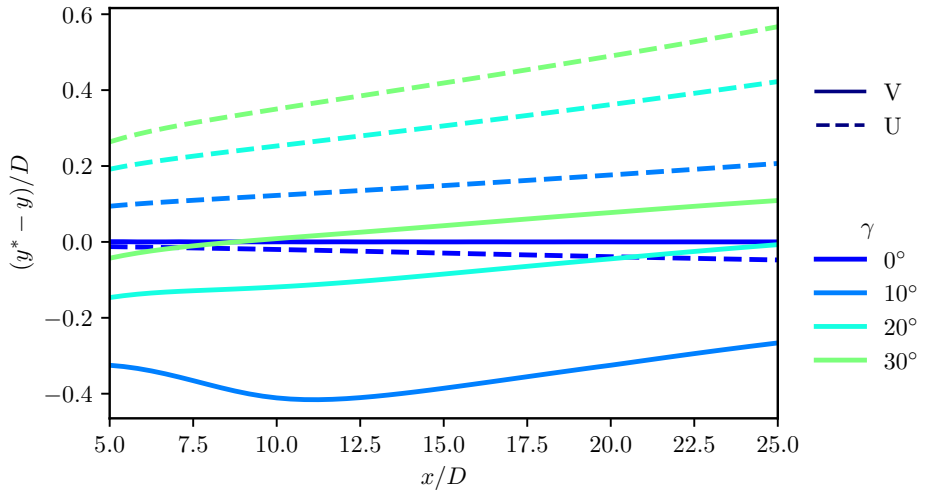


Figure 3.21: Variation of normalised streamwise (dashed line) and lateral (solid line) wake centre location with downstream distance for various yaw angles. $C_T = 0.8$ and $I_{\infty,h} = 6\%$.

As with the wake widths, there are notable differences between the wake centre locations of the streamwise and lateral wakes. Whilst increasing yaw angle appears to translate all the wake centres towards positive y, the lateral wake is offset towards negative y when transitioning from unyawed ($\gamma = 0°$) to yawed ($\gamma = 10°$). This can be linked to the behaviour seen in Figure 3.5, where the peak of the velocity profile is not centred at $y = 0$ for smaller yaw angles, but moves towards it with increasing yaw angle. Despite this, for both the streamwise and lateral wakes the wake centre has a positive slope with downstream distance for yawed turbines in the far wake. The exceptions are at $\gamma = 0°$, where both velocity components appear to exhibit the initial horizontal wake deflection described by Jacobs [70], a product of the interaction between wake rotation and wind shear; and at 10°, where before $10D$ the profile is determined by a combination of the flow divergence around the turbine and the lateral forcing and therefore lies within the difficult transition in the definition of wake centre.

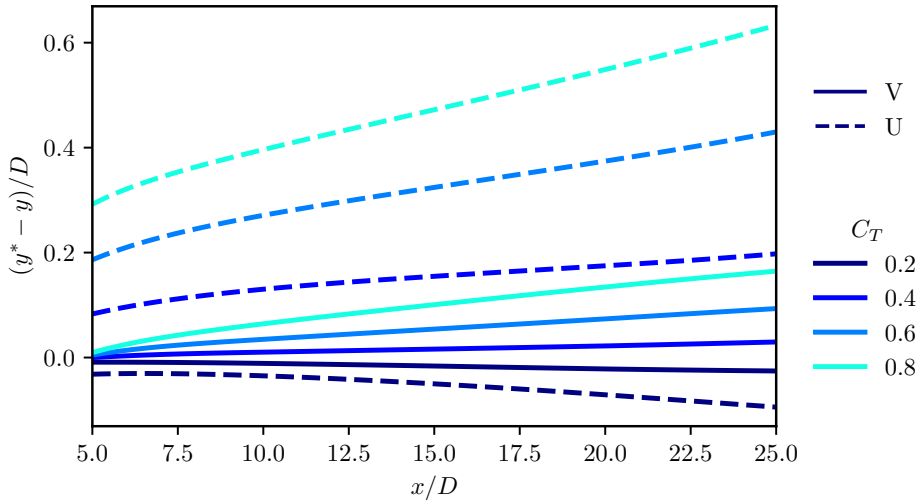


Figure 3.22: Variation of normalised streamwise (dashed line) and lateral (solid line) wake centre location with downstream distance for various thrust coefficients. $\gamma = 25°$ and $I_{\infty,h} = 6\%$.

There are also differences in the effect of thrust coefficient on the streamwise and lateral wake centres, most notably the magnitude of the deflection. Whilst C_T affects both the centre location at $x/D = 5$ and the slope of the location with downstream distance for the streamwise wake, the location at $x/D = 5$ is relatively unaffected for the lateral wake. The differences can at least be partially attributed to the absolute size of the thrust component, as at this yaw angle of $\gamma = 25°$, the streamwise thrust component is still significantly larger than the lateral component, thus the scaling is larger in absolute terms. An investigation of more yaw angles could confirm this but is outside of the scope of the current work.

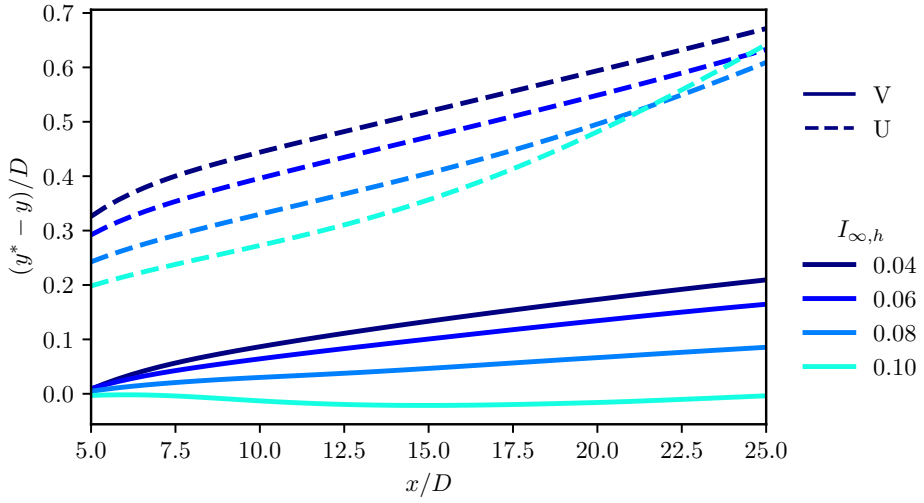


Figure 3.23: Variation of normalised streamwise (dashed line) and lateral (solid line) wake centre location with downstream distance for various hub-height total incoming turbulence intensities. $\gamma = 25°$ and $C_T = 0.8$.

Lastly, there is the effect of turbulence intensity. Much as has been observed previously

in this chapter, compared to thrust coefficient, an increase in turbulence intensity tends to decrease the deflection of the wake centre. However, it is also interesting to note that for high values of $I_{\infty,h}$, there can be some non-linearity introduced in the very far wake, i.e. $x/D > 15$ for $I_{\infty,h} = 0.04$ in Figure 3.23. One should note though that at these high turbulence intensities, the enhanced wake recovery means that the velocity profiles can be more difficult to extract wake centres from, and so this non-linearity may well be a matter of post-processing rather than flow physics.

CHAPTER 4
Derivation of Analytical Solution

As was mentioned in Section 1.3, Bastankhah et al. [14] developed an analytical solution for the streamwise velocity within a wind farm of arbitrary layout. This thesis seeks to extend that work, applying the same methods to develop an analytical solution for the lateral velocities within a wind farm, now with arbitrary layout and arbitrarily yawed wind turbines. To that end, the full derivation employed by Bastankhah et al. [14] is set out in Appendix B, including some steps which were glossed over in the original paper for the sake of brevity.

With yawed turbines, the lateral velocity becomes important as it primarily dictates the lateral deflection of the wake. Whilst the original method was capable of accounting for wake steering, this was only in so much as to be capable of accepting a varying wake centre location with downstream distance, not calculating the deflection itself. This chapter derives a means of calculating that deflection using the same logic as in Bastankhah et al. [14]. Some modifications to the solution for the streamwise velocity are also necessary, but these are addressed in Chapter 5.

4.1 Integral form of governing equations for turbine lateral wakes within a wind farm

Let us assume a wind farm with an arbitrary layout of n wind turbines (WT_1, WT_2, …, WT_i, …, WT_n) immersed in a turbulent boundary layer flow with a lateral velocity profile denoted by V_0. The position of WT_i is denoted by $\boldsymbol{X}_i = (x_i, y_i, z_i)$, where x, y, and z are the streamwise, spanwise, and vertical directions in the coordinate system, respectively. Turbines are labelled with respect to their streamwise positions such that $x_i \geq x_{i-1}$, where $i = \{2, 3, \ldots, n\}$. The RANS equation in the lateral direction at high Reynolds numbers (neglecting viscosity effects) is given by

$$U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + W\frac{\partial V}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial y} - \frac{\partial \overline{uv}}{\partial x} - \frac{\partial \overline{v^2}}{\partial y} - \frac{\partial \overline{vw}}{\partial z} + \sum_{i=1}^{n} f_{i,2}, \tag{4.1}$$

where U, V, and W are the time-averaged streamwise (x), lateral (y), and vertical (z) velocity components, respectively. Turbulent velocity fluctuations are represented by u, v, and w and the overbar denotes time averaging. Also, P is the time-averaged static pressure and ρ is the air density. The term $f_{i,2}$ represents the effect of the lateral component of the thrust force of WT_i on the lateral momentum.

Using the incoming boundary-layer profile $V_0(z)$, Equation 4.1 can be written as

$$U\frac{\partial (V - V_0)}{\partial x} + V\frac{\partial (V - V_0)}{\partial y} + W\frac{\partial (V - V_0)}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial y} - \frac{\partial \overline{uv}}{\partial x} - \frac{\partial \overline{v^2}}{\partial y} - \frac{\partial \overline{vw}}{\partial z} - W\frac{dV_0}{dz} + \sum_{i=1}^{n} f_{i,2}, \tag{4.2}$$

where $W\frac{dV_0}{dz}$ is the only additional term on the right-hand side because $V_0(z)$ and thus $\frac{dU_0}{dx} = \frac{dU_0}{dy} = 0$. From the continuity equation, we know that

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} = 0. \tag{4.3}$$

Multiplying Equation 4.3 by $(V - V_0)$ and adding the product to the left-hand side of Equation 4.2 yields

$$\begin{aligned}\frac{\partial U}{\partial x}(V - V_0) + \frac{\partial V}{\partial y}(V - V_0) + \frac{\partial W}{\partial z}(V - V_0) + U\frac{\partial (V - V_0)}{\partial x} + V\frac{\partial (V - V_0)}{\partial y} + W\frac{\partial (V - V_0)}{\partial z} \\ = -\frac{1}{\rho}\frac{\partial P}{\partial y} - \frac{\partial \overline{uv}}{\partial x} - \frac{\partial \overline{v^2}}{\partial y} - \frac{\partial \overline{vw}}{\partial z} - W\frac{dV_0}{dz} + \sum_{i=1}^{n} f_{i,2}.\end{aligned} \tag{4.4}$$

The terms on the left-hand side can then be combined using the product rule (e.g. $U\frac{\partial (V-V_0)}{\partial x} + (V - V_0)\frac{\partial U}{\partial x} = \frac{\partial U(V-V_0)}{\partial x}$) to give

$$\begin{aligned}\frac{\partial U(V - V_0)}{\partial x} + \frac{\partial V(V - V_0)}{\partial y} + \frac{\partial W(V - V_0)}{\partial z} \\ = -\frac{1}{\rho}\frac{\partial P}{\partial y} - \frac{\partial \overline{uv}}{\partial x} - \frac{\partial \overline{v^2}}{\partial y} - \frac{\partial \overline{vw}}{\partial z} - W\frac{dV_0}{dz} + \sum_{i=1}^{n} f_{i,2}.\end{aligned} \tag{4.5}$$

Next, Equation 4.5 is integrated from x_a to x_b with respect to x, from y_a to y_b with respect to y, and from z_a to z_b with respect to z, where $x_a \ll x_n < x_b$, $y_a \ll y_n \ll y_b$ and $0 \ll z_a < z_n \ll z_b$. Note that $z_a \gg 0$ to ensure that the assumption of negligible viscous forces is valid. The value of z_a can be equal to zero if the Reynolds shear stresses in Equation 4.5 are replaced with the total shear stresses (i.e. the sum of turbulent and viscous shear stresses). Without loss of generality, we assume that the integration box includes WT_n and an arbitrary number of upwind turbines as shown in Figure 4.1.

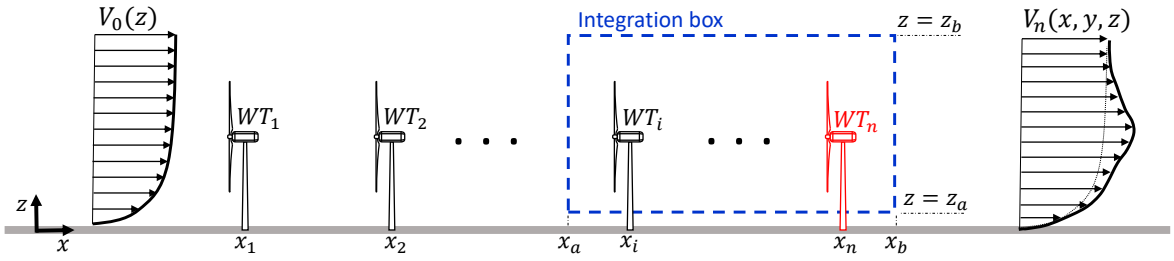


Figure 4.1: Schematic of a wind farm with an arbitrary layout and arbitrary yaw angles consisting of n wind turbines (WT_1, WT_2, ..., WT_n) immersed in a turbulent boundary-layer flow. The momentum equation Equation 4.5 is integrated over the shown box. The integration is performed with and without WT_n, shown in the figure in red. Figure adapted from Bastankhah et al. [14].

Integrating Equation 4.5 yields

$$\sum_{i\in B}\frac{T_i\sin\gamma_i}{\rho}=\int U_n(V_n-V_0)\Big|_{x_a}^{x_b}\,\mathrm{d}A+\frac{1}{\rho}\int_{z_a}^{z_b}\int_{x_a}^{x_b}P_n\Big|_{y_a}^{y_b}\,\mathrm{d}x\,\mathrm{d}z+\int\overline{uv}_n\Big|_{x_a}^{x_b}\,\mathrm{d}A$$
$$+\int_{y_a}^{y_b}\int_{x_a}^{x_b}\overline{vw}_n\Big|_{z_a}^{z_b}\,\mathrm{d}x\,\mathrm{d}y+\int\frac{dV_0}{dz}W_n\,\mathrm{d}V,\quad(4.6)$$

where γ_i is the yaw angle of WT_i, and i is a member of set B if WT_i is inside the integration box. Also $\mathrm{d}A$ is $\mathrm{d}y\,\mathrm{d}z$ and $\mathrm{d}V$ is $\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$. In Equation 4.6 and hereafter, any velocity or pressure term with a subscript i denotes the value of the given variable in the presence of WT_1, WT_2, …, WT_i. Now, we perform the same integration once more but this time in the absence of WT_n. This leads to

$$\sum_{i\in B'}\frac{T_i\sin\gamma_i}{\rho}=\int U_{n-1}(V_{n-1}-V_0)\Big|_{x_a}^{x_b}\,\mathrm{d}A+\frac{1}{\rho}\int_{z_a}^{z_b}\int_{x_a}^{x_b}P_{n-1}\Big|_{y_a}^{y_b}\,\mathrm{d}x\,\mathrm{d}z+\int\overline{uv}_{n-1}\Big|_{x_a}^{x_b}\,\mathrm{d}A$$
$$+\int_{y_a}^{y_b}\int_{x_a}^{x_b}\overline{vw}_{n-1}\Big|_{z_a}^{z_b}\,\mathrm{d}x\,\mathrm{d}y+\int\frac{dV_0}{dz}W_{n-1}\,\mathrm{d}V,\quad(4.7)$$

where set B' is equal to set B excluding n (i.e. $B\setminus\{n\}=\{i:i\in B\text{ and }i\notin\{n\}\}$). As $x_a\ll x_n$, surface integrals at $x=x_a$ provide the same results in Equation 4.6 and Equation 4.7. By subtracting Equation 4.7 from Equation 4.6, we obtain

$$\underbrace{\frac{T_n\sin\gamma_n}{\rho}}_{\text{Lateral thrust}}=\underbrace{\left[\int U_n(V_n-V_0)\,\mathrm{d}A-\int U_{n-1}(V_{n-1}-V_0)\,\mathrm{d}A\right]}_{\text{Lateral momentum surplus}}$$
$$\underbrace{+\frac{1}{\rho}\int_{z_a}^{z_b}\int_{x_a}^{x_b}(P_n-P_{n-1})\Big|_{y_b}\,\mathrm{d}x\,\mathrm{d}z}_{\text{Pressure}}+\underbrace{\int(\overline{uv}_n-\overline{uv}_{n-1})\,\mathrm{d}A}_{\text{Lateral Reynolds shear stress}}$$
$$\underbrace{+\int_{y_a}^{y_b}\int_{x_a}^{x_b}(\overline{vw}_n-\overline{vw}_{n-1})\Big|_{z_b}\,\mathrm{d}x\,\mathrm{d}y}_{\text{Vertical Reynolds shear stress}}+\underbrace{\int\frac{dV_0}{dz}(W_n-W_{n-1})\,\mathrm{d}V}_{\text{Mean flow shear}},\quad(4.8)$$

where $\mathrm{d}A$ in Equation 4.8 is $\mathrm{d}y\,\mathrm{d}z$ at $x=x_b$. Note that some terms, for example the two other convective terms and the Reynolds normal stress, drop out of consideration in the integration of Equation 4.5. The reasons for this are fully outlined in Appendix C, along with the full integration of Equation 4.5.

4.2 General solution

We seek a solution for V_n in the following 'conservation of lateral momentum surplus' equation,

$$\rho\int U_n(V_n-V_0)\,\mathrm{d}A-\rho\int U_{n-1}(V_{n-1}-V_0)\,\mathrm{d}A\approx T_n\sin\gamma_n.\quad(4.9)$$

Using the definition for the streamwise velocity deficit from Bastankhah et al. [14, (4.2)],

$$\Delta U_n(\boldsymbol{X}) = U_{n-1}(\boldsymbol{X}) - U_n(\boldsymbol{X}), \tag{4.10}$$

one could similarly define the lateral velocity surplus at a given position $\boldsymbol{X} = (x, y, z)$ downwind of WT_n, as the difference of the lateral velocity in absence and presence of WT_n at $\boldsymbol{X}$; i.e.

$$\Delta V_n(\boldsymbol{X}) = V_n(\boldsymbol{X}) - V_{n-1}(\boldsymbol{X}). \tag{4.11}$$

Bastankhah et al. [14] presented results from LES that demonstrated that, with the definition of velocity deficit in Equation 4.10, the wake of a turbine within a wind farm exhibits a good degree of self-similarity, akin to a standalone turbine. Figure 4.2 demonstrates the degree of self-similarity of the lateral velocity surplus according to Equation 4.11 for: (a) a standalone wind turbine, (b) the middle turbine in the last row of the aligned wind farm (i.e. WT_{15}), and (c) the middle turbine in the last row of the slanted wind farm (i.e. WT_{15}). Whilst the self-similarity is not as marked as for the streamwise velocity deficit, as explained in Section 3.2, it is deemed to be sufficient for use in the subsequent derivation.

The fact that both $(U_{n-1} - U_n)$ and $(V_n - V_{n-1})$ are self-similar means they can be written as

$$U_{n-1} - U_n = C_{n,1}(x) f_{n,1}(\boldsymbol{X}) \tag{4.12}$$

and

$$V_n - V_{n-1} = C_{n,2}(x) f_{n,2}(\boldsymbol{X}). \tag{4.13}$$

where $f_{n,1}$ and $f_{n,2}$ are self-similar functions describing the streamwise and lateral velocity deficits in space, respectively, with $C_{n,1}$ and $C_{n,2}$ as the peak deficit and surplus, respectively.

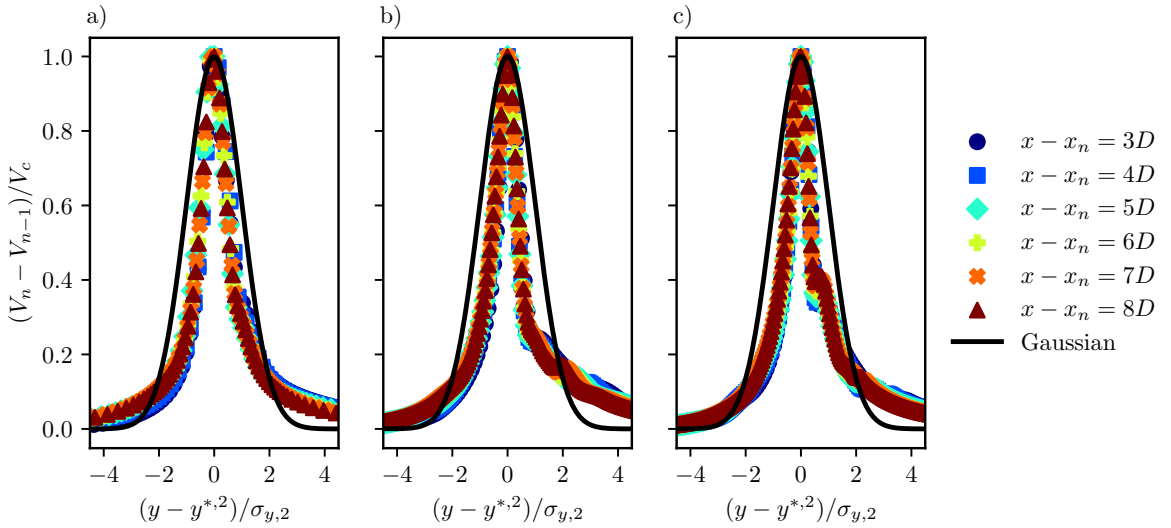


Figure 4.2: Self-similar lateral distributions of normalised lateral velocity surplus at several positions downwind of WT_n for (a) a single turbine ($n = 1$) yawed at 25°, (b) the turbine in the fifth row (middle column) of the aligned wind farm ($n = 15$), where all turbines are yawed at 25° and (c) the turbine in the fifth row (middle column) of the slanted wind farm ($n = 15$), where all turbines are yawed at 25°.

Shifting the index n in Equation 4.12 to $n-1, n-2, \ldots, 1$ leads to a set of equations as follows:

$$\begin{aligned} U_{n-2} - U_{n-1} &= C_{n-1,1} f_{n-1,1}, \\ U_{n-3} - U_{n-2} &= C_{n-2,1} f_{n-2,1}, \\ &\vdots \\ U_0 - U_1 &= C_{1,1} f_{1,1}. \end{aligned} \tag{4.14}$$

Adding Equation 4.12 and Equation 4.14 together then yields

$$\begin{aligned} (U_0 - U_1) + (U_1 - U_2) + \ldots + (U_{n-2} - U_{n-1}) + (U_{n-1} - U_n) \\ = C_{1,1} f_{1,1} + C_{2,1} f_{2,1} + \ldots + C_{n-2,1} f_{n-2,1} + C_{n-1,1} f_{n-1,1} + C_{n,1} f_{n,1}, \end{aligned}$$

$$U_0 - U_n = \sum_{i=1}^{n} C_{i,1} f_{i,1} = \sum_{i=1}^{n-1} C_{i,1} f_{i,1} + C_{n,1} f_{n,1}, \tag{4.15}$$

which can be rearranged to give

$$U_n = U_0 - \sum_{i=1}^{n} C_{i,1} f_{i,1} = U_0 - \sum_{i=1}^{n-1} C_{i,1} f_{i,1} - C_{n,1} f_{n,1}. \tag{4.16}$$

Similar expressions can be derived for the lateral velocity, i.e.

$$V_n - V_0 = \sum_{i=1}^{n} C_{i,2} f_{i,2} = \sum_{i=1}^{n-1} C_{i,2} f_{i,2} + C_{n,2} f_{n,2}. \tag{4.17}$$

Substituting $n-1$ for n in Equations 4.16 and 4.17, one can also state that

$$U_{n-1} = U_0 - \sum_{i=1}^{n-1} C_{i,1} f_{i,1} = U_0 - \sum_{i=1}^{n} C_{i,1} f_{i,1} + C_{n,1} f_{n,1}, \tag{4.18}$$

and

$$V_{n-1} - V_0 = \sum_{i=1}^{n-1} C_{i,2} f_{i,2} = \sum_{i=1}^{n} C_{i,2} f_{i,2} - C_{n,2} f_{n,2}. \tag{4.19}$$

Equations 4.16 to 4.19 can then be substituted into Equation 4.9 to give

$$\begin{aligned} \int \left(U_0 - \sum_{i=1}^{n-1} C_{i,1} f_{i,1} - C_{n,1} f_{n,1} \right) \left(\sum_{i=1}^{n-1} C_{i,2} f_{i,2} + C_{n,2} f_{n,2} \right) \, \mathrm{d}A \\ - \int \left(U_0 - \sum_{i=1}^{n-1} C_{i,1} f_{i,1} \right) \left(\sum_{i=1}^{n-1} C_{i,2} f_{i,2} \right) \, \mathrm{d}A \approx \frac{T_n \sin \gamma_n}{\rho}. \end{aligned} \tag{4.20}$$

Expanding this to

$$\int U_0 \sum_{i=1}^{n-1} C_{i,2} f_{i,2} + U_0 C_{n,2} f_{n,2} - \sum_{i=1}^{n-1} C_{i,1} f_{i,1} \sum_{i=1}^{n-1} C_{i,2} f_{i,2} - C_{n,2} f_{n,2} \sum_{i=1}^{n-1} C_{i,1} f_{i,1} - C_{n,1} f_{n,1} \sum_{i=1}^{n-1} C_{i,2} f_{i,2} - C_{n,1} f_{n,1} C_{n,2} f_{n,2} \, \mathrm{d}A - \int U_0 \sum_{i=1}^{n-1} C_{i,2} f_{i,2} + \sum_{i=1}^{n-1} C_{i,1} f_{i,1} \sum_{i=1}^{n-1} C_{i,2} f_{i,2} \, \mathrm{d}A \approx \frac{T_n \sin \gamma_n}{\rho}, \quad (4.21)$$

and simplifying it to

$$\int U_0 C_{n,2} f_{n,2} - C_{n,1} f_{n,1} \sum_{i=1}^{n-1} C_{i,2} f_{i,2} - C_{n,2} f_{n,2} \sum_{i=1}^{n-1} C_{i,1} f_{i,1} - C_{n,1} f_{n,1} C_{n,2} f_{n,2} \, \mathrm{d}A = \frac{T_n \sin \gamma_n}{\rho}, \quad (4.22)$$

before factorising it as

$$\int U_0 C_{n,2} f_{n,2} + \sum_{i=1}^{n-1} C_{i,1} f_{i,1} \sum_{i=1}^{n-1} C_{i,2} f_{i,2} - \left(\sum_{i=1}^{n-1} C_{i,1} f_{i,1} + C_{n,1} f_{n,1} \right) \left(\sum_{i=1}^{n-1} C_{i,2} f_{i,2} + C_{n,2} f_{n,2} \right) = \frac{T_n \sin \gamma_n}{\rho}, \quad (4.23)$$

leads, via the additional assumption that the equality is exact and not approximate, to the general solution

$$\boxed{\int U_0 C_{n,2} f_{n,2} + \sum_{i=1}^{n-1} C_{i,1} f_{i,1} \sum_{i=1}^{n-1} C_{i,2} f_{i,2} - \sum_{i=1}^{n} C_{i,1} f_{i,1} \sum_{i=1}^{n} C_{i,2} f_{i,2} \, \mathrm{d}A = \frac{T_n \sin \gamma_n}{\rho}}. \quad (4.24)$$

By rearranging this into an alternative form, namely

$$\int \underbrace{U_0 C_{n,2} f_{n,2}}_{\text{Term I}} \underbrace{- C_{n,1} f_{n,1} \sum_{i=1}^{n-1} Ci, 2 f_{i,2} - C_{n,2} f_{n,2} \sum_{i=1}^{n-1} C_{i,1} f_{i,1}}_{\text{Term II}} \underbrace{- C_{n,1} f_{n,1} C_{n,2} f_{n,2}}_{\text{Term III}} \, \mathrm{d}A = \frac{T_n \sin \gamma_n}{\rho}, \quad (4.25)$$

and multiplying out the terms in the original form of the streamwise solution (Equation B.29) to express it in a similar form, i.e.

$$\int \underbrace{U_0 C_{n,1} f_{n,1}}_{\text{Term I}} \underbrace{- 2 \sum_{i=1}^{n-1} C_{i,1} f_{i,1}}_{\text{Term II}} \underbrace{- (C_{n,1} f_{n,1})^2}_{\text{Term III}} \, \mathrm{d}A = \frac{T_n \cos \gamma_n}{\rho}, \quad (4.26)$$

one can make direct comparisons. For example, Term I of both equations is of the form $\int U_0 C_{n,*} f_{n,*}$, where $* = 1$ or 2 for the streamwise or lateral Gaussian terms, respectively. This term is the isolated turbine term, which is to say it represents the effect of turbine n itself on its wake, and ignores any interaction effects introduced by the presence of other turbines. These interaction effects are carried by the convection velocities U_n and U_{n-1} in Equation 4.9. This can be demonstrated by assuming that, rather than being convected at $\boldsymbol{U}_n$, the wake is convected at $U_0(z)$, the freestream velocity. This first-order approximation yields a general solution of the form

$$\int U_0 C_{n,2} f_{n,2} \, \mathrm{d}A = \frac{T_n \sin \gamma_n}{\rho}, \tag{4.27}$$

i.e. only term I in Equation 4.25. The full derivation of Equation 4.27 is presented in Appendix D. Whilst term II in Equation 4.25 must remain as two separate summations because of the presence of both $C_{n,1} f_{n,1}$ and $C_{n,2} f_{n,2}$, the streamwise solution contains only $C_{n,1} f_{n,1}$, so can aggregate these into a single sum with a factor 2 outside. The same aggregation is true of term III in each equation, where the product in the lateral solution becomes an exponent in the streamwise.

These latter two terms point to a fundamental difference in the nature of the streamwise and lateral problems. Unlike its streamwise counterpart, the left-hand side of the 'conservation of lateral momentum surplus' equation, Equation 4.9, contains both the lateral and streamwise velocities. This means that this is a coupled equation, which makes it more difficult to solve. The streamwise velocity appears here and in the equation for the vertical velocity because it is the dominant term in the wake convection velocity, $\boldsymbol{U}_n$. As a result of this coupling, any specific solution for the lateral velocity, V_n, or vertical velocity, W_n, will also require information regarding the streamwise velocity, U_n.

In the case of the streamwise velocity the convection velocity was the same as the velocity one is attempting to solve for, that is to say in the 'conservation of momentum deficit' equation, Equation B.21, there is a U_n (or U_{n-1}) term both inside the velocity deficit term, and outside, as the convection velocity. As a result, the solution for C_n becomes a quadratic one (see Equation B.66). The lateral problem involves U and V, however, U can be found from the streamwise solution, and thus the problem becomes a linear one, as will be shown in the following chapter.

4.3 Specific solution

If we assume that $C_{i,1}$, where $i = 1, 2, \ldots, n$, is known from the streamwise solution [14]

$$\frac{C_{n,1}}{U_h} = \left(1 - \sum_{i=1}^{n-1} \frac{C_{i,1}}{U_h} \lambda_{i,1}^{n,1}\right) \left(1 - \sqrt{1 - \frac{c_{t,n} \left(\frac{\langle U_{n-1} \rangle_{(n,x_n)}}{U_h}\right)^2}{8 \left(\sigma_{n,1}/D\right)^2 \left(U_0 - \sum_{i=1}^{n-1} \lambda_{i,1}^{n,1} \frac{C_{i,1}}{U_h}\right)^2}}\right), \tag{4.28}$$

where $\lambda_{i,1}^{n,1}$ is a dimensionless coefficient that describes the contribution of WT_i to C_n and is defined as

$$\lambda_{i,1}^{n,1} = \frac{2\sigma_{i,1}^2}{\sigma_{n,1}^2 + \sigma_{i,1}^2} \exp\left(-\frac{(y_n - y_i)^2}{2(\sigma_{n,1}^2 + \sigma_{i,1}^2)}\right) \exp\left(-\frac{(z_n - z_i)^2}{2(\sigma_{n,1}^2 + \sigma_{i,1}^2)}\right), \tag{4.29}$$

and $c_{t,n}$ is the thrust coefficient of WT_n and is given as

$$c_{t,n} = \frac{8T_n}{\pi \rho D^2 \langle U_{n-1} \rangle_{(n,x_n)}^2}, \tag{4.30}$$

where D is the rotor diameter and $\langle\rangle_{(i,x_j)}$ denotes spatial averaging over the frontal projected area of WT_i at $x = x_j$.

Starting with Equation 4.22, rearranging terms gives

$$\int U_0 C_{n,2} f_{n,2} - C_{n,2} f_{n,2} \sum_{i=1}^{n-1} C_{i,1} f_{i,1} - C_{n,1} f_{n,1} C_{n,2} f_{n,2} \, \mathrm{d}A = \frac{T_n \sin \gamma_n}{\rho} + \int C_{n,1} f_{n,1} \sum_{i=1}^{n-1} C_{i,2} f_{i,2} \, \mathrm{d}A, \quad (4.31)$$

which can be factorised for the unknown $C_{n,2}$ as

$$C_{n,2} \left(\int U_0 f_{n,2} - f_{n,2} \sum_{i=1}^{n-1} C_{i,1} f_{i,1} - C_{n,1} f_{n,1} f_{n,2} \, \mathrm{d}A \right) = \frac{T_n \sin \gamma_n}{\rho} + \int C_{n,1} f_{n,1} \sum_{i=1}^{n-1} C_{i,2} f_{i,2} \, \mathrm{d}A. \quad (4.32)$$

Separating the integrals into

$$C_{n,2} \left(\int U_0 f_{n,2} \, \mathrm{d}A - \int f_{n,2} \sum_{i=1}^{n-1} C_{i,1} f_{i,1} \, \mathrm{d}A - \int C_{n,1} f_{n,1} f_{n,2} \, \mathrm{d}A \right) = \frac{T_n \sin \gamma_n}{\rho} + \int C_{n,1} f_{n,1} \sum_{i=1}^{n-1} C_{i,2} f_{i,2} \, \mathrm{d}A, \quad (4.33)$$

then allows the expression to be simplified to

$$C_{n,2} \left(U_0 \int f_{n,2} \, \mathrm{d}A - \sum_{i=1}^{n-1} C_{i,1} \int f_{i,1} f_{n,2} \, \mathrm{d}A - C_{n,1} \int f_{n,1} f_{n,2} \, \mathrm{d}A \right) = \frac{T_n \sin \gamma_n}{\rho} + C_{n,1} \sum_{i=1}^{n-1} C_{i,2} \int f_{n,1} f_{i,2} \, \mathrm{d}A. \quad (4.34)$$

Assuming both the normalised streamwise velocity deficit and the normalised lateral velocity profiles can be described by a Gaussian, i.e.

$$f_{i,\alpha} = \exp\left(-\frac{(y - y_{i,\alpha})^2}{2\sigma_{i,\alpha}^2} \right) \exp\left(-\frac{(z - z_{i,\alpha})^2}{2\sigma_{i,\alpha}^2} \right), \quad (4.35)$$

then the following integrals can be defined

$$\begin{aligned} \int f_{i,\alpha} &= 2\pi\sigma_{i,\alpha}^2 \\ \int f_{i,\alpha} f_{j,\beta} &= \pi \sigma_{i,\alpha}^2 \lambda_{j,\beta}^{i,\alpha} \end{aligned} \quad (4.36)$$

where

$$\boxed{\lambda_{j,\beta}^{i,\alpha} = \frac{2\sigma_{j,\beta}^2}{\sigma_{i,\alpha}^2 + \sigma_{j,\beta}^2} \exp\left(-\frac{(y_{i,\alpha} - y_{j,\beta})^2}{2(\sigma_{i,\alpha}^2 + \sigma_{j,\beta}^2)} \right) \exp\left(-\frac{(z_{i,\alpha} - z_{j,\beta})^2}{2(\sigma_{i,\alpha}^2 + \sigma_{j,\beta}^2)} \right)}. \quad (4.37)$$

Substituting Equation 4.36 into Equation 4.34 yields

$$C_{n,2}\left(2\pi U_0\sigma_{n,2}^2 - \pi\sigma_{n,2}^2\sum_{i=1}^{n-1} C_{i,1}\lambda_{i,1}^{n,2} - \pi\sigma_{n,2}^2 C_{n,1}\lambda_{n,1}^{n,2}\right) = \frac{T_n\sin\gamma_n}{\rho} + \pi\sigma_{n,1}^2 C_{n,1}\sum_{i=1}^{n-1} C_{i,2}\lambda_{i,2}^{n,1}. \quad (4.38)$$

This can be simplified by noting that $\sum_{i=1}^{n-1} C_{i,1}\lambda_{i,1}^{n,2} + C_{n,1}\lambda_{n,1}^{n,2} = \sum_{i=1}^{n} C_{i,1}\lambda_{i,1}^{n,2}$. Combining these terms and dividing through by π leads to

$$C_{n,2}\left(2U_0\sigma_{n,2}^2 - \sigma_{n,2}^2\sum_{i=1}^{n} C_{i,1}\lambda_{i,1}^{n,2}\right) = \frac{T_n\sin\gamma_n}{\pi\rho} + \sigma_{n,1}^2 C_{n,1}\sum_{i=1}^{n-1} C_{i,2}\lambda_{i,2}^{n,1}, \quad (4.39)$$

which can be rearranged for $C_{n,2}$ to yield the specific Gaussian solution

$$\boxed{C_{n,2} = \frac{T_n\sin\gamma_n + \pi\rho\sigma_{n,1}^2 C_{n,1}\sum_{i=1}^{n-1} C_{i,2}\lambda_{i,2}^{n,1}}{\pi\rho\sigma_{n,2}^2\left(2U_h - \sum_{i=1}^{n} C_{i,1}\lambda_{i,1}^{n,2}\right)}}, \quad (4.40)$$

as with the streamwise solution assuming $U_0 \approx U_h$ [14].

