TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATIONS

ABL	Atmospheric Boundary Layer
AD	Actuator Disk
BEM	Blade Element/Momentum theory
CFL	Courant-Friedrichs-Lewy condition
DES	Detached-Eddy Simulation
DNS	Direct Numerical Simulation
HIT	Homogeneous-Isotropic Turbulence
HAWT	Horizontal-Axis Wind Turbine
HWA	Hot-Wire Anemometry
LDV	Laser Doppler Velocimetry
LES	Large-Eddy Simulations
LFT	Longtudinal Flow Times
LIDAR	Large Detection and Ranging
PISO	Pressure-Implicit Split-Operator
PIV	Particle Image Velocimetry
PSD	Power Spectral Density
QUICK	Quadratic Upstream Interpolation for Convective Kinetics
RANS	Reynolds-Averaged Navier-Stokes simulation
RDT	Rapid Distortion Theory

RPM	Revolutions Per Minute
SGÉ	Simulation aux Grandes Échelles
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
SGS	Sub-Grid Scale
TP	Turbulence Plane
URANS	Unsteady Reynolds-Averaged Navier-Stokes simulation

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

General notation

$\langle a \rangle$	Time or space average
a_i, a_{ij}	vector or tensor expressed in index notation
$\boldsymbol{a}$	vector or tensor expressed in vector notation ($\boldsymbol{a} = a_i \hat{e}_i$)
$\overline{a}$ or a_{res}	Filtered or resolved value
a_{SGS}	Sub-grid or modelled value
a_{tot}	Total value computed by the LES ($a_{tot} = a_{\text{SGS}} + a_{res}$)
a^*	Complex conjugate of a

Upper-case Roman

A_D	Area covered by the rotor or actuator disk [m^2]
B	Number of blades in the rotor
C_D	Drag coefficient [-]
C_L	Lift coefficient [-]
C_m	Weighting parameter in mixed-scale SGS model (eq. 2.7) [-]
C_s	Smagorinsky constant [-]
C_T	Thrust coefficient [-]
D	Diameter of the rotor or actuator disk [m]
$E(\kappa)$	Three-dimensional turbulence energy spectrum as function of κ [m^3/s^2]

$F_1(\kappa),\ F_2(\kappa)$	One-dimensional longitudinal and transversal turbulence energy spectra as function of κ [m^3/s^2]
$\boldsymbol{F}$	Vector of aerodynamic forces at the blade section [N]
F_x	Thrust force [N]
F_D	Drag force [N]
F_L	Lift force [N]
F_θ	Tangential force [N]
I_d	Moment of inertia of the drivetrain [kg·m^2]
I_{gen}	Moment of inertia of the generator [kg·m^2]
I_{rot}	Moment of inertia of the rotor [kg·m^2]
F_θ	Tangential force [N]
$L_{ij}^{(d)}$	Integral lengthscale from correlations of velocity components i and j in direction d (eq. 2.46) [m]
L_1	Longitudinal integral lengthscale in the streamwise direction [m]
L_2	Transversal integral lengthscale in the streamwise direction [m]
$L_{1,\mathrm{B}}$	L_1 of the synthetic velocity field (volume average) [m]
M	Spacing of grid used for turbulence generation in a wind tunnel [m]
P	Power [W]
Pe	Péclet number (eq. 2.12) [-]
Q_{aero}	Aerodynamic Torque [Nm]
Q_{gen}	Generator Torque [Nm]

R	Radius of the rotor or actuator disk [m]
R_{ij}	Covariance tensor (eq. 2.44) [m^2/s^2]
$\mathcal{R}_{ij}$	Correlation coefficient (eq. 2.45) [-]
Re_λ	Taylor scale based Reynolds number [-]
S_{ij}	Rate-of-strain tensor (eq. 2.4) [$1/\mathrm{s}$]
$S_{11}(\kappa)$	One-dimensional longitudinal PSD spectrum as function of κ [m^2/s]
TI	Streamwise turbulence intensity $\mathrm{TI} = \frac{u_{rms}}{\langle U \rangle}$ [-]
$\mathrm{TI_g}$	Global turbulence intensity (eq. 1.4) [-]
$\mathrm{TI_B}$	TI of the synthetic turbulence box (volume average) [-]
U_0	Inflow velocity [m/s]
U_∞	Freestream velocity [m/s]
U_{rel}	Relative velocity at the blade section (eq. 2.28) [m/s]
U_θ	Velocity component at the plane of rotation [m/s]
V_D	Volume of the Actuator Disk [m^3]