As with the general solution, one can draw parallels between the specific solutions for the streamwise (Equation B.71) and the lateral velocity. Besides the fact that both include the thrust forcing normalised by $\pi\rho\sigma_{n,*}^2$, where $* = 1$ or 2 for the streamwise and lateral solutions, respectively, they both also include a form of deficit, $U_h - \sum_{i=1}^{n-1} C_{i,1}\lambda_{i,1}^{n,1}$ in the streamwise case, and $2U_h - \sum_{i=1}^{n} C_{i,1}\lambda_{i,1}^{n,2}$ in the lateral.

Moreover, the λ term has evolved to incorporate the effect of interaction between components wakes. In Bastankhah et al. [14], the dimensionless coefficient λ_{ni}, as it is called, 'quantifies the contribution of WT_i on the value of C_n [$C_{n,1}$ in this thesis]'. Equation 4.37 extends this definition to quantify the contribution of the component wake β (where $\beta = 1$ implies the streamwise wake and $\beta = 2$ the lateral wake) of WT_j to the value of $C_{i,\alpha}$. This definition also includes the deflection of each component wake.

One notable extraction from the solution for the lateral wake is that of secondary wake steering. If the contribution of WT_n to $C_{n,2}$ is set to zero, via the lateral forcing term $T_n\sin\gamma_n$, i.e. by unyawing the turbine making $\gamma_n = 0$, then the solution reduces to

$$C_{n,2} = \left(\frac{\sigma_{n,1}}{\sigma_{n,2}}\right)^2 \frac{C_{n,1}\sum_{i=1}^{n-1} C_{i,2}\lambda_{i,2}^{n,1}}{2U_h - \sum_{i=1}^{n} C_{i,1}\lambda_{i,1}^{n,2}}. \quad (4.41)$$

That is to say, that even when WT_n is unyawed, provided there are turbines operating upstream of it which make some contribution to $C_{n,2}$ via $\lambda_{i,1}^{n,2}$, then $C_{n,2}$ is non-zero, i.e. the lateral velocities in the wake of WT_n are greater than they would be were WT_n absent. This appears to be mathematical proof of the concept of 'secondary wake steering' developed by Fleming et al. [43]. The decoupled solution presented in Section 4.2, where the convection velocity was assumed to be freestream, leads to a specific solution of the form

$$\frac{C_{n,2}}{U_h} = \frac{T_n\sin\gamma_n}{2\pi\rho\sigma_{n,2}^2 U_h^2}, \quad (4.42)$$

which represents Equation 4.40, but with all the interaction terms removed. Appendix D gives the full derivation of Equation 4.42.

CHAPTER 5
Model Implementation

Typically in the literature, analytical derivations are performed that arrive at expressions such as Equation 4.40, and then computational results are presented which demonstrate the abilities of the expression to describe the corresponding physics. However, there is a tendency to omit the details of the means by which the expression is implemented. Whilst this normally has little bearing on the results themselves, it can make replication of the results and cross-validation by intrigued readers very difficult. For this reason, this chapter will cover the structure of the Python scripts used to implement the model described in Chapter 4. The full function as it was at the time of publication is also included in Appendix F, and an up-to-date version can be found at `https://github.com/NilsGaukroger/Analytical-solution-for-the-cumulative-wake-of-yawed-wind-turbines/blob/main/lateralSolution.py`.

Besides the mechanics of programming the equations from Chapter 4, implementation of the model also involves several other aspects. This is evident from the structure of applications for analytical wind farm modelling such as PyWake [38] and FLORIS [107], where modules for wake velocities and superposition methods are but one of many modules, including wind farm layout, inflow, added turbulence intensity, wake recovery rate, etc. Whilst these other modules are outside of the scope of this thesis, they are necessary in order to obtain the results presented in Chapter 6, and so an explanation of the simple methods used for them is given in this chapter.

5.1 Setup

The setup of the model involves three steps: (i) defining the wind farm layout, (ii) defining the flow domain, and (iii) initialising the velocity fields with an inflow model.

The wind farm layout can be defined using the x-, y-, and z-coordinates of the turbine hubs. At this stage, the yaw angles of each turbine are also specified. As the algorithm computes the contributions of each turbine from upstream to downstream, it is then necessary to sort these turbine positions and yaw angles by increasing x-coordinate. The flow domain is simply defined as the x-, y-, and z-coordinates of the nodes in a 3D Cartesian grid. Whilst in Listing F.1 this is given as an arbitrary domain, when comparing to CFD results such as in Chapter 6, this is set using the dimensions and discretisations of the CFD domain. Finally, through Equations 4.15 and 4.17, the model is designed to sum the contributions of each turbine, calculated using Equation 4.40, and subtract the total from the corresponding inflow, U_0 or V_0. Therefore, it is necessary to specify an inflow velocity field. This is then used to initialise the velocity field of the solution, from which the contributions are subtracted in turn. The inflow could take a vast number of forms, for example Appendix F uses an adiabatic log law with no veer, whereas Chapter 6 extracts the inflow profile from the corresponding CFD simulation. Whatever the form, once this velocity field has been initialised, the solution can be run.

5.2 Solution

For each turbine, the solution process itself involves four steps: (i) thrust force lookup using computed disc velocity, (ii) calculation of the inflow turbulence intensity at each turbine, accounting for added turbulence intensity where necessary, (iii) determination of the wake recovery rate, and (iv) calculation of the velocities and deflection.

The thrust force of the NREL 5 MW turbine used in this thesis is computed using a lookup table. The lookup table contains the values of thrust force corresponding to the disc velocity as shown in Figure 5.1(a). As the model assumes information only travels downstream, and computes from upstream to downstream, the disc velocity can be extracted from the current velocity field. In Listing F.2 this is done by extracting the yawed plane containing the actuator disc, masking the points outside of the rotor and taking the mean of each velocity component. These components can then be amalgamated into the mean velocity vector, and the dot product of this vector with the disc normal vector yields the disc velocity. This disc velocity is then used to interpolate for a value of C_T, and C_P if required, which can be converted to thrust and power in the usual fashion.

Where the thrust coefficient is important for determining the magnitude of the velocity deficits and the initial streamwise wake width, the turbulence intensity is the determining factor for the wake recovery rate, k, which determines the wake width with downstream distance according to Equation 5.6. The magnitude of incoming turbulence intensity at each turbine is taken to be

$$I_{in} = \sqrt{I_\infty^2 + \Delta I_{in}^2}, \tag{5.1}$$

where I_∞ is the freestream turbulence intensity, and ΔI_{in} is the turbulence intensity added by upwind turbines. Several engineering models exist for calculating the added turbulence intensity from individual turbines [31, 115]. The solution for the streamwise velocity in Bastankhah et al. [14], for instance, uses a version of the model proposed by Crespo and Hernández [31] (hereafter referred to as the Crespo model) adjusted to better fit the results from LES. The adjusted

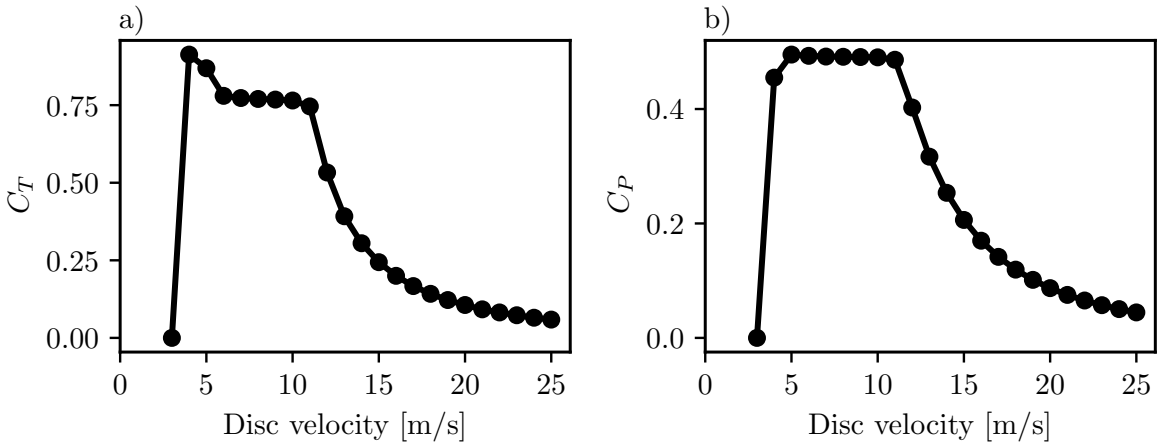


Figure 5.1: a) Thrust coefficient and b) power coefficient of the NREL 5 MW actuator disc model from PyWakeEllipSys [39].

model predicts the value of ΔI_{in} for WT_n due to upwind turbine WT_i, where $i < n$, as

$$\Delta I_{in,i} = 0.66 a_i^{0.83} I_\infty^{0.03} \left[(x_n - x_i)/D\right]^{-0.32} . \tag{5.2}$$

Within wind farms, the contribution of individual turbines must be combined in some way, much like the superposition models discussed in Section 1.3. A plethora of models also exist for this purpose [4, 49, 69, 106]. Bastankhah et al. [14] uses the geometric method suggested by Niayifar and Porté-Agel [106]. This method combines the overlap area of the wake of the upstream turbine with the downstream turbine's rotor, A_w, with Equation 5.2 to determine the added turbulence intensity at WT_n as the maximum of the added turbulence intensity due to each turbine, $I_{in,i}$, such that

$$\Delta I_{in} = \max \left(\frac{4 A_w}{\pi D^2} \Delta I_{in,i} \right) . \tag{5.3}$$

This approach has met with some criticism, for example Qian and Ishihara [114] points out that it assumes a top-hat distribution of turbulence intensity, despite the distribution having been shown to be dual-Gaussian [113, 93]. These are valid criticisms, but, as with several other decisions in this thesis, this represents a trade-off between fidelity and simplicity where the preference is for a simpler, faster model with acceptable sacrifices in terms of accuracy.

Given this thesis is intended as an extension to the model developed by Bastankhah et al. [14], the same method of determining added turbulence intensity is to be used. However, some modifications are necessary for the case of yawed turbines. Firstly, for modelling the added turbulence intensity due to a single turbine, $I_{in,i}$, Equation 5.2 is intended for unyawed turbines. The axial induction factor, a_i, for example, is derived from axial momentum theory, which assumes an unyawed turbine. Moreover, Crespo and Hernández [31] derived Equation 5.2 from an empirical fit of numerical simulations validated against field data, both for an unyawed turbine. In spite of this, the effect of the choice of model here is beyond the scope of the current work, and thus Equation 5.2 is also applied for the yawed turbines considered in this thesis. The consideration of added turbulence intensity within a wind farm described by Equation 5.3 was also derived and verified for unyawed turbines. However, by instead considering the overlap between the elliptical wake and the elliptical rotor of the downstream turbine, one can generalise this model. Whilst unverified, this should make the model more applicable in this case. Other more suitable approaches exist, for instance Qian and Ishihara [114] recently developed a method for determining I_{in} assuming a dual-Gaussian turbulence intensity distribution. This model could prove more accurate, and is a suitable candidate for further investigation.

5.2.1 Wake recovery and turbulence intensity

In addition to an estimation of the wake recovery rate for the streamwise wake, the model for the lateral velocity requires a single empirical input, namely an equivalent estimate of the wake recovery rate for the lateral wake. Bastankhah et al. [14] analysed the relationship between incoming turbulence intensity and streamwise wake recovery rate indirectly by computing the streamwise wake half-width, $\sigma_{n,1}$, in the wake of the last turbine in an aligned wind farm with varying number of rows, see Figure 6.9(a). Several authors have suggested an affine relationship between the wake recovery rate of a single turbine and the incoming turbulence intensity, both through numerical simulations [14, 106, 129] and lidar campaigns [148]. Bastankhah et al. [14] uses this assumption to determine the relationship between the streamwise wake recovery rate, $k_{w,1}$, and incoming turbulence intensity, I_{in}, by making a linear fit of the wake widths.

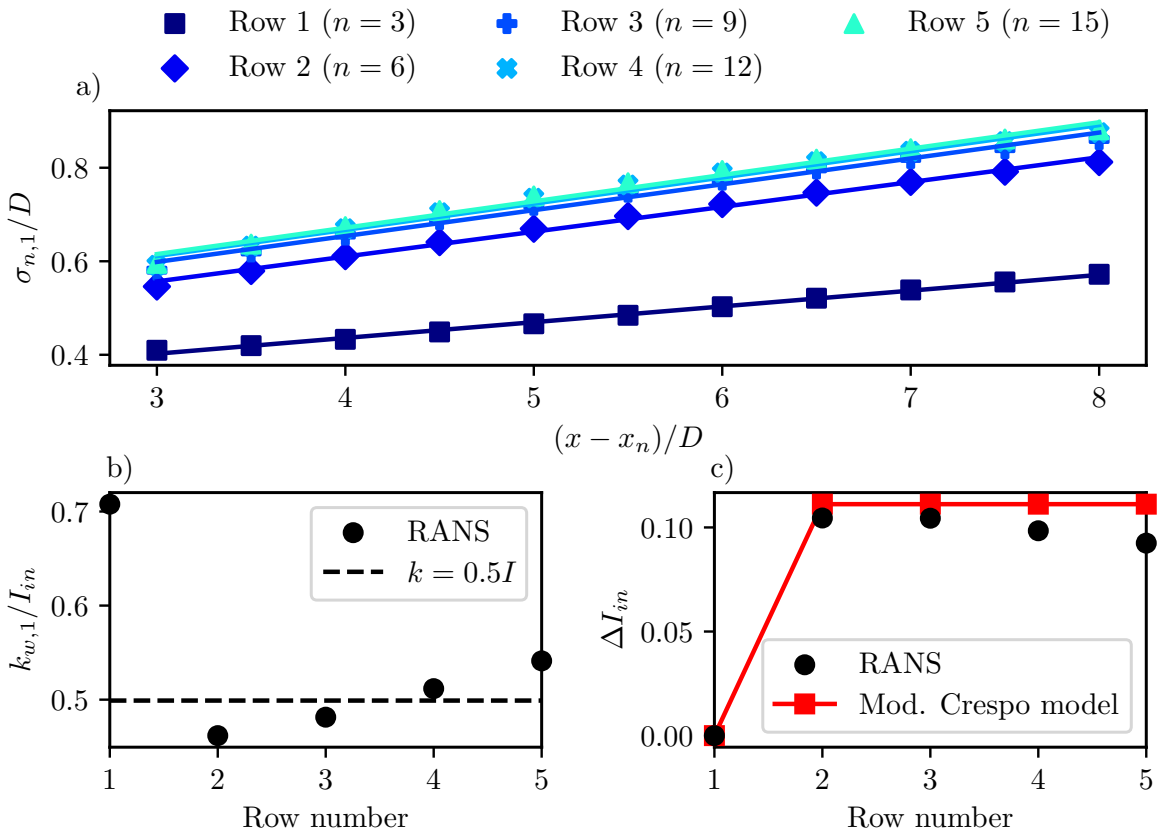


Figure 5.2: a) Variation of the normalised streamwise wake half-width with streamwise distance downstream of the the last turbine in the middle column of aligned wind farms with differing numbers of rows and all turbines unyawed ($\gamma = 0°$). b) The ratio of streamwise wake recovery rate $k_{w,1}$, to the incoming turbulence intensity I_{in} for turbines in different rows based on data from RANS simulations. c) Comparison of incoming added turbulence intensity predicted by the modified Crespo model [14] with RANS data for turbines in different rows. Note that this figure is equivalent to Figure 7 in Bastankhah et al. [14].

Figure 5.2 presents an analysis using the same methods, this time for the RANS results from PyWakeEllipSys, as opposed to SOWFA LES results [14]. Results from Section 3.4 suggest the streamwise wake half-width to be linear with downstream distance, and indeed the linear fit shown in Figure 5.2(a) matches the points very well. One can then determine the inflow turbulence intensity for each turbine by taking the the disc-averaged turbulent kinetic energy from the simulation in the manner already presented for the velocity, and converting it to total turbulence intensity following Equation 1.4, i.e.

$$I_{in} = \frac{\sqrt{\frac{2}{3}k_{AD}}}{|\boldsymbol{U}|}. \tag{5.4}$$

However, the analysis performed by Bastankhah et al. [14] assumes the use of the hub-height streamwise turbulence intensity, $I_{h,u}$, therefore it is necessary to convert total to streamwise turbulence intensity. This can be done using the standard ratios of the standard deviations

measured by Panofsky and Dutton [109] and implemented in *IEC 61400-1* [68], which leads to the relationship

$$I_{h,u} = 0.8I_{\infty,h}. \tag{5.5}$$

Taking the gradient of the linear fit of the wake half-widths from Figure 5.2(a) and dividing by the streamwise incoming turbulence intensity at the disc, one can then the proposed linear coefficient of proportionality between incoming turbulence intensity and wake recovery rate as shown in Figure 5.2(b). However, unlike the results presented for the LES data in Figure 7(b) of Bastankhah et al. [14], the RANS results do not show the same linearity between incoming turbulence intensity and streamwise wake recovery rate. There are two notable differences. Firstly, for a single row wind farm, the wake recovery rate is larger in this case than it was for LES data, and secondly, following a drop from row number 1 to 2, there is a steady increase of the ratio $k_{w,1}/I_{in}$ with row number. Further investigation is required to determine the exact reasons for these differences, but the ability of the RANS model to predict turbulence intensities, particularly cumulative turbulence intensities, when compared to LES, is surely a factor. Nevertheless, by neglecting the first row as an outlier and taking an average of the values from rows 2-5, one finds a relationship of $k_{w,1} = 0.5I_{in,u}$, which is larger than previously values observed for LES [27, 14, 106, 148], but useful in comparing the model predictions to the RANS simulations in Chapter 6.

Bastankhah et al. [14] also uses the same analysis to determine the suitability of the Crespo model for added turbulence intensity presented in the previous section. Figure 5.2(c) presents the added turbulence intensity experienced by the middle turbine of each row of the aligned wind farm. The behaviour of the added turbulence predicted by RANS is different to that predicted by LES, again due to the nature of the ability to predict added turbulence values. Despite this, the modified Crespo model employed in the streamwise solution shows a reasonably good fit to the data and will therefore be used in the predictions in Chapter 6.

Of course, this thesis aims to extend the model of Bastankhah et al. [14] to yawed turbines through the development of a model for the lateral velocity, therefore this analysis should also be applied to yawed turbines and the lateral wake. This analysis can be seen in Appendix E. However, the use of this analysis in those contexts highlights the lack of generality of the approach, and as such it is difficult to draw useful conclusions. Whilst the approach of Bastankhah et al. [14] was directly relevant to the flow cases they examined, it suffers from a number of shortcomings that make it difficult to generalise. Firstly, the thrust coefficients of the turbines, which are known to affect the wake recovery, are not all the same as downstream turbines are subject to the velocity deficits caused by upstream turbines. Furthermore, and as the authors themselves concede to further work, the wake recovery also likely has a dependence on for instance the integral length scale. A more general approach would perform simulations with different inflow turbulence intensities and examine the effect on wake recovery rate in such a way as to isolate the effects of incoming turbulence intensity, turbulent length scale, thrust coefficient, etc. This study is outside of the scope of this work but is undoubtedly a matter for further research as presented in Chapter 7.

5.2.2 Velocities and deflection

With a value for streamwise and lateral wake recovery rates determined, one can then calculate the width of the wake of any given turbine at any given streamwise position. Still within the turbine loop, the solution is now required to loop over each streamwise position, x, downstream

of WT_n within the flow domain. To calculate the wake widths - streamwise and lateral - for the given x-position, this implementation uses

$$\begin{aligned}\sigma_{n,1} &= k_{w,1}(x - x_n) + \varepsilon D \\ \sigma_{n,2} &= k_{w,2}(x - x_n) + \varepsilon D,\end{aligned} \tag{5.6}$$

where $\varepsilon(c_t)$ is the normalised initial wake half-width [11] given by $0.2\sqrt{(}\beta)$ and

$$\beta = \frac{1 + \sqrt{1 - c_{t,n}}}{2\sqrt{1 - c_{t,n}}}. \tag{5.7}$$

The use of this function also for the lateral wake is not rigorously justified, as the determination of an expression for the initial lateral wake half-width would require work beyond the scope of this thesis. However, results suggest this is a reasonable first approximation.

The interaction terms present in Equation 4.40 must then be calculated for each turbine upstream of turbine n. This can be done by again looping over these turbines and adding each contribution to a sum term $\Lambda_{i,\alpha}^{j,\beta}$, for both the streamwise and lateral interactions. This leads to the following terms:

$$\begin{aligned}
\Lambda_{i,1}^{n,1} &= \sum_{i=1}^{n-1} C_{i,1}\lambda_{i,1}^{n,1} = \sum_{i=1}^{n-1} C_{i,1}\frac{2\sigma_{i,1}^2}{\sigma_{n,1}^2 + \sigma_{i,1}^2}\exp\left(-\frac{(y_n - (y_i + \delta_i))^2}{2(\sigma_{n,1}^2 + \sigma_{i,1}^2)}\right)\exp\left(-\frac{(z_n - z_i)^2}{2(\sigma_{n,1}^2 + \sigma_{i,1}^2)}\right) \\
\Lambda_{i,2}^{n,1} &= \sum_{i=1}^{n-1} C_{i,2}\lambda_{i,2}^{n,1} = \sum_{i=1}^{n-1} C_{i,2}\frac{2\sigma_{i,2}^2}{\sigma_{n,1}^2 + \sigma_{i,2}^2}\exp\left(-\frac{(y_{n,1} - y_{i,2})^2}{2(\sigma_{n,1}^2 + \sigma_{i,2}^2)}\right)\exp\left(-\frac{(z_{n,1} - z_{i,2})^2}{2(\sigma_{n,1}^2 + \sigma_{i,2}^2)}\right) \\
\Lambda_{i,1}^{n,2} &= \sum_{i=1}^{n-1} C_{i,1}\lambda_{i,1}^{n,2} = \sum_{i=1}^{n-1} C_{i,1}\frac{2\sigma_{i,1}^2}{\sigma_{n,2}^2 + \sigma_{i,1}^2}\exp\left(-\frac{(y_{n,2} - (y_i + \delta_i))^2}{2(\sigma_{n,2}^2 + \sigma_{i,1}^2)}\right)\exp\left(-\frac{(z_{n,2} - z_{i,1})^2}{2(\sigma_{n,2}^2 + \sigma_{i,1}^2)}\right)
\end{aligned} \tag{5.8}$$

Using these three terms and known information about the turbines and wind farm, one can then calculate the peak streamwise deficit and lateral surplus of the given turbine for the given streamwise location using the two analytical solutions

$$\begin{aligned}
C_{n,1} &= \left(U_h - \Lambda_{i,1}^{n,1}\right)\left(1 - \sqrt{1 - \frac{T_n \cos\gamma_n}{\pi\rho\sigma_{n,1}^2\left(U_h - \Lambda_{i,1}^{n,1}\right)^2}}\right) \\
C_{n,2} &= \frac{\frac{T_n \sin\gamma_n}{\pi\rho} + \sigma_{n,1}^2 C_{n,1}\Lambda_{i,2}^{n,1}}{\sigma_{n,2}^2\left(2U_h - \Lambda_{i,1}^{n,2} - C_{n,1}\frac{2\sigma_{n,1}^2}{\sigma_{n,2}^2 + \sigma_{n,1}^2}\right)}.
\end{aligned} \tag{5.9}$$

Note that the streamwise solution now adjusts the streamwise component of the thrust force by the cosine of the yaw angle. This is of course the same as the original solution presented in Bastankhah et al. [14] for a yaw angle of zero. The final term in the denominator of the solution for $C_{n,2}$ appears because Equation 4.40 calls for $\sum_{i=1}^{n} C_{i,1}\lambda_{i,1}^{n,2}$, but $\Lambda_{i,1}^{n,2} = \sum_{i=1}^{n-1} C_{i,1}\lambda_{i,1}^{n,2}$ to be consistent with the other sum terms.

Following this, use the Gaussian to define the distributions of velocities in y and z, including the lateral wake centre deflection δ_n within $f_{n,1}$. The lateral wake centre deflection was shown to be very small in Section 3.5 and is therefore assumed to be negligible. In the absence of tilt,

the vertical deflection is also assumed to be zero. The two Gaussian functions describing the streamwise and lateral wake distributions are therefore

$$\begin{aligned} f_{n,1} &= \exp\left(-\frac{(y-(y_n+\delta_n))^2}{2\sigma_{n,1}^2}\right)\exp\left(-\frac{(z-z_n)^2}{2\sigma_n^2}\right) \\ f_{n,2} &= \exp\left(-\frac{(y-y_n)^2}{2\sigma_{n,2}^2}\right)\exp\left(-\frac{(z-z_n)^2}{2\sigma_{n,2}^2}\right) \end{aligned} \tag{5.10}$$

The perturbations can then be added to the flow fields according to Equations 4.15 and 4.17 as

$$\begin{aligned} U_n &= U_0 - \sum_{i=1}^{n} C_{i,1} f_{i,1} \\ V_n &= V_0 + \sum_{i=1}^{n} C_{i,2} f_{i,2} \end{aligned} \tag{5.11}$$

Finally, these flow fields can be used to calculate the deflection. The wake centre velocities can be found by interpolation to be

$$\begin{aligned} U_c &= U_n(x, y_h, z_h) - C_{n,1}(x) \\ V_c &= V_n(x, y_h, z_h) + C_{n,2}(x) \end{aligned} \tag{5.12}$$

Assuming the wake centre trajectory is tangential to the wake centre streamline, the wake centre deflection can be found from these velocities to be

$$\frac{\mathrm{d}\delta_n}{\mathrm{d}x} = \tan\left(\frac{V_{c,n}}{U_{c,n}}\right) \approx \frac{V_{c,n}}{U_{c,n}}. \tag{5.13}$$

In practice, this differential equation is implemented using a forward differencing scheme such that

$$\delta_n(x_{i+1}) = \delta_n(x_i) + \frac{V_{c,n}}{U_{c,n}}(x_{i+1} - x_i). \tag{5.14}$$

Once the turbine loop is completed, the resulting velocity fields are final and can be extracted alongside values such as power.

CHAPTER 6
Model Predictions

In order to evaluate the performance of the analytical model as derived in Chapter 4 and implemented in Chapter 5, it is important to first establish what the model is expected to be capable of. These capabilities can be summarised as three requisites, the analytical model should be able to predict: (i) the lateral wake velocities of a single turbine for various yaw angles, thrust coefficients, and turbulence intensities; (ii) the presence or absence of secondary wake effects as described in Fleming et al. [43] and King et al. [78]; and lastly, (iii) the lateral velocities within a wind farm, i.e. cumulative wake effects, for any distribution of yaw angles, thrust coefficients, and at any turbulence intensity.

To test each of these capabilities, three different flow cases have been chosen. The first features a single turbine. This setup is very similar to the investigations of the lateral wake in Chapter 3, except that the thrust coefficient of the turbine is determined by the thrust curve of the NREL 5 MW, and the turbulence intensity is kept fixed at $I_{\infty,h} = 6\%$ throughout. To assess the ability of the model to capture cumulative effects, the second set of simulations introduces a downstream turbine. In Fleming et al. [47], the authors examine a two-turbine case in both SOWFA and FLORIS (FLow Redirection and Induction In Steady State) [53] to demonstrate the 'secondary wake steering' phenomenon. That setup has an inter-turbine spacing of $7D$ and presents results for the upstream turbine unyawed or with a yaw angle of 25°, whilst keeping the downstream turbine unyawed. For ease of comparison, the same setup is adopted here. Finally, there are the wind farm cases. These are based on the two wind farm cases studies in Bastankhah et al. [14] and include an aligned and a staggered wind farm. Further details are given in Section 6.3.

This chapter seeks to compare two analytical models with the RANS predictions for the various flow cases. The baseline is the model for the streamwise velocity developed by Bastankhah et al. [14], hereafter referred to in the figures and analysis as 'Bastankhah et al.', which is built upon with the predictions of lateral velocity developed in this thesis in what is called the 'Analytical' model in this chapter. Section 6.1 also presents a correction for the near-wake deflection, which leads to the use of a third analytical model called the 'Corrected' model. Finally, the Bastankhah et al. model is omitted from the lateral velocity analyses because it makes no predictions of lateral velocity.

6.1 Single-turbine case

In essence, this thesis attempts to test both a new single turbine wake model for the lateral wake, and an improved approach to cumulative wake modelling. In order to establish the degree of success for each of these items separately, it is important to first consider just a single turbine. Moreover, as the verification data is now from PyWakeEllipSys, which is different from the LES data used in Bastankhah et al. [14], a baseline should be set which examines the differences between the original model for just the streamwise velocity of unyawed turbines from Bastankhah et al. [14] and the RANS data.

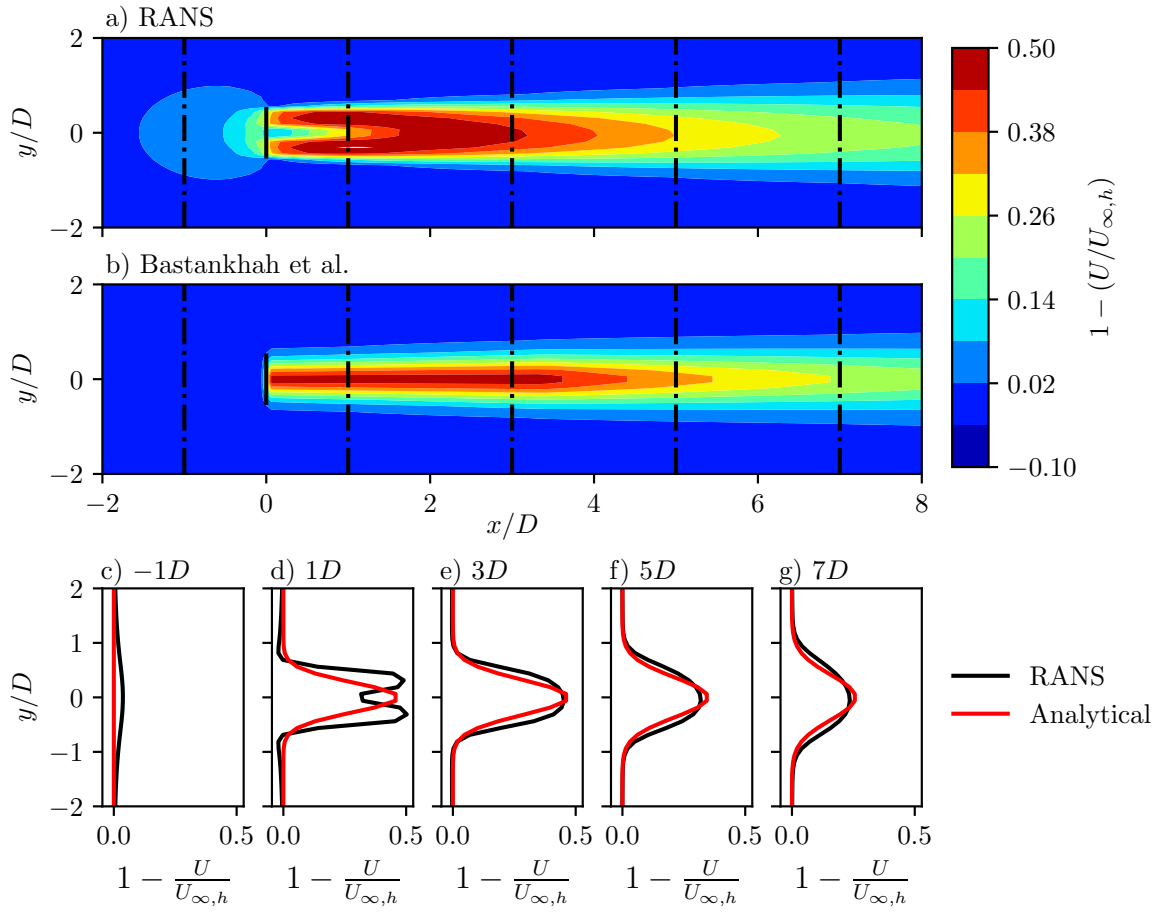


Figure 6.1: Contour plots of normalised hub-height streamwise velocity deficit for a single unyawed turbine from (a) RANS data and (b) analytical model for streamwise velocity from Bastankhah et al. [14] with comparison of streamwise velocity deficit profiles at $x =$ (c) $-1D$, (d) $1D$, (e) $3D$, (f) $5D$, and (g) $7D$ downstream. $I_{\infty,h} = 6\%$.

Figure 6.1 presents these differences first in terms of hub-height normalised streamwise velocity deficit. The first item to note is that the model proposed by Bastankhah et al. [14] is a wake model, therefore does not predict any upstream induction effects, hence the discrepancies around $x = -1D$. Secondly, the Gaussian wake model is a far wake model. It has been established previously that the near wake is closer to a dual-Gaussian, and this can be seen in Figure 6.1(d). Given most turbines in commercial wind farms operate in the far wake of upstream turbines, and the layouts tested in this chapter follow that trend, it is not important that the near wake is not modelled entirely accurately. Beyond the near wake, for $x/D \geq 3$, the profiles match very well. The RANS results present a profile slightly wider than Gaussian, but the peak velocities match well regardless.

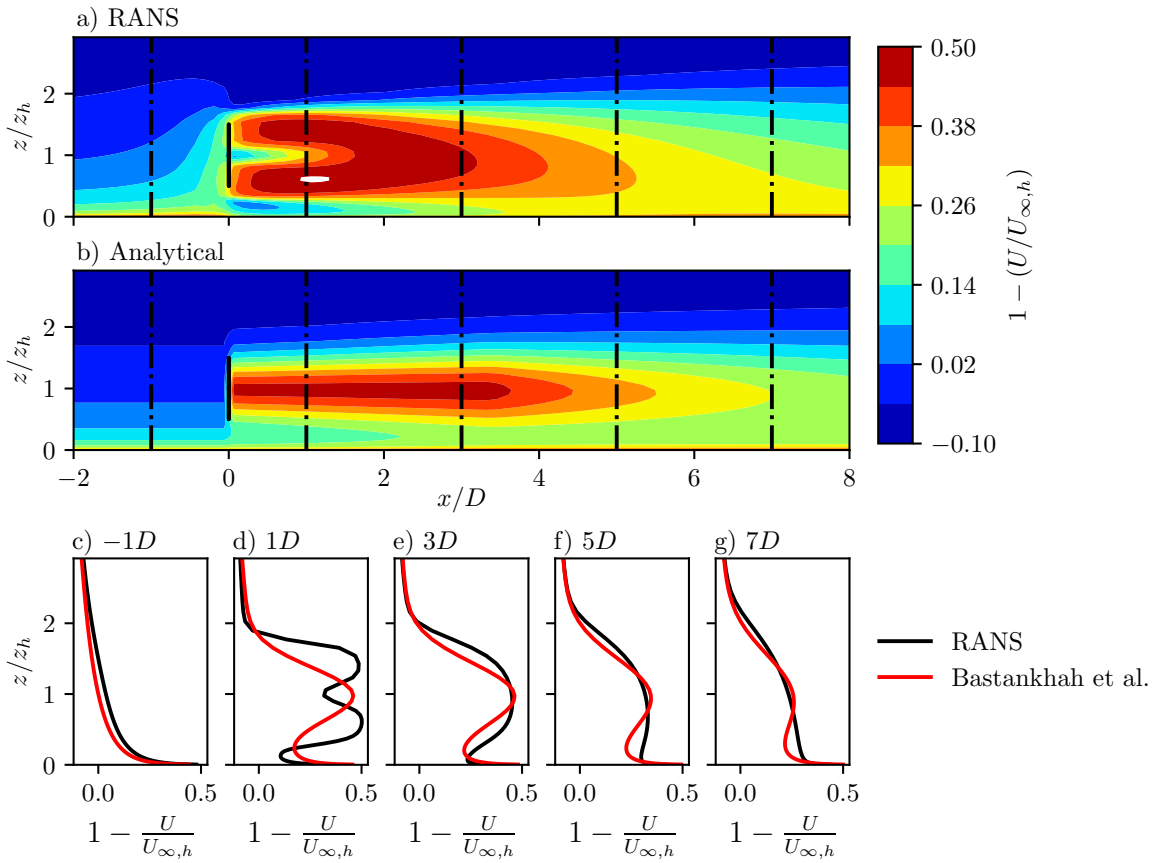


Figure 6.2: Contour plots of normalised streamwise velocity deficit in a x-z plane at the turbine y-coordinate for a single unyawed turbine from (a) RANS data and (b) analytical model for streamwise velocity from Bastankhah et al. [14] with comparison of vertical streamwise velocity deficit profiles at $x =$ (c) $-1D$, (d) $1D$, (e) $3D$, (f) $5D$, and (g) $7D$ downstream. $I_{\infty,h} = 6\%$.

The same discrepancies in the near upstream and near wake can be seen for a vertical plane at the turbine y-position in Figure 6.2. There are also some macro-scale effects the analytical model does not capture. For example, the tendency of the flow to go over the turbine and force the wake towards the ground around $x = 6D$. However, the far wake is also again well modelled, with the slight non-Gaussian nature of the RANS occurring again, but otherwise matching well. Furthermore, as the analytical model adds the contribution of the rotor to the preallocated inflow field, the ground effects are reasonably well replicated.

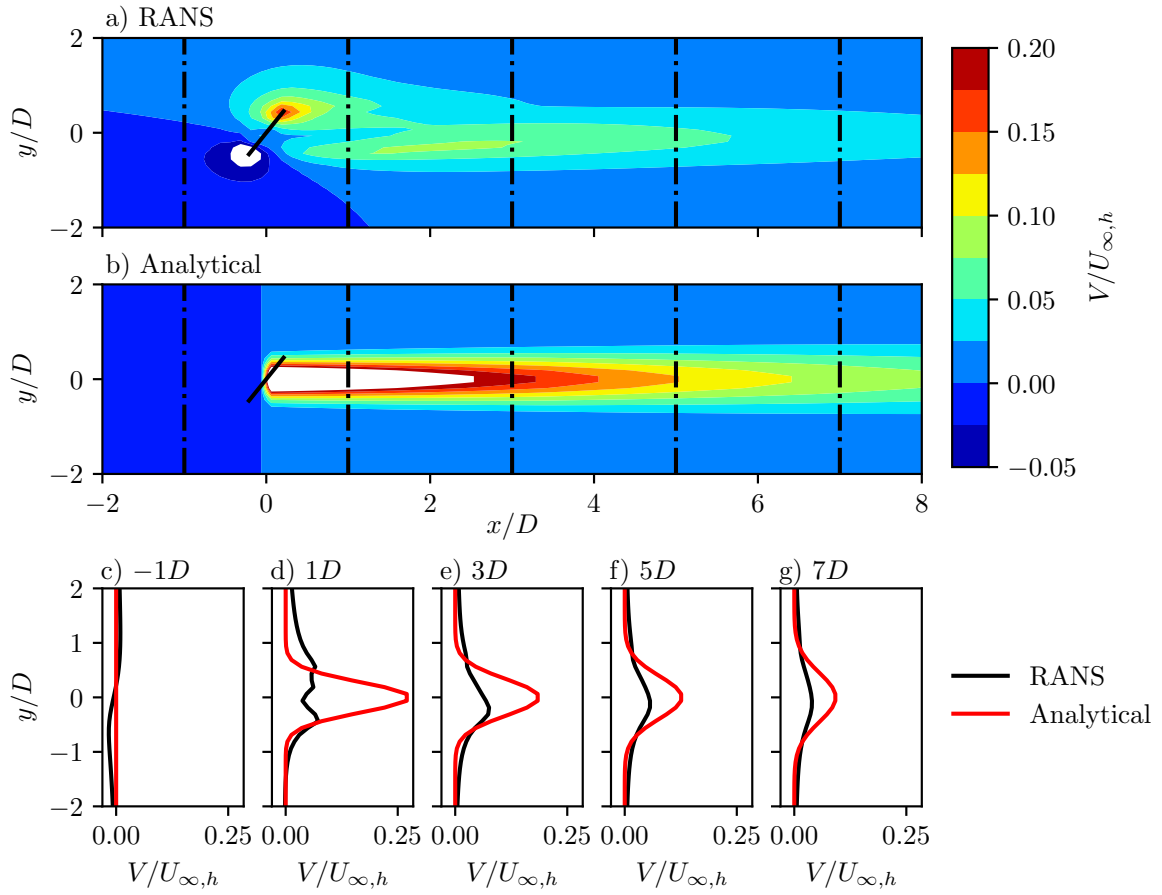


Figure 6.3: Contour plots of normalised hub-height lateral velocity for a single turbine yawed at $\gamma = 25°$ from (a) RANS data and (b) analytical model with comparison of lateral velocity profiles at $x =$ (c) $-1D$, (d) $1D$, (e) $3D$, (f) $5D$, and (g) $7D$ downstream. $I_{\infty,h} = 6\%$.

Turning attention to the lateral velocities, the analytical model again does not attempt to make any prediction of upstream effects, which in this case manifest as flow divergence and a sinusoidal profile at $x = -1D$ from the RANS. The analytical model predicts just zero velocity. Moreover, as was discussed in Chapter 3, the Gaussian fits is a far wake model, not just for the streamwise, but also for the lateral wake. The near wake profile predicted by the RANS in for example Figure 6.3(d) is not Gaussian, as it is still affected by the flow divergence process, but further downstream the predictions of the analytical model are improved. Despite this, there is a consistent overprediction of the wake centre velocity, $C_{n,2}$ evident even in the far wake, for which the model is intended to be valid. This could be due to the assumptions made in the budget analysis that lead to the 'conservation of lateral momentum deficit', Equation 4.9. The magnitudes of the velocities in the lateral wake are significantly smaller than the streamwise velocity deficit (almost a factor of 10 when comparing Figure 6.1(a) with Figure 6.3(a) for example), and the lateral wake is a combination of wake divergence, lateral forcing, and wake recovery whereas the streamwise wake is mostly determined by thrust forcing and wake recovery. This may mean that the momentum deficit in the lateral case is less clearly defined by just the lateral thrust forcing, and that the terms in Equation 4.8 that could be neglected in the streamwise case are less easily neglected for the lateral case.

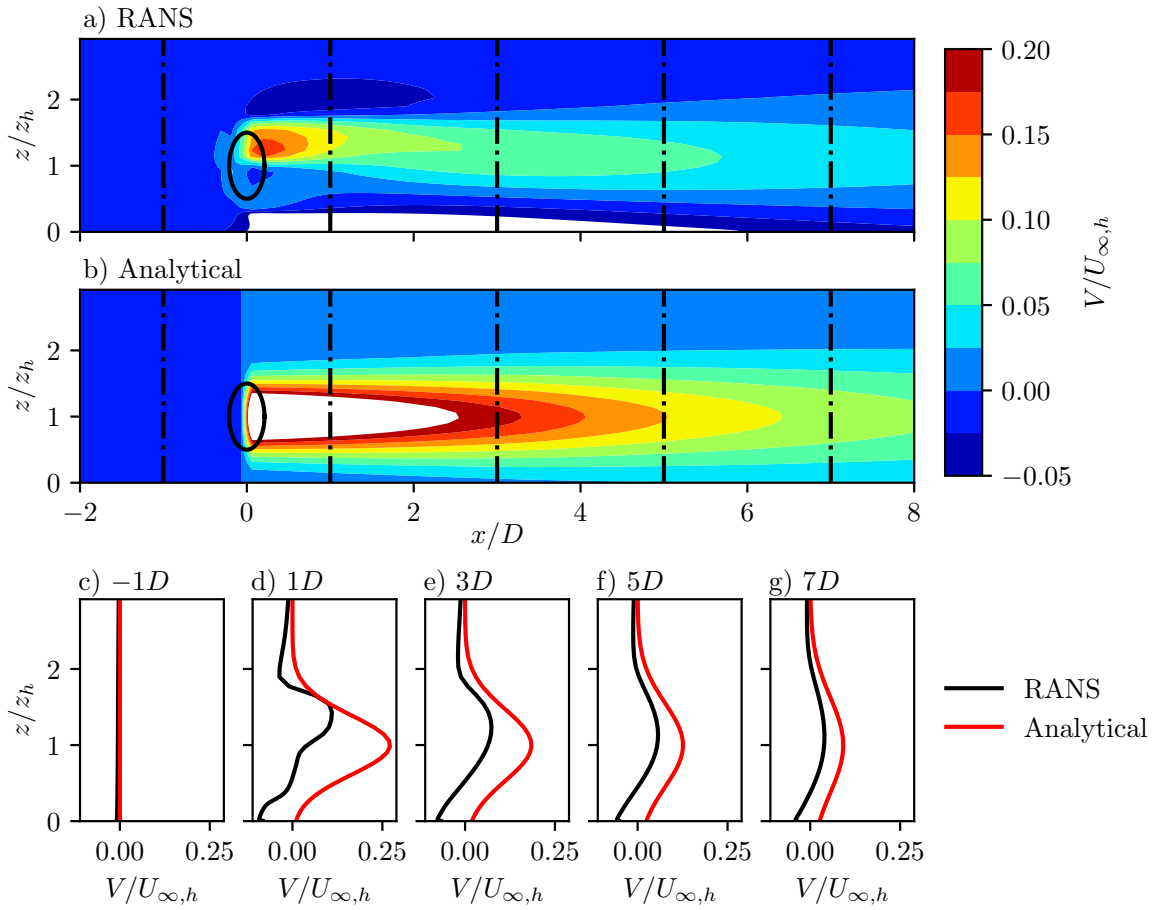


Figure 6.4: Contour plots of normalised lateral velocity in a x-z plane at the turbine y-coordinate for a single turbine yawed at $\gamma = 25°$ from (a) RANS data and (b) analytical model with comparison of vertical lateral velocity profiles at $x =$ (c) $-1D$, (d) $1D$, (e) $3D$, (f) $5D$, and (g) $7D$ downstream. $I_{\infty,h} = 6\%$.

Viewing the velocity fields in a vertical plane as in Figure 6.4 emphasises the effects of wake rotation, which are not evident in the hub-height plots. The analytical model does not account for wake rotation, and thus predicts $V|_{z=0} = 0$. However, wake rotation combined with ground effects in fact leads to negative velocities for $z/z_h < 0.5$ which persist into the far wake in the RANS data. Wake rotation effects are most evident in the near wake, for instance in Figure 6.4(d), where the RANS profile exhibits negative values of V for significant ranges of z. On the other hand, where lateral forcing dominates, such as at $x = 5D$ and $7D$, there is much better agreement between the profiles. Finally, the overprediction in $C_{n,2}$ also affects this vertical profile.

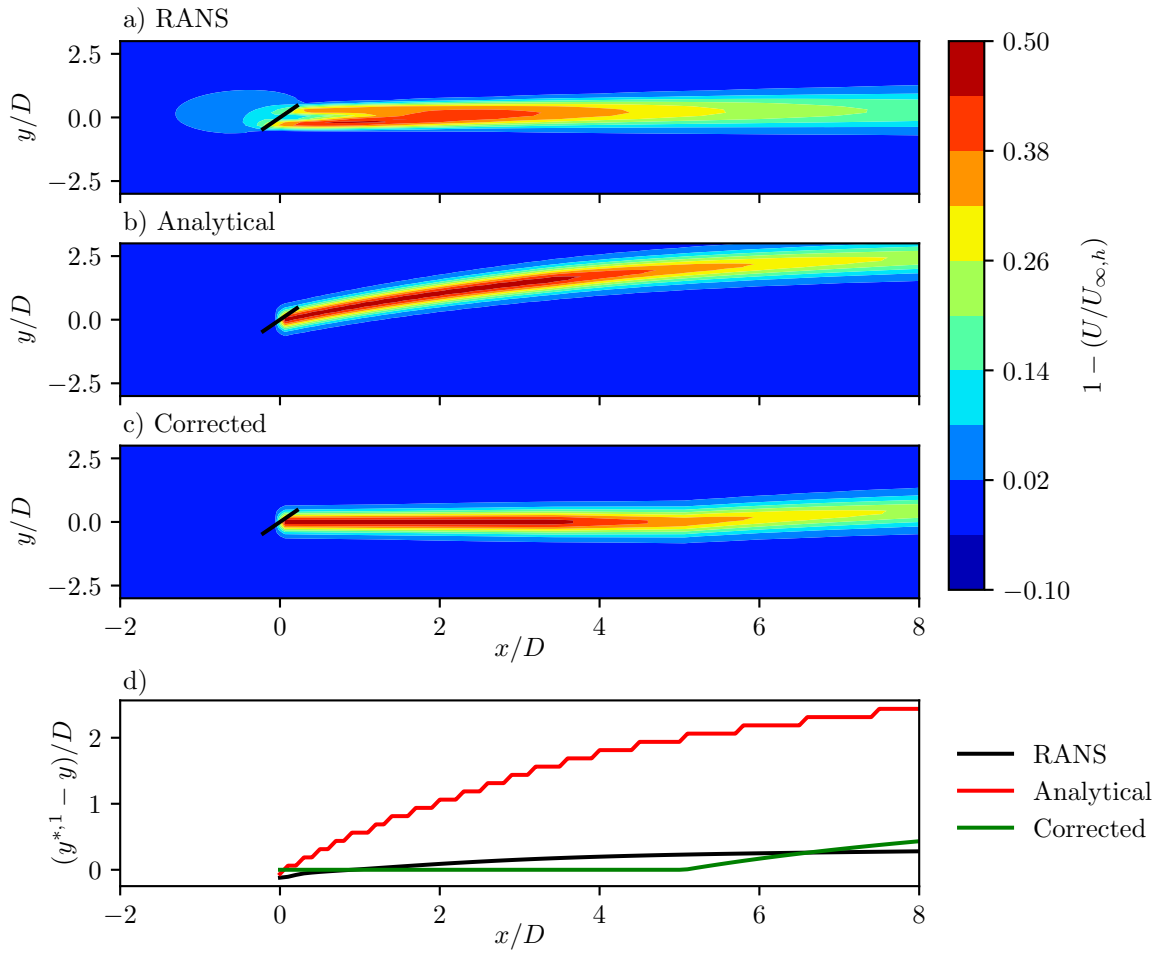


Figure 6.5: Contour plots of normalised hub-height streamwise velocity deficit for a single turbine yawed at $\gamma = 25°$ from (a) RANS data, (b) analytical model, and (c) analytical model with wake deflection correction applied with (d) comparison of streamwise wake deflections. $I_{\infty,h} = 6\%$.

As has been alluded to, the relevance of the lateral wake is mainly in determining the deflection of the streamwise wake. Using the implementation presented in Section 5.2.2, Figure 6.5 shows a comparison of the trajectories of the streamwise wakes from the analytical model and the RANS data. The major observation from the analytical model in the form presented so far in this thesis is that the deflection is considerably overestimated. This is also clear from the comparison of the wake centre locations in Figure 6.5(d). Note that the stepwise motion of the wake deflection of the Analytical model is due to a post-processing issue with Gaussian fitting and it should rather be smooth like the other two lines. As has been discussed, the wake model presented in this thesis is intended as a far wake model, in the same way as the Gaussian is a far wake model in the streamwise case. However, unlike the streamwise case, for the calculation of wake deflection, the near wake actually makes a more significant contribution than the far wake. Furthermore, as the wake deflection is a cumulative process, the error in the near wake - in this case the overprediction - of the lateral velocity, is compounded in this cumulative process to lead to a significant overestimate of the wake deflection.

In order to best demonstrate the potential of the model, a near-wake correction has been applied which removes the contributions to deflection for $x/D < 5$. This ensures the deflection is roughly correct at $x/D = 7$, as can be seen in Figure 6.5(d), the streamwise spacing used throughout these results. Better modelling of the lateral velocity in the near wake, and addressing the overpredictions of C_n could fix this issue and this correction is by no means intended as a permanent solution, but is simply used to demonstrate the potential capabilities of the model.

6.2 Two-turbine case

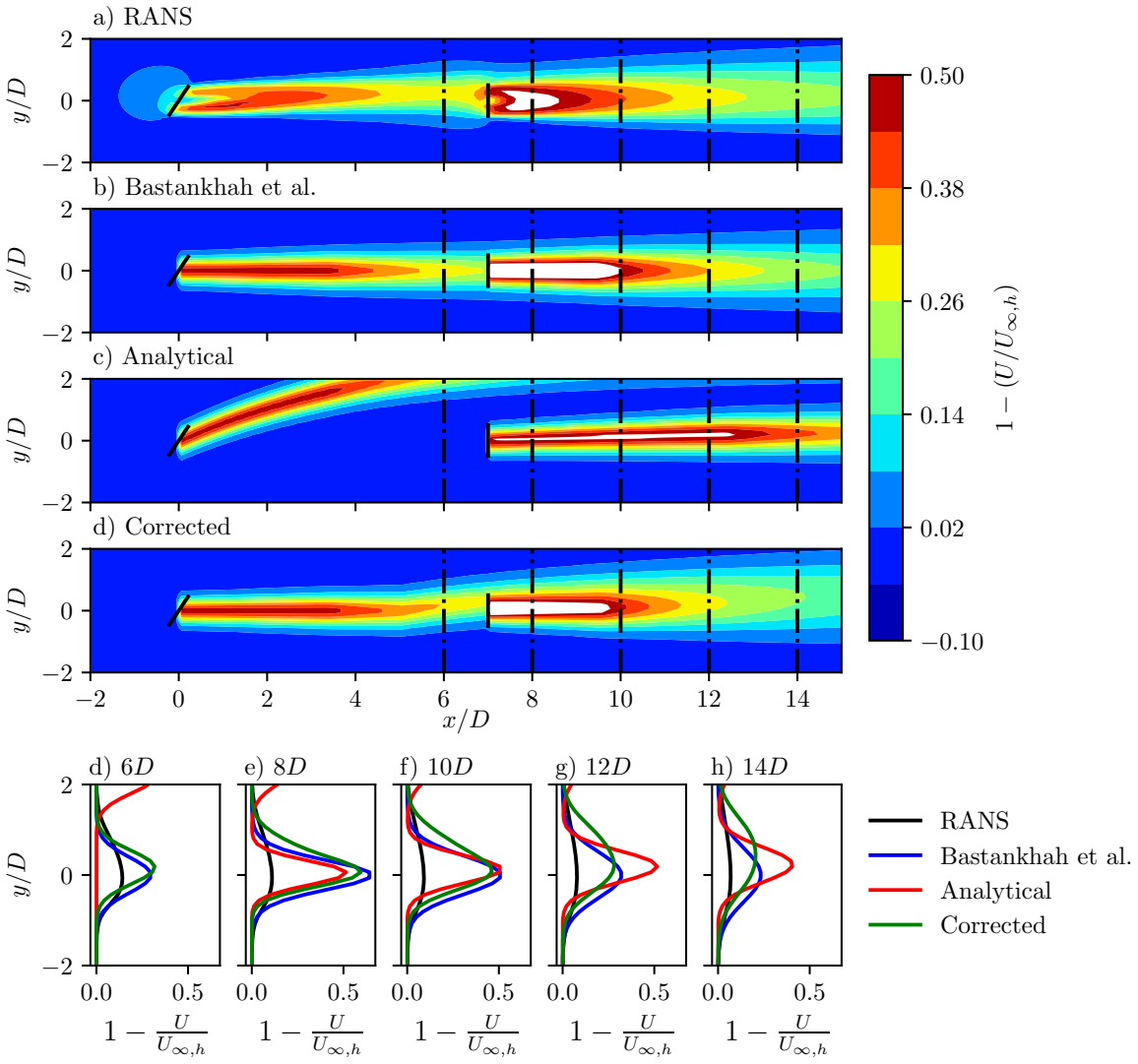


Figure 6.6: Contour plots of normalised hub-height streamwise velocity deficit for a two-turbine layout with the first turbine yawed at $\gamma_1 = 25°$ and the second unyawed from (a) RANS data and (b) analytical model with comparison of lateral velocity profiles at $x =$ (c) $-1D$, (d) $1D$, (e) $3D$, (f) $5D$, and (g) $7D$ downstream of the downstream turbine. $I_{\infty,h} = 6\%$.

The accuracy of the wake trajectory prediction is important for turbine predicting turbine interactions such as in this two turbine case. From Figure 6.6(c)-(d) one can see that without the near wake correction, the wake of the upstream turbine would entirely bypass the downstream turbine. In reality in this aligned setup there is very little deflection predicted by the RANS data, and so the predictions of the Bastankhah et al. model are reasonably accurate, with very minor improvements seen in the Corrected model. Examining the profiles, which are taken at $\{-1, 1, 3, 5, 7\}$ diameters upstream of the downstream turbine, it is immediately evident that there remains an overprediction of the velocity deficit. This can be attributed to a combination of the overprediction of $C_{n,1}$ seen for a single turbine in Figure 6.1, and the lack of induction effects in the modelled prediction. While the corrected profile perhaps fits the RANS result better in the wake of the downstream turbine, the overprediction is such that the difference from the Bastankhah et al. profile is negligible.

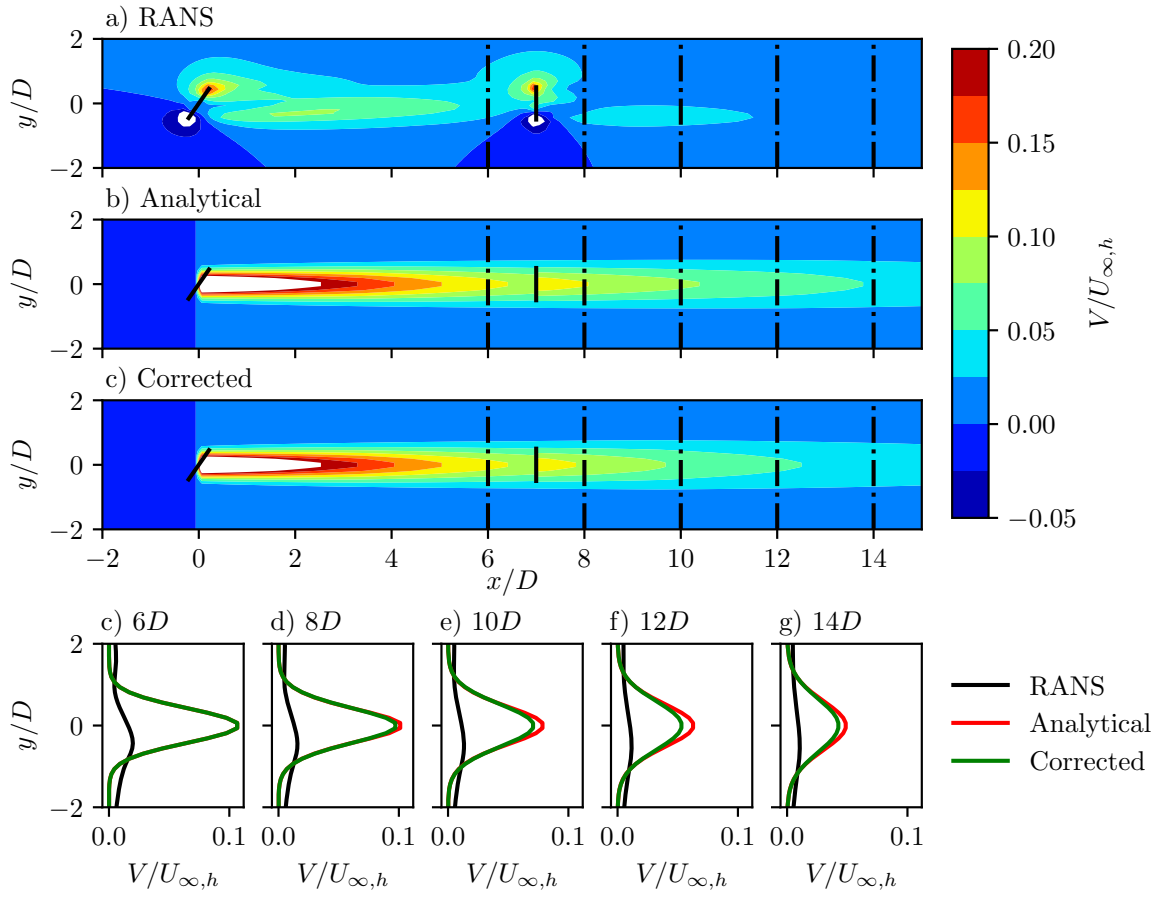


Figure 6.7: Contour plots of normalised hub-height lateral velocity for a two-turbine layout with the first turbine yawed at $\gamma_1 = 25°$ and the second unyawed from (a) RANS data and (b) analytical model with comparison of lateral velocity profiles at $x =$ (c) $-1D$, (d) $1D$, (e) $3D$, (f) $5D$, and (g) $7D$ downstream of the downstream turbine. $I_{\infty,h} = 6\%$.

The compounding of the overpredictions is also a feature of the lateral velocity. The overprediction of lateral velocity peak by analytical model is larger than was seen for a single turbine in Figure 6.3. This is because the downstream turbine has some upstream induction

effects, i.e. the flow attempts to go around the turbine, which redistributes the lateral velocities and cause a decrease in the peak velocity which persists downstream of the second turbine. These effects are not modelled by the analytical solution. The overprediction of lateral velocities in the wake of the downstream turbine are also partially caused by the contribution of the downstream turbine itself, even with the correction, which improves the prediction of disc velocity, and so thrust force, at the downstream turbine.

The downstream turbine also contributes as a result of the cumulative effects in Equation 4.40, however, from the setup in Figure 6.7 it is difficult to discern the specific effect of introducing the downstream turbine. In order to model this contribution, one can simulate a single turbine, then add in the second turbine and rerun the simulation before subtracting the latter flow field from the former. This method forms the basis for the definition of streamwise velocity deficit used in Bastankhah et al. [14] and lateral velocity surplus used in Chapter 4.

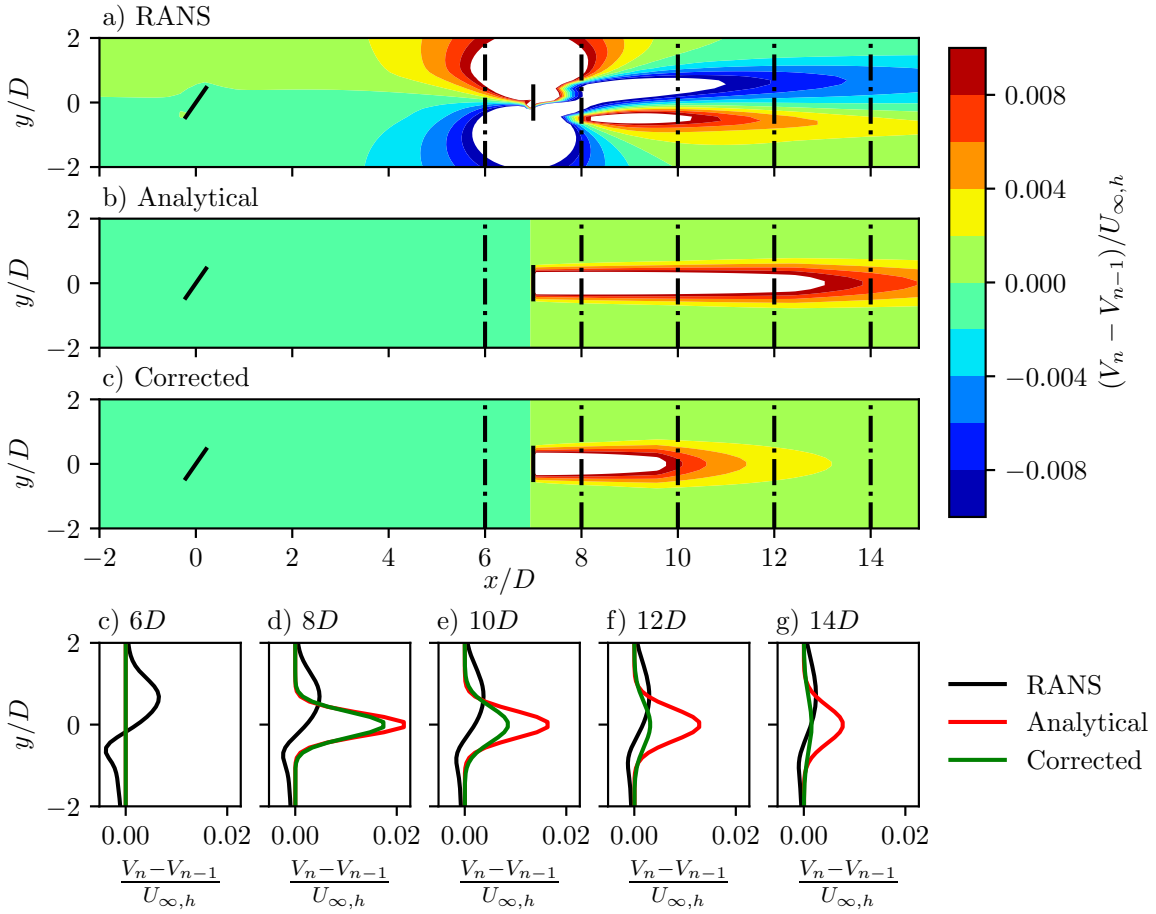


Figure 6.8: Contour plots of normalised hub-height lateral velocity surplus due to WT_2 (i.e. $n = 2$) for a two-turbine layout with the first turbine yawed at $\gamma_1 = 25°$ and the second unyawed from (a) RANS data and (b) analytical model with comparison of lateral velocity profiles at $x =$ (c) $-1D$, (d) $1D$, (e) $3D$, (f) $5D$, and (g) $7D$ downstream of the downstream turbine. $I_{\infty,h} = 6\%$.

Although this approach does allow one to see the effect of introducing the downstream

turbine, given it is unyawed its contribution is dominated by wake divergence and recovery effects, thus the analysis remains difficult. However, it is evident from the analytical results that the downstream turbine contributes to the lateral velocities, that is to say that the analytical model does indeed predict secondary wake steering. It is not possible to say whether the RANS data also exhibits secondary wake steering though, as the secondary wake steering effects are negligible when compared to the flow divergence and wake recovery effects.

This approach also demonstrates the effect of introducing the downstream turbine, but the concept of secondary wake steering is based upon the effect on the downstream turbine of yawing the upstream turbine, not of introducing the downstream turbine. A better approach would therefore be to subtract from the setup in Figure 6.7 a flowfield that includes just the upstream turbine, yawed as it is, and a flowfield that includes just the downstream turbine, unyawed as it is. This would allow one to remove the contributions of each individual turbine and examine simply the interaction effects. It would therefore produce a clearer picture from CFD results of any evidence of secondary wake steering. Unfortunately this requires some considerable thought and time in the post-processing that was not available to the author at the time of conceiving the idea.

6.3 Wind farm cases

Having examined the performance of the Gaussian as a single turbine lateral wake, followed by the ability of the full analytical model to predict turbine-turbine interaction effects, the next step is to assess the performance of the model over wind-farm scale.

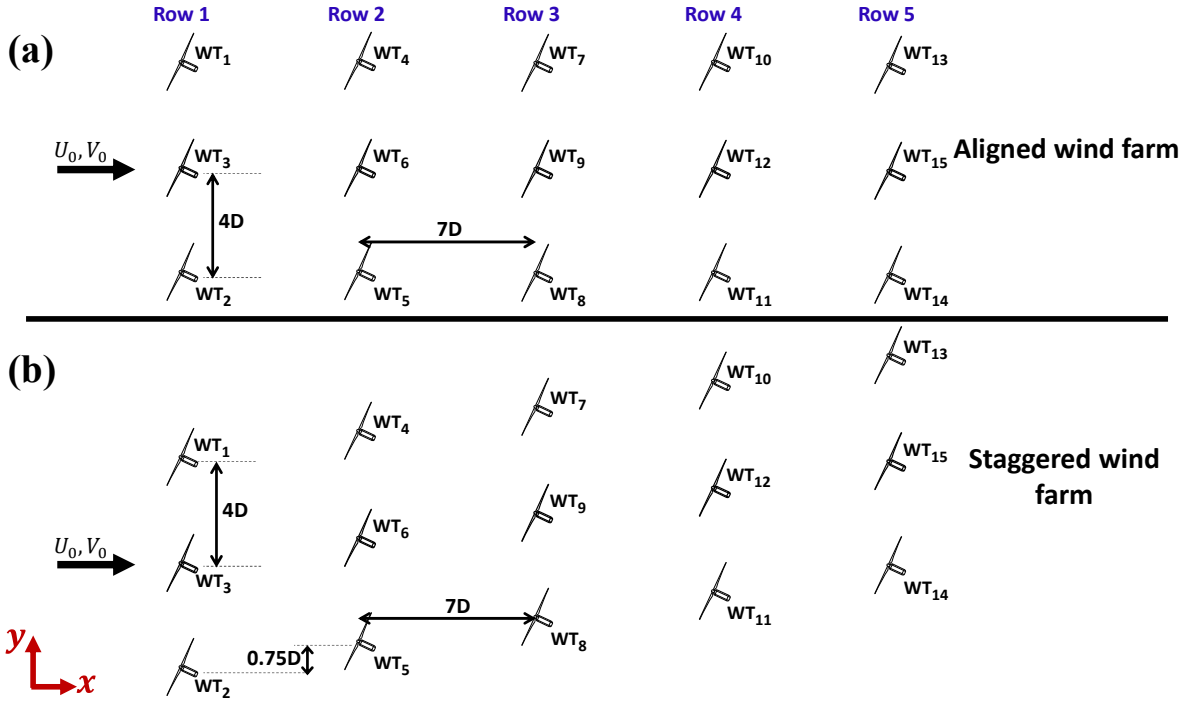


Figure 6.9: Schematic top view of the two wind farm layouts used for analysing model predictions: (a) aligned and (b) staggered. Figure adapted from Bastankhah et al. [14].

Figure 6.9 presents the two wind farm layouts to be investigated. These layouts are the same as used in Bastankhah et al. [14], except for the inter-turbine spacings, which were chosen to be consistent with the two-turbine case and Fleming et al. [43] and King et al. [78]. They also reflect the fact that this thesis uses a lower turbulence intensity than in Bastankhah et al. [14] to better represent typical offshore wind farm T.I. values. The lower turbulence intensity is also expected to somewhat cancel out the increase in inter-turbine spacings from Bastankhah et al. [14] via a slower wake recovery rate.

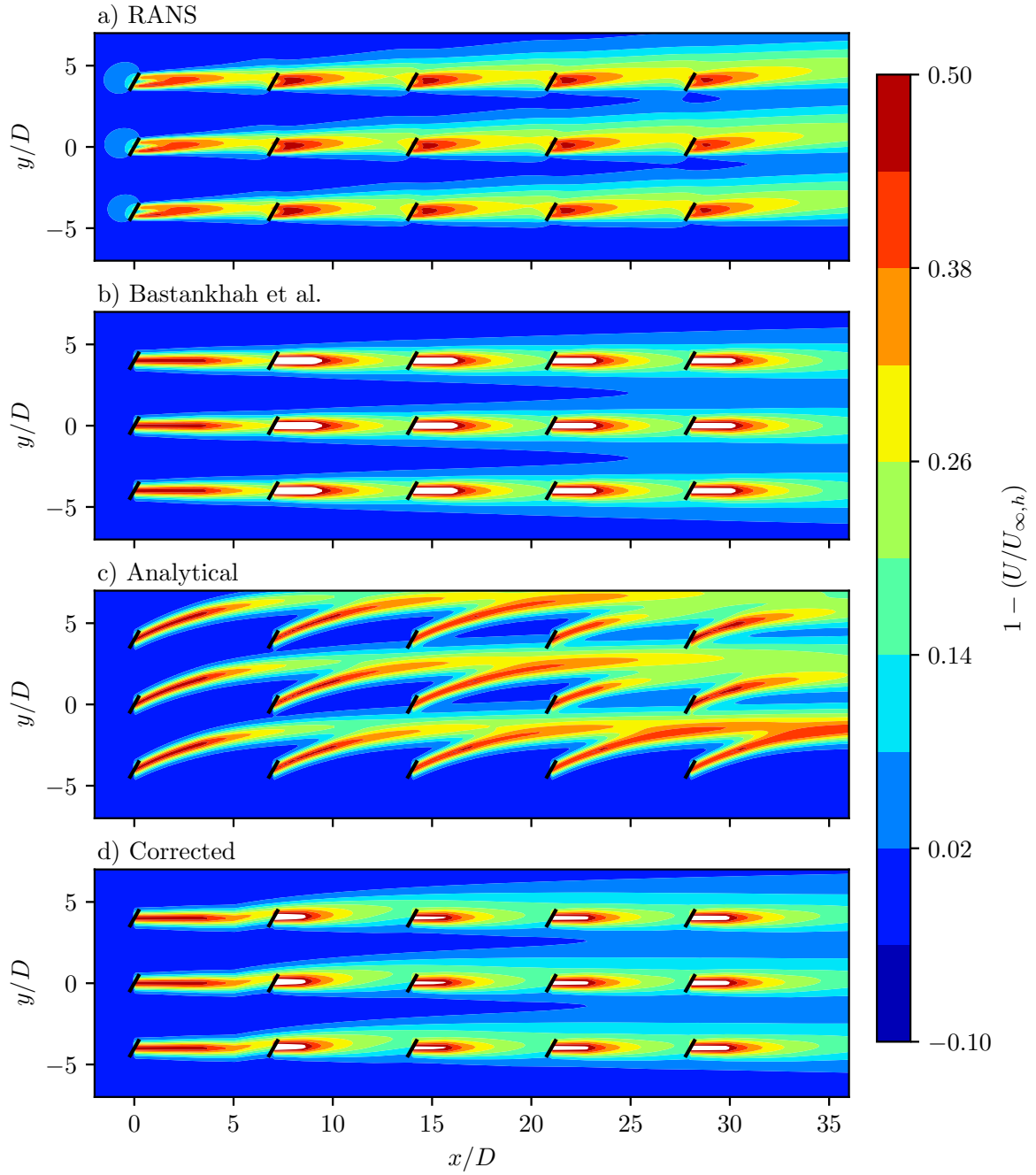


Figure 6.10: Contour plots of normalised hub-height streamwise velocity deficit with the aligned wind farm layout for (a) RANS, (b) analytical model for streamwise velocity from Bastankhah et al. [14], (c) new analytical model, and (d) new analytical model with near-wake correction.

Examining first the streamwise velocity deficits in Figure 6.10, the effect of the overprediction of the lateral velocity, combined with the underprediction when using the Gaussian as a model of the streamwise velocity deficit in the near wake, on the wake deflection is immediately apparent from Figure 6.10. Without the near wake correction, this error leads to significant

less accurate predictions than simply assuming zero lateral velocities, as in the Bastankhah et al. model. Implementing the correction does lead to predictions much closer to the RANS data, perhaps even closer than the Bastankhah et al. results, though it is difficult to tell from these plots - later analysis shows this better.