Lower-case Roman

c	Chord [m]
c_A	Constant for decaying turbulence parametrization (eq. 2.58 and 2.59) [-]
c_{B_1}	Constant for parametrization of integral lengthscale evolution (eq. 2.61) [-]
c_{B_2}	Constant for parametrization of Taylor lengthscale evolution (eq. 2.62) [-]
$\mathbf{e}_D$	Unitary vector in the direction of the drag force

$\mathbf{e}_L$	Unitary vector in the direction of the lift force
f_i	External force or source term in the N-S/LES momentum equation [m/s^2]
$g(x)$	Gaussian distribution in the $x-$direction
k	Turbulent kinetic energy $\frac{1}{2}\left(\langle u^2\rangle + \langle v^2\rangle + \langle w^2\rangle\right)$ per unit mass [m^2/s^2]
$\overline{k}_c$	Resolved turbulent kinetic energy per unit mass of highest frequencies (eq. 2.8) [m^2/s^2]
ℓ	Wavelength $\ell = 2\pi/\kappa$ [m]
p	Pressure [N/m]
r	Radial position in the rotor or actuator disk [m]
t	Time [s]
$x,\ y,\ z$	Streamwise, vertical and spanwise directions in a cartesian frame of reference [m]
u, v, w	Streamwise, vertical and spanwise components of the instantaneous velocity [m/s]
u_{rms}	Root-mean-square velocity [m/s]
u^t	Test field velocity (eq. 2.9) [m/s]
u_*	Friction velocity [m/s]
x_0	Virtual origin (eq. 2.58) [m]
x_D	Target location (simulations without AD) or location of AD
y_0	Roughness length [m]

Greek

α	Kolmogorov constant [-]
α_a	Angle of attack [rad]
α_m	Weighting parameter in mixed-scale SGS model (eq. 2.7) [-]
δ_{ij}	Kronecker delta
Δ_i	Filter size [m]
$\widetilde{\Delta}_i$	Test filter size [m]
ε	Dissipation of turbulent kinetic energy [m^2/s^3]
κ	Wavenumber [1/m]
κ_c	Cutoff wavenumber [1/m]
κ^*	von Kármán constant [-]
λ_1	Longitudinal Taylor lengthscale (eq. 2.47a) [m]
λ_2	Transversal Taylor lengthscale (eq. 2.47b) [m]
Λ	Tip-speed ratio $\Lambda = \Omega R/U_0$ [-]
ν	Kinematic viscosity [m^2/s]
ν_t	Turbulent viscosity [m^2/s]
ν_{SGS}	Sub-grid or turbulent viscosity in LES [m^2/s]
ν_{eff}	Effective viscosity $\nu_{eff} = \nu + \nu_{\text{SGS}}$ [m^2/s]
η	Kolmogorov lengthscale [m]
Φ_{ij}	Spectral tensor

φ	Angle between the place of rotation and the relative vel. (eq. 2.29) [rad]
σ	Standard deviation
τ_{ij}^{SGS}	Sub-grid stress tensor [$1/(\text{ms}^2)$]
θ	Angular coordinate [rad]
θ_p	Pitch angle [rad]
ϑ	Blade tip correction factor [-]
Ω	Angular velocity of the rotor [cycles/s] or [RPM]

INTRODUCTION

The wind industry has been under a period of great expansion for some years. After a slow-down in added capacity in 2013, the next two years set again new records, with more than 50 GW and 60 GW of new installations (World Wind Energy Association, 2015, 2016). Currently, the contribution of wind power is approaching 5% of the total electricity demand worldwide and in some countries (Denmark, Spain, Portugal, Ireland, the United Kingdom and Germany), it contributes to at least 10% of their energy needs (World Wind Energy Association, 2015). After the recent economic recession in many parts of the world (circa 2010), the outlook for wind energy is positive and it is expected that new turbines will be installed at a higher pace in the forthcoming years (Global Wind Energy Council, 2015). However, due to the reduced availability not only of ideal sites —flat and obstacle-free— but of land in general, wind turbines are located close to each other in wind farms. Evidently, economic reasons can also play a role in such occurrence, to maximize the profitability of the land. This causes interference problems owing to the interactions between the turbines themselves and with the wind flow which in turn reduce the efficiency of the array.