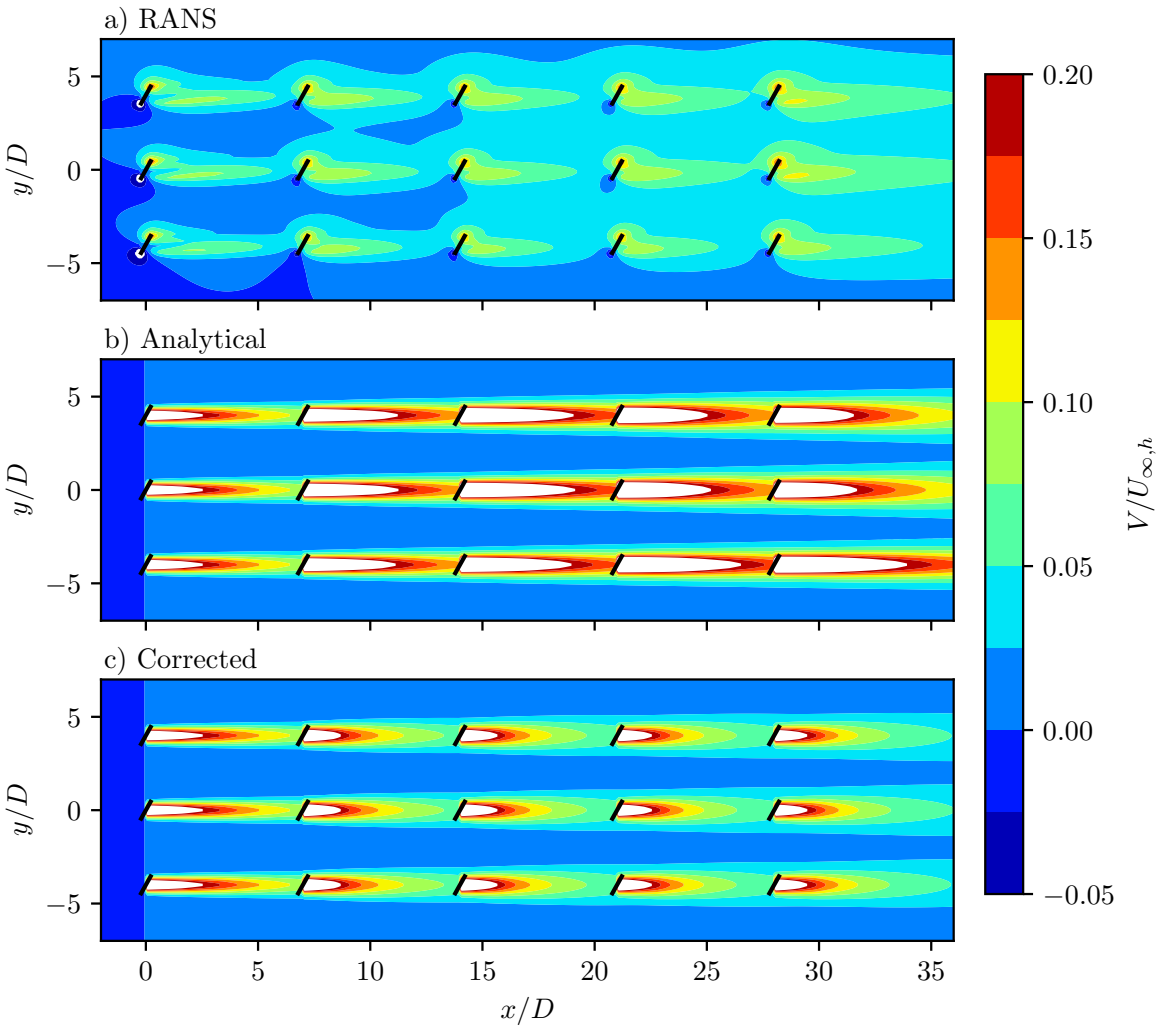


Figure 6.11: Contour plots of normalised hub-height lateral velocity with the aligned wind farm layout for (a) RANS, (b) new analytical model, and (c) new analytical model with near-wake correction.

The hub-height lateral velocity fields shed some more light on the accuracy of the model. Again, the error in wake deflection leads to large excesses in lateral velocity. However, with the correction implemented there are still notable differences from the RANS results. One interesting feature is the apparent different in wake widths. The RANS results suggest the influence of the turbines on lateral velocities extends much further laterally than the analytical results, particular at the rotor location. This is likely a result of the flow divergence behaviour, but it also affects the width of the lateral wake further downstream, and the effect appears to accumulate. This again suggests that the behaviour of the near wake is not as easily negligible

as has been assumed through the use of a Gaussian.

To put this model into a practical context given its intended application in wind resource assessment, one can look at the power predictions of each model.

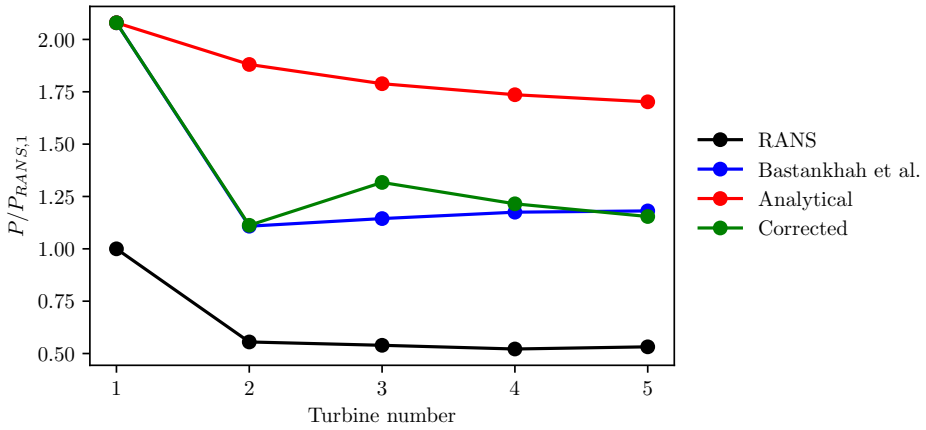


Figure 6.12: Power for turbines in the middle column of the aligned wind farm (turbines WT_i for $i = \{3, 6, 9, 12, 15\}$) normalised by the power of WT_3 predicted by RANS for the RANS data and three analytical models.

Figure 6.12 shows the so-called 'power-down-the-line' for the middle column of the aligned wind farm. Here the values have been normalised by the power predicted for the first turbine in the column by the RANS data. However, this normalisation technique makes it difficult to compare the models as there is a difference in the power predicted for the first turbine. This can be attributed to a difference in the disc velocity predicted for the first turbine by each approach, as seen in Figure 6.13.

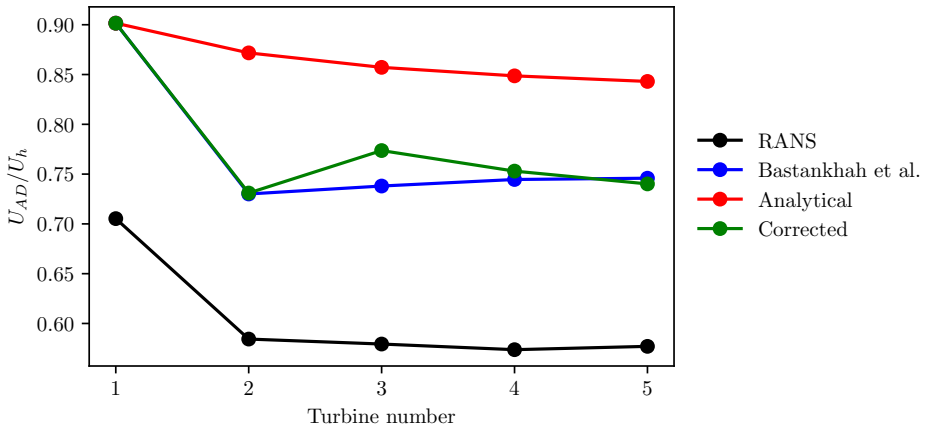


Figure 6.13: Normalised disc velocity for turbines in the middle column of the aligned wind farm (turbines WT_i for $i = \{3, 6, 9, 12, 15\}$) for the RANS data and three analytical models.

This difference is due to the inclusion of induction effects by the RANS model, which are not included by the analytical model. However, by instead normalising by the power at the first

turbine predicted by each model respectively, as in Figure 6.14, one can much more easily observe the performance of each model.

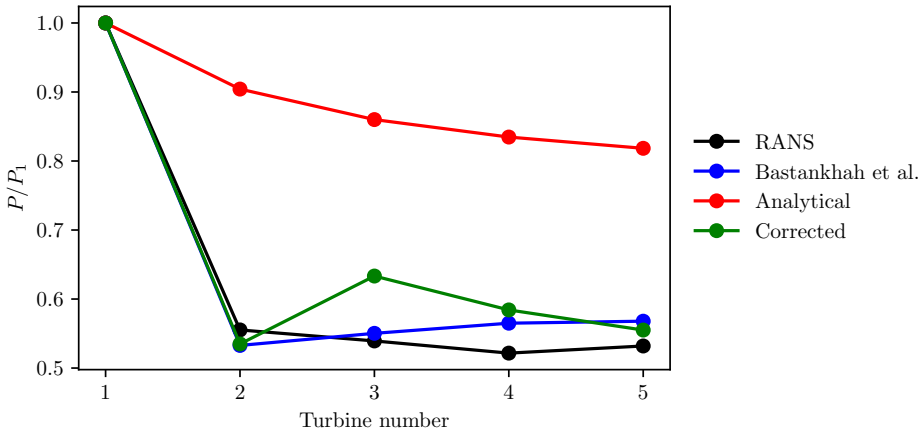


Figure 6.14: Power for turbines in the middle column of the aligned wind farm (turbines WT_i for $i = \{3, 6, 9, 12, 15\}$) normalised by the power of WT_3 predicted by each respective model for the RANS data and three analytical models.

Firstly, there is a clear overestimate by the Analytical model, as was expected given the wake deflection led to almost no waking of downstream turbines in Figure 6.10. Among the other two models overall the Bastankhah et al. model better fits the RANS predictions. At turbines 3 and 4, particularly the former, the Corrected model overpredicts the power. This must be a subtle effect of the wake steering or due to something else as the difference is not immediately evident from Figure 6.10.

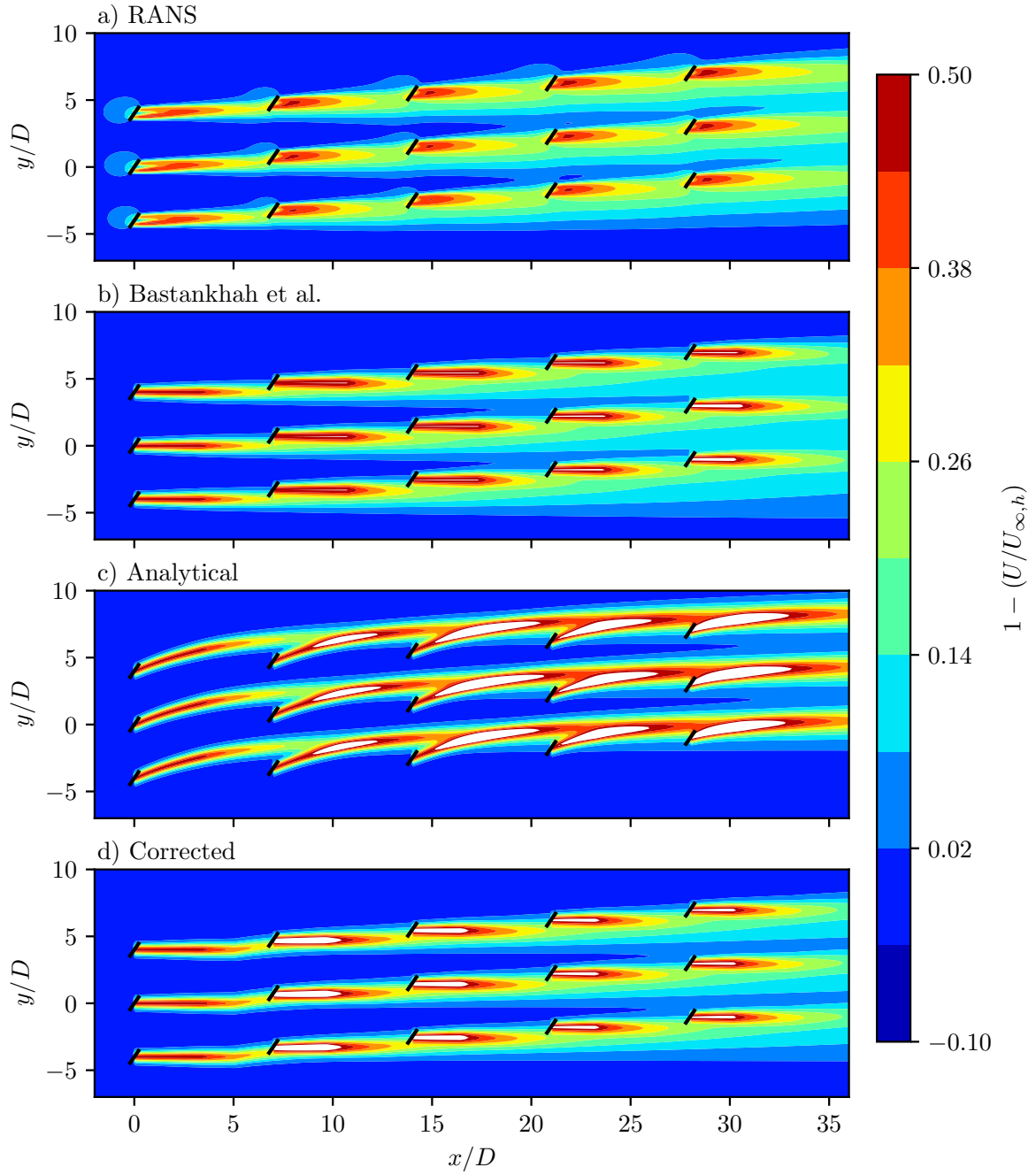


Figure 6.15: Contour plots of normalised hub-height streamwise velocity deficit with the staggered wind farm layout for (a) RANS, (b) analytical model for streamwise velocity from Bastankhah et al. [14], (c) new analytical model, and (d) new analytical model with near-wake correction.

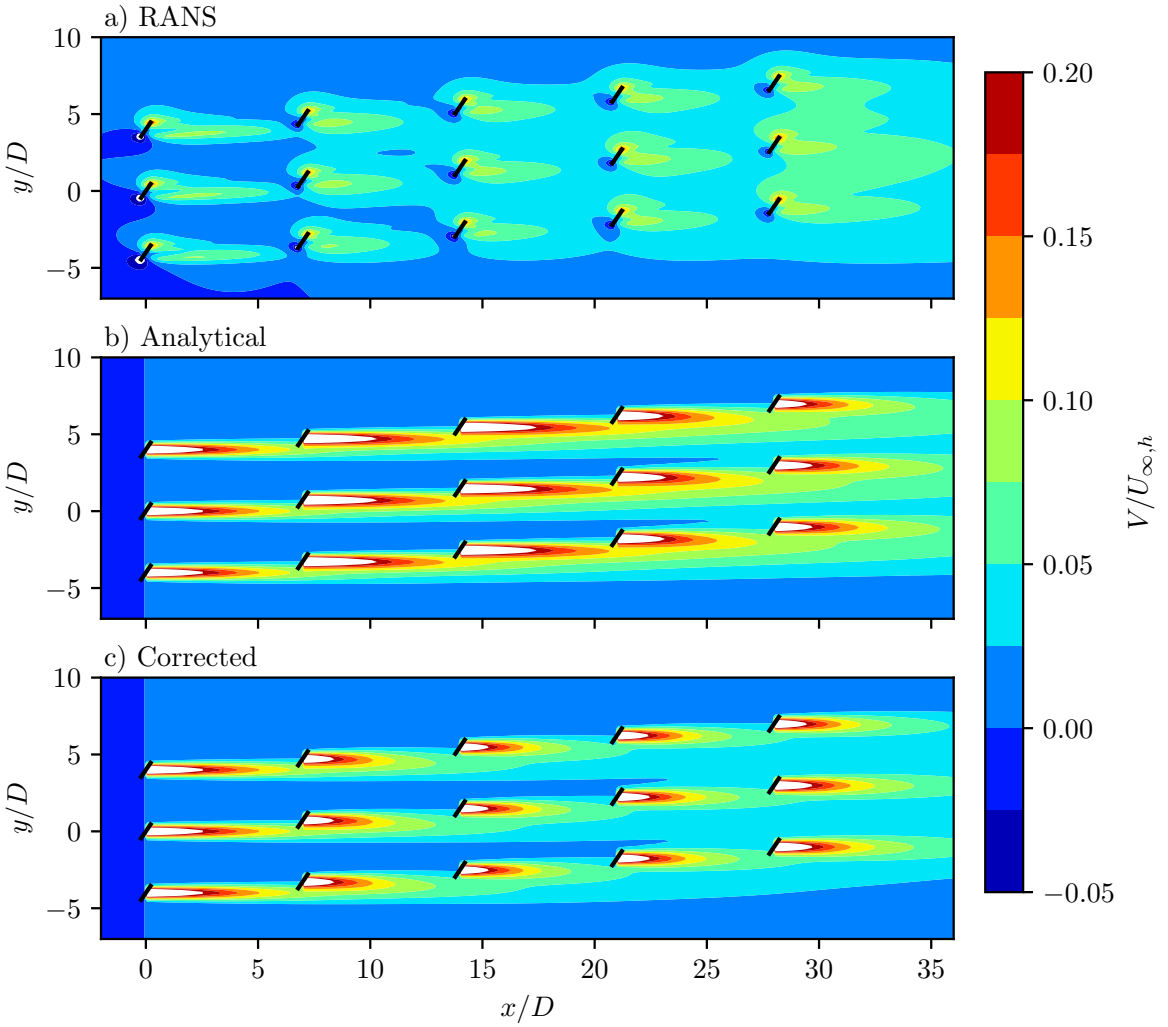


Figure 6.16: Contour plots of normalised hub-height lateral velocity with the staggered wind farm layout for (a) RANS, (b) new analytical model, and (c) new analytical model with near-wake correction.

Performing the same analysis for the staggered wind farm shows primarily the strong dependence of the predictions on the wind farm layout. Again, from Figure 6.15 the benefits of the Corrected model over the Analytical model are clear, but compared to the Bastankhah et al. model they are less obvious, as wake deflection is relatively subtle. The comparison of Figure 6.16(a) and (c) with their counterparts in Figure 6.11 is an interesting one. Again the wake width in the RANS data is larger, but the effect is a complex one, most evident downstream of the last row, where the individual wakes are not easily distinguishable. This is not predicted by the analytical model.

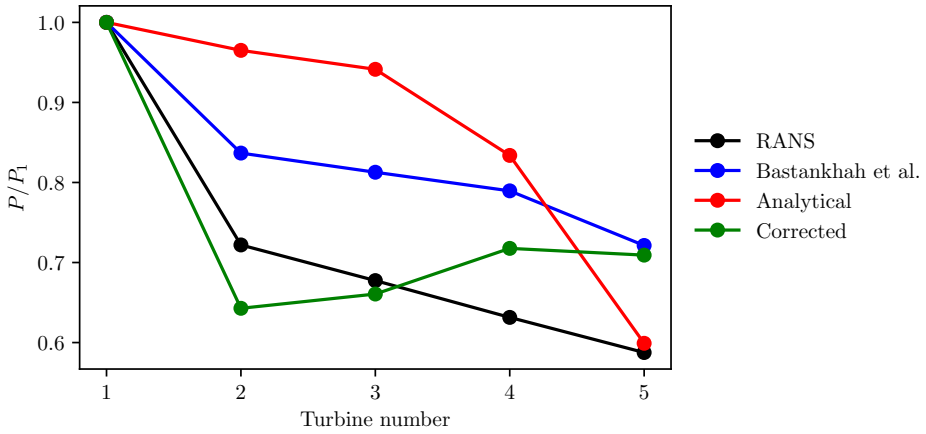


Figure 6.17: Power for turbines in the middle column of the slanted wind farm (turbines WT_i for $i = \{3, 6, 9, 12, 15\}$) normalised by the power of WT_3 predicted by each respective model for the RANS data and three analytical models.

The same issues with disc velocity in the first row apply in the slanted case, so the normalisation is again per model. However, for this layout the differences in power-down-the-line predicted by the models in Figure 6.17 are more stark. The Analytical model is again not very accurate, and given the analysis that has come before the accuracy at turbine 5 must be considered more fortunate than physical. Nevertheless, with the near-wake correction applied, it appears that the Corrected model performs better than the original Bastankhah et al. model.

Finally, aggregating all of the effects into a wind farm power, and noting that the percentage differences are strongly affected by the error in disc velocity at the first row of turbines, Table 6.1 shows that, as was expected, the Bastankhah et al. model, with no lateral velocity predictions, performs better than the Analytical model for the aligned farm. Notably, however, it also performs better than the Corrected model. Given the subtlety of the effect of wake steering in this layout, the accuracy required of the lateral velocity prediction to improve the predictions is greater than the model in its current form is capable of providing. In the staggered layout, where the wake steering appears to play a greater role, the Corrected model improves predictions, though it should be noted that the Analytical model predicts even closer to the RANS data, which could again suggest both predictions are coincidental. A more rigorous assessment of these models, including for example AEP predictions for various wind directions and speeds, could be performed were they implemented into setups such as PyWake or FLORIS.

Table 6.1: Wind farm power predictions of each model for the two wind farm layouts as a percentage difference from the RANS result given in megawatts.

	RANS [MW]	Bastankhah et al. [% diff.]	Analytical [% diff.]	Corrected [% diff.]
Aligned	6.39	113	185	120
Staggered	7.13	148	116	120

CHAPTER 7
Conclusions and Further Research

The main objective of this thesis is:

> "To improve the physical grounding and prediction accuracy of cumulative wake effect modelling within wind farms with yawed wind turbines, by means of an analytical solution for the lateral velocity field within a wind farm of arbitrary layout and arbitrarily yawed wind turbines that compares well with CFD".

In this chapter, Section 7.1 first provides answers to the research questions set out in Section 1.5 which define this objective, before Section 7.2 provides recommendations regarding the most important areas for further research.

7.1 Conclusions

RQ1.1 What is the influence of yaw angle on the lateral wake of a single turbine

Generally speaking, the influence of yaw angle on the lateral wake of a single turbine is to change the balance of the three driving forces, namely flow divergence, wake recovery, and lateral forcing, which determine the lateral wake. More specifically, an increase in yaw angle increases the influence of the lateral thrust forcing within the wake. This leads to more uniformity in the direction of the lateral wake velocity, and a more self-similar velocity profile. It also affects the width and location of the wake as it travels downstream. Yawed turbines have larger wake widths than unyawed ones, though it should be noted that the difference in the velocity profiles means that the definition of lateral wake width must also changed when yawing a turbine. The wake centre location of the hub-height velocity profile tends also to approach the hub centreline with increasing yaw angle.

Overall, the yaw angle is the major parameter in determining the nature of the lateral wake, however, it is very important to note that the behaviour of lateral velocities upstream and in the near wake is much less affected by yaw angle than the far wake, particularly when considering wake deflection, as here the near wake contributes significantly.

RQ1.2 What is the influence of thrust coefficient on the lateral wake of a single turbine?

The thrust coefficient plays a similar role in the lateral wake that it does in the streamwise wake, at least for yawed turbines. That is to say that the lateral velocity scales with the thrust coefficient. However, for the lateral wake of a turbine yawed at 25°, this thesis has shown this scaling to be linear across the range of typical range of thrust coefficients experienced by the NREL 5 MW ($C_T = 0.2$ to 0.8). Regarding wake width and centre location, the thrust coefficient is not as decisive as the yaw angle in determining the wake width of a yawed turbine, but the change in wake centre location with downstream distance does scale with thrust coefficient.

Interestingly, the contribution of the near wake to wake centre location does not appear to be influenced by thrust coefficient. Finally, thrust coefficient does not affect the self-similarity of the far wake lateral velocity profile.

RQ1.3 What is the influence of turbulence intensity on the lateral wake of a single turbine?

The effect of turbulence intensity is similar to that of thrust coefficient, with a few notable differences. Firstly, as has been observed more generally for the effect of turbulence intensity, its influence occurs at the interaction between the actuator and the flow, that is to say its influence on the lateral wake does not extend as far laterally as that of the thrust coefficient. Moreover, the scaling with thrust coefficient is the inverse of that with thrust coefficient, i.e. smaller turbulence intensity leads to larger lateral wake velocities. This tallies with the theory regarding the relationship between turbulence intensity and wake recovery rate. Incoming turbulence intensity was also shown not to affect the self-similarity of the far wake lateral velocity profile.

RQ1.4 Under which conditions is the normalised lateral wake velocity profile self-similar, and which function is most suitable to describe the self-similar profile?

Given the independence of the self-similarity of the far wake lateral velocity profile from thrust coefficient and turbulence intensity that has been established, the self-similarity can be said to primarily be determined by the yaw angle. The analysis in Section 3.2 demonstrates that the lateral wake of an unyawed turbine is self-similarity for an unyawed turbine and for a yawed turbine with a sufficiently large yaw angle. This distinction is somewhat nebulous but nevertheless important one. The larger the yaw angle, the greater the relative influence of lateral forcing and the more self-similar the profile. There is therefore a range of yaw angles where the self-similarity is detrimentally affected by the interplay of the driving forces discussed, but the degree of self-similarity for the smallest, 10° case was considered sufficient in this work.

Section 3.2 also examines the appropriateness of the Gaussian as a function to describe the self-similar profile. The findings are that a Gaussian is not a suitable profile to fit the lateral velocity profile in the wake of an unyawed turbine. For yawed turbines, the fit is not as good as for the streamwise velocity deficit, but given the ease of solution of the Gaussian integral, and the current infeasibility of applying other functions such as that described by vortex modelling, the Gaussian is considered adequate.

RQ2 How can the method used to derive the analytical solution for the streamwise velocity in Bastankhah et al. [14] be applied for the lateral velocity of yawed turbines?

Assuming the lateral wake can be described using a Gaussian, the method used in Bastankhah et al. [14] can be applied in a similar manner to the lateral velocity. This thesis assumes the budget analysis for the streamwise RANS momentum equation can be carried over directly to the lateral equation, which leads to a similar approximate form of conservation of mass and momentum as seen in Bastankhah et al. [14]. In the lateral case however, the equation is coupled, as the convection velocity is dominated by U. Employing the use of the streamwise solution allows one to form a linear equation for the lateral wake centre velocity $C_{n,2}$, which also describes the interaction between turbines within a wind farm, without the need for an ad hoc superposition method.

Based on this model, the velocity distribution downwind of a wind farm consisting of n

wind turbines is given by

$$\begin{aligned} U_n &= U_0 - \sum_{i=1}^{n} C_{i,1} \exp\left(-\frac{(y-(y_i+\delta_i))^2 + (z-z_i)^2}{2\sigma_{i,1}^2}\right) \\ V_n &= V_0 + \sum_{i=1}^{n} C_{i,2} \exp\left(-\frac{(y-y_i)^2 + (z-z_i)^2}{2\sigma_{i,2}^2}\right), \end{aligned} \tag{7.1}$$

where the wake half-widths, $\sigma_{i,1}$ and $\sigma_{i,2}$ are estimated from Equation 5.6, and the deflection, δ_i is calculated from

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}x} = \tan\left(\frac{C_{i,2}}{C_{i,1}}\right) \approx \frac{C_{i,2}}{C_{i,1}}. \tag{7.2}$$

The value of $C_{i,1}$ comes from Bastankhah et al. [14], and $C_{i,2}$ is determined as

$$C_{i,2} = \frac{T_i \sin\gamma_n + \pi\rho\sigma_{i,1}^2 C_{i,1} \sum_{j=1}^{i-1} C_{j,2}\lambda_{j,2}^{i,1}}{\pi\rho\sigma_{i,2}^2 \left(2U_h - \sum_{j=1}^{i} C_{j,1}\lambda_{j,1}^{i,2}\right)} \tag{7.3}$$

where T_i is the thrust force of WT_i. For $i = 1$, the equation is reduced to a single wake model by setting $\sum_{j=1}^{i-1} C_{j,2}\lambda_{j,2}^{i,1} = \sum_{j=1}^{i} C_{j,1}\lambda_{j,1}^{i,2} = 0$. For $i > 1$, the value of $\lambda_{j,\beta}^{i,\alpha}$ is given by

$$\lambda_{j,\beta}^{i,\alpha} = \frac{2\sigma_{j,\beta}^2}{\sigma_{i,\alpha}^2 + \sigma_{j,\beta}^2} \exp\left(-\frac{(y_{i,\alpha} - y_{j,\beta})^2}{2(\sigma_{i,\alpha}^2 + \sigma_{j,\beta}^2)}\right) \exp\left(-\frac{(z_{i,\alpha} - z_{j,\beta})^2}{2(\sigma_{i,\alpha}^2 + \sigma_{j,\beta}^2)}\right). \tag{7.4}$$

The value of $\lambda_{j,\beta}^{i,\alpha}$ depends on the wind farm layout and inflow conditions. This model has been implemented in Python and the scripts are available at `https://github.com/NilsGaukroger/Analytical-solution-for-the-cumulative-wake-of-yawed-wind-turbines`.

RQ3 How do the predictions of the resulting analytical model compare to RANS simulations from PyWakeEllipSys?

There are three main conclusions from the analysis in Chapter 6:

(i) The model to a limited extent overestimates $C_{n,2}$ in far wake. This is suspected to be a result of both the flow divergence and wake recovery processes leading to a lower wake centre velocity in the RANS, and the error in assuming the wake profile to be Gaussian.

(ii) The model also overestimates $C_{n,2}$ in near wake, though the single turbine lateral wake model (Gaussian) was not intended to be able to accurate model the near wake, so this was expected. However, this, combined with the inaccuracy of the Gaussian for the near wake of the streamwise velocity deficit, leads to a significant overestimate of the wake deflection, particularly in the near wake. Importantly, the effect is cumulative, so near wake predictions are also significant further downstream. This speaks to a wider issue with lateral wake modelling. That is to say, if its utility is in the prediction of wake steering, which it almost exclusively is, then contrary to the logic of streamwise wake modelling, modelling of the near wake is more important than that of the far wake for lateral wake modelling. This is because the near wake contributes relatively more to the wake deflection than the far wake.

(iii) If the near wake error can be reduced, there is potential for the new model to improve predictions of wind farm power for certain layouts and yaw angles. Though the range of flow cases here is limited due to time constraints.

7.2 Further research

There remain some important limitations to the implementation of such a lateral wake model in wake steering predictions for a variety of wind farm layouts and inflow conditions. Bastankhah et al. [14, §8] mentions some important issues for the streamwise model, and though with the lateral model there are many avenues one could take, below are outlined the three the author considers to be most pressing:

(i) Near wake modelling. The results of Chapter 6 have shown that the focus for lateral wake modelling for the prediction of wake deflection needs to be on the near wake. Vortex models have shown some promise for this Branlard and Gaunaa [22], and could, with some simplification, be implemented into a derivation such as presented in this thesis.

(ii) Neglected terms in the governing equation, Equation 4.8. A budget analysis for the this equation for various inflow conditions and layout geometries would highlight any potential issues in assuming the dominance of the thrust term.

(iii) Relationship between wake recovery rate and turbulence intensity. Several authors have pointed out the need for a better knowledge of the factors that drive turbulence intensity distributions in wind farms [14, 114], for example turbulent length scale and turbine operating conditions.

(iv) Model implementation in PyWake/FLORIS for AEP calculations. To best evaluate and compare the performance of the multitude of available wake, turbulence, and superposition models, they must be implemented in frameworks such as PyWake or FLORIS to allow a rigorous assessment over a range of conditions in for example AEP predictions.

APPENDIX A
Derivation of normalised governing equations

As set out in Laan et al. [86], both the Navier-Stokes and RANS equations can be set out in a normalised (non-dimensionalised) form, which can be shown, in the presence of certain conditions, to be independent of the characteristic scales of velocity and length. This appendix sets out to derive Equations 2 and 3 in Laan et al. [86] and explain there relevance and utility.

A.1 Incompressible Navier-Stokes equations

The governing equations of fluid flow in wind farms are as set out in Equation 2.1, $\frac{\partial u_i}{\partial x_i} = 0$, and Equation 2.2, which can also be written as

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{1}{\rho} f_i, \tag{A.1}$$

where u_i are the velocity components, p the pressure, f_i the external forces, x_i the spatial dimensions, and t the temporal.

Using the following non-dimensional variables

$$u_i' \equiv \frac{u_i}{\mathcal{U}}, \quad p' \equiv \frac{p}{\rho \mathcal{U}^2}, \quad f_i' \equiv \frac{f_i \mathcal{L}}{\rho \mathcal{U}^2}, \quad x_i' \equiv \frac{x_i}{\mathcal{L}}, \quad t' \equiv \frac{t}{\mathcal{T}}, \tag{A.2}$$

with $\mathcal{U}$, $\mathcal{L}$, and $\mathcal{T}$ the characteristic scales of velocity, length, and time, respectively, Equation A.1 can be written in non-dimensional form.

Substituting the non-dimensional variables in Equation A.1,

$$\frac{\partial(\mathcal{U}u_i')}{\partial(\mathcal{T}t')} + \mathcal{U}u_j' \frac{\partial(\mathcal{U}u_i')}{\partial(\mathcal{L}x_i')} = -\frac{1}{\rho}\frac{\partial(\rho \mathcal{U}^2 p')}{\partial(\mathcal{L}x_j')} + \nu \frac{\partial}{\partial(\mathcal{L}x_j')}\left(\frac{\partial(\mathcal{U}u_i')}{\partial(\mathcal{L}x_j')} + \frac{\partial(\mathcal{U}u_j')}{\partial(\mathcal{L}x_i')}\right) + \frac{1}{\rho}\frac{\rho \mathcal{U}^2}{\mathcal{L}} f_i'.$$

One can then take the characteristic scales outside of the derivatives to arrive at

$$\frac{\mathcal{U}}{\mathcal{T}}\frac{\partial u_i'}{\partial t'} + \frac{\mathcal{U}^2}{\mathcal{L}} u_j' \frac{\partial u_i'}{\partial x_i'} = -\frac{\mathcal{U}^2}{\mathcal{L}}\frac{\partial p'}{\partial x_i'} + \frac{\nu}{\mathcal{L}}\frac{\partial}{\partial x_j'}\left(\frac{\mathcal{U}}{\mathcal{L}}\frac{\partial u_i'}{\partial x_j'} + \frac{\mathcal{U}}{\mathcal{L}}\frac{\partial u_j'}{\partial x_i'}\right) + \frac{\mathcal{U}^2}{\mathcal{L}} f_i'$$

which can be further simplified by factorising the viscous term to give

$$\frac{\mathcal{U}}{\mathcal{T}}\frac{\partial u_i'}{\partial t'} + \frac{\mathcal{U}^2}{\mathcal{L}} u_j' \frac{\partial u_i'}{\partial x_i'} = -\frac{\mathcal{U}^2}{\mathcal{L}}\frac{\partial p'}{\partial x_i'} + \frac{\nu \mathcal{U}}{\mathcal{L}^2}\frac{\partial}{\partial x_j'}\left(\frac{\partial u_i'}{\partial x_j'} + \frac{\partial u_j'}{\partial x_i'}\right) + \frac{\mathcal{U}^2}{\mathcal{L}} f_i'.$$

Multiplying both sides by $\mathcal{L}/\mathcal{U}^2$ then yields

$$\frac{\mathcal{L}}{\mathcal{U}\mathcal{T}}\frac{\partial u_i'}{\partial t'} + u_j' \frac{\partial u_i'}{\partial x_i'} = -\frac{\partial p'}{\partial x_i'} + \frac{\nu}{\mathcal{U}\mathcal{L}}\left(\frac{\partial u_i'}{\partial x_j'} + \frac{\partial u_j'}{\partial x_i'}\right) + f_i'.$$

Substituting $\mathrm{Re} = {}^{\mathcal{U}\mathcal{L}}/_{\nu}$, this becomes Equation 2 of Laan et al. [86],

$$\frac{\mathcal{L}}{\mathcal{U}\mathcal{T}}\frac{\partial u_i'}{\partial t'} + u_j'\frac{\partial u_i'}{\partial x_i'} = -\frac{\partial p'}{\partial x_i'} + \frac{1}{\mathrm{Re}}\left(\frac{\partial u_i'}{\partial x_j'} + \frac{\partial u_j'}{\partial x_i'}\right) + f_i'. \tag{A.3}$$

One should notice that for high-Re conditions, such as those in which wind turbines operate,

$$\lim_{\mathrm{Re}\to\infty}\left(\frac{1}{\mathrm{Re}}\left(\frac{\partial u_i'}{\partial x_j'} + \frac{\partial u_j'}{\partial x_i'}\right)\right) = 0,$$

and, therefore, one can neglect the viscous term. Moreover, provided the forcing term, f_i, does not include a viscous component, e.g. if

$$|f_i| = |f_T| \propto \frac{\frac{1}{2}C_T\rho\mathcal{U}^2}{\mathcal{L}}, \tag{A.4}$$

and so

$$|f_i'| = \frac{\frac{1}{2}C_T\rho\mathcal{U}^2}{\mathcal{L}} \times \frac{\mathcal{L}}{\rho\mathcal{U}^2} = \text{constant}. \tag{A.5}$$

More generally, if

1. Re is sufficiently large,
$$\mathrm{Re} \to \infty, \tag{A.6}$$
2. all external forces scale with ${}^{\rho\mathcal{U}^2}/_{\mathcal{L}}$
$$f_i \propto \frac{\rho\mathcal{U}^2}{\mathcal{L}}, \tag{A.7}$$
3. and the characteristic time scale, $\mathcal{T}$, of the flow is equal to ${}^{\mathcal{L}}/_{\mathcal{U}}$
$$\mathcal{T} = \frac{\mathcal{L}}{\mathcal{U}}, \tag{A.8}$$

then the Navier-Stokes equations in normalised form, Equation A.3, do not depend on the characteristic scales of velocity or length [86].