In particular, the wind turbine wakes increase the level of turbulence and cause a momentum defect within the wind farm, which may lead to an increase of structural loads on the rotors and to a reduction of the power output, respectively. Moreover, the rise of turbulence along with the apparition of dynamic loads can induce fatigue that may produce considerable damage in the turbines (Jiménez *et al.*, 2008). For these reasons, the prediction of turbulence characteristics in the wakes and its interaction with other turbines is a crucial element to predict the everyday as well as the long-term production of the park. Notably, the accurate prediction of turbulence levels in the wake contributes to improve the estimation of the wake recovery and with it the forecasting of wind resources for the downstream turbines. This issue is of particular relevance if it is considered that linearized models often employed in the industry are not adequate to assess these effects (Palma *et al.*, 2008), due to their inability to model turbulence. For instance, since turbulence enhances mixing in the wake, which in turn permits a faster recovery of the wind speed, neglecting it can produce an underestimation of power production in a park.

When numerical simulations are made of large wind parks, computational limitations oblige us to employ simpler rotor models that, while numerically less expensive, are requested to produce a minimum level of detail in wake features that yields an acceptable reproduction of the interaction of these and the downstream wind turbines as well as other wakes. Amongst the simplest formulations of the rotor model is the Actuator Disk (AD) (Sørensen and Myken, 1992; Ammara *et al.*, 2002) which reproduces the effect of the rotor in the incoming flow by means of a permeable surface in the shape of a disk where flow momentum is lost. In its most basic conception, the AD constitutes a one-dimensional force opposite to the flow, perpendicular to the rotor plane, with no rotation or airfoil properties. It has been shown experimentally (Aubrun *et al.*, 2013) and numerically (Jiménez *et al.*, 2008; Porté-Agel *et al.*, 2011) that the characteristics of the turbulence in the *far wake* region can be adequately reproduced employing uniformly loaded disks when compared to either a three-bladed model turbine or an actuator line method (Sørensen and Shen, 2002) . Conversely, the numerical representation of the wake field closer to the rotor can noticeably benefit from the introduction of the tangential velocity component, as it has been shown in comparisons with experimental wake data (Porté-Agel *et al.*, 2011) or more sophisticated rotor models (Troldborg *et al.*, 2015).

Therefore, when these rotor models are employed for the simulation of wakes, it is particularly important to assess the accuracy in the representation of the turbulence features yielded by the computations. For this reason, the present work concentrates on the simulation and characterization of the turbulence produced by a wind turbine model. This requires the development of a methodology to produce an adequate turbulence inflow, next to the implementation of the rotor model. The results of this process are validated using experimental data as well as design parameters of a concept turbine. After this, the performance of the flow modelling technique is assessed both in the absence of disks as well as within the wake field. Finally, the evolution of diverse turbulence features in the wake are evaluated and studied under different inflow and rotor conditions.

0.1. Objectives and methodology

The main objective of this work is *the modelling of turbulence characteristics in the wake flow of a wind turbine model in an homogeneous inflow*.

To achieve our main goal, we formulate three specific objectives:

- To implement a method of turbulence generation to reproduce the characteristics of a homogeneous turbulence field
- To assess the reproduction of turbulence characteristics in the wake of an actuator disk
- To evaluate the effects on the turbulence field of the wake due to the inclusion of rotation and non-uniform load distribution

To fulfil the goals of this work, a methodology is developed in this work to replicate: 1) the turbulence characteristics of a homogeneous wind tunnel flow and 2) the wake field arising from the introduction of porous disks representing the wind turbine. In Chapters 3 and 4, this procedure is employed to reproduce the inflow and wake characteristics measured in the experimental campaign carried out by G. Espana and S. Aubrun (Espana, 2009; Sumner *et al.*, 2013), so the comparison with wind tunnel data serves as a process of validation of the implemented methodology. The study of diverse features of the turbulence field both in the free decaying flow and in the wake is presented next to such comparison. Later, in Chapter 5, we assess the differences in the turbulent wake field obtained with two versions of the AD model. Additionally, the performance yielded by the modelled rotor is examined.

The computational platform employed is OpenFOAM® (Weller *et al.*, 1998; The OpenFOAM Foundation, 2016), an open-source code amply used in flow simulations, chosen for its availability and access to apply ad-hoc modifications to existing solvers and utilities. A synthetic turbulence field is created with an implementation of the method of Mann (1998) and used as an inflow for our computations of decaying turbulence. To replicate the transient

turbulence features in both the turbulence and the wake fields, Large-Eddy Simulations (LES) are performed. The methodology implemented follows—in part—a procedure developed on EllipSys3D (Michelsen, 1992, 1994; Sørensen, 1995) to simulate wake flows with turbulent inflows (Troldborg, 2008). It has been shown that this method provides good results to introduce ABL as well as homogeneous turbulence conditions (see Sec. 1.7) in LES computations. For that reason, the study presented in Chapters 3 and 4 is complemented by performing the same set of computations in EllipSys3D to compare its results with those obtained with OpenFOAM, besides the comparison with wind tunnel data. This election is also justified since EllipSys3D is a platform widely used and tested for wind turbine wakes simulations, so results obtained with it can be used as a reference. Both codes comprise Computational Fluid Dynamics (CFD) solvers that employ the finite-volume approach in collocated grids. However, since distinctive numerical setups are employed in each code, our approach emphasizes the reproduction of the main inflow characteristics extracted from the experiments independently in each code. In this way, the wake features obtained from each platform can be later compared. While the numerical setup in OpenFOAM has been chosen for its adequacy to this type of study, the setup in EllipSys3D is taken from previous works in wake simulations, in what can be considered a *common practice* configuration for wake computations. As such, the results obtained with EllipSys3D represent a significant reference point in our study. It is important to note that EllipSys3D simulations were perfomed in collaboration with other research group (see Sec. 2.2.4) and that we did not have access to it. Likewise, the turbulence generator employed in EllipSys3D was not accessible for its use in the LES computations performed with OpenFOAM.

With the approach presented above, the fulfilment of the objectives should permit answering the following questions:

a. How well can we reproduce the development of the main turbulence characteristics (e.g. turbulence intensity and integral lengthscale) along the wind tunnel with LES ?

b. How well can the turbulence features in the wake be predicted by the LES ?

c. How do the main turbulence characteristics in the wind tunnel change due to the presence of the wind turbine model ?

d. How does the LES modelling change along the wake compared to the undisturbed flow ?

e. What are the main differences in the wake representation when uniformly loaded and rotating AD models are used?

f. How well is the performance of the modelled Horizontal Axis Wind Turbine (HAWT) reproduced by the rotating AD implementation?

Questions a and b are answered with the results obtained in Chapter 3, c and d with those of Chapter 4 while Chapter 5 concerns the answer of e and f. It should be noted that these questions are formulated in the context of a *limited mesh resolution*, which makes it more relevant for the wind energy field since it is often desired to minimize the computational requirements while successfully reproducing the requested flow features, which in the case of the reproduction of wind tunnel measurements consists mainly of the integral lengthscale. Make note that although computations of Chapters 3 and 4 are carried out with two codes, no fundamental modifications are performed with the intent of approaching the numeric implementations from one program to the other. Furthermore, some of the procedures applied in each code remain different, such as the methods employed to introduce the synthetic turbulence field into the computational domain or the SGS model.