A.2 RANS equations

The same logic can be applied to the RANS (momentum) equations. Starting with Equation 2.6, where the Boussinesq approximation in Equation 2.5 has been applied to the original RANS equations (Equation 2.4), and which can also be written as

$$\frac{\partial U_i}{\partial t} + U_j\frac{\partial U_i}{\partial x_j} = -\frac{\partial P^*}{\partial x_i} + \frac{\partial}{\partial x_j}\left((\nu+\nu_t)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right) + \frac{1}{\rho}F_i, \tag{A.9}$$

where P^* is the modified pressure, defined to include the turbulent kinetic energy as

$$P^* = P + \frac{2}{3}k. \tag{A.10}$$

Again, a set of non-dimensionalised variables can be defined,

$$U_i' \equiv \frac{U_i}{\mathcal{U}}, \quad P^{*\prime} \equiv \frac{P^*}{\rho\mathcal{U}^2}, \quad \nu_t' \equiv \frac{\nu_t}{\mathcal{U}\mathcal{L}}, \quad F_i' \equiv \frac{F_i\mathcal{L}}{\rho\mathcal{U}^2}, \quad x_i' \equiv \frac{x_i}{\mathcal{L}}, \quad t' \equiv \frac{t}{\mathcal{T}}, \tag{A.11}$$

and then substituted into Equation A.9 to give

$$\frac{\partial(\mathcal{U}U_i')}{\partial(\mathcal{T}t')} + \mathcal{U}U_j'\frac{\partial(\mathcal{U}U_i')}{\partial(\mathcal{L}x_j')} = -\frac{1}{\rho}\frac{\partial(\rho\mathcal{U}^2P^{*\prime})}{\partial(\mathcal{L}x_i')} + \frac{\partial}{\partial(\mathcal{L}x_j')}\left((\nu + \mathcal{U}\mathcal{L}\nu_t')\left(\frac{\partial(\mathcal{U}U_i')}{\partial(\mathcal{L}x_j')} + \frac{\partial(\mathcal{U}U_j')}{\partial x_i'}\right)\right) + \frac{1}{\rho}\frac{\rho\mathcal{U}^2}{\mathcal{L}}F_i'.$$

Again one can take the characteristic scales outside of the derivatives to come to

$$\frac{\mathcal{U}}{\mathcal{T}}\frac{\partial U_i'}{\partial t'} + \frac{\mathcal{U}^2}{\mathcal{L}}U_j'\frac{\partial U_i'}{\partial x_j'} = -\frac{\mathcal{U}^2}{\mathcal{L}}\frac{\partial P^{*\prime}}{\partial x_i'} + \frac{1}{\mathcal{L}}\frac{\partial}{\partial x_j'}\left((\nu + \mathcal{U}\mathcal{L}\nu_t')\left(\frac{\mathcal{U}}{\mathcal{L}}\frac{\partial U_i'}{\partial x_j'} + \frac{\mathcal{U}}{\mathcal{L}}\frac{\partial U_j'}{\partial x_i'}\right)\right) + \frac{\mathcal{U}^2}{\mathcal{L}}F_i',$$

which can be further simplified to

$$\frac{\mathcal{U}}{\mathcal{T}}\frac{\partial U_i'}{\partial t'} + \frac{\mathcal{U}^2}{\mathcal{L}}U_j'\frac{\partial U_i'}{\partial x_j'} = -\frac{\mathcal{U}^2}{\mathcal{L}}\frac{\partial P^{*\prime}}{\partial x_i'} + \frac{\mathcal{U}}{\mathcal{L}^2}\frac{\partial}{\partial x_j'}\left((\nu + \mathcal{U}\mathcal{L}\nu_t')\left(\frac{\partial U_i'}{\partial x_j'} + \frac{\partial U_j'}{\partial x_i'}\right)\right) + \frac{\mathcal{U}^2}{\mathcal{L}}F_i'.$$

Again, multiplying through by $\mathcal{L}/\mathcal{U}^2$,

$$\frac{\mathcal{L}}{\mathcal{U}\mathcal{T}}\frac{\partial U_i'}{\partial t'} + U_j'\frac{\partial U_i'}{\partial x_j'} = -\frac{\partial P^{*\prime}}{\partial x_i'} + \frac{1}{\mathcal{U}\mathcal{L}}\frac{\partial}{\partial x_j'}\left((\nu + \mathcal{U}\mathcal{L}\nu_t')\left(\frac{\partial U_i'}{\partial x_j'} + \frac{\partial U_j'}{\partial x_i'}\right)\right) + F_i',$$

and manipulating the viscous term into the form

$$\frac{\mathcal{L}}{\mathcal{U}\mathcal{T}}\frac{\partial U_i'}{\partial t'} + U_j'\frac{\partial U_i'}{\partial x_j'} = -\frac{\partial P^{*\prime}}{\partial x_i'} + \frac{\partial}{\partial x_j'}\left(\left(\frac{\nu}{\mathcal{U}\mathcal{L}} + \nu_t'\right)\left(\frac{\partial U_i'}{\partial x_j'} + \frac{\partial U_j'}{\partial x_i'}\right)\right) + F_i',$$

allows the whole expression to be given in the same form as Equation 3 in Laan et al. [86],

$$\frac{\mathcal{L}}{\mathcal{U}\mathcal{T}}\frac{\partial U_i'}{\partial t'} + U_j'\frac{\partial U_i'}{\partial x_j'} = -\frac{\partial P^{*\prime}}{\partial x_i'} + \frac{\partial}{\partial x_j'}\left(\left(\frac{1}{\mathrm{Re}} + \nu_t'\right)\left(\frac{\partial U_i'}{\partial x_j'} + \frac{\partial U_j'}{\partial x_i'}\right)\right) + F_i', \tag{A.12}$$

with the addition of the unsteady term for its applications in URANS and converging numerically unstable flows.

The normalised RANS equations are also independent of the velocity and length scales if the same conditions apply as for the Navier-Stokes equations, and

4. The eddy viscosity, ν_t, scales with $\mathcal{U}\mathcal{L}$

$$\nu_t \propto \mathcal{U}\mathcal{L}, \tag{A.13}$$

i.e. ν_t' is non-dimensional.

One can show that this condition does hold for the k-ε-f_P EVM for atmospheric flows by examining the scaling of the terms with the turbulent velocity scale, $u_* \propto \mathcal{U}$, and length, $\ell \propto \mathcal{L}$. The turbulent kinetic energy is given by Equation 2.15 as

$$k = \frac{u_*^2}{\sqrt{C_\mu}} \propto \mathcal{U}^2, \tag{A.14}$$

and the dissipation of k is given by Equation 2.16 as

$$\varepsilon = \frac{u_*^3}{\kappa(z + z_0)} \propto \frac{\mathcal{U}^3}{\mathcal{L}}. \tag{A.15}$$

These terms are sufficient for the standard k-ε EVM model but the k-ε-f_P EVM model includes a flow-dependent C_μ^* in the equation for nu_t. This $C_\mu^* = C_\mu f_P$ can be shown, however, is based on the dimensionless shear parameter, σ, as, per Equation 2.11

$$f_P = \frac{2f_0}{1 + \sqrt{1 + 4f_0(f_0 - 1)\left(\frac{\sigma}{\tilde{\sigma}}\right)}}, \quad \text{where} \quad f_0 = \frac{C_R}{C_R - 1}. \tag{A.16}$$

Removing the constants from this expression, one can show that

$$f_p \propto \frac{1}{\sqrt{\left(\frac{\sigma}{\tilde{\sigma}}\right)^2}} = \frac{\tilde{\sigma}}{\sigma} \tag{A.17}$$

where

$$\tilde{\sigma} = \frac{1}{\sqrt{C_\mu}}, \quad \text{and} \quad \sigma \equiv \frac{k}{\varepsilon}\sqrt{(U_{i,j})^2} = \frac{k}{\varepsilon}\sqrt{\frac{\partial U_i}{\partial x_j}\frac{\partial U_i}{\partial x_j}}. \tag{A.18}$$

From the scaling in Equations A.14 and A.15, $k/\varepsilon \propto \mathcal{L}/\mathcal{U}$. This then cancels with $\sqrt{(U_{i,j})^2} \propto \mathcal{U}/\mathcal{L}$ in Equation A.18, which means σ, and, via Equation A.16 and Equation 2.10, f_P and C_μ^*, are independent of $\mathcal{U}$ and $\mathcal{L}$. As the turbulent eddy viscosity is defined as

$$\nu_t = C_\mu^* \frac{k^2}{\varepsilon} \propto \frac{\mathcal{U}^4}{\frac{\mathcal{U}^3}{\mathcal{L}}} = \mathcal{U}\mathcal{L}. \tag{A.19}$$

APPENDIX B
Derivation of streamwise analytical solution

Bastankhah et al. [14] presents a solution for the streamwise velocity distribution within a wind farm with arbitrary layout. Within the confines of the journal article the full derivation cannot be presented, however, the streamwise solution is an integral part of the derivation of the lateral velocity solution in this thesis, and therefore the full derivation is deemed necessary for the integrity of the proposed lateral solution. This appendix presents that derivation. Please note that the notation used here is the same as that used in Bastankhah et al. [14] and is thus different from the rest of this thesis. This is to make clear the steps in the context of the original paper. Furthermore, this appendix is by no means intended as a novel piece of work, in fact it seeks to use the same language as the original paper wherever possible, such that the additional steps not shown there are more evident. This appendix seeks only to fill out that original derivation so the steps taken are clear when the same method is applied for the lateral velocity in this paper. In addition, whilst this thesis, like Bastankhah et al. [14] assumes a Gaussian wake profile, this is by no means the only available wake profile, and this derivation has also been performed for a super-Gaussian profile by Blondel [19]. Finally, this model has been implemented in Python as a part of this thesis and the implementation is available at `https://github.com/NilsGaukroger/Analytical-solution-for-the-cumulative-wake-of-yawed-wind-turbines/blob/main/streamwiseSolution.py`.

B.1 Integral form of governing equations for turbine wakes within a wind farm

Let us assume a wind farm with an arbitrary layout of n wind turbines (WT_1, WT_2, …, WT_i, …, WT_n) immersed in a turbulent boundary layer flow with a velocity profile denoted by U_0. The position of WT_i is denoted by $\boldsymbol{X}_i = (x_i, y_i, z_i)$, where x, y, and z are the streamwise, spanwise, and vertical directions in the coordinate system, respectively. Turbines are labelled with respect to their streamwise positions such that $x_i \geq x_{i-1}$, where $i = \{2, 3, \ldots, n\}$. The Reynolds-averaged Navier-Stokes (known as RANS) equation in the streamwise direction at high Reynolds numbers (neglecting viscosity effects) is given by [128, 129]

$$U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} + W\frac{\partial U}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial x} - \frac{\partial \overline{u^2}}{\partial x} - \frac{\partial \overline{uv}}{\partial y} - \frac{\partial \overline{uw}}{\partial z} + \sum_{i=1}^{n} f_i, \tag{B.1}$$

where U, V, and W are the time-averaged streamwise (x), lateral (y), and vertical (z) velocity components, respectively. Turbulent velocity fluctuations are represented by u, v, and w and the overbar denotes time averaging. Also, P is the time-averaged static pressure and ρ is the air density. The term f_i represents the effect of the thrust force of WT_i on the horizontal

momentum and is given by

$$f_i = -T_i/(\rho\pi R^2)\delta(x - x_i)H\left(R^2 - \left[(y - y_i)^2 + (z - z_i)^2\right]\right), \tag{B.2}$$

where T_i is the magnitude of the thrust force of the WT_i in the streamwise direction, R is the turbine radius, $\delta(x)$ is the Dirac delta function and $H(x)$ is the Heaviside step function. Using the incoming boundary-layer profile $U_0(z)$, Equation B.1 can be written as

$$U\frac{\partial(U_0 - U)}{\partial x} + V\frac{\partial(U_0 - U)}{\partial y} + W\frac{\partial(U_0 - U)}{\partial z} = \frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{\partial\overline{u^2}}{\partial x} + \frac{\partial\overline{uv}}{\partial y} + \frac{\partial\overline{uw}}{\partial z} + W\frac{dU_0}{dz} - \sum_{i=1}^{n} f_i, \tag{B.3}$$

where $W\frac{dU_0}{dz}$ is the only additional term on the right-hand side because $U_0(z)$ and thus $\frac{dU_0}{dx} = \frac{dU_0}{dy} = 0$. From the continuity equation, we know that

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} = 0. \tag{B.4}$$

Multiplying Equation B.4 by $(U_0 - U)$ and adding the product to the left-hand side of Equation B.3 yields

$$\frac{\partial U}{\partial x}(U_0 - U) + \frac{\partial V}{\partial y}(U_0 - U) + \frac{\partial W}{\partial z}(U_0 - U) + U\frac{\partial(U_0 - U)}{\partial x} + V\frac{\partial(U_0 - U)}{\partial y} + W\frac{\partial(U_0 - U)}{\partial z} = \frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{\partial\overline{u^2}}{\partial x} + \frac{\partial\overline{uv}}{\partial y} + \frac{\partial\overline{uw}}{\partial z} + W\frac{dU_0}{dz} - \sum_{i=1}^{n} f_i. \tag{B.5}$$

The terms on the left-hand side can then be combined using the product rule (e.g. $U\frac{\partial(U_0-U)}{\partial x} + (U_0 - U)\frac{\partial U}{\partial x} = \frac{\partial U(U_0-U)}{\partial x}$) to give

$$\frac{\partial U(U_0 - U)}{\partial x} + \frac{\partial V(U_0 - U)}{\partial y} + \frac{\partial W(U_0 - U)}{\partial z} = \frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{\partial\overline{u^2}}{\partial x} + \frac{\partial\overline{uv}}{\partial y} + \frac{\partial\overline{uw}}{\partial z} + W\frac{dU_0}{dz} - \sum_{i=1}^{n} f_i. \tag{B.6}$$

Next, Equation B.6 is integrated from x_a to x_b with respect to x, from y_a to y_b with respect to y, and from z_a to z_b with respect to z, where $x_a \ll x_n < x_b$, $y_a \ll y_n \ll y_b$ and $0 \ll z_a < z_n \ll z_b$. Note that $z_a \gg 0$ to ensure that the assumption of negligible viscous forces is valid. The value of z_a can be equal to zero if the Reynolds shear stresses in Equation B.6 are replaced with the total shear stresses (i.e. the sum of turbulent and viscous shear stresses).

Without loss of generality, we assume that the integration box includes WT_n and an arbitrary number of upwind turbines as shown in Figure B.1. Integrating Equation B.6 first from x_a to x_b with respect to x produces

$$U_n(U_0 - U_n)\Big|_{x_a}^{x_b} + \int_{x_a}^{x_b}\frac{\partial V_n(U_0 - U_n)}{\partial y}\,\mathrm{d}x + \int_{x_a}^{x_b}\frac{\partial W_n(U_0 - U_n)}{\partial z}\,\mathrm{d}x = \frac{1}{\rho}P_n\Big|_{x_a}^{x_b} + \overline{u^2}_n\Big|_{x_a}^{x_b} + \int_{x_a}^{x_b}\frac{\partial\overline{uv}_n}{\partial y}\,\mathrm{d}x + \int_{x_a}^{x_b}\frac{\partial\overline{uw}_n}{\partial z}\,\mathrm{d}x + \int_{x_a}^{x_b} W_n\frac{dU_0}{dz}\,\mathrm{d}x - \int_{x_a}^{x_b}\sum_{i=1}^{n} f_i. \tag{B.7}$$

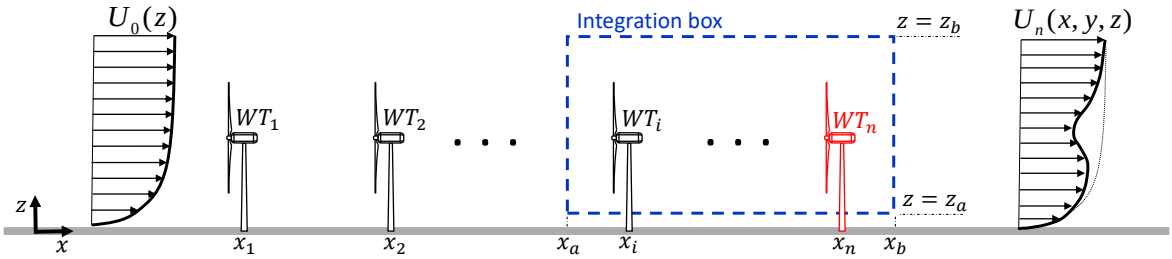


Figure B.1: Schematic of a wind farm with an arbitrary layout consisting of n wind turbines (WT_1, WT_2, …, WT_n) immersed in a turbulent boundary-layer flow. The momentum equation Equation B.6 is integrated over the shown box. The integration is performed with and without WT_n, shown in the figure in red. Figure reprinted from Bastankhah et al. [14].

The final term can be developed by substituting the definition of f_i from Equation B.2, such that

$$\begin{aligned}
&\int_{x_a}^{x_b} \sum_{i=1}^{n} \frac{-T_i}{\rho \pi R^2} \delta(x - x_i) H\left(R^2 - \left[(y - y_i)^2 + (z - z_i)^2\right]\right) \, \mathrm{d}x \\
&= \sum_{i=1}^{n} \frac{-T_i}{\rho \pi R^2} H\left(R^2 - \left[(y - y_i)^2 + (z - z_i)^2\right]\right) \int_{x_a}^{x_b} \delta(x - x_i) \, \mathrm{d}x \\
&= \sum_{i=1}^{n} \frac{-T_i}{\rho \pi R^2} H\left(R^2 - \left[(y - y_i)^2 + (z - z_i)^2\right]\right),
\end{aligned} \tag{B.8}$$

and so

$$\begin{aligned}
&U_n(U_0 - U_n)\Big|_{x_a}^{x_b} + \int_{x_a}^{x_b} \frac{\partial V_n (U_0 - U_n)}{\partial y} \, \mathrm{d}x + \int_{x_a}^{x_b} \frac{\partial W_n (U_0 - U_n)}{\partial z} \, \mathrm{d}x \\
&\qquad = \frac{1}{\rho} P_n \Big|_{x_a}^{x_b} + \overline{u^2}_n \Big|_{x_a}^{x_b} + \int_{x_a}^{x_b} \frac{\partial \overline{uv}_n}{\partial y} \, \mathrm{d}x + \int_{x_a}^{x_b} \frac{\partial \overline{uw}_n}{\partial z} \, \mathrm{d}x \\
&\qquad\quad + \int_{x_a}^{x_b} W_n \frac{dU_0}{dz} \, \mathrm{d}x + \sum_{i=1}^{n} \frac{T_i}{\rho \pi R^2} H\left(R^2 - \left[(y - y_i)^2 + (z - z_i)^2\right]\right).
\end{aligned} \tag{B.9}$$

Integrating Equation B.9 from y_a to y_b with respect to y then yields

$$\begin{aligned}
&\int_{y_a}^{y_b} U_n(U_0 - U_n)\Big|_{x_a}^{x_b} \, \mathrm{d}y + \int_{x_a}^{x_b} V_n(U_0 - U_n)\Big|_{y_a}^{y_b} \, \mathrm{d}x + \int_{y_a}^{y_b} \int_{x_a}^{x_b} \frac{\partial W_n (U_0 - U_n)}{\partial z} \, \mathrm{d}x \, \mathrm{d}y \\
&\qquad = \frac{1}{\rho} \int_{y_a}^{y_b} P_n \Big|_{x_a}^{x_b} \, \mathrm{d}y + \int_{y_a}^{y_b} \overline{u^2}_n \Big|_{x_a}^{x_b} \, \mathrm{d}y + \int_{x_a}^{x_b} \overline{uv}_n \Big|_{y_a}^{y_b} \, \mathrm{d}x + \int_{y_a}^{y_b} \int_{x_a}^{x_b} \frac{\partial \overline{uw}_n}{\partial z} \, \mathrm{d}x \, \mathrm{d}y \\
&\qquad\quad + \int_{y_a}^{y_b} \int_{x_a}^{x_b} W_n \frac{dU_0}{dz} \, \mathrm{d}x \, \mathrm{d}y + \sum_{i=1}^{n} \frac{T_i}{\rho \pi R^2} \int_{y_a}^{y_b} H\left(R^2 - \left[(y - y_i)^2 + (z - z_i)^2\right]\right) \, \mathrm{d}y.
\end{aligned} \tag{B.10}$$

The second term on the left-hand side, $\int_{x_a}^{x_b} V_n(U_0 - U_n)\Big|_{y_a}^{y_b} \mathrm{d}x$, is in fact equal to zero, i.e.

$$V_n(U_0 - U_n)|_{y_b \gg y_n} - V_n(U_0 - U_n)|_{y_a \ll y_n} = 0, \tag{B.11}$$

because the velocity deficit, $(U_0 - U_n)$, tends to zero as the influence of the presence of the turbine diminishes with increasing lateral distance from the turbine at y_n. Moreover, by relating it to mean flow shear via the Boussinesq hypothesis in Equation 2.5, one can show that

$$\int_{x_a}^{x_b} \overline{uv}_n \Big|_{y_a}^{y_b} \mathrm{d}x = 0. \tag{B.12}$$

This is because this Reynolds shear stress is proportional to $\frac{\partial U}{\partial y}$ according to the Boussinesq hypothesis, which also tends to zero with increasing distance from the turbine at y_n, thus

$$\left.\frac{\partial U}{\partial y}\right|_{y_b \gg y_n} - \left.\frac{\partial U}{\partial y}\right|_{y_a \ll y_n} = \overline{uv}_n|_{y_b \gg y_n} - \overline{uv}_n|_{y_a \ll y_n} = 0. \tag{B.13}$$

The same cannot be said of the other Reynolds stress terms, the normal stress corresponds to $\frac{\partial U}{\partial x}$, which is affected by the turbine thrust forces, and the vertical shear stress corresponds to $\frac{\partial U}{\partial z}$, which is determined by the inflow profile. Given these simplifications, Equation B.10 then becomes

$$\begin{aligned}
&\int_{y_a}^{y_b} U_n(U_0 - U_n)\Big|_{x_a}^{x_b} \mathrm{d}y + \int_{y_a}^{y_b}\int_{x_a}^{x_b} \frac{\partial W_n(U_0 - U_n)}{\partial z} \,\mathrm{d}x\,\mathrm{d}y \\
&\qquad = \frac{1}{\rho}\int_{y_a}^{y_b} P_n\Big|_{x_a}^{x_b} \mathrm{d}y + \int_{y_a}^{y_b} \overline{u^2}_n\Big|_{x_a}^{x_b} \mathrm{d}y + \int_{y_a}^{y_b}\int_{x_a}^{x_b} \frac{\partial \overline{uw}_n}{\partial z} \,\mathrm{d}x\,\mathrm{d}y \\
&\quad + \int_{y_a}^{y_b}\int_{x_a}^{x_b} W_n \frac{dU_0}{dz} \,\mathrm{d}x\,\mathrm{d}y + \sum_{i=1}^{n} \frac{T_i}{\rho\pi R^2} \int_{y_a}^{y_b} H\left(R^2 - \left[(y - y_i)^2 + (z - z_i)^2\right]\right) \mathrm{d}y.
\end{aligned} \tag{B.14}$$

Lastly, integrating Equation B.14 from z_a to z_b with respect to z yields

$$\begin{aligned}
&\int_{z_a}^{z_b}\int_{y_a}^{y_b} U_n(U_0 - U_n)\Big|_{x_a}^{x_b} \mathrm{d}y\,\mathrm{d}z + \int_{y_a}^{y_b}\int_{x_a}^{x_b} W_n(U_0 - U_n)\Big|_{z_a}^{z_b} \mathrm{d}x\,\mathrm{d}y \\
&\qquad = \frac{1}{\rho}\int_{z_a}^{z_b}\int_{y_a}^{y_b} P_n\Big|_{x_a}^{x_b} \mathrm{d}y\,\mathrm{d}z + \int_{z_a}^{z_b}\int_{y_a}^{y_b} \overline{u^2}_n\Big|_{x_a}^{x_b} \mathrm{d}y\,\mathrm{d}z + \int_{y_a}^{y_b}\int_{x_a}^{x_b} \overline{uw}_n\Big|_{z_a}^{z_b} \mathrm{d}x\,\mathrm{d}y \\
&+ \int_{z_a}^{z_b}\int_{y_a}^{y_b}\int_{x_a}^{x_b} W_n \frac{dU_0}{dz} \,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z + \sum_{i=1}^{n} \frac{T_i}{\rho\pi R^2} \int_{z_a}^{z_b}\int_{y_a}^{y_b} H\left(R^2 - \left[(y - y_i)^2 + (z - z_i)^2\right]\right) \mathrm{d}y\,\mathrm{d}z.
\end{aligned} \tag{B.15}$$

The final term on the right-hand side can also now be simplified. Physically, the Heaviside step function represents the rotor disc in the y-z plane, and so the integration here over y and z

with the given limits means

$$\begin{aligned}&\sum_{i=1}^{n}\frac{T_i}{\rho\pi R^2}\int_{z_a}^{z_b}\int_{y_a}^{y_b}H\left(R^2-\left[(y-y_i)^2+(z-z_i)^2\right]\right)\,\mathrm{d}y\,\mathrm{d}z\\&=\sum_{i=1}^{n}\frac{T_i}{\rho\pi R^2}(\pi R^2)=\sum_{i=1}^{n}\frac{T_i}{\rho}.\end{aligned}\tag{B.16}$$

Equation B.15 can then be written as

$$\begin{aligned}&\int_{z_a}^{z_b}\int_{y_a}^{y_b}U_n(U_0-U_n)\Big|_{x_a}^{x_b}\,\mathrm{d}y\,\mathrm{d}z+\int_{y_a}^{y_b}\int_{x_a}^{x_b}W_n(U_0-U_n)\Big|_{z_a}^{z_b}\,\mathrm{d}x\,\mathrm{d}y\\&\quad=\frac{1}{\rho}\int_{z_a}^{z_b}\int_{y_a}^{y_b}P_n\Big|_{x_a}^{x_b}\,\mathrm{d}y\,\mathrm{d}z+\int_{z_a}^{z_b}\int_{y_a}^{y_b}\overline{u^2}_n\Big|_{x_a}^{x_b}\,\mathrm{d}y\,\mathrm{d}z+\int_{y_a}^{y_b}\int_{x_a}^{x_b}\overline{uw}_n\Big|_{z_a}^{z_b}\,\mathrm{d}x\,\mathrm{d}y\\&\qquad+\int_{z_a}^{z_b}\int_{y_a}^{y_b}\int_{x_a}^{x_b}W_n\frac{dU_0}{dz}\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z+\sum_{i=1}^{n}\frac{T_i}{\rho}.\end{aligned}\tag{B.17}$$

Rearranging this for the thrust force, one obtains (2.6) from Bastankhah et al. [14],

$$\begin{aligned}\sum_{i\in B}\frac{T_i}{\rho}&=\int U_n(U_0-U_n)\,\mathrm{d}A\Big|_{x_a}^{x_b}-\frac{1}{\rho}\int P_n\Big|_{x_a}^{x_b}\,\mathrm{d}A\\&\quad-\int\overline{u^2}_n\,\mathrm{d}A\Big|_{x_a}^{x_b}-\int\overline{uw}_n\,\mathrm{d}x\,\mathrm{d}y\Big|_{z_a}^{z_b}-\int\frac{dU_0}{dz}W_n\,\mathrm{d}V,\end{aligned}\tag{B.18}$$

where i is a member of set B if WT_i is inside the integration box, $\mathrm{d}A$ is $\mathrm{d}y\,\mathrm{d}z$ and $\mathrm{d}V$ is $\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$. In Equation B.18 and hereafter, any velocity or pressure term with a subscript i denotes the value of the given variable in the presence of WT_1, WT_2, …, WT_i. Now, we perform the same integration once more but this time in the absence of WT_n. This leads to

$$\begin{aligned}\sum_{i\in B'}\frac{T_i}{\rho}&=\int U_{n-1}(U_0-U_{n-1})\,\mathrm{d}A\Big|_{x_a}^{x_b}-\frac{1}{\rho}\int P_{n-1}\Big|_{x_a}^{x_b}\,\mathrm{d}A\\&\quad-\int\overline{u^2}_{n-1}\,\mathrm{d}A\Big|_{x_a}^{x_b}-\int\overline{uw}_{n-1}\,\mathrm{d}x\,\mathrm{d}y\Big|_{z_a}^{z_b}-\int\frac{dU_0}{dz}W_{n-1}\,\mathrm{d}V,\end{aligned}\tag{B.19}$$

where set B' is equal to set B excluding n (i.e. $B\setminus\{n\}=\{i:i\in B\text{ and }i\notin\{n\}\}$). As $x_a\ll x_n$, surface integrals at $x=x_a$ provide the same results in both Equation B.18 and Equation B.19.

By subtracting Equation B.19 from Equation B.18, we obtain

$$\underbrace{\frac{T_n}{\rho}}_{\text{Thrust}} = \underbrace{\left[\int U_n(U_0 - U_n)\ \mathrm{d}A - \int U_{n-1}(U_0 - U_{n-1})\ \mathrm{d}A\right]}_{\text{Momentum deficit}} \underbrace{-\frac{1}{\rho}\int (P_n - P_{n-1})\ \mathrm{d}A}_{\text{Pressure}}$$
$$\underbrace{-\int \left(\overline{u^2}_n - \overline{u^2}_{n-1}\right)\ \mathrm{d}A}_{\text{Reynolds normal stress}} \underbrace{-\int \left(\overline{uw}_n - \overline{uw}_{n-1}\right)\Big|_{z_a}^{z_b}\ \mathrm{d}x\,\mathrm{d}y}_{\text{Reynolds shear stress}}$$
$$\underbrace{-\int \frac{dU_0}{dz}\left(W_n - W_{n-1}\right)\ \mathrm{d}V}_{\text{Mean flow shear}}, \quad \text{(B.20)}$$

where $\mathrm{d}A$ in Equation B.20 is $\mathrm{d}y\,\mathrm{d}z$ at $x = x_b$. Bastankhah et al. [14] then uses an LES budget analysis to make the approximation that the momentum deficit is the dominant term in this equation, and thus that Equation B.20 can be approximated as Equation B.21.

B.2 General solution

We seek a solution for U_n in the following 'conservation of momentum deficit' equation

$$\rho \int U_n(U_0 - U_n)\ \mathrm{d}A - \rho \int U_{n-1}(U_0 - U_{n-1})\ \mathrm{d}A \approx T_n. \quad \text{(B.21)}$$

Unlike for single, isolated turbines, the definition of velocity deficit with respect to the incoming flow (i.e. $U_{in,n}$) is not suitable for turbines within wind farms [14]. Instead, we define the velocity deficit at a given position $\mathbf{X} = (x, y, z)$ downwind of WT_n as the difference between the streamwise velocity in the absence, and in the presence, of WT_n at $\mathbf{X}$; i.e.

$$\Delta U_n(\mathbf{X}) = U_{n-1}(\mathbf{X}) - U_n(\mathbf{X}). \quad \text{(B.22)}$$

As $(U_{n-1} - U_n)$ is self-similar [14], we can write

$$U_{n-1} - U_n = C_n(x) f_n(\mathbf{X}), \quad \text{(B.23)}$$

where f_n is the self-similar function. Shifting the index n in Equation B.23 to $n-1, n-2, \ldots, 1$ leads to a set of equations as follows:

$$\begin{aligned} U_{n-2} - U_{n-1} &= C_{n-1} f_{n-1}, \\ U_{n-3} - U_{n-2} &= C_{n-2} f_{n-2}, \\ &\vdots \\ U_0 - U_1 &= C_1 f_1. \end{aligned} \quad \text{(B.24)}$$

Adding Equation B.23 and Equation B.24 results in

$$\begin{aligned} (U_0 - U_1) + (U_1 - U_2) + \ldots + (U_{n-2} - U_{n-1}) + (U_{n-1} - U_n) \\ = C_1 f_1 + C_2 f_2 + \ldots + C_{n-2} f_{n-2} + C_{n-1} f_{n-1} + C_n f_n, \end{aligned}$$

$$U_0 - U_n = \sum_{i=1}^{n} C_i f_i = \sum_{i=1}^{n-1} C_i f_i + C_n f_n. \tag{B.25}$$

This can also be written as

$$U_n = U_0 - \sum_{i=1}^{n} C_i f_i = U_0 - \sum_{i=1}^{n-1} C_i f_i - C_n f_n, \tag{B.26}$$

or, substituting $n = n - 1$, as

$$U_0 - U_{n-1} = \sum_{i=1}^{n-1} C_i f_i = \sum_{i=1}^{n} C_i f_i - C_n f_n, \tag{B.27}$$

or

$$U_{n-1} = U_0 - \sum_{i=1}^{n-1} C_i f_i = U_0 - \sum_{i=1}^{n} C_i f_i + C_n f_n. \tag{B.28}$$

Substituting Equation B.25 to Equation B.28 into Equation B.21 yields

$$\int \left(U_0 - \sum_{i=1}^{n-1} C_i f_i - C_n f_n\right) \left(\sum_{i=1}^{n-1} C_i f_i + C_n f_n\right) \, \mathrm{d}A - \int \left(U_0 - \sum_{i=1}^{n-1} C_i f_i\right) \left(\sum_{i=1}^{n-1} C_i f_i\right) \, \mathrm{d}A \approx \frac{T_n}{\rho}.$$

Expanding all the terms then produces

$$\int U_0 \sum_{i=1}^{n-1} C_i f_i + U_0 C_n f_n - \left(\sum_{i=1}^{n-1} C_i f_i\right)^2 - C_n f_n \sum_{i=1}^{n-1} C_i f_i - C_n f_n \sum_{i=1}^{n-1} C_i f_i - (C_n f_n)^2 - U_0 \sum_{i=1}^{n-1} C_i f_i + \left(\sum_{i=1}^{n-1} C_i f_i\right)^2 \, \mathrm{d}A \approx \frac{T_n}{\rho},$$

which simplifies to

$$\int U_0 C_n f_n - 2 C_n f_n \sum_{i=1}^{n-1} C_i f_i - (C_n f_n)^2 \, \mathrm{d}A \approx \frac{T_n}{\rho}.$$

Factorising by $C_n f_n$ and assuming the two sides are exactly, not approximately, equal, yields [14, (4.6)]

$$\int C_n f_n \left(U_0 - C_n f_n - 2 \sum_{i=1}^{n-1} C_i f_i\right) \, \mathrm{d}A = \frac{T_n}{\rho}. \tag{B.29}$$

B.3 General solution: Modified form

Bastankhah et al. [14] also proposes a modified form of Equation B.21, which reads

$$\rho \int U_n (U_{n-1} - U_n) \, \mathrm{d}A \approx T_n. \tag{B.30}$$

Substituting Equation B.23 and Equation B.26 into Equation B.30, assuming the approximation is in fact exact, and rearranging, yields an equation of the same form as Equation B.29,

$$\int C_n f_n \left(U_0 - C_n f_n - \sum_{i=1}^{n-1} C_i f_i \right) \, \mathrm{d}A = \frac{T_n}{\rho}. \tag{B.31}$$

The solution obtained from this modified form is almost the same as that from the original form (see Appendix B.2, the only difference being that λ_{ni} is modified to

$$\lambda_{ni}^{modified} = \frac{\lambda_{ni}}{2}. \tag{B.32}$$

B.4 Specific solution: Gaussian model

The self-similar function used to describe the wake profile can take many forms. From the top-hat distribution of Jensen [72], to the Super-Gaussian of Blondel and Cathelain [18], each leads to a slightly different solution for C_n. Following Bastankhah et al. [14], we start with the Gaussian,

$$f_i = \exp\left(-\frac{(y - y_i)^2}{2\sigma_i^2} \right) \exp\left(-\frac{(z - z_i)^2}{2\sigma_i^2} \right). \tag{B.33}$$

Note that, for simplicity, this work assume that the wake width is the same both the lateral and vertical directions, but substituting σ^2 with the product of σ_y and σ_z hereafter would allow for this assumption to be overcome.