Original contribution

The investigation performed for this work is expected to contribute to the understanding of the turbulence behaviour in the wakes of HAWTs. More precisely, our contribution is made through the following aspects:

- The development of a methodology in OpenFOAM to model wind turbine wakes. This comprises the implementation of a method for turbulence generation, the introduction of turbulence into the computational domain and the implementation of a rotor model

- The implementation in OpenFOAM of an AD with rotation, based on the BEM theory which includes a rotational control method
- To study the reproducibility of the turbulence scales with a restricted resolution, we assess how well the turbulence characteristics are reproduced in this context by comparing our results with experimental data
- The assessment of the performance of two different LES models. The evaluation of the results from the two employed codes can be useful for other researchers when studying the evolution of turbulence features, either inside or outside the wake envelope.

Thesis overview

The work presented here is organized as follows: in Chapter 1, a brief literature review about the different components that take part in our research is presented, such as turbulence, wind turbine wakes and rotor models. This is complemented by a summary of previous work specific to our problem. In Chapter 2, we discuss the methods employed in this investigation, which concern principally the numerical modelling of the inflow turbulence, rotor and wake flow, as well as some definitions to be used in the analysis of the evolution of turbulence features. In Chapter 3, after introducing the experimental campaign and the numerical setup employed to reproduce the corresponding measurements, we assess the methodology employed for the turbulent inflow generation, as well as the characteristics of the decaying turbulence (in the absence of the rotor). Chapter 4 presents the study of the turbulence characteristics along the wakes produced when the AD model is introduced, next to an evaluation of the LES modelling. In Chapter 5 we present an implementation of a rotating AD model to compare the obtained wake characteristics with those of a uniformly loaded AD. In addition, an evaluation of the performance yielded by the rotating AD in comparison with the design parameters is provided. Next, the overall conclusions of this investigation are presented (partial conclusions are included at the end of each Chapter). Finally, Appendix I contains an example of the effects of the mesh resolution on the representation of decaying turbulence features while Appendix II

consists of some OpenFOAM libraries that indicate the main numerical methods employed in our computations.

CHAPTER 1

WIND TURBINE WAKES

In this first Chapter, a review of the literature relevant to the subject of this work is presented. To this aim, the content is divided in sections where the principal topics that comprise the research are introduced (Sections 1.2 to 1.6). Finally, a summary of previous work performed in the specific subject is included (Sec. 1.7).

1.1 Description

The wake is defined as the region behind a wind turbine where the velocity of the wind flow is lower compared to that in front of the turbine. In general terms, a wake is generated by the loss of momentum due to the force of the body on the fluid, acting in opposite direction to the flow. Conversely, the force of opposite sign applied on the body itself is know as *thrust*. The characteristics of the wake depend largely on the geometry of the body and the Reynolds number of the flow. Whereas for streamlined objects the wake is small and of the order of the boundary layer thickness, the wakes of wind turbines can extend over many turbine-diameters behind the rotor.

The wake perimeter is drawn by the *shear layer*, conceived as the cylindrical region shaped by the helical vortices arising from the tip of the blades, which represents the boundary between the slow wake flow and the outside flow (Crespo *et al.*, 1999). The wake region can be divided in two parts: the *near wake* is the zone just behind the rotor, of about a few rotor diameters in length (e.g, 1 in Vermeer *et al.*, 2003; 1-3 in Manwell *et al.*, 2002; 2-5 in Crespo *et al.*, 1999; or 1-5 in Medici, 2005). In this region, the blade airfoil design largely influences the flow dynamics as opposed to the downstream zone. This part extends until the shear layer increases its thickness (due to turbulence diffusion) and reaches the centreline of the wake (Vermeer *et al.*, 2003). Beyond this zone is the *far wake*, where mixing and diffusion continue until the turbulence generated by the turbine and velocity deficit with respect to the free stream flow have disappeared (Manwell *et al.*, 2002). Self-similarity is reached at a considerable distance in the

far wake. Using experimental results in the wake of solid disks of diameter D as example, self-similarity is reached for distances of $x/D > 30$ (Johansson and George, 2006) or $x/D = 50$ (Medici, 2005). Therefore, this type of analysis is of little interest in practical wind energy applications (Medici, 2005). The far wake research is dominated by turbulence processes and interactions of the wake with the ground, the Atmospheric Boundary Layer (ABL) and other turbines (in wind farms). On the other hand, the determination of the near wake characteristics can be influenced by the aerodynamic design, performance and physical processes of power extraction.