A few integrals should be defined before proceeding. The first is the surface integral of the self-similar function,

$$\int f_n \, \mathrm{d}A = \iint_{-\infty}^{+\infty} \exp\left(-\frac{(y - y_n)^2}{2\sigma_n^2} \right) \exp\left(-\frac{(z - z_n)^2}{2\sigma_n^2} \right) \mathrm{d}y \, \mathrm{d}z. \tag{B.34}$$

This integral can be split into

$$\int_{-\infty}^{+\infty} \exp\left(-\frac{(y - y_n)^2}{2\sigma_n^2} \right) \, \mathrm{d}y \int_{-\infty}^{+\infty} \exp\left(-\frac{(z - z_n)^2}{2\sigma_n^2} \right) \, \mathrm{d}z, \tag{B.35}$$

and then reformulated as

$$\int_{-\infty}^{+\infty} \exp\left(-\left(\frac{(y - y_n)}{\sqrt{2}\sigma_n} \right)^2 \right) \, \mathrm{d}y \int_{-\infty}^{+\infty} \exp\left(-\left(\frac{(z - z_n)}{\sqrt{2}\sigma_n} \right)^2 \right) \, \mathrm{d}z, \tag{B.36}$$

to allow for the substitution of the solution

$$\int_{-\infty}^{+\infty} e^{-x^2} \, \mathrm{d}x = \sqrt{\pi}, \tag{B.37}$$

known as the Gaussian integral. The creation of two temporary variables

$$\xi = \frac{y - y_n}{\sqrt{2}\sigma_n} \quad \text{and} \quad \zeta = \frac{z - z_n}{\sqrt{2}\sigma_n}, \tag{B.38}$$

where

$$\mathrm{d}y = \sqrt{2}\sigma_n \,\mathrm{d}\xi \quad \text{and} \quad \mathrm{d}z = \sqrt{2}\sigma_n \,\mathrm{d}\zeta, \tag{B.39}$$

allows Equation B.36 to be written in the form

$$\int f_n \,\mathrm{d}A = \sqrt{2}\sigma_n \int_{-\infty}^{+\infty} \exp\left(-\xi^2\right) \mathrm{d}\xi \cdot \sqrt{2}\sigma_n \int_{-\infty}^{+\infty} \exp\left(-\zeta^2\right) \mathrm{d}\zeta, \tag{B.40}$$

and, from the definition of the Gaussian integral, Equation B.37, this yields

$$\boxed{\int f_n \,\mathrm{d}A = \left(\sqrt{2}\sigma_n\sqrt{\pi}\right)^2 = 2\pi\sigma_n^2}. \tag{B.41}$$

The second of these useful integrals is

$$\int f_n^2 \,\mathrm{d}A = \iint_{-\infty}^{+\infty} \left(\exp\left(-\frac{(y-y_n)^2}{2\sigma_n^2}\right)\exp\left(-\frac{(z-z_n)^2}{2\sigma_n^2}\right)\right)^2 \mathrm{d}y\,\mathrm{d}z, \tag{B.42}$$

which can again be divided into

$$\int_{-\infty}^{+\infty} \left(\exp\left(-\frac{(y-y_n)^2}{2\sigma_n^2}\right)\right)^2 \mathrm{d}y \int_{-\infty}^{+\infty} \left(\exp\left(-\frac{(z-z_n)^2}{2\sigma_n^2}\right)\right)^2 \mathrm{d}z. \tag{B.43}$$

Using the rules of exponentials this can simplified to

$$\int_{-\infty}^{+\infty} \exp\left(-\frac{(y-y_n)^2}{\sigma_n^2}\right) \mathrm{d}y \int_{-\infty}^{+\infty} \exp\left(-\frac{(z-z_n)^2}{\sigma_n^2}\right) \mathrm{d}z, \tag{B.44}$$

where the argument of the exponential is the same as in Equation B.36 but for a factor of 1/2. Thus very similar temporary variables, this time

$$\xi = \frac{y-y_n}{\sigma_n} \quad \text{and} \quad \zeta = \frac{z-z_n}{\sigma_n}, \tag{B.45}$$

where now

$$\mathrm{d}y = \sigma_n \,\mathrm{d}\xi \quad \text{and} \quad \mathrm{d}z = \sigma_n \,\mathrm{d}\zeta, \tag{B.46}$$

allow Equation B.44 to be written in the form

$$\int f_n^2 \,\mathrm{d}A = \sigma_n \int_{-\infty}^{+\infty} \exp\left(-\xi^2\right) \mathrm{d}\xi \cdot \sigma_n \int_{-\infty}^{+\infty} \exp\left(-\zeta^2\right) \mathrm{d}\zeta. \tag{B.47}$$

Finally, inserting the Gaussian integral solution yields

$$\boxed{\int f_n^2 \,\mathrm{d}A = \left(\sigma_n\sqrt{\pi}\right)^2 = \pi\sigma_n^2}. \tag{B.48}$$

The last useful integral is

$$\int f_n f_i \,\mathrm{d}A = \iint_{-\infty}^{+\infty} \exp\left(-\frac{(y-y_n)^2}{2\sigma_n^2}\right)\exp\left(-\frac{(z-z_n)^2}{2\sigma_n^2}\right) \exp\left(-\frac{(y-y_i)^2}{2\sigma_i^2}\right)\exp\left(-\frac{(z-z_i)^2}{2\sigma_i^2}\right) \mathrm{d}y\,\mathrm{d}z, \tag{B.49}$$

which can also be written

$$\int_{-\infty}^{+\infty} \exp\left(-\left(\frac{(y-y_n)^2}{2\sigma_n^2}+\frac{(y-y_i)^2}{2\sigma_i^2}\right)\right)\,\mathrm{d}y \int_{-\infty}^{+\infty} \exp\left(-\left(\frac{(z-z_n)^2}{2\sigma_n^2}+\frac{(z-z_i)^2}{2\sigma_i^2}\right)\right)\,\mathrm{d}z. \quad \text{(B.50)}$$

Combining the fractions in the arguments then gives

$$\int_{-\infty}^{+\infty} \exp\left(-\left(\frac{\sigma_i^2(y-y_n)^2+\sigma_n^2(y-y_i)^2}{2\sigma_n^2\sigma_i^2}\right)\right)\,\mathrm{d}y \int_{-\infty}^{+\infty} \exp\left(-\left(\frac{\sigma_i^2(z-z_n)^2+\sigma_n^2(z-z_i)^2}{2\sigma_n^2\sigma_i^2}\right)\right)\,\mathrm{d}z. \quad \text{(B.51)}$$

At this point, for the sake of simplicity we will assume that all turbines are aligned both laterally and vertically, i.e. $y_1 = y_2 = \ldots = y_n$ and $z_1 = z_2 = \ldots = z_n$. This assumption means one can substitute $y_i = y_n$ and $z_i = z_n$ into Equation B.51, which then simplifies to

$$\int_{-\infty}^{+\infty} \exp\left(-\frac{\left(\sigma_n^2+\sigma_i^2\right)\left(y-y_n\right)^2}{2\sigma_i^2\sigma_n^2}\right)\,\mathrm{d}y \int_{-\infty}^{+\infty} \exp\left(-\frac{\left(\sigma_n^2+\sigma_i^2\right)\left(z-z_n\right)^2}{2\sigma_i^2\sigma_n^2}\right)\,\mathrm{d}z. \quad \text{(B.52)}$$

Again substituting temporary variables into Equation B.52

$$\xi = \frac{\sqrt{\sigma_n^2+\sigma_i^2}(y-y_n)}{\sqrt{2}\sigma_i\sigma_n} \quad \text{and} \quad \zeta = \frac{\sqrt{\sigma_n^2+\sigma_i^2}(z-z_n)}{\sqrt{2}\sigma_i\sigma_n}, \quad \text{(B.53)}$$

where

$$\mathrm{d}y = \frac{\sqrt{2}\sigma_i\sigma_n}{\sqrt{\sigma_n^2+\sigma_i^2}}\,\mathrm{d}\xi \quad \text{and} \quad \mathrm{d}z = \frac{\sqrt{2}\sigma_i\sigma_n}{\sqrt{\sigma_n^2+\sigma_i^2}}\,\mathrm{d}\zeta, \quad \text{(B.54)}$$

yields

$$\int f_n f_i\,\mathrm{d}A = \frac{\sqrt{2}\sigma_i\sigma_n}{\sqrt{\sigma_n^2+\sigma_i^2}}\int_{-\infty}^{+\infty}\exp\left(-\xi^2\right)\,\mathrm{d}\xi \cdot \frac{\sqrt{2}\sigma_i\sigma_n}{\sqrt{\sigma_n^2+\sigma_i^2}}\int_{-\infty}^{+\infty}\exp\left(-\zeta^2\right)\,\mathrm{d}\zeta. \quad \text{(B.55)}$$

Whereupon the substitution of the Gaussian integral solution this becomes

$$\int f_n f_i\,\mathrm{d}A = \left(\frac{\sqrt{2}\sigma_i\sigma_n}{\sqrt{\sigma_n^2+\sigma_i^2}\sqrt{pi}}\right)^2 = \frac{2\pi\sigma_i\sigma_n}{\sqrt{\sigma_n^2+\sigma_i^2}}. \quad \text{(B.56)}$$

Removing the assumption that all turbines are aligned adds two exponential terms to Equation B.56 such that

$$\boxed{\int f_n f_i\,\mathrm{d}A = \frac{2\pi\sigma_i^2\sigma_n^2}{\sigma_n^2+\sigma_i^2}\exp\left(-\frac{(y_n-y_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right)\exp\left(-\frac{(z_n-z_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right)}. \quad \text{(B.57)}$$

It should be noted that this last step was performed with the use of a computational integral calculator.

Considering then the original, unmodified, 'conservation of momentum deficit', Equation B.29, developing it leads to

$$\int C_n U_0 \,\mathrm{d}A \int f_n \,\mathrm{d}A - \int C_n^2 \,\mathrm{d}A \int f_n^2 \,\mathrm{d}A - \int 2C_n \sum_{i=1}^{n-1} C_i f_n f_i \,\mathrm{d}A = \frac{T_n}{\rho}, \tag{B.58}$$

into which can be substituted the integrals Equation B.41, Equation B.48, and Equation B.57 to produce

$$2\pi \int C_n \sigma_n^2 U_0 \,\mathrm{d}A - \pi \int C_n^2 \sigma_n^2 \,\mathrm{d}A - 2\pi \int C_n \sum_{i=1}^{n-1} C_i \,\mathrm{d}A \sum_{i=1}^{n-1} \frac{2\pi\sigma_n^2\sigma_i^2}{\sigma_n^2+\sigma_i^2} \exp\left(-\frac{(y_n-y_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right) \exp\left(-\frac{(z_n-z_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right) = \frac{T_n}{\rho}. \tag{B.59}$$

This can be simplified by fact that $C_n(x)$, $\sigma_n(x)$, and U_0 can be approximated with $U_h = U_0(z = z_h)$, to become

$$2\pi C_n U_0 \sigma_n^2 - \pi C_n \sigma_n^2 - 2\pi C_n \sigma_n^2 \sum_{i=1}^{n-1} C_i \frac{2\sigma_i^2}{\sigma_i+\sigma_n^2} \exp\left(-\frac{(y_n-y_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right) \exp\left(-\frac{(z_n-z_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right) = \frac{T_n}{\rho}. \tag{B.60}$$

Dividing through by $\pi\sigma_n^2$ and rearranging,

$$C_n^2 - 2C_n U_0 + 2C_n \sum_{i=1}^{n-1} C_i \frac{2\sigma_i^2}{\sigma_n^2+\sigma_i^2} \exp\left(-\frac{(y_n-y_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right) \exp\left(-\frac{(z_n-z_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right) = -\frac{T_n}{\rho\pi\sigma_n^2}. \tag{B.61}$$

Factorising then gives [14, (4.8)],

$$C_n^2 - 2\left(U_0 - \sum_{i=1}^{n-1} C_i \lambda_{ni}\right) C_n + \frac{T_n}{\rho\pi\sigma_n^2} = 0, \tag{B.62}$$

where, as per [14, (4.9)]

$$\boxed{\lambda_{ni} = \frac{2\sigma_i^2}{\sigma_n^2+\sigma_i^2} \exp\left(-\frac{(y_n-y_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right) \exp\left(-\frac{(z_n-z_i)^2}{2(\sigma_n^2+\sigma_i^2)}\right)}. \tag{B.63}$$

Now we introduce the thrust coefficient,

$$\boxed{c_{t,n} = \frac{8T_n}{\pi\rho D^2 \langle U_{n-1}\rangle^2_{(n,x_n)}}}, \tag{B.64}$$

where D is the rotor diameter and $\langle\rangle_{(i,x_j)}$ denotes spatial averaging over the frontal projected area of WT_i at $x = x_j$, and substitute into Equation B.62

$$C_n^2 - 2C_n\left(U_0 - \sum_{i=1}^{n-1} \lambda_{ni} C_i\right) + \frac{c_{t,n}\pi\rho D^2 \langle U_{n-1}\rangle^2_{(n,x_n)}}{8\pi\rho\sigma_n^2} = 0, \tag{B.65}$$

which simplifies to

$$C_n^2 - 2C_n\left(U_0 - \sum_{i=1}^{n-1}\lambda_{ni}C_i\right) + \frac{c_{t,n}\langle U_{n-1}\rangle^2_{(n,x_n)}}{8\left(\sigma_n/D\right)^2} = 0. \tag{B.66}$$

We now solve for C_n. The roots of the polynomials are given by

$$C_n = \left(U_0 - \sum_{i=1}^{n-1}\lambda_{ni}C_i\right) \pm \frac{\sqrt{4\left(U_0 - \sum_{i=1}^{n-1}\lambda_{ni}C_i\right)^2 - \frac{c_{t,n}\langle U_{n-1}\rangle^2_{(n,x_n)}}{2(\sigma_n/D)^2}}}{2}, \tag{B.67}$$

which can be written

$$C_n = \left(U_0 - \sum_{i=1}^{n-1}\lambda_{ni}C_i\right) \pm \sqrt{\left(U_0 - \sum_{i=1}^{n-1}\lambda_{ni}C_i\right)^2 - \frac{c_{t,n}\langle U_{n-1}\rangle^2_{(n,x_n)}}{8\left(\sigma_n/D\right)^2}}. \tag{B.68}$$

Factorising this leads to

$$C_n = \left(U_0 - \sum_{i=1}^{n-1}\lambda_{ni}C_i\right)\left(1 \pm \sqrt{1 - \frac{c_{t,n}\langle U_{n-1}\rangle^2_{(n,x_n)}}{8\left(\sigma_n/D\right)^2\left(U_0 - \sum_{i=1}^{n-1}\lambda_{ni}C_i\right)^2}}\right) \tag{B.69}$$

Dividing both sides by the reference wind velocity, $U_h(\approx U_0)$, yields the final form

$$\frac{C_n}{U_h} = \left(1 - \sum_{i=1}^{n-1}\frac{C_i}{U_h}\lambda_{ni}\right)\left(1 \pm \sqrt{1 - \frac{c_{t,n}\left(\frac{\langle U_{n-1}\rangle_{(n,x_n)}}{U_h}\right)^2}{8\left(\sigma_n/D\right)^2\left(U_0 - \sum_{i=1}^{n-1}\lambda_{ni}\frac{C_i}{U_h}\right)^2}}\right). \tag{B.70}$$

Of the two solutions, the physically acceptable one - where C_n decreases with an increase in σ_n - is

$$\boxed{\frac{C_n}{U_h} = \left(1 - \sum_{i=1}^{n-1}\frac{C_i}{U_h}\lambda_{ni}\right)\left(1 - \sqrt{1 - \frac{c_{t,n}\left(\frac{\langle U_{n-1}\rangle_{(n,x_n)}}{U_h}\right)^2}{8\left(\sigma_n/D\right)^2\left(U_0 - \sum_{i=1}^{n-1}\lambda_{ni}\frac{C_i}{U_h}\right)^2}}\right)}, \tag{B.71}$$

which is identical to (4.10) in Bastankhah et al. [14].

APPENDIX C

Integral of lateral momentum equation over integration box within wind farm

From Equation 4.1, via the introduction of the incoming boundary layer profile, $V_0(z)$, one can arrive at the expression

$$\frac{\partial U(V-V_0)}{\partial x}+\frac{\partial V(V-V_0)}{\partial y}+\frac{\partial W(V-V_0)}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial y}-\frac{\partial \overline{uv}}{\partial x}-\frac{\partial \overline{v^2}}{\partial y}-\frac{\partial \overline{vw}}{\partial z}-W\frac{dV_0}{dz}+\sum_{i=1}^{n} f_{i,2}. \quad \text{(C.1)}$$

To progress to Equation 4.6 from this, one must then integrated over the integration box shown in Figure C.1. This involves first an integration from x_a to x_b with respect to x, where $x_a \ll x_n < x_b$, i.e.

$$\int_{x_a}^{x_b}\left(\frac{\partial U_n(V_n-V_0)}{\partial x}+\frac{\partial V_n(V_n-V_0)}{\partial y}+\frac{\partial W_n(V_n-V_0)}{\partial z}\right)\mathrm{d}x = \int_{x_a}^{x_b}\left(-\frac{1}{\rho}\frac{\partial P_n}{\partial y}-\frac{\partial \overline{uv}_n}{\partial x}-\frac{\partial \overline{v^2}_n}{\partial y}-\frac{\partial \overline{vw}_n}{\partial z}-W_n\frac{dV_0}{dz}+\sum_{i=1}^{n} f_{i,2}\right)\mathrm{d}x, \quad \text{(C.2)}$$

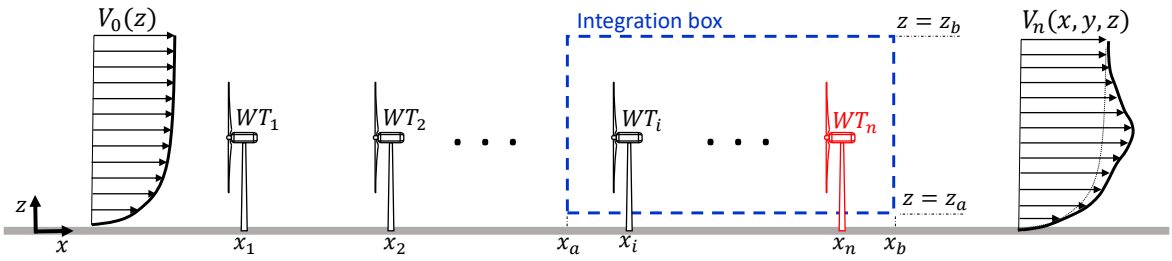


Figure C.1: Schematic of a wind farm with an arbitrary layout and arbitrary yaw angles consisting of n wind turbines (WT_1, WT_2, ..., WT_n) immersed in a turbulent boundary-layer flow. The momentum equation Equation 4.5 is integrated over the shown box. The integration is performed with and without WT_n, shown in the figure in red.

which evaluates to

$$U_n(V_n - V_0)\Big|_{x_a}^{x_b} + \int_{x_a}^{x_b} \frac{\partial V_n(V_n - V_0)}{\partial y}\,\mathrm{d}x + \int_{x_a}^{x_b} \frac{\partial W_n(V_n - V_0)}{\partial z}\,\mathrm{d}x$$
$$= -\frac{1}{\rho}\int_{x_a}^{x_b} \frac{\partial P_n}{\partial y}\,\mathrm{d}x - \overline{uv}_n\Big|_{x_a}^{x_b} - \int_{x_a}^{x_b} \frac{\partial \overline{v^2}_n}{\partial y}\,\mathrm{d}x - \int_{x_a}^{x_b} \frac{\partial \overline{vw}_n}{\partial z}\,\mathrm{d}x$$
$$- \int_{x_a}^{x_b} W_n \frac{dV_0}{dz}\,\mathrm{d}x + \sum_{i=1}^{n} \int_{x_a}^{x_b} f_{i,2}\,\mathrm{d}x. \quad \text{(C.3)}$$

Integrating then between y_a and y_b with respect to y, where $y_a \ll y_n \ll y_b$,

$$\int_{y_a}^{y_b} U_n(V_n - V_0)\Big|_{x_a}^{x_b}\,\mathrm{d}y + \int_{x_a}^{x_b} V_n(V_n - V_0)\Big|_{y_a}^{y_b}\,\mathrm{d}x + \int_{y_a}^{y_b}\int_{x_a}^{x_b} \frac{\partial W_n(V_n - V_0)}{\partial x}\,\mathrm{d}x\,\mathrm{d}y$$
$$= -\frac{1}{\rho}\int_{x_a}^{x_b} P_n\Big|_{y_a}^{y_b}\,\mathrm{d}x - \int_{y_a}^{y_b} \overline{uv}_n\Big|_{x_a}^{x_b}\,\mathrm{d}y - \int_{x_a}^{x_b} \overline{v^2}_n\Big|_{y_a}^{y_b}\,\mathrm{d}x - \int_{y_a}^{y_b}\int_{x_a}^{x_b} \frac{\partial \overline{vw}_n}{\partial z}\,\mathrm{d}x\,\mathrm{d}y$$
$$- \int_{y_a}^{y_b}\int_{x_a}^{x_b} W_n \frac{dV_0}{dz}\,\mathrm{d}x\,\mathrm{d}y + \sum_{i=1}^{n} \int_{y_a}^{y_b}\int_{x_a}^{x_b} f_{i,2}\,\mathrm{d}x\,\mathrm{d}y. \quad \text{(C.4)}$$

The second term on the left-hand side, $\int_{x_a}^{x_b} V_n(V_n - V_0)\Big|_{y_a}^{y_b}\,\mathrm{d}x$, is in fact equal to zero, i.e.

$$V_n(V_n - V_0)|_{y_b \gg y_n} - V_n(V_n - V_0)|_{y_a \ll y_n} = 0, \quad \text{(C.5)}$$

because the lateral velocity surplus, $(V_n - V_0)$, tends to zero as the influence of the presence of the turbine diminishes with increasing lateral distance from the turbine at y_n. Equation C.4 then becomes

$$\int_{y_a}^{y_b} U_n(V_n - V_0)\Big|_{x_a}^{x_b}\,\mathrm{d}y + \int_{y_a}^{y_b}\int_{x_a}^{x_b} \frac{\partial W_n(V_n - V_0)}{\partial x}\,\mathrm{d}x\,\mathrm{d}y$$
$$= -\frac{1}{\rho}\int_{x_a}^{x_b} P_n\Big|_{y_a}^{y_b}\,\mathrm{d}x - \int_{y_a}^{y_b} \overline{uv}_n\Big|_{x_a}^{x_b}\,\mathrm{d}y - \int_{x_a}^{x_b} \overline{v^2}_n\Big|_{y_a}^{y_b}\,\mathrm{d}x - \int_{y_a}^{y_b}\int_{x_a}^{x_b} \frac{\partial \overline{vw}_n}{\partial z}\,\mathrm{d}x\,\mathrm{d}y$$
$$- \int_{y_a}^{y_b}\int_{x_a}^{x_b} W_n \frac{dV_0}{dz}\,\mathrm{d}x\,\mathrm{d}y + \sum_{i=1}^{n} \int_{y_a}^{y_b}\int_{x_a}^{x_b} f_{i,2}\,\mathrm{d}x\,\mathrm{d}y. \quad \text{(C.6)}$$

Finally, integration with respect to z, where $0 \ll z_a < z_n \ll z_b$, yields

$$\int_{z_a}^{z_b}\int_{y_a}^{y_b} U_n(V_n - V_0)\Big|_{x_a}^{x_b} \,\mathrm{d}y\,\mathrm{d}z + \int_{y_a}^{y_b}\int_{x_a}^{x_b} W_n(V_n - V_0)\Big|_{z_a}^{z_b} \,\mathrm{d}x\,\mathrm{d}y$$
$$= -\frac{1}{\rho}\int_{z_a}^{z_b}\int_{x_a}^{x_b} P_n\Big|_{y_a}^{y_b} \,\mathrm{d}x\,\mathrm{d}z - \int_{z_a}^{z_b}\int_{y_a}^{y_b} \overline{uv}_n\Big|_{x_a}^{x_b} \,\mathrm{d}y\,\mathrm{d}z - \int_{z_a}^{z_b}\int_{x_a}^{x_b} \overline{v^2}_n\Big|_{y_a}^{y_b} \,\mathrm{d}x\,\mathrm{d}z - \int_{y_a}^{y_b}\int_{x_a}^{x_b} \overline{vw}_n\Big|_{z_a}^{z_b} \,\mathrm{d}x\,\mathrm{d}y$$
$$- \int_{z_a}^{z_b}\int_{y_a}^{y_b}\int_{x_a}^{x_b} W_n \frac{dV_0}{dz} \,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z + \sum_{i=1}^{n}\int_{z_a}^{z_b}\int_{y_a}^{y_b}\int_{x_a}^{x_b} f_{i,2} \,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z. \quad \text{(C.7)}$$

Again, the second term on the left-hand side,

$$W_n(V_n - V_0)|_{z_b \gg z_n} - W_n(V_n - V_0)|_{z_a \ll z_n} = 0, \quad \text{(C.8)}$$

because the lateral velocity surplus, $(V_n - V_0)$, also tends to zero as with increasing vertical distance from the turbine at y_n. Naturally the distance below the turbine, i.e. $(z_n - z_a)$, is limited by the distance to the ground, and the fact that too close to the ground and the assumption of negligible viscous forces would be invalid. However, this is deemed to be a safe assumption. Furthermore, the normal Reynolds stress term

$$-\int_{z_a}^{z_b}\int_{x_a}^{x_b} \overline{v^2}_n\Big|_{y_a}^{y_b} \,\mathrm{d}x\,\mathrm{d}z = 0. \quad \text{(C.9)}$$

This can be understood by relating the Reynolds stress to the mean velocity gradient via the Boussinesq hypothesis (Equation 2.5). This normal stress is then proportional to $\frac{\partial V}{\partial y}$, which, like the lateral velocity surplus, tends to zero with increasing lateral distance from the turbine such that

$$\left.\frac{\partial V}{\partial y}\right|_{y_b \gg y_n} - \left.\frac{\partial V}{\partial y}\right|_{y_a \ll y_n} = \overline{v^2}_n|_{y_b \gg y_n} - \overline{v^2}_n|_{y_a \ll y_n} = 0. \quad \text{(C.10)}$$

The same does not apply to the two other Reynolds stress terms, because $\frac{\partial V}{\partial x}$ varies according to the lateral thrust forcing within the farm, and $\frac{\partial V}{\partial z}$ is affected by the mean flow shear. For a inflow profile with no veer, i.e. $V_0 = 0$, this second, vertical Reynolds shear stress could be neglected. Lastly, the definition of $f_{i,2}$ as the lateral thrust force per unit volume means that the final term on the right-hand side can be evaluated as the lateral thrust force itself, meaning Equation C.7 becomes

$$\int U_n(V_n - V_0)\Big|_{x_a}^{x_b} \,\mathrm{d}A = -\frac{1}{\rho}\int_{z_a}^{z_b}\int_{x_a}^{x_b} P_n\Big|_{y_a}^{y_b} \,\mathrm{d}x\,\mathrm{d}z$$
$$- \int \overline{uv}_n\Big|_{x_a}^{x_b} \,\mathrm{d}A - \int_{z_a}^{z_b}\int_{x_a}^{x_b} \overline{v^2}_n\Big|_{y_a}^{y_b} \,\mathrm{d}x\,\mathrm{d}z - \int_{y_a}^{y_b}\int_{x_a}^{x_b} \overline{vw}_n\Big|_{z_a}^{z_b} \,\mathrm{d}x\,\mathrm{d}y$$
$$- \int W_n \frac{dV_0}{dz} \,\mathrm{d}V + \sum_{i=1}^{n} \frac{T_i \sin\gamma_i}{\rho}, \quad \text{(C.11)}$$

where $\mathrm{d}A$ is $\mathrm{d}y\,\mathrm{d}z$, $\mathrm{d}V$ is $\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$, T_i the thrust force of WT_i, γ_i the yaw angle of WT_i, and ρ the air density. Rearranging this for the lateral thrust yields

$$\sum_{i\in B}\frac{T_i\sin\gamma_i}{\rho} = \int U_n(V_n - V_0)\Big|_{x_a}^{x_b}\,\mathrm{d}A + \frac{1}{\rho}\int_{z_a}^{z_b}\int_{x_a}^{x_b} P_n\Big|_{y_a}^{y_b}\,\mathrm{d}x\,\mathrm{d}z + \int \overline{uv}_n\Big|_{x_a}^{x_b}\,\mathrm{d}A + \int_{y_a}^{y_b}\int_{x_a}^{x_b} \overline{vw}_n\Big|_{z_a}^{z_b}\,\mathrm{d}x\,\mathrm{d}y + \int W_n\frac{dV_0}{dz}\,\mathrm{d}V, \quad \text{(C.12)}$$

which is identical to Equation 4.6.

APPENDIX D

Decoupled solution to the 'conservation of lateral momentum surplus' equation

As is discussed in Section 4.2, the 'conservation of lateral momentum surplus', unlike its streamwise counterpart, is a coupled system. The proposed simplification is to assume that instead of being convected at $\boldsymbol{U}_n$, the wake is convected at $U_0(z)$, the inflow velocity. Whilst this is undoubtedly physically inaccurate, as it violates momentum theory, it decouples the system and so makes the solution of Equation 4.9 significantly more straightforward, providing a first-order approximation of the solution to the coupled equations.

Applying this assumption to Equation 4.9 then provides a simplified form of the 'conservation of lateral momentum surplus',

$$\rho \int U_0(V_0 - V_n)\ \mathrm{d}A - \rho \int U_0(V_0 - V_{n-1})\ \mathrm{d}A \approx T_n \sin\gamma. \tag{D.1}$$

Again using the definition of velocity deficit in Equation 4.11, and substituting Equation 4.17 and Equation 4.19 into Equation D.1 yields

$$\int U_0 \left(\sum_{i=1}^{n-1} C_{i,2} f_{i,2} + C_{n,2} f_{n,2} \right)\ \mathrm{d}A - \int U_0 \left(\sum_{i=1}^{n-1} C_{i,2} f_{i,2} \right)\ \mathrm{d}A = \frac{T_n \sin\gamma_n}{\rho}. \tag{D.2}$$

Expanding this to

$$\int U_0 \sum_{i=1}^{n-1} C_{i,2} f_{i,2} + U_0 C_{n,2} f_{n,2} - U_0 \sum_{i=1}^{n-1} C_{i,2} f_{i,2}\ \mathrm{d}A = \frac{T_n \sin\gamma_n}{\rho}, \tag{D.3}$$

and cancelling terms yields the decoupled general solution,

$$\boxed{\int U_0 C_{n,2} f_{n,2}\ \mathrm{d}A = \frac{T_n \sin\gamma_n}{\rho}}. \tag{D.4}$$

Realising that $C_n(x)$, and U_0 can be approximated as $U_0(z = z_h) = U_h$, Equation D.4 can be rewritten as

$$U_h C_{n,2} \int f_{n,2}\ \mathrm{d}A = \frac{T_n \sin\gamma_n}{\rho}. \tag{D.5}$$

Assuming then that $f_{n,2}$ is Gaussian, i.e.

$$f_{i,2} = \exp\left(-\frac{(y - y_i)^2}{2\sigma_{i,2}^2} \right) \exp\left(-\frac{(z - z_i)^2}{2\sigma_{i,2}^2} \right), \tag{D.6}$$

and as a result

$$\int f_n \, \mathrm{d}A = 2\pi\sigma_{n,2}^2. \tag{D.7}$$

Equation D.5 can now be written as

$$U_h C_{n,2}(2\pi\sigma_{n,2}^2) = \frac{T_n \sin\gamma_n}{\rho}, \tag{D.8}$$

which can be rearranged and normalised to give the decoupled specific Gaussian solution

$$\boxed{\frac{C_{n,2}}{U_h} = \frac{T_n \sin\gamma_n}{2\pi\rho\sigma_{n,2}^2 U_h^2}}. \tag{D.9}$$

The thrust force can also be given in terms of the the thrust coefficient as

$$T_n = \frac{\pi\rho D^2 \langle U_{n-1}\rangle_{(n,x_n)}^2 c_{t,n}}{8}. \tag{D.10}$$

Substituting this into Equation D.9 as such,

$$\frac{C_{n,2}}{U_h} = \frac{\pi\rho D \langle U_{n-1}\rangle_{(n,x_n)}^2 c_{t,n} \sin\gamma_n}{16\pi\rho\sigma_{n,2}^2 U_h^2}, \tag{D.11}$$

and cancelling terms yields the decoupled specific solution

$$\frac{C_{n,2}}{U_h} = \frac{\langle U_{n-1}\rangle_{(n,x_n)}^2 c_{t,n} \sin\gamma_n}{16\left(\frac{\sigma_{n,2}}{D}\right)^2 U_h^2}. \tag{D.12}$$

APPENDIX E
Wake width and turbulence intensity

As discussed in Section 5.2.1, the approach taken by Bastankhah et al. [14] to determine a linear relationship between the streamwise incoming turbulence intensity and the wake recovery of a given turbine is, for varying numbers of rows in an aligned wind farm such as in Figure 6.9(b), to first determine the wake half-width downstream of the middle turbine in the row by fitting a Gaussian and then to divide this by the incoming turbulence intensity at the turbine itself. Section 5.2.1 presents an identical analysis but for the RANS simulations performed in this thesis. This appendix then extends the same analysis to yawed turbines and the lateral wake. A number of criticisms of the approach are also set out at the end of Section 5.2.1 and are highlighted in this appendix.

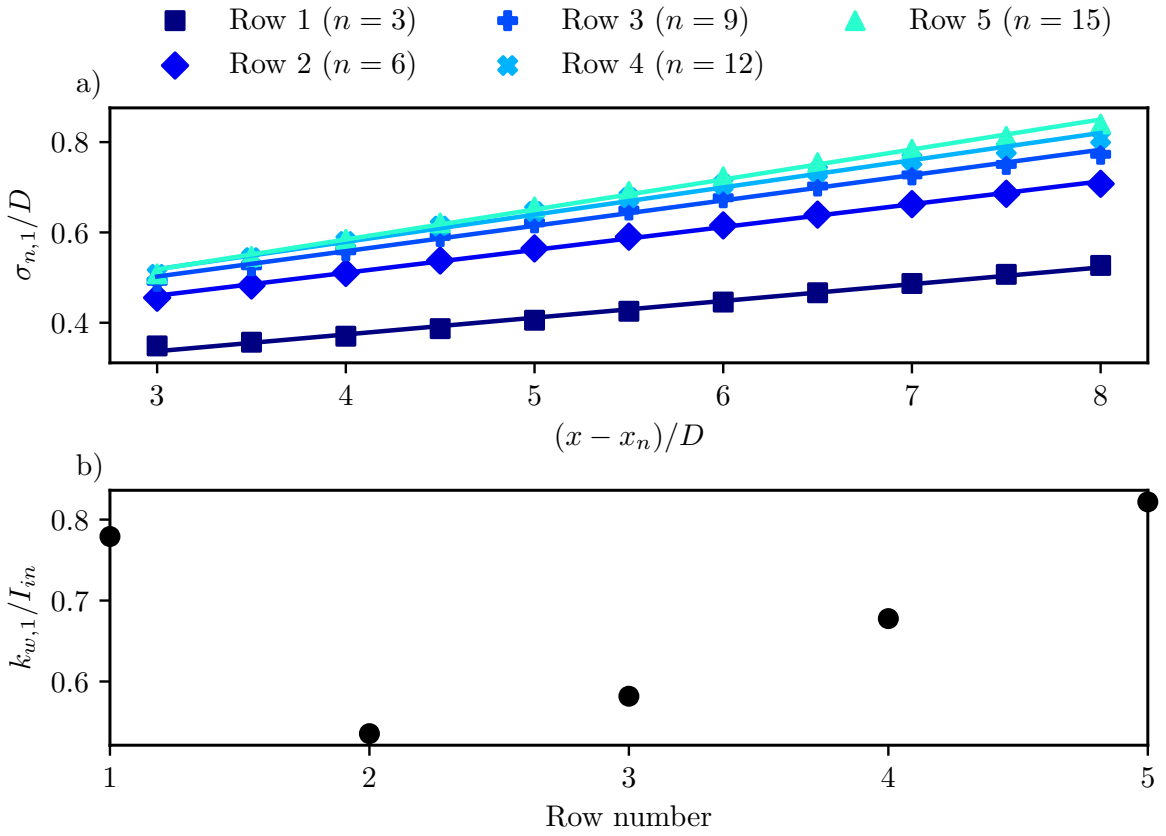


Figure E.1: a) Variation of the normalised streamwise wake half-width with streamwise distance downstream of the the last turbine in the middle column of aligned wind farms with differing numbers of rows and all turbines yawed at $\gamma = 25°$. b) The ratio of streamwise wake recovery rate $k_{w,1}$, to the incoming turbulence intensity I_{in} for turbines in different rows based on data from RANS simulations.

The first item to address is how the relationship between streamwise wake recovery and incoming turbulence intensity is affected by yaw. Figure E.1 presents the same analysis as in Figure 5.2, but now for an aligned wind farm where all turbines are yawed at an angle of 25°. As is shown in Figure 3.17, for the range $3 < x/D < 8$, the streamwise wake half-width is smaller for a yawed turbine than for an unyawed one. However, Figure E.1(a) shows that a linear fit of the wake half-width with downstream distance is still suitable. There is a marked difference in the results for $k_{w,1}/I_{in}$, however, as shown in Figure E.1(b). Unlike for the unyawed turbines, there does not appear to be a discernable linear relationship between the streamwise wake recovery and incoming turbulence intensity. Herein lies one of the flaws of applying this approach to yawed turbines. Yawed turbines introduce wake steering, which will likely cause partial wake situations at downstream turbines. This approach does not allow one to divorce the effects of differing incoming turbulence intensities from for example partial wake scenarios. A more rigorous approach would examine an individual turbine and vary the inflow conditions to avoid this situation.

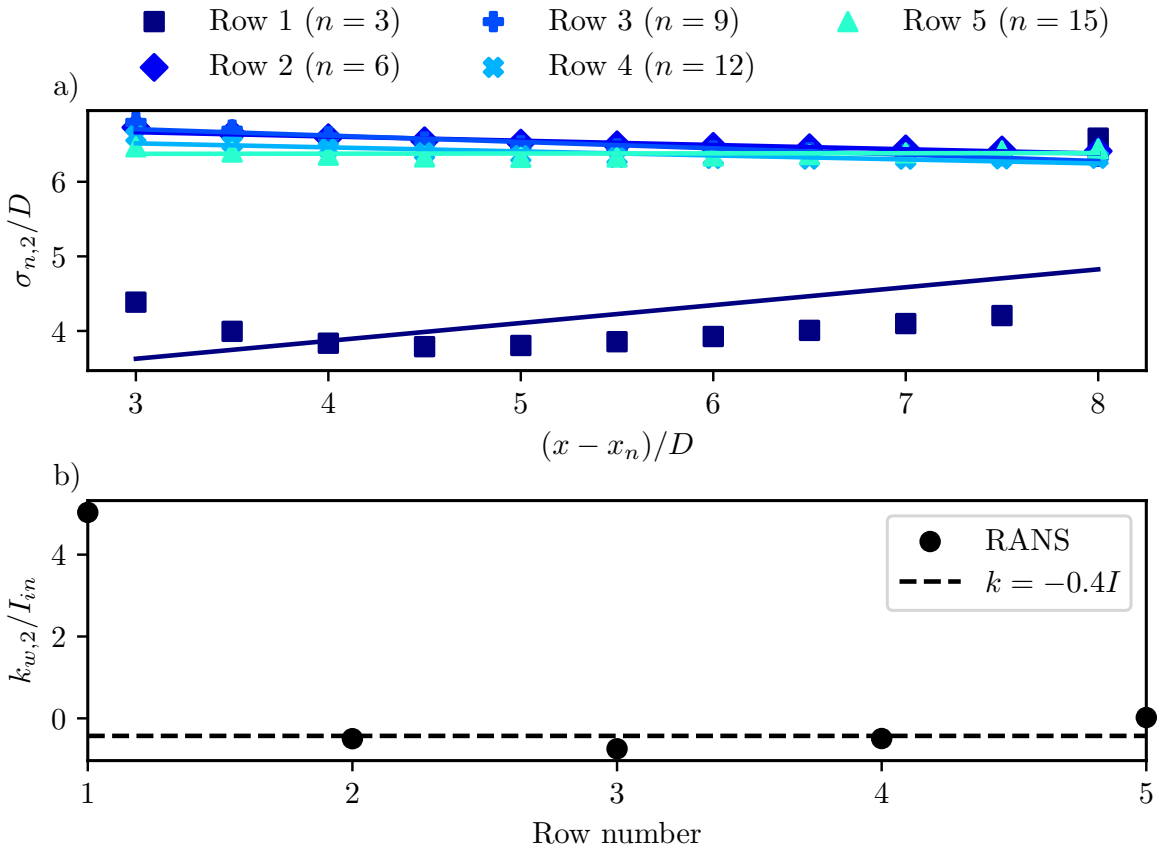


Figure E.2: a) Variation of the normalised lateral wake half-width with streamwise distance downstream of the the last turbine in the middle column of aligned wind farms with differing numbers of rows and all turbines unyawed ($\gamma = 0°$). b) The ratio of lateral wake recovery rate $k_{w,2}$, to the incoming turbulence intensity I_{in} for turbines in different rows based on data from RANS simulations. c) Incoming added turbulence intensity from RANS data for turbines in different rows.

Directly transposing the approach to the lateral wake also provides some useful insight. Firstly for an unyawed turbine, the linear fit of wake half-width appears to be appropriate for rows 2 to 5, but struggles with row 1. Perhaps due to some cumulative lateral wake effects. Moreover, the lateral wake recovery rate is every so slightly negative, with a reasonable proportionality to the incoming turbulence intensity. This was predicted for a yawed turbine in Section 3.4, but is evident here even for an unyawed turbine. A generalised approach would allow for easier conclusions to be drawn, but there is at least no effect of wake steering here.

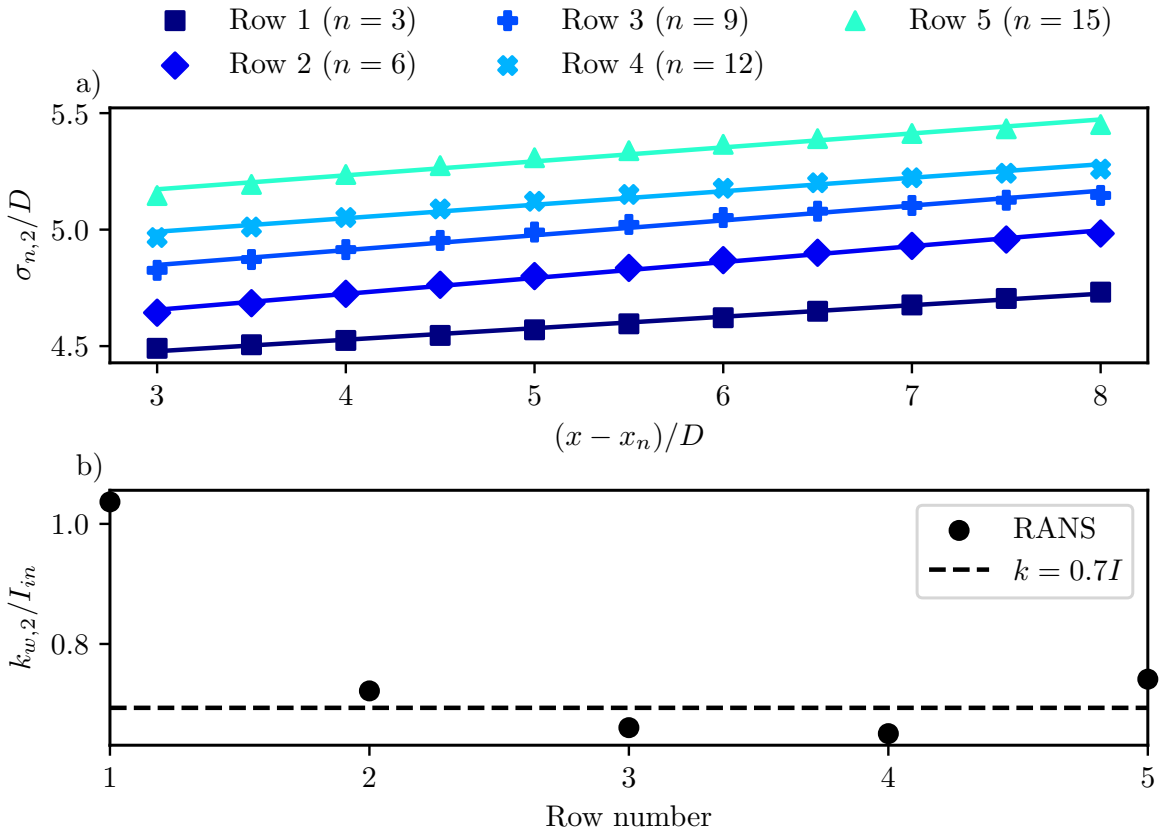


Figure E.3: a) Variation of the normalised streamwise wake half-width with streamwise distance downstream of the the last turbine in the middle column of aligned wind farms with differing numbers of rows and all turbines yawed at $\gamma = 25°$. b) The ratio of lateral wake recovery rate $k_{w,2}$, to the incoming turbulence intensity I_{in} for turbines in different rows based on data from RANS simulations. c) Comparison of incoming added turbulence intensity predicted by the Crespo model [31] and the suggested Crespo model with RANS data for turbines in different rows.

Whilst Section 3.4 showed that the width of the lateral wake of a yawed turbine is non-linear with downstream distance over $5 < x/D < 25$, Figure E.2(a) demonstrates that, at least for rows 2 to 5, it can be approximated as linear over the range $3 < x/D < 8$ like the streamwise wake. Section 3.4 also demonstrated that the lateral wake width of a yawed turbine is primarily dependent on the incoming turbulence intensity. This is somewhat evident here in Figure E.2(b), but again the first turbine is a significant outlier, and the proportionality varies.

APPENDIX F

Python implementation of analytical solution

Below are the Python scripts used to implement the analytical solution described in this thesis as they are at the time of publication. For the most recent version, please see `https://github.com/NilsGaukroger/Analytical-solution-for-the-cumulative-wake-of-yawed-wind-turbines/blob/main/lateralSolution.py`. The first script contains the main function, and the second some useful sub-functions.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun 14 11:10:39 2022

@author: nilsg
"""

def lateralSolution(method, n_t, x_t, y_t, z_t, yaws, x, y, z, U0, U_h, V0, I0,
     near_wake_correction, rho=1.225, zh=90.0, D=126.0):
    '''
    Analytical solution for the cumulative wake of yawed wind turbines. Yields
        streamwise and lateral velocity fields for a given turbine type, wind
        farm layout and set of yaw angles.

    Parameters
    ----------
    method : string
        Version of the conservation of momentum deficit to be used, either "
            original" or "modified". See Bastankhah et al., 2021, for further
            details.
    n_t : int
        Number of turbines, should match len(x_t) = len(y_t) = len(z_t).
    x_t : Array of float64
        x-coordinates of turbines.
    y_t : Array of float64
        y-coordinates of turbines.
    z_t : Array of float64
        z-coordinates of turbines.
    yaws : Array of int32
        Yaw angles of turbines in radians, positive is clockwise rotation when
            viewed from above.
    x : Array of float64
        x-coordinates of cells within flow domain.
    y : Array of float64
        y-coordinates of cells within flow domain.
    z : Array of float64
        z-coordinates of cells within flow domain.
    U0 : Array of float64 (nx,ny,nz)
        Initial streamwise velocity field.
```

```
    V0 : Array of float64 (nx,ny,nz)
        Initial lateral velocity field.
    rho : float, optional
        Air density [kg/m^3]. The default is 1.225.
    zh : float, optional
        Hub height [m]. The default is 90.0.
    D : float, optional
        Rotor diameter [m]. The default is 126.0.

    Returns
    -------
    U : Array of float64 (nx,ny,nz)
        Streamwise velocity field.
    V : Array of float64 (nx,ny,nz)
        Lateral velocity field.

    '''

    #%% Module imports
    import numpy as np
    from analytical_functions import vel_disc, NREL5MW, turb_add, epsilon
    from numpy.lib.scimath import sqrt as csqrt
    from copy import copy
    import xarray
    from scipy.interpolate import interp1d

    #%% Setup checks
    # Sort turbine x, y, z positions by increasing x
    idx = np.argsort(x_t)
    x_t = x_t[idx]
    y_t = y_t[idx]
    z_t = z_t[idx]

    # Initialise velocity fields
    U = copy(U0)
    V = copy(V0)

    # Make velocities into xarray DataArray
    flowdata = xarray.Dataset(
        data_vars = dict(
            U  = (["x","y","z"], U),
            V  = (["x","y","z"], V),
            U0 = (["x","y","z"], U0),
            V0 = (["x","y","z"], V0)
        ),
        coords = dict(
            x = ("x", x),
            y = ("y", y),
            z = ("z", z)),
        attrs = dict(description = "Analytical solution"),
    )

    #%% Flow domain
    nx = len(x)
    Y, Z = np.meshgrid(y, z)
    Y = Y.T; Z = Z.T

```

```
    #%% Analytical solution for streamwise velocity
    # Preallocation
    U_d  = np.zeros((n_t))
    CT   = np.zeros((n_t))
    CP   = np.zeros((n_t))
    T    = np.zeros((n_t))
    P    = np.zeros((n_t))
    I    = np.zeros((n_t))
    H    = np.zeros((n_t))
    k_w  = np.zeros((n_t)) # Streamwise wake recovery
    k_w2 = np.zeros((n_t)) # Lateral wake recovery
    s    = np.zeros((n_t)) # Streamwise wake width
    s2   = np.zeros((n_t)) # Lateral wake width

    C    = np.zeros((n_t, nx), dtype=np.complex_) # Streamwise
    C2   = np.zeros((n_t, nx), dtype=np.complex_) # Lateral

    # Initialise deflection
    delta_D = np.zeros((n_t, nx))

    print('Computing velocity field...')
    for n in range(n_t):
        #%% Thrust force
        # Calculate rotor-averaged streamwise velocity
        U_d[n] = vel_disc(flowdata, x_t[n]*D, y_t[n]*D, z_t[n], yaws[n], D)
        # Look up corresponding thrust and power coefficients for NREL 5MW
        CT[n], CP[n] = NREL5MW(U_d[n])
        # Calculate thrust force from CT
        T[n] = (np.pi * rho * CT[n] * U_d[n]**2 * D**2) / 8
        # Calculate power from CP
        P[n] = (np.pi * rho * CP[n] * U_d[n]**3 * D**2) / 8

        #%% Turbulence intensity
        if n == 0:
            # For first turbine, use inflow turbulence intensity
            I[n] = I0
        else:
            # For all other turbines, include added turbulence from other
                turbines
            for i in range(n):
                # Added T.I. of all other turbines on turbine n
                H[i] = turb_add(y,
                                z,
                                (y_t[i] + np.interp(x_t[n]*D,x,delta_D[i,:]))*D
                                    ,
                                z_t[i],
                                yaws[i],
                                k_w[i] * (x_t[n] - x_t[i])*D + epsilon(CT[i]) *
                                     D,
                                CT[i],
                                y_t[n]*D,
                                z_t[n],
                                yaws[n],
                                D,
                                (x_t[n] - x_t[i])*D,
                                I0)
            # Total T.I. at turbine n
```

```
145             I[n] = np.sqrt(I0**2 + max(H[:n], default=0)**2)
146
147         #%% Wake recovery rate
148         # k_w[n] = 0.38 * I[n] # Niayifar & Porté-Agel, 2016
149         # k_w[n] = 0.35 * I[n] # Carbajo Fuertes et al., 2018
150         # k_w[n] = 0.34 * I[n] # Zhan et al., 2020
151         # k_w[n] = 0.31 * I[n] # Bastankhah et al., 2021
152         k_w[n] = 0.4 * I[n] # Gaukroger, 2022
153         k_w2[n] = 0.6 * I[n] # Gaukroger, 2022
154
155         #%% Velocity
156         # For all streamwise points in CFD domain
157         downstream = x > x_t[n]*D
158         first      = np.argmax(downstream)
159         x_n        = x[downstream]
160
161         # Interpolate for centreline velocities
162         U0_c = flowdata.U0.interp(y=y_t[n]*D, z=z_t[n])
163         V0_c = flowdata.V0.interp(y=y_t[n]*D, z=z_t[n])
164         Uc_int = interp1d(x, U0_c)
165         Vc_int = interp1d(x, V0_c)
166
167         for i in range(len(x_n)):
168             # Add offset to x-coordinate
169             i = i + first
170             #%% Sum of lambda * C_i/U_h
171             # Compute wake width of turbine n at position x[i]
172             s[n] = k_w[n] * (x[i] - x_t[n]*D) + epsilon(CT[n]) * D # streamwise
173             s2[n] = k_w2[n] * (x[i] - x_t[n]*D) + epsilon(CT[n]) * D # lateral
174             # Set total for sum of lambda * C_i/U_h from (4.10) = 0
175             L = 0; L1 = 0; L2 = 0
176             if n > 0: # because lambda = 0 for the first turbine?
177                 # Compute sum of lambda * C_i/U_h term from (4.10)
178                 for ii in range(n-1, -1, -1):
179                     s[ii] = k_w[ii] * (x[i] - x_t[ii]*D) + epsilon(CT[ii]) * D
180                     l_dict = {
181                         "original" : ((2*s[ii]**2)/(s[n]**2 + s[ii]**2)) * np.
                                exp(-((((y_t[n] - (y_t[ii] + delta_D[ii,i]))*D)**2)
                                /(2*(s[n]**2 + s[ii]**2)))) * np.exp(-(((z_t[n] -
                                z_t[ii])**2)/(2*(s[n]**2 + s[ii]**2)))),
182                         "modified" : ((s[ii]**2)/(s[n]**2 + s[ii]**2)) * np.exp
                                (-((((y_t[n] - (y_t[ii] + delta_D[ii,i]))*D)**2)
                                /(2*(s[n]**2 + s[ii]**2)))) * np.exp(-(((z_t[n] -
                                z_t[ii])**2)/(2*(s[n]**2 + s[ii]**2))))
183                         }
184                     l = l_dict.get(method)
185                     # Add contribution to sum
186                     L += l * (np.real(C[ii,i]))
187                     # Lateral
188                     s2[ii] = k_w2[ii] * (x[i] - x_t[ii]*D)
189                     l1 = ((2*s2[ii]**2)/(s[n]**2 + s2[ii]**2)) * np.exp(-((((
                            y_t[n] - y_t[ii])*D)**2)/(2*(s[n]**2 + s2[ii]**2)))) *
                            np.exp(-(((z_t[n] - z_t[ii])**2)/(2*(s[n]**2 + s2[ii
                            ]**2))))
190                     # Add contribution to sum
191                     L1 += l1 * np.real(C2[ii,i])
192                     l2 = ((2*s[ii]**2)/(s2[n]**2 + s[ii]**2)) * np.exp(-((((y_t
```

```
                    [n] - (y_t[ii] + delta_D[ii,i]))*D)**2)/(2*(s2[n]**2 +
                    s[ii]**2)))) * np.exp(-(((z_t[n] - z_t[ii])**2)/(2*(s2[
                    n]**2 + s[ii]**2))))
193                 # Add contribution to sum
194                 L2 += l2 * np.real(C[ii,i])
195
196             #%% Solve for Cn
197             # New proposed model
198             C[n,i] = (U_h-L) * (1 - csqrt(1 - ((T[n] * np.cos(yaws[n])) / (rho
                    * np.pi * s[n]**2 * (U_h-L)**2) )))
199             # Linear superposition with local velocity as reference velocity
200             # C[n,i] = U_d[n] * (1 - np.sqrt(1 - ( CT[n] / (8 * s[n]**2/D**2) )
                    ))
201             # Near wake region
202             if ((np.imag(C[n,i] != 0)) or ((C[n,i]/U_d[n]) > (1 - np.sqrt(1-CT[
                    n])))):
203                 C[n,i] = (1 - np.sqrt(1-CT[n])) * U_d[n]
204
205             #%% Solve for C2n
206             # Coupled solution
207             C2[n,i] = (((T[n] * np.sin(yaws[n])) / (np.pi * rho)) + C[n,i] * s[
                    n]**2 * L1) / (s2[n]**2 * (2*U_h - L2 - C[n,i] * (2*s[n]**2/(s2
                    [n]**2 + s[n]**2))))
208
209             # Near wake correction
210             # if C2[n,i]/U_h > 0.08:
211             #     C2[n,i] = 0.08 * U_h
212
213             # Decoupled solution
214             # C2[n,i] = (T[n] * np.sin(yaws[n])) / (2 * np.pi * rho * s2[n]**2
                    * U_h)
215
216             #%% Solve for Un through domain using (4.5)
217             # Streamwise
218             f = np.exp(-(((Y - (y_t[n] + delta_D[n,i])*D)**2)/(2*s[n]**2))) *
                    np.exp(-(((Z - z_t[n])**2)/(2*s[n]**2)))
219             flowdata.U[i,:,:] += -np.real(C[n,i]) * f
220
221             # Lateral
222             f2 = np.exp(-(((Y - y_t[n]*D)**2)/(2*s2[n]**2))) * np.exp(-(((Z -
                    z_t[n])**2)/(2*s2[n]**2)))
223             flowdata.V[i,:,:] += np.real(C2[n,i]) * f2
224
225             #%% Compute wake centre deflection
226             # Wake centre velocities (option 1)
227             Uc = Uc_int(x[i]) - np.real(C[n,i])
228             Vc = Vc_int(x[i]) + np.real(C2[n,i])
229
230             # Wake centre velocities (option 2)
231             # Uc = interpolate.interpn((x,y,z), U, (x[i], y_t[n]*D, z_t[n]))
232             # Vc = interpolate.interpn((x,y,z), V, (x[i], y_t[n]*D, z_t[n]))
233
234             # Deflection
235             if i < len(x_n)+first-1:
236                 # delta_D[n,i+1] = delta_D[n,i] + ((Vc/U_h)*(x[i+1]-x[i]))/D
237                 if near_wake_correction:
238                     if (x[i] - x_t[n]*D) > 5*D:
```

```
                        delta_D[n,i+1] = delta_D[n,i] + ((Vc/Uc)*(x[i+1]-x[i]))
                            /D
                else:
                    delta_D[n,i+1] = delta_D[n,i] + ((Vc/Uc)*(x[i+1]-x[i]))/D

        #%% Status check
        print('turbine {:d} of {:d} ({:.0f}% complete)'
              .format(n+1, n_t, ((n+1)/n_t)*100))

    return flowdata, U_d

#%% if __name__ = '__main__':
if __name__ == '__main__':
    #%% Module imports
    import numpy as np
    import matplotlib.pyplot as plt
    import time
    from scipy.interpolate import interp1d

    def vel_zh(z, zh, U, V):
        U_int = interp1d(z, U, axis=2)
        V_int = interp1d(z, V, axis=2)
        U_zh  = U_int(zh)
        V_zh  = V_int(zh)
        return U_zh, V_zh

    #%% Setup
    # Layout alignment
    alignment = 'aligned' # 'aligned' or 'slanted'

    # Choice of method
    method = "original" # "original" or "modified" (see Bastankhah et al.,
        2020)

    # Define constants
    rho = 1.225 # air density [kg/m^3]
    zh  = 90.0  # turbine hub height [m]
    D   = 126.0 # turbine rotor diameter [m]

    # Define number of turbines
    n_t = 15

    # Define layout parameters
    s_x     = 7    # streamwise inter-turbine spacing [D]
    s_y     = 4    # lateral inter-turbine spacing [D]
    stagger = 0.75 # stagger [D]

    # Define turbine x and y positions
    x_t = np.empty((n_t))
    y_t = np.empty((n_t))
    for i_t in range(n_t):
        x_t[i_t] = np.floor(i_t/3) * s_x
        if alignment == 'aligned':
            y_t[i_t] = ((i_t-1)%3) * s_y - s_y
        elif alignment == 'slanted':
            y_t[i_t] = ((i_t-1)%3) * s_y + np.floor(i_t/3) * stagger - s_y
        else:
```

```
            print('Incorrect alignment specification.')
            break

    # Plot layout
    fig, ax = plt.subplots(1,1)
    ax.scatter(x_t,y_t)
    ax.set_xlabel('x/D')
    ax.set_ylabel('y/D')

    # Define turbine z positions
    z_t = zh * np.ones((n_t))

    # Define yaw angles
    yaws = np.deg2rad(np.asarray([25]*n_t))

    # Sort turbine x, y, z positions by increasing x
    idx = np.argsort(x_t)
    x_t = x_t[idx]
    y_t = y_t[idx]
    z_t = z_t[idx]
    yaws = yaws[idx]

    print('Setup complete.')

    #%% Flow domain
    x = np.linspace(min(x_t)-2, max(x_t)+10,  100)*D
    y = np.linspace(min(y_t)-1,  max(y_t)+1,  100)*D
    z = np.linspace(         0,           2,  100)*zh

    nx, ny, nz = len(x), len(y), len(z)

    #%% Inflow
    # Define inflow parameters
    U_h = 8    # streamwise hub height inflow velocity [m/s]
    V_h = 0    # lateral hub height inflow velocity [m/s]
    I0  = 0.10 # hub height total inflow turbulence intensity [-]

    # Preallocate analytical velocity field
    U0 = np.zeros((nx, ny, nz)) # streamwise velocity
    V0 = np.zeros((nx, ny, nz)) # lateral velocity

    # Preallocate streamwise velocity with adiabatic log law inflow
    u_s   = 0.5  # friction velocity [m/s]
    kappa = 0.4  # von Kármán constant [-]
    z0    = 0.15 # roughness length [m]

    U_in = np.zeros((nz))
    U_in[1:] = (u_s/kappa) * np.log(z[1:]/z0)

    U0[:,:,:] = U_in

    print('Inflow preallocated.')

    #%% Run solution
    start_time = time.time()
    flowdata, P = lateralSolution("original", n_t, x_t, y_t, z_t, yaws, x, y, z
        , U0, U_h, V0, I0, rho=1.225, zh=90.0, D=126.0)
```

```
    end_time = time.time()

    #%% Timing
    execution_time = end_time - start_time
    mins = execution_time // 60
    secs = execution_time % 60
    print("Solution execution time: {:.0f}m {:.1f}s".format(mins, secs))

    #%% Hub height velocities
    U_zh = flowdata.U.interp(z=zh)
    V_zh = flowdata.V.interp(z=zh)

    #%% Plotting
    fig, ax = plt.subplots(1, 1)
    levels = np.linspace(0,0.6,101)
    p = ax.contourf(x/D,y/D,1-(U_zh.T/U_h),cmap='jet',levels=levels)
    ax.set_xlabel('$x/D$')
    ax.set_ylabel('$y/D$')
    cbar = fig.colorbar(p, ticks=[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6])
    cbar.set_label('$1-(U/U_h)$')

    fig, ax = plt.subplots(1, 1)
    # levels = np.linspace(0,0.32,101)
    p = ax.contourf(x/D,y/D,V_zh.T/U_h,cmap='jet')
    ax.set_xlabel('$x/D$')
    ax.set_ylabel('$y/D$')
    cbar = fig.colorbar(p)
    cbar.set_label('$V_h/U_h$')

    # fig, ax = plt.subplots(1,1)
    # for i, i_t in enumerate(list(range(0,n_t,3))):
    #     ax.plot(delta[i_t,:]/D, label=str(i+1 ))
    # ax.set_xlabel('$x/D$')
    # ax.set_ylabel('$\delta_1/D$')
    # ax.legend(title='Row', bbox_to_anchor=(1.05,0.5), loc='center left')
```

Listing F.1: Main run script.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun 14 11:32:06 2022

@author: nilsg
"""

import numpy as np
import xarray as xr
import warnings

# Suppress FutureWarnings (for xarray)
warnings.simplefilter(action='ignore', category=FutureWarning)

def vel_disc(flowdata, x_t, y_t, z_t, yaw, D, n=128):
    '''
    Function to calculate disc velocity of a turbine at (x_t, y_t, z_t) with
        yaw angle 'yaw' and diameter 'D' from velocity fields in 'flowdata'.

    Parameters
```

```
    ----------
    flowdata : xarray.DataSet
        xarray dataset containing velocities and coordinates.
    x_t : float
        x-coordinate of turbine rotor centre.
    y_t : float
        y-coordinate of turbine rotor centre.
    z_t : float
        z-coordinate of turbine rotor centre.
    yaw : float
        Yaw angle of rotor in radians.
    D : float
        Rotor diameter [m].
    n : int, optional
        Spatial discretisation. The default is 128.

    Returns
    -------
    vel_AD : float
        Disc velocity [m/s].

    '''
    # Find edges of rotor in x, y, z
    lower = (x_t - (D/2)*np.sin(yaw),
             y_t - (D/2)*np.cos(yaw),
             z_t - (D/2)
             )
    upper = (x_t + (D/2)*np.sin(yaw),
             y_t + (D/2)*np.cos(yaw),
             z_t + (D/2)
             )

    # Rotated rotor coordinate axis
    r_rotor = np.linspace(y_t - D/2, y_t + D/2, n)

    # Discretise rotor surface as rectangle
    x_rotor = xr.DataArray(np.linspace(lower[0], upper[0], n),
                           dims="r", coords={"r": r_rotor})
    y_rotor = xr.DataArray(np.linspace(lower[1], upper[1], n),
                           dims="r", coords={"r": r_rotor})
    z_rotor = xr.DataArray(np.linspace(lower[2], upper[2], n),
                           dims="z")

    # 3D interpolation of rectangle around rotor
    flowdata_rec = flowdata.interp(x=x_rotor, y=y_rotor, z=z_rotor)

    # Filter rotor disc from rectangle
    flowdata_rotor = flowdata_rec.where(np.sqrt((flowdata_rec["r"] - y_t)**2 +
        (flowdata_rec.z - z_t)**2) <= (D/2),
                        other=np.nan)

    # Create disc averaged velocity vector (assuming W=0)
    vel_d = np.asarray([flowdata_rotor.U.mean(), flowdata_rotor.V.mean(), 0])

    # Create disc normal vector (pointing downstream)
    n = np.asarray([np.cos(yaw), -np.sin(yaw), 0])

```

```
76      # Dot product disc velocity vector with rotor normal vector
77      vel_AD = np.dot(vel_d, n)
78
79      return vel_AD
80
81  def turb_add(y, z, y1, z1, yaw1, s, CT, y2, z2, yaw2, D, sep_x, I0):
82      '''
83
84
85      Parameters
86      ----------
87      y : Array of float64
88          y-coordinates of flow domain.
89      z : Array of float64
90          z-coordinates of flow domain.
91      y1 : float
92          y-coordinate of upstream turbine's wake centre.
93      z1 : float
94          z-coordinate of upstream turbine's wake centre.
95      yaw1 : float
96          Yaw angle of upstream turbine [rad].
97      s : float
98          Width of wake of upstream turbine at x-position of downstream position.
99      CT : float
100         Thrust coefficient of upstream turbine.
101     y2 : float
102         y-coordinate of downstream turbine.
103     z2 : float
104         z-coordinate of downstream turbine.
105     yaw1 : float
106         Yaw angle of downstream turbine [rad].
107     D : float
108         Rotor diameter of both turbines.
109     sep_x : float
110         x-separation of two turbines.
111     I0 : float
112         Freestream turbulence intensity.
113
114     Returns
115     -------
116     H : float
117         Contribution to added turbulence of turbine 1 on turbine 2.
118
119     '''
120     if sep_x < 0.1: # because I_p tends to infinity as separation tends to zero
121         H = 0
122     else:
123         # axial induction factor of turbine 1
124         a = 0.5 * (1 - np.sqrt(1 - CT*np.cos(yaw1)))
125         # Added turbulence intensity due to turbine 1 according to modified
                Crespo model (see Bastankhah et al., 2021)
126         I_p = 0.66 * a**0.83 * I0**0.03 * (sep_x/D)**(-0.32)
127
128         ## Radii within yz-plane at turbine 2 x-position
129         # turbine 1 (upstream) wake diameter
130         rw = 2 * s # why 2*sigma? see Porté-Agel et al., 2013, "A numerical..."
131         # turbine 2 (downstream) rotor diameter
```

```
132         r2 = D/2
133 
134         ## Overlap area
135         if (yaw1 == 0) & (yaw2 == 0):
136             # Rotor centre separation in yz-plane at turbine 2 x-position
137             sep_yz = np.sqrt((y1-y2)**2 + (z1-z2)**2)
138 
139             # If turbine 2 rotor not within turbine 1 wake
140             if sep_yz >= (rw + r2):
141                 A = 0 # overlap area = zero
142             # If turbine 2 fully enveloped by turbine 1 wake
143             elif sep_yz <= (rw - r2):
144                 A = np.pi * r2**2 # overlap area = rotor area
145             # If there is partial overlap
146             else:
147                 # Use formula for partial overlap of two circles with different
                        radii
148                 d1 = (rw**2 - r2**2 + sep_yz**2) / (2*sep_yz)
149                 d2 = sep_yz - d1
150                 A  = rw**2*np.arccos(d1/rw) - d1*np.sqrt(rw**2-d1**2) + r2**2*
                        np.arccos(d2/r2) - d2*np.sqrt(r2**2-d2**2)
151 
152         else:
153             # Meshgrid y and z coordinates of plane
154             Y, Z = np.meshgrid(y, z)
155 
156             # Create mask for turbine 2 rotor
157             in_rotor = (((Y - y2)/(r2*np.cos(yaw2)))**2 + ((Z - z2)/r2)**2) <=
                    1
158             # Create mask for turbine 1 wake
159             in_wake  = (((Y - y1)/(rw*np.cos(yaw1)))**2 + ((Z - z1)/rw)**2) <=
                    1
160             # Create mask for overlap
161             in_both  = in_rotor & in_wake
162 
163             # Count number of points within masks
164             n_rotor = np.count_nonzero(in_rotor)
165             n_both  = np.count_nonzero(in_both)
166 
167             # Overlap area
168             A = (n_both/n_rotor) * np.pi * r2**2
169 
170         # Contribution of turbulence intensity (see Niayifar & Porté-Agel,
                2016, Eq. 18)
171         H = ((A * 4) / (np.pi * D**2)) * I_p
172 
173     return H
174 
175 def epsilon(CT):
176     beta = 0.5 * (1 + np.sqrt(1-CT)) / (np.sqrt(1-CT))
177     e    = 0.2 * np.sqrt(beta)
178     return e
179 
180 def NREL5MW(U_d):
181     # PyWakeEllipSys
182     CT_curve = np.asarray([[3.0,   0.0  ],
183                            [4.0,   0.913],
```

```
                        [5.0,   0.869],
                        [6.0,   0.780],
                        [7.0,   0.773],
                        [8.0,   0.770],
                        [9.0,   0.768],
                        [10.0,  0.765],
                        [11.0,  0.746],
                        [12.0,  0.533],
                        [13.0,  0.392],
                        [14.0,  0.305],
                        [15.0,  0.244],
                        [16.0,  0.200],
                        [17.0,  0.167],
                        [18.0,  0.142],
                        [19.0,  0.122],
                        [20.0,  0.106],
                        [21.0,  0.092],
                        [22.0,  0.082],
                        [23.0,  0.073],
                        [24.0,  0.065],
                        [25.0,  0.059]])

    P_curve = np.asarray([[3.0,   0.0  ],
                        [4.0,   209.2],
                        [5.0,   444.8],
                        [6.0,   765.2],
                        [7.0,   1212.0],
                        [8.0,   1807.7],
                        [9.0,   2571.6],
                        [10.0,  3523.8],
                        [11.0,  4649.7],
                        [12.0,  5000.0],
                        [13.0,  5000.0],
                        [14.0,  5000.0],
                        [15.0,  5000.0],
                        [16.0,  5000.0],
                        [17.0,  5000.0],
                        [18.0,  5000.0],
                        [19.0,  5000.0],
                        [20.0,  5000.0],
                        [21.0,  5000.0],
                        [22.0,  5000.0],
                        [23.0,  5000.0],
                        [24.0,  5000.0],
                        [25.0,  5000.0]])

    CP_curve = ((P_curve[:,1]*1e3/(1-0.059)) / (0.125 * 1.225 * 126.0**2 * np.
        pi * P_curve[:,0]**3))

    CP_curve = np.vstack((P_curve[:,0], CP_curve)).T

    # SOWFA
    # CT_curve = np.asarray([[2.0,   1.4612728464576872],
    #                     [2.5,   1.3891500248600195],
    #                     [3.0,   1.268082754962957 ],
    #                     [3.5,   1.1646999475504172],
    #                     [4.0,   1.0793803926905128],
```

```
240     #                       [4.5,  1.0098020917279509],
241     #                       [5.0,  0.9523253671258429],
242     #                       [5.5,  0.9048200632193146],
243     #                       [6.0,  0.8652746358037285],
244     #                       [6.5,  0.8317749797630494],
245     #                       [7.0,  0.8032514305647592],
246     #                       [7.5,  0.7788892341777304],
247     #                       [8.0,  0.7730863447173755],
248     #                       [8.5,  0.7726206761501038],
249     #                       [9.0,  0.7721934195205071],
250     #                       [9.5,  0.7628473779358198],
251     #                       [10.0, 0.7459330274762097],
252     #                       [10.5, 0.7310049480450205],
253     #                       [11.0, 0.7177914274917664],
254     #                       [11.5, 0.799361832581412 ],
255     #                       [12.0, 0.8871279360742889],
256     #                       [12.5, 0.9504655842078242],
257     #                       [13.0, 1.0000251651970853],
258     #                       [13.5, 1.0390424010487957],
259     #                       [14.0, 1.0701572223736   ],
260     #                       [14.5, 1.0945877239199593]])
261
262     # SOWFA
263     # CP_curve = np.asarray([[2.0,  -0.2092219804533027],
264     #                       [2.5,  0.2352391893638198 ],
265     #                       [3.0,  0.46214453324002824],
266     #                       [3.5,  0.5476677311380832 ],
267     #                       [4.0,  0.5772456648046942],
268     #                       [4.5,  0.5833965967255043],
269     #                       [5.0,  0.5790298877294793],
270     #                       [5.5,  0.5701467792599509],
271     #                       [6.0,  0.5595564940228319],
272     #                       [6.5,  0.5480479331210222],
273     #                       [7.0,  0.5366246493538858],
274     #                       [7.5,  0.5258303873334416],
275     #                       [8.0,  0.5229191014420005],
276     #                       [8.5,  0.5224657416437077],
277     #                       [9.0,  0.5220516710065948],
278     #                       [9.5,  0.5175531496262384],
279     #                       [10.0, 0.5092952304943719],
280     #                       [10.5, 0.5016730194861562],
281     #                       [11.0, 0.4946298748497652],
282     #                       [11.5, 0.5326349577484786],
283     #                       [12.0, 0.5597671514540806],
284     #                       [12.5, 0.5679550280111124],
285     #                       [13.0, 0.5659876382489049],
286     #                       [13.5, 0.5572755521043566],
287     #                       [14.0, 0.5441595739848516],
288     #                       [14.5, 0.5280326705762761]])
289
290     CT = np.interp(U_d, CT_curve[:,0], CT_curve[:,1])
291     CP = np.interp(U_d, CP_curve[:,0], CP_curve[:,1])
292
293     return CT, CP
```

Listing F.2: Sub-functions.

Bibliography

[1] Tanvir Ahmad et al. “Implementation and Analyses of Yaw Based Coordinated Control of Wind Farms”. In: *Energies* 12.7 (April 2019), page 1266. ISSN: 1996-1073. DOI: 10.3390/en12071266.

[2] J.F. Ainslie. “Calculating the flowfield in the wake of wind turbines”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 27.1-3 (January 1988), pages 213–224. ISSN: 01676105. DOI: 10.1016/0167-6105(88)90037-2. URL: https://linkinghub.elsevier.com/retrieve/pii/0167610588900372.

[3] D.D. Apsley and M.A. Leschziner. “A new low-Reynolds-number nonlinear two-equation turbulence model for complex flows”. In: *International Journal of Heat and Fluid Flow* 19.3 (June 1998), pages 209–222. ISSN: 0142727X. DOI: 10.1016/S0142-727X(97)10007-8. URL: https://linkinghub.elsevier.com/retrieve/pii/S0142727X97100078.

[4] Peter Argyle et al. “Modelling turbulence intensity within a large offshore wind farm”. In: *Wind Energy* 21.12 (December 2018), pages 1329–1343. ISSN: 10954244. DOI: 10.1002/we.2257. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.2257.

[5] G.T. Atkinson and D.M.A. Wilson. “Unsteadiness and Structure in the Wake of a Wind Turbine”. In: *Wind Engineering* 10.3 (1986), pages 150–162.