Due to the interaction of the turbulent inflow with the turbine blades, a highly complex flow field within the wake region is expected. As torque is produced, rotation is also induced, adding an angular momentum component to the wake flow. Furthermore, changes in the bound circulation along the blades give place to the shedding of strong tip vortices that follow helical trajectories (Ivanell, 2009). Due to different instability mechanisms, vortices break down and form small scale turbulence structures. The primary cause of wind turbine wake destabilization is mutual inductance (vortex pairing) by the tip vortices (Sarmast, 2014).

1.2 Rotor modelling

The process of momentum extraction of the flow by the rotor is the central point when modelling the wind turbine. This is described by the *momentum theory* (Betz, 1926), where a stream tube representing a control volume surrounds a non-rotating actuator disk of uniform thrust that creates a pressure discontinuity in the crossing flow. Making a series of assumptions and applying the principle of conservation of momentum, the thrust, power and the induction factor a (the fractional decrease of the velocity between the freestream velocity U_∞ and the velocity at the rotor plane) can be found. In this way, the performance of the turbine can be defined with respect to the available force and power available in the wind. Specifically, by calculating the fraction of thrust and power in the wind extracted by the rotor, represented by the thrust (C_T) and power (C_P) coefficients. The theory of Betz also establishes the theoretical limit of the power production of the turbine which is found to occur when $a = 1/3$, which

results in $C_T = 8/9$ and $C_P = 16/27$ (Manwell *et al.*, 2002). This theoretical limit of power is further reduced once factors such as wake rotation, blades surface, aerodynamic drag, etc., are taken into account. In particular, it is found that the power increases as a function of the tip-speed ratio Λ, until it reaches the Betz limit.

An analysis that comprises the modelling of blades is made by the *blade element theory*, which allows to express the forces on the blades as a function of lift (C_L) and drag (C_D) coefficients as well as the angle of attack (α_a). By considering a division of the blade into a N number of sections, the thrust and the tangential force (torque) are found for each of partition as a function of these set of parameters. The Blade Element Momentum theory (BEM) is built by combining these results with the momentum theory, so the overall performance of the turbine is found. The principles of this method are stated and used in Sec. 2.3.2. Since this model is simple and fast to run on a computer, it is highly popular and one of the most popular design schemes used by the wind industry. However, the model is limited concerning the representation of complex flow conditions around the wind turbine, especially regarding off-design conditions. These include dynamic inflow, yaw misalignment, tip loss and heavily loaded rotors (Sørensen and Shen, 2002).

Although the BEM theory permits a more descriptive solution of the blade, the solution of the flow is still based on the principle of its division in annular control volumes and the application of momentum balance and energy conservation in each section. In order to attain a computationally affordable model that nonetheless reproduces the physical characteristics required for this study, the *generalized* Actuator Disk model (Madsen, 1982; Rajagopalan and Fanucci, 1985; Sørensen and Myken, 1992) referred to as AD, is employed in this work. This model is an extension of the BEM method, as it makes use of airfoil data and conservation laws. The main difference is that the annular independence is replaced by a full set of Euler or Navier-Stokes equations to simulate the flow field. Then, unlike the classical BEM, no physical restrictions are imposed over the kinematics of the flow. The generalized AD model is widely used in wind energy research to calculate the loads, power output and wakes of wind turbines. As such, the treatment of the rotor characteristics vary in complexity, according to the goals of the compu-

tation. The AD disk is conceived as a permeable surface normal to the freestream direction on which a distribution of forces acts upon the flow.

Early works with the AD have shown good agreement between experimental data and computations in a range of settings: on axisymmetric flow conditions with constant loading, using the Euler equations (Sørensen and Myken, 1992), the Navier-Stokes equations (Madsen, 1997; Sørensen and Michelsen, 2000) or in unsteady flow using a blade-element approach (Sørensen and Kock, 1995), considering also conned rotors (Masson *et al.*, 2001; Ammara *et al.*, 2002) as well as including turbulent flow, modelled with RANS equations (Crespo *et al.*, 1985; El Kasmi and Masson, 2008) and for constant loading and turbulent flow using LES (Jiménez *et al.*, 2007, 2008). From these early works, the last group (Crespo *et al.*, 1985; El Kasmi and Masson, 2008; Jiménez *et al.*, 2007, 2008) concentrates on wake—and turbulence—modelling, whereas the rest focuses on the performance of the wind turbine through improvements of the rotor modelling.