[6] G.T. Atkinson and D.M.A. Wilson. “Wake Measurements on a Free Yawing 5m HAWT”. In: *Proceedings of the BWEA Wind Energy Conference*. Cambridge: Cavendish Lab, 1985, pages 75–83.

[7] Maulidi Barasa et al. “The balance effects of momentum deficit and thrust in cumulative wake models”. In: *Energy* 246 (May 2022), page 123399. ISSN: 03605442. DOI: 10.1016/j.energy.2022.123399. URL: https://linkinghub.elsevier.com/retrieve/pii/S0360544222003024.

[8] Paul Bartholomew et al. “Unified formulation of the momentum-weighted interpolation for collocated variable arrangements”. In: *Journal of Computational Physics* 375 (December 2018), pages 177–208. ISSN: 00219991. DOI: 10.1016/j.jcp.2018.08.030. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021999118305539.

[9] Jan Bartl, Franz Mühle, and Lars Sætran. “Wind tunnel study on power output and yaw moments for two yaw-controlled model wind turbines”. In: *Wind Energy Science* 3.2 (August 2018), pages 489–502. ISSN: 2366-7451. DOI: 10.5194/wes-3-489-2018.

[10] Jan Bartl et al. “Wind tunnel experiments on wind turbine wakes in yaw: effects of inflow turbulence and shear”. In: *Wind Energy Science* 3.1 (June 2018), pages 329–343. ISSN: 2366-7451. DOI: 10.5194/wes-3-329-2018.

[11] Majid Bastankhah and Fernando Porté-Agel. “A new analytical model for wind-turbine wakes”. In: *Renewable Energy* 70 (October 2014), pages 116–123. ISSN: 09601481. DOI: 10.1016/j.renene.2014.01.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960148114000317.

[12] Majid Bastankhah and Fernando Porté-Agel. “Experimental and theoretical study of wind turbine wakes in yawed conditions”. In: *Journal of Fluid Mechanics* 806 (2016), pages 506–541. ISSN: 14697645. DOI: 10.1017/jfm.2016.595.

[13] Majid Bastankhah and Fernando Porté-Agel. “Wind farm power optimization via yaw angle control: A wind tunnel study”. In: *Journal of Renewable and Sustainable Energy* 11.2 (2019). ISSN: 19417012. DOI: 10.1063/1.5077038. URL: http://dx.doi.org/10.1063/1.5077038.

[14] Majid Bastankhah et al. “Analytical solution for the cumulative wake of wind turbines in wind farms”. In: *Journal of Fluid Mechanics* 911 (March 2021), A53. ISSN: 0022-1120. DOI: 10.1017/jfm.2020.1037. URL: https://www.cambridge.org/core/product/identifier/S002211202001037X/type/journal_article.

[15] Mads Christian Baungaard. *Modeling of aerodynamic rotor interaction for multi-rotor wind turbines*. Kongens Lyngby, July 2019.

[16] Albert Betz. “Das Maximum der theoretisch möglichen Ausnützung des Windes durch Windmotoren”. In: *Zeitschrift für gesamte Turbinewesten* 20 (1920).

[17] Albert Betz. “Schraubenpropeller mit geringstem Energieverlust. Mit einem Zusatz von L. Prandtl”. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Matematisch-Physicalische Klasse* (1919), pages 193–217.

[18] F Blondel and M Cathelain. “An alternative form of the super-Gaussian wind turbine wake model”. 2020. DOI: 10.5194/wes-2019-99. URL: https://www.wind-energ-sci-discuss.net/wes-2019-99/.

[19] Frédéric Blondel. “Brief communication: A momentum-conserving superposition method applied to the super-Gaussian wind turbine wake model”. In: (June 2022). DOI: 10.5194/wes-2022-44. URL: https://doi.org/10.5194/wes-2022-44.

[20] E Branlard, M Gaunaa, and E Machefaux. “Investigation of a new model accounting for rotors of finite tip-speed ratio in yaw or tilt”. In: *Journal of Physics: Conference Series* 524.1 (June 2014), page 012124. ISSN: 1742-6596. DOI: 10.1088/1742-6596/524/1/012124. URL: https://iopscience.iop.org/article/10.1088/1742-6596/524/1/012124.

[21] E. Branlard and M. Gaunaa. “Cylindrical vortex wake model: right cylinder”. In: *Wind Energy* 18.11 (November 2015), pages 1973–1987. ISSN: 10954244. DOI: 10.1002/we.1800. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.1800.

[22] E. Branlard and M. Gaunaa. “Cylindrical vortex wake model: skewed cylinder, application to yawed or tilted rotors”. In: *Wind Energy* 19.2 (February 2016), pages 345–358. ISSN: 10954244. DOI: 10.1002/we.1838. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.1838.

[23] E. Branlard and M. Gaunaa. "Superposition of vortex cylinders for steady and unsteady simulation of rotors of finite tip-speed ratio". In: *Wind Energy* 19.7 (July 2016), pages 1307–1323. ISSN: 10954244. DOI: 10.1002/we.1899. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.1899.

[24] Filippo Campagnolo et al. "Wind tunnel testing of a closed-loop wake deflection controller for wind farm power maximization". In: *Journal of Physics: Conference Series*. Volume 753. 3. Institute of Physics Publishing, October 2016. DOI: 10.1088/1742-6596/753/3/032006.

[25] Filippo Campagnolo et al. "Wind tunnel testing of wake control strategies". In: *2016 American Control Conference (ACC)*. IEEE, July 2016, pages 513–518. ISBN: 978-1-4673-8682-1. DOI: 10.1109/ACC.2016.7524965. URL: http://ieeexplore.ieee.org/document/7524965/.

[26] Filippo Campagnolo et al. "Wind tunnel testing of wake steering with dynamic wind direction changes". In: *Wind Energy Science* 5.4 (October 2020), pages 1273–1295. ISSN: 2366-7451. DOI: 10.5194/wes-5-1273-2020. URL: https://wes.copernicus.org/articles/5/1273/2020/.

[27] Fernando Carbajo Fuertes, Corey Markfort, and Fernando Porté-Agel. "Wind Turbine Wake Characterization with Nacelle-Mounted Wind Lidars for Analytical Wake Model Validation". In: *Remote Sensing* 10.5 (April 2018), page 668. ISSN: 2072-4292. DOI: 10.3390/rs10050668. URL: http://www.mdpi.com/2072-4292/10/5/668.

[28] Ismail B. Celik et al. "Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications". In: *Journal of Fluids Engineering* 130.7 (July 2008), page 078001. ISSN: 00982202. DOI: 10.1115/1.2960953. URL: http://FluidsEngineering.asmedigitalcollection.asme.org/article.aspx?articleid=1434171.

[29] Matthew J Churchfield. *A Review of Wind Turbine Wake Models and Future Directions (Presentation)*. Technical report. Boulder, CO: NREL (National Renewable Energy Laboratory), August 2013. URL: https://www.osti.gov/biblio/1132152.

[30] Umberto Ciri, Mario A. Rotea, and Stefano Leonardi. "Effect of the turbine scale on yaw control". In: *Wind Energy* 21.12 (December 2018), pages 1395–1405. ISSN: 1095-4244. DOI: 10.1002/we.2262. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.2262.

[31] A. Crespo and J. Hernández. "Turbulence characteristics in wind-turbine wakes". In: *Journal of Wind Engineering and Industrial Aerodynamics* 61.1 (June 1996), pages 71–85. ISSN: 01676105. DOI: 10.1016/0167-6105(95)00033-X. URL: https://linkinghub.elsevier.com/retrieve/pii/016761059500033X.

[32] A. Crespo, J. Hernández, and S. Frandsen. "Survey of modelling methods for wind turbine wakes and wind farms". In: *Wind Energy* 2.1 (January 1999), pages 1–24. ISSN: 1095-4244. DOI: 10.1002/(SICI)1099-1824(199901/03)2:1<1::AID-WE16>3.0.CO;2-7. URL: https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1099-1824(199901/03)2:1%3C1::AID-WE16%3E3.0.CO;2-7.

[33] Antonio Crespo et al. "Numerical analysis of wind turbine wakes". In: *Workshop on Wind Energy Applications, Delphi, Greece* (1985), pages 15–25.

[34] Rick Damiani et al. “Assessment of wind turbine component loads under yaw-offset conditions”. In: *Wind Energy Science* 3.1 (April 2018), pages 173–189. ISSN: 2366-7451. DOI: 10.5194/wes-3-173-2018. URL: https://wes.copernicus.org/articles/3/173/2018/.

[35] Jean M. Delery. “Aspects of vortex breakdown”. In: *Progress in Aerospace Sciences* 30.1 (January 1994), pages 1–59. ISSN: 03760421. DOI: 10.1016/0376-0421(94)90002-7.

[36] Bart M. Doekemeijer et al. “Field experiment for open-loop yaw-based wake steering at a commercial onshore wind farm in Italy”. In: *Wind Energy Science* 6.1 (January 2021), pages 159–176. ISSN: 2366-7451. DOI: 10.5194/wes-6-159-2021.

[37] Paula Doubrawa et al. “A stochastic wind turbine wake model based on new metrics for wake characterization”. In: *Wind Energy* 20.3 (March 2017), pages 449–463. ISSN: 10991824. DOI: 10.1002/we.2015.

[38] DTU Wind Energy. *PyWake*. July 2022. URL: https://topfarm.pages.windenergy.dtu.dk/PyWake/.

[39] DTU Wind Energy. *PyWakeEllipSys*. June 2022. URL: https://topfarm.pages.windenergy.dtu.dk/cuttingedge/pywake/pywake_ellipsys/.

[40] Brandon L Ennis, Jonathan R White, and Joshua A Paquette. “Wind turbine blade load characterization under yaw offset at the SWiFT facility”. In: *Journal of Physics: Conference Series* 1037 (June 2018), page 052001. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1037/5/052001. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1037/5/052001.

[41] G. España et al. “Spatial study of the wake meandering using modelled wind turbines in a wind tunnel”. In: *Wind Energy* 14.7 (October 2011), pages 923–937. ISSN: 10954244. DOI: 10.1002/we.515. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.515.

[42] Joel H. Ferziger, Milovan Perić, and Robert L. Street. *Computational Methods for Fluid Dynamics*. Cham: Springer International Publishing, 2020. ISBN: 978-3-319-99691-2. DOI: 10.1007/978-3-319-99693-6. URL: http://link.springer.com/10.1007/978-3-319-99693-6.

[43] Paul Fleming et al. “A simulation study demonstrating the importance of large-scale trailing vortices in wake steering”. In: *Wind Energy Science* 3.1 (2018), pages 243–255. ISSN: 23667451. DOI: 10.5194/wes-3-243-2018.

[44] Paul Fleming et al. “Continued results from a field campaign of wake steering applied at a commercial wind farm – Part 2”. In: *Wind Energy Science* 5.3 (July 2020), pages 945–958. ISSN: 2366-7451. DOI: 10.5194/wes-5-945-2020.

[45] Paul Fleming et al. “Field test of wake steering at an offshore wind farm”. In: *Wind Energy Science* 2.1 (May 2017), pages 229–239. ISSN: 2366-7451. DOI: 10.5194/wes-2-229-2017. URL: https://wes.copernicus.org/articles/2/229/2017/.

[46] Paul Fleming et al. “Initial results from a field campaign of wake steering applied at a commercial wind farm – Part 1”. In: *Wind Energy Science* 4.2 (May 2019), pages 273–285. ISSN: 2366-7451. DOI: 10.5194/wes-4-273-2019.

[47] Paul Fleming et al. “Simulation comparison of wake mitigation control strategies for a two-turbine case”. In: *Wind Energy* 18.12 (2015), pages 2135–2143.

[48] Sten Frandsen et al. “Analytical modelling of wind speed deficit in large offshore wind farms”. In: *Wind Energy* 9.1-2 (January 2006), pages 39–53. ISSN: 1095-4244. DOI: 10.1002/we.189. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.189.

[49] Sten Tronæs Frandsen. “Turbulence and turbulence-generated structural loading in wind turbine clusters”. PhD thesis. Roskilde: Risø National Lab, January 2007. URL: https://www.osti.gov/etdeweb/biblio/20685756.

[50] Joeri Alexis Frederik et al. “Periodic dynamic induction control of wind farms: proving the potential in simulations and wind tunnel experiments”. In: *Wind Energy Science* 5.1 (February 2020), pages 245–257. ISSN: 2366-7451. DOI: 10.5194/wes-5-245-2020.

[51] Joeri Alexis Frederik et al. “The helix approach: Using dynamic individual pitch control to enhance wake mixing in wind farms”. In: *Wind Energy* 23.8 (August 2020), pages 1739–1751. ISSN: 1095-4244. DOI: 10.1002/we.2513.

[52] W Froude. *On the elementary relation between pitch, slip, and propulsion efficiency.* Technical report. Paris: NACA, 1920.

[53] P. M. O. Gebraad et al. “Wind plant power optimization through yaw control using a parametric model for wake effects-a CFD simulation study”. In: *Wind Energy* 19.1 (January 2016), pages 95–114. ISSN: 10954244. DOI: 10.1002/we.1822. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.1822.

[54] Hermann Glauert. *The Elements of Aerofoil and Airscrew Theory.* Cambridge University Press, 1983.

[55] Tuhfe Göçmen et al. “Wind turbine wake models developed at the technical university of Denmark: A review”. In: *Renewable and Sustainable Energy Reviews* 60 (July 2016), pages 752–769. ISSN: 13640321. DOI: 10.1016/j.rser.2016.01.113. URL: https://linkinghub.elsevier.com/retrieve/pii/S136403211600143X.

[56] K. Gunn et al. “Limitations to the validity of single wake superposition in wind farm yield assessment”. In: *Journal of Physics: Conference Series* 749.1 (2016). ISSN: 17426596. DOI: 10.1088/1742-6596/749/1/012003.

[57] Martin Hansen. *Aerodynamics of Wind Turbines.* Routledge, May 2015. ISBN: 9781317671039. DOI: 10.4324/9781315769981. URL: https://www.taylorfrancis.com/books/9781317671039.

[58] Francis H. Harlow and J. Eddie Welch. “Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface”. In: *Physics of Fluids* 8.12 (1965), page 2182. ISSN: 00319171. DOI: 10.1063/1.1761178. URL: https://aip.scitation.org/doi/10.1063/1.1761178.

[59] T G Herges et al. “Detailed analysis of a waked turbine using a high-resolution scanning lidar”. In: *Journal of Physics: Conference Series* 1037 (June 2018), page 072009. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1037/7/072009. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1037/7/072009.

[60] Daan van der Hoek et al. “Effects of axial induction control on wind farm energy production - A field test”. In: *Renewable Energy* 140 (September 2019), pages 994–1003. ISSN: 09601481. DOI: 10.1016/j.renene.2019.03.117.

[61] Daan van der Hoek et al. “Experimental analysis of the effect of dynamic induction control on a wind turbine wake”. In: *Wind Energy Science* 7.3 (June 2022), pages 1305–1320. ISSN: 2366-7451. DOI: 10.5194/wes-7-1305-2022.

[62] Peng Hou et al. “A review of offshore wind farm layout optimization and electrical system design methods”. In: *Journal of Modern Power Systems and Clean Energy* 7.5 (2019), pages 975–986.

[63] Daniel R. Houck. “Review of wake management techniques for wind turbines”. In: *Wind Energy* 25.2 (February 2022), pages 195–220. ISSN: 1095-4244. DOI: 10.1002/we.2668. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.2668.

[64] Michael Howland et al. “Optimal closed-loop wake steering, Part 2: Diurnal cycle atmospheric boundary layer conditions”. In: *Wind Energy Science Discussions* (2021), pages 1–30. ISSN: 2366-7443. DOI: 10.5194/wes-2021-85.

[65] Michael F. Howland, Sanjiva K. Lele, and John O. Dabiri. “Wind farm power optimization through wake steering”. In: *Proceedings of the National Academy of Sciences* 116.29 (July 2019), pages 14495–14500. ISSN: 0027-8424. DOI: 10.1073/pnas.1903680116. URL: https://pnas.org/doi/full/10.1073/pnas.1903680116.

[66] Michael F. Howland et al. “Wake structure in actuator disk models of wind turbines in yaw under uniform inflow conditions”. In: *Journal of Renewable and Sustainable Energy* 8.4 (July 2016), page 043301. ISSN: 1941-7012. DOI: 10.1063/1.4955091. URL: http://dx.doi.org/10.1063/1.4955091%20http://aip.scitation.org/doi/10.1063/1.4955091.

[67] IEA. *Global Energy Review 2021*. Technical report. Paris, 2021. URL: https://www.iea.org/reports/global-energy-review-2021.

[68] *IEC 61400-1*. Technical report. International Electrotechnical Commission, 1999.

[69] Takeshi Ishihara and Guo-Wei Qian. “A new Gaussian-based analytical wake model for wind turbines considering ambient turbulence intensities and thrust coefficient effects”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 177 (June 2018), pages 275–292. ISSN: 01676105. DOI: 10.1016/j.jweia.2018.04.010. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167610517306396.

[70] Pim Anthonius Everhart Petrus Allegonda Jacobs. *A parametric study of shear effects on wind turbine wake deflection using CFD*. 2020.

[71] Hrvoje Jasak. “Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows”. PhD thesis. London: Imperial College of Science, Technology and Medicine, June 1996.

[72] Niels Otto Jensen. *A note on wind generator interaction*. 1983.

[73] Ángel Jiménez, Antonio Crespo, and Emilio Migoya. “Application of a LES technique to characterize the wake deflection of a wind turbine in yaw”. In: *Wind Energy* 13.6 (December 2009), pages 559–572. ISSN: 10954244. DOI: 10.1002/we.380. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.380.

[74] J. Jonkman et al. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*. Technical report. Golden, CO: National Renewable Energy Laboratory (NREL), February 2009. DOI: 10.2172/947422. URL: http://www.osti.gov/servlets/purl/947422-nhrlni/.

[75] Stoyan Kanev, Edwin Bot, and Jack Giles. "Wind Farm Loads under Wake Redirection Control". In: *Energies* 13.16 (August 2020), page 4088. ISSN: 1996-1073. DOI: 10.3390/en13164088. URL: https://www.mdpi.com/1996-1073/13/16/4088.

[76] I Katić, Jørgen Højstrup, and Niels Otto Jensen. "A simple model for cluster efficiency". In: *European Wind Energy Association Conference and Exhibition*. A. Raguzzi. 1987, pages 407–410.

[77] Ali C. Kheirabadi and Ryozo Nagamune. "A quantitative review of wind farm control with the objective of wind farm power maximization". In: *Journal of Wind Engineering and Industrial Aerodynamics* 192 (September 2019), pages 45–73. ISSN: 01676105. DOI: 10.1016/j.jweia.2019.06.015. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167610519305240.

[78] Jennifer King et al. "Control-oriented model for secondary effects of wake steering". In: *Wind Energy Science* 6.3 (2021), pages 701–714. ISSN: 23667451. DOI: 10.5194/wes-6-701-2021.

[79] Gijs A.M. van Kuik. "The Lanchester–Betz–Joukowsky limit". In: *Wind Energy* 10.3 (May 2007), pages 289–291. ISSN: 10954244. DOI: 10.1002/we.218. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.218.

[80] Gijs A.M. van Kuik, Jens Nørkær Sørensen, and Valery L Okulov. "Rotor theories by Professor Joukowsky: Momentum theories". In: *Progress in Aerospace Sciences* 73 (February 2015), pages 1–18. ISSN: 03760421. DOI: 10.1016/j.paerosci.2014.10.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S0376042114000955.

[81] Pijush K Kundu, Ira M Cohen, and David R Dowling. *Fluid Mechanics*. 6th edition. Elsevier, 2016. ISBN: 9780124059351. DOI: 10.1016/C2012-0-00611-4. URL: https://linkinghub.elsevier.com/retrieve/pii/C20120006114.

[82] Maarten Paul van der Laan and Søren Juhl Andersen. "The turbulence scales of a wind turbine wake: A revisit of extended k-epsilon models". In: *Journal of Physics: Conference Series* 1037.7 (June 2018), page 072001. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1037/7/072001. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1037/7/072001.

[83] Maarten Paul van der Laan, Mark C. Kelly, and Niels N. Sørensen. "A new k-epsilon model consistent with Monin-Obukhov similarity theory". In: *Wind Energy* 20.3 (March 2017), pages 479–489. ISSN: 10954244. DOI: 10.1002/we.2017. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.2017.

[84] Maarten Paul van der Laan et al. "An improved k - ϵ model applied to a wind turbine wake in atmospheric turbulence". In: *Wind Energy* 18.5 (May 2015), pages 889–907. ISSN: 10954244. DOI: 10.1002/we.1736. URL: https://onlinelibrary-wiley-com.tudelft.idm.oclc.org/doi/epdf/10.1002/we.1736.

[85] Maarten Paul van der Laan et al. *Efficient Turbulence Modeling for CFD Wake Simulations*. Technical report. DTU Wind Energy PhD, 2014. URL: https://orbit.dtu.dk/en/publications/efficient-turbulence-modeling-for-cfd-wake-simulations.

[86] Maarten Paul van der Laan et al. "Fluid scaling laws of idealized wind farm simulations". In: *Journal of Physics: Conference Series* 1618.6 (September 2020), page 062018. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1618/6/062018. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1618/6/062018.

[87] Maarten Paul van der Laan et al. “Nonlinear Eddy Viscosity Models applied to Wind Turbine Wakes”. In: *Proceedings of the 2013 International Conference on Aerodynamics of Offshore Wind Energy Systems and Wakes*. Lyngby: Technical University of Denmark, 2013, pages 514–525.

[88] Maarten Paul van der Laan et al. “The k - ϵ - f P model applied to double wind turbine wakes using different actuator disk force methods”. In: *Wind Energy* 18.12 (December 2015), pages 2223–2240. ISSN: 10954244. DOI: 10.1002/we.1816. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.1816.

[89] Maarten Paul van der Laan et al. “The k - ϵ - f P model applied to wind farms”. In: *Wind Energy* 18.12 (December 2015), pages 2065–2084. ISSN: 10954244. DOI: 10.1002/we.1804. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.1804.

[90] Frederick W Lanchester. “A contribution to the theory of propulsion and the screw propeller”. In: *Journal of the American Society for Naval Engineers* 27.2 (1915), pages 509–510.

[91] Gunner Christian Larsen. *A Simple Wake Calculation Procedure*. Technical report. 1988. URL: https://orbit.dtu.dk/en/publications/a-simple-wake-calculation-procedure.

[92] Gunner Christian Larsen et al. *Dynamic wake meandering modeling Title: Dynamic wake meandering modeling*. Technical report. June 2007. URL: https://www.osti.gov/etdeweb/biblio/20941220.

[93] Mou Lin and Fernando Porté-Agel. “Large-Eddy Simulation of Yawed Wind-Turbine Wakes: Comparisons with Wind Tunnel Measurements and Analytical Wake Models”. In: *Energies* 12.23 (November 2019), page 4574. ISSN: 1996-1073. DOI: 10.3390/en12234574. URL: https://www.mdpi.com/1996-1073/12/23/4574.

[94] P. B. S. Lissaman. “Energy Effectiveness of Arbitrary Arrays of Wind Turbines”. In: *Journal of Energy* 3.6 (November 1979), pages 323–328. ISSN: 0146-0412. DOI: 10.2514/3.62441. URL: https://arc.aiaa.org/doi/10.2514/3.62441.

[95] Luis A. Martínez-Tossas et al. “A Comparison of Actuator Disk and Actuator Line Wind Turbine Models and Best Practices for Their Use”. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Reston, Virigina: American Institute of Aeronautics and Astronautics, January 2012. ISBN: 978-1-60086-936-5. DOI: 10.2514/6.2012-900.

[96] Luis A. Martínez-Tossas et al. “The aerodynamics of the curled wake: a simplified model in view of flow control”. In: *Wind Energy Science* 4.1 (March 2019), pages 127–138. ISSN: 2366-7451. DOI: 10.5194/wes-4-127-2019. URL: https://wes.copernicus.org/articles/4/127/2019/.

[97] Luis A. Martínez-Tossas et al. “The curled wake model: a three-dimensional and extremely fast steady-state wake solver for wind plant flows”. In: *Wind Energy Science* 6.2 (April 2021), pages 555–570. ISSN: 2366-7451. DOI: 10.5194/wes-6-555-2021. URL: https://wes.copernicus.org/articles/6/555/2021/.

[98] D. Medici et al. “The upstream flow of a wind turbine: blockage effect”. In: *Wind Energy* 14.5 (July 2011), pages 691–697. ISSN: 10954244. DOI: 10.1002/we.451. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.451.

[99] Charles Meneveau. "Big wind power: seven questions for turbulence research". In: *Journal of Turbulence* 20.1 (2019), pages 2–20. ISSN: 14685248. DOI: 10.1080/14685248.2019.1584664. URL: https://doi.org/10.1080/14685248.2019.1584664.

[100] Johan Meyers et al. "Wind farm flow control: prospects and challenges". In: *Wind Energy Science (Preprint)* (2022). DOI: 10.5194/wes-2022-24. URL: https://doi.org/10.5194/wes-2022-24.

[101] Jess Andreas Michelsen. *Basis3D - A platform for development of multiblock PDE solvers: β - release*. Technical report. Lyngby, Denmark: Technical University of Denmark, 1992.

[102] Jess Andreas Michelsen. *Block structured Multigrid solution of 2D and 3D elliptic PDE's*. Technical report. Technical University of Denmark, 1994.

[103] Robert Flemming Mikkelsen. "Actuator Disc Methods Applied to Wind Turbines". PhD thesis. Kgs. Lyngby: Technical University of Denmark, January 2004.

[104] Andrei Sergeevich Monin and Aleksandr Mikhailovich Obukhov. "Basic laws of turbulent mixing in the surface layer of the atmosphere". In: *Originally published in Tr. Akad. Nauk SSSR Geophiz. Inst* 24.151 (1954), pages 163–187.

[105] Wim Munters and Johan Meyers. "Towards practical dynamic induction control of wind farms: analysis of optimally controlled wind-farm boundary layers and sinusoidal induction control of first-row turbines". In: *Wind Energy Science* 3.1 (June 2018), pages 409–425. ISSN: 2366-7451. DOI: 10.5194/wes-3-409-2018.

[106] Amin Niayifar and Fernando Porté-Agel. "Analytical Modeling of Wind Farms: A New Approach for Power Prediction". In: *Energies* 9.9 (September 2016), page 741. ISSN: 1996-1073. DOI: 10.3390/en9090741. URL: http://www.mdpi.com/1996-1073/9/9/741.

[107] NREL. *FLORIS. Version 2.4*. 2021. URL: https://github.com/NREL/floris.

[108] Søren Ott, Jacob Berg, and Morten Nielsen. *Linearised CFD Models for Wakes*. Technical report. 2011. URL: https://www.osti.gov/etdeweb/biblio/1033688.

[109] Hans A Panofsky and J A Dutton. *Atmospheric Turbulence*. New York, US: Wiley-Interscience, 1984.

[110] Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, August 2000. ISBN: 9780521591256. DOI: 10.1017/CBO9780511840531. URL: https://www.cambridge.org/core/product/identifier/9780511840531/type/book.

[111] Fernando Porté-Agel, Majid Bastankhah, and Sina Shamsoddin. "Wind-Turbine and Wind-Farm Flows: A Review". In: *Boundary-Layer Meteorology* 174.1 (January 2020), pages 1–59. ISSN: 0006-8314. DOI: 10.1007/s10546-019-00473-0. URL: http://link.springer.com/10.1007/s10546-019-00473-0.

[112] Fernando Porté-Agel et al. "Large-eddy simulation of atmospheric boundary layer flow through wind turbines and wind farms". In: *Journal of Wind Engineering and Industrial Aerodynamics* 99.4 (April 2011), pages 154–168. ISSN: 01676105. DOI: 10.1016/j.jweia.2011.01.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167610511000134.

[113] Guo Wei Qian and Takeshi Ishihara. "A new analytical wake model for yawed wind turbines". In: *Energies* 11.3 (2018). ISSN: 19961073. DOI: 10.3390/en11030665.

[114] Guo Wei Qian and Takeshi Ishihara. "Wind farm power maximization through wake steering with a new multiple wake model for prediction of turbulence intensity". In: *Energy* 220 (April 2021). ISSN: 03605442. DOI: 10.1016/j.energy.2020.119680.

[115] D.C. Quarton and J.F. Ainslie. "Turbulence in Wind Turbine Wakes". In: *Wind Engineering* 14.1 (1990), pages 15–23. URL: https://www.jstor.org/stable/43749409.

[116] William John Macquorn Rankine. "On the mechanical principles of the action of propellers". In: *Transactions of the Institution of Naval Architects* 6 (1865).

[117] Pierre-Elouan Réthoré and Niels N Sørensen. "A discrete force allocation algorithm for modelling wind turbines in computational fluid dynamics". In: *Wind Energy* 15.7 (October 2012), pages 915–926. ISSN: 10954244. DOI: 10.1002/we.525. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.525.

[118] Pierre-Elouan Réthoré et al. "Verification and validation of an actuator disc model". In: *Wind Energy* 17.6 (June 2014), pages 919–937. ISSN: 10954244. DOI: 10.1002/we.1607. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.1607.

[119] C. M. Rhie and W. L. Chow. "Numerical study of the turbulent flow past an airfoil with trailing edge separation". In: *AIAA Journal* 21.11 (November 1983), pages 1525–1532. ISSN: 0001-1452. DOI: 10.2514/3.8284. URL: https://arc.aiaa.org/doi/10.2514/3.8284.

[120] P.J Richards and R.P Hoxey. "Appropriate boundary conditions for computational wind engineering models using the k-ϵ turbulence model". In: *Journal of Wind Engineering and Industrial Aerodynamics* 46-47 (August 1993), pages 145–153. ISSN: 01676105. DOI: 10.1016/0167-6105(93)90124-7. URL: https://linkinghub.elsevier.com/retrieve/pii/0167610593901247.

[121] Lewis Fry Richardson. "IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 210.459-470 (January 1911), pages 307–357. ISSN: 0264-3952. DOI: 10.1098/rsta.1911.0009. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.1911.0009.

[122] Javier Criado Risco. *A CFD surrogate model for annual energy production calculations of wind farms*. 2021.

[123] Christopher J. Roy. "Grid Convergence Error Analysis for Mixed-Order Numerical Schemes". In: *AIAA Journal* 41.4 (April 2003), pages 595–604. ISSN: 0001-1452. DOI: 10.2514/2.2013. URL: https://arc.aiaa.org/doi/10.2514/2.2013.

[124] B. Sanderse, S. P. Van Der Pijl, and B. Koren. "Review of computational fluid dynamics for wind turbine wake aerodynamics". In: *Wind Energy* 14.7 (2011), pages 799–819. ISSN: 10954244. DOI: 10.1002/we.458.

[125] Rabia Shakoor et al. "Wake effect modeling: A review of wind farm layout optimization using Jensen's model". In: *Renewable and Sustainable Energy Reviews* 58 (May 2016), pages 1048–1059. ISSN: 13640321. DOI: 10.1016/j.rser.2015.12.229. URL: https://linkinghub.elsevier.com/retrieve/pii/S1364032115016123.

[126] K. Shaler, M. Debnath, and J. Jonkman. "Validation of FAST.Farm Against Full-Scale Turbine SCADA Data for a Small Wind Farm". In: *Journal of Physics: Conference Series* 1618.6 (September 2020), page 062061. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1618/6/062061. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1618/6/062061.

[127] Kelsey Shaler et al. "Loads assessment of a fixed-bottom offshore wind farm with wake steering". In: *Wind Energy* (July 2022). ISSN: 1095-4244. DOI: 10.1002/we.2756. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.2756.

[128] Carl R. Shapiro, Dennice F. Gayme, and Charles Meneveau. "Modelling yawed wind turbine wakes: A lifting line approach". In: *Journal of Fluid Mechanics* 841 (2018), R11–R112. ISSN: 14697645. DOI: 10.1017/jfm.2018.75.

[129] Carl R. Shapiro et al. "A wake modeling paradigm for wind farm design and control". In: *Energies* 12.15 (2019). ISSN: 19961073. DOI: 10.3390/en12152956.

[130] Siemens Gamesa Renewable Energy. *Wake Adapt Press Release*. November 2019. URL: https://www.siemensgamesa.com/en-int/newsroom/2019/11/191126-siemens-gamesa-wake-adapt-en.

[131] Eric Simley et al. "Characterization of wind velocities in the upstream induction zone of a wind turbine using scanning continuous-wave lidars". In: *Journal of Renewable and Sustainable Energy* 8.1 (January 2016), page 013301. ISSN: 1941-7012. DOI: 10.1063/1.4940025. URL: http://aip.scitation.org/doi/10.1063/1.4940025.

[132] Eric Simley et al. "Results from a wake-steering experiment at a commercial wind plant: investigating the wind speed dependence of wake-steering performance". In: *Wind Energy Science* 6.6 (November 2021), pages 1427–1453. ISSN: 2366-7451. DOI: 10.5194/wes-6-1427-2021. URL: https://wes.copernicus.org/articles/6/1427/2021/.

[133] Jens Nørkær Sørensen et al. "Analytical body forces in numerical actuator disc model of wind turbines". In: *Renewable Energy* 147 (March 2020), pages 2259–2271. ISSN: 09601481. DOI: 10.1016/j.renene.2019.09.134. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960148119314788.

[134] Niels N Sørensen. "General Purpose Flow Solver Applied to Flow over Hills". PhD thesis. Roskilde, Denmark: Risø National Laboratory, December 2003.

[135] Niels N Sørensen et al. "Identification of severe wind conditions using a Reynolds Averaged Navier-Stokes solver". In: *Journal of Physics: Conference Series* 75.1 (July 2007), page 012053. ISSN: 1742-6596. DOI: 10.1088/1742-6596/75/1/012053. URL: https://iopscience.iop.org/article/10.1088/1742-6596/75/1/012053.

[136] Richard J.A.M. Stevens and Charles Meneveau. *Flow Structure and Turbulence in Wind Farms*. Volume 49. 2017, pages 311–339. ISBN: 0108160602. DOI: 10.1146/annurev-fluid-010816-060206.

[137] Henk Tennekes and John L. Lumley. *A First Course in Turbulence*. 2020. DOI: 10.7551/mitpress/3014.001.0001.

[138] Albert M. Urbán et al. "Optimal yaw strategy for optimized power and load in various wake situations". In: *Journal of Physics: Conference Series* 1102 (October 2018), page 012019. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1102/1/012019.

[139] L.J. Vermeer, J.N. Sørensen, and A. Crespo. "Wind turbine wake aerodynamics". In: *Progress in Aerospace Sciences* 39.6-7 (August 2003), pages 467–510. ISSN: 03760421. DOI: 10.1016/S0376-0421(03)00078-2. URL: https://linkinghub.elsevier.com/retrieve/pii/S0376042103000782.

[140] P. E. J. Vermeulen. "An experimental analysis of wind turbine wakes". In: *Precision Engineering* (1980), pages 431–450. ISSN: 04016359 and 01416359.

[141] Christopher R. Vogel and Richard H. J. Willden. "Investigation of wind turbine wake superposition models using Reynolds-averaged Navier-Stokes simulations". In: *Wind Energy* 23.3 (March 2020), pages 593–607. ISSN: 1095-4244. DOI: 10.1002/we.2444. URL: https://onlinelibrary.wiley.com/doi/10.1002/we.2444.

[142] S Voutsinas, K Rados, and A Zervos. *On the Analysis of Wake Effects in Wind Parks*. Technical report 4. 1990, pages 204–219. URL: http://www.jstor.org/stable/43749429.

[143] R. E. Wilson, P. B.S. Lissaman, and S. N. Walker. *Aerodynamic performance of wind turbines. Final report*. Technical report. Historical Energy Database (United States), June 1976. DOI: 10.2172/7315651. URL: http://www.osti.gov/servlets/purl/7315651/.

[144] Aidan Wimshurst. *Rhie & Chow Interpolation*. September 2021. URL: https://www.youtube.com/watch?v=yqZ59Xn_aF8&list=PLnJ8lIgfDbkp5DtCPtP2rcqEEUJk-PM8N&index=8.

[145] J. W. van Wingerden et al. "Expert Elicitation on Wind Farm Control". In: *Journal of Physics: Conference Series* 1618.2 (September 2020), page 022025. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1618/2/022025. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1618/2/022025.

[146] Yu-Ting Wu and Fernando Porté-Agel. "Large-Eddy Simulation of Wind-Turbine Wakes: Evaluation of Turbine Parametrisations". In: *Boundary-Layer Meteorology* 138.3 (March 2011), pages 345–366. ISSN: 0006-8314. DOI: 10.1007/s10546-010-9569-x.

[147] Daniel S. Zalkind and Lucy Y. Pao. "The fatigue loading effects of yaw control for wind plants". In: *2016 American Control Conference (ACC)*. IEEE, July 2016, pages 537–542. ISBN: 978-1-4673-8682-1. DOI: 10.1109/ACC.2016.7524969. URL: http://ieeexplore.ieee.org/document/7524969/.

[148] Lu Zhan, Stefano Letizia, and Giacomo Valerio Iungo. "Optimal tuning of engineering wake models through lidar measurements". In: *Wind Energy Science* 5.4 (November 2020), pages 1601–1622. ISSN: 2366-7451. DOI: 10.5194/wes-5-1601-2020.

[149] Nikolaj Yegorovich Zhukovsky. "Windmill of the NEJ type". In: *Transactions of the Central Institute for Aero-Hydrodynamics of Moscow* 1 (1920), page 57.

[150] Haohua Zong and Fernando Porté-Agel. "A momentum-conserving wake superposition method for wind farm power prediction". In: *Journal of Fluid Mechanics* 889 (2020). ISSN: 14697645. DOI: 10.1017/jfm.2020.77.