Nevertheless, the underlying principle of the AD model leads to disadvantages in the accurate representation of the near wake flow produced by a real turbine (in particular, from the blade's tips) since the forces are distributed on the azimuthal direction over the disk, so the influence of the blades is spread over their swept area. To overcome these limitations, Sørensen and Shen (2002) developed a model in which body forces are distributed radially along each of the rotor blades. While the flow solution is determined by the Navier-Stokes equations, the influence of the rotating blades on the flow field is parametrized using tabulated airfoil data (usually corrected for three-dimensional effects). This technique is referred to as the *Actuator Line* (AL). The model is used to determine features in the near wake, such as axial interference factor and the position of tip vortices. Due to its capabilities to model the shed vortices behind the rotor, this technique is particularly useful to represent the interaction of the tip vortices along the wake flow (Ivanell, 2009; Sarmast, 2014). Other work has also demonstrated that this model is capable of representing the global flow field around the blades more accurately than with the AD model (Troldborg *et al.*, 2015), although at a heavier computational expense.

A representation of the full geometry of the rotor is made in the model by Zahle *et al.* (2007), where unlike the AD and AL models, the boundary layer that develops over the blades is also simulated. This methodology was initially applied for the modelling of downwind turbines and to the study of rotor-tower interactions, showing good agreement with experimental results (Zahle and Sørensen, 2007). This technique has been recently compared to AD and AL also by Troldborg *et al.* (2015), observing that appreciable differences in the estimation of shear levels in the wake exist in the absence of inflow turbulence. Conversely, the same work demonstrates the capabilities of the AD and AL models to simulate wind turbine wakes in real (turbulent) conditions. On the other hand, Sibuet Watters and Masson (2010) developed a concept consisting of porous surfaces that carry velocity and pressure discontinuities to model the action of lifting surfaces on the flow. The so-called *actuator surface* model is not exclusively applied to represent wind turbines as it is employed to model non-rotating wings as well. Comparisons with experimental data show that the model can reproduce accurately C_T and C_P in near wake velocity measurements. Moreover, it was found that this technique is able to reproduce the flow structure of a vorticial wake. However, in its current state, this technique has been only applied to inviscid flows.

1.3 About turbulence

Turbulence is understood in this work as the significant and irregular variation of the velocity field, both in position and time, characterized by the apparition of eddies, in a wide range of scales, in the fluid flow (Pope, 2000). One of the main features of turbulence is its ability to transport (mass and heat) and mix fluids which is largely increased compared to non-turbulent flows. This has a considerable impact on areas such as weather prediction, the mixing of pollutants (Kim and Patel, 2000) or the power production in a wind park (Nilsson, 2015). The apparition of turbulence is characterized by a scenario where the inertial forces in the flow prevail over the viscosity forces, estimated through the Reynolds number (Re), which measures their relative strength. Hence, the onset of turbulence is generally associated to the overpassing

of a limit Re. The Re value can also be used as an indication of the level of fluctuations in the flow when a characteristic lengthscale of the eddies is employed.

Turbulence is a nonlinear phenomena, where a large range of lengthscales in the flow interact with themselves. These are identified as whirling eddies in the flow, referred to as *turbulence structures*. While the largest structures carry most of the momentum transport, the smallest eddies act primarily as a dissipative source, vanishing when their size is small enough to interact with the molecular viscosity ν, dissipating the remaining kinetic energy. The largest turbulence structures are affected by the boundary conditions of the flow and thus display an anisotropic shape, but at the dissipation range, the structures are considered isotropic (Pope, 2000). Between these two ends, eddies transfer their energy to subsequently smaller eddies in a process known as the energy cascade. This process is assumed to occur in equilibrium, so the energy transfer between scales is equal to the viscous dissipation. The distribution of energy along the different turbulent scales is discernable by means of a Fourier analysis. This way, the energy spectra is computed from the turbulence velocity field to reveal the energy content along the range of wavenumbers (or frequencies) of the velocity fluctuations. Consequently, the largest energies are obtained for the smallest wavenumbers (that identify the big eddies) and on the opposite end, the minimum energies are yielded by the smallest eddies where viscous dissipation takes place. In between lays the energy transfer region called the *inertial (sub)range* which for "sufficiently large" Re numbers[1] takes the shape of a straight line with a slope of $-5/3$, as first recognized by Kolmogorov (1991).

For a given flow, the instantaneous velocity vector is referred to as u_i (using Einstein summation convention) whose components in the streamwise, vertical and spanwise directions (x, y, z) are $u_i = (u, v, w)$. In the analysis of turbulent flows, statistical quantities are used to characterize variations from the mean flow. In the case of the longitudinal component u, the

[1] Mydlarski and Warhaft (1996) and Mydlarski and Warhaft (1998) studied the development of the inertial subrange in decaying turbulence, finding that above $Re_\lambda \sim 200$ the slope of this region shows a clear slope of $-5/3$ (see also Sec. 2.4.1).

mean velocity is defined as the simple average over time:

$$\langle U \rangle = \frac{1}{T} \int_0^T u dt, \tag{1.1}$$

where T is the total time of the sample. Evidently, T should be larger than the turbulence lengthscales so that average is statistically significant. Make note that in this work, capitalized letters (U, V, W) are used to denote average magnitudes, besides the average operator "$\langle \cdot \rangle$". Alternatively, this operation can be performed over a volume to obtain an spatial average (replacing the integral by sums for discrete cases). The *fluctuation* of the flow is defined as:

$$u' = u - \langle U \rangle . \tag{1.2}$$

The second statistical moment is one of the most useful tools to obtain information about turbulence. In this way, the root-mean-square of the velocity fluctuations $\sqrt{\langle u'^2 \rangle}$ (abbreviated as u_{rms}) is used to define the streamwise Turbulence Intensity (TI):

$$\mathrm{TI} = \frac{\sqrt{\langle u'^2 \rangle}}{\langle U \rangle} = \frac{\sigma(u)}{\langle U \rangle}, \tag{1.3}$$

where σ corresponds to the standard deviation. In addition, the global turbulence intensity is calculated as

$$\mathrm{TI_g} = \frac{\sqrt{\langle u'^2 \rangle + \langle v'^2 \rangle + \langle w'^2 \rangle}}{\frac{1}{3}\left(\langle U \rangle + \langle V \rangle + \langle W \rangle\right)} = \frac{\sqrt{2k}}{\langle u_i \rangle}, \tag{1.4}$$

where k corresponds to the *turbulent kinetic energy*.

1.4 Turbulent flow equations

The dynamics of the fluid flow are dictated by the Navier-Stokes equations. Considering an incompressible Newtonian fluid of constant density and dynamic viscosity ν, the equation for

the conservation of momentum takes the well-known form:

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + f_i, \tag{1.5}$$

(see for example Bechmann, 2006) where p is the pressure and f_i represents the external forces acting over the fluid. The continuity equation is $\partial u_i/\partial x_i = 0$.

The solution to the Navier-Stokes equations is not know in analytical form but for a handful of examples where a series of simplifications can be made (Schlichting and Gersten, 2003). For the vast majority of physical problems and in particular those involving turbulent flows, models are constructed to approach the solution, followed by a numerical implementation to simulate the results.

1.5 Numerical modelling of turbulent wind flow

The main difficulty of the modelling of the wind turbine wakes resides in the representation of the stochastic and non-linear characteristics of turbulence. However, the solution of the Navier-Stokes equations for turbulent flows has to be approximated by computational means. Precisely, the turbulence phenomena can only be correctly represented by either resolving the non-linear convective term of the instantaneous velocities or approximating its effect (i.e. modelling) in the fluid flow.

When the velocity field is completely solved, without attempting to model any quantity, a Direct Numerical Simulation (DNS) is performed. In a DNS of turbulence, the Navier-Stokes equations are solved numerically with the appropriate initial and boundary conditions. The entire range of spatial and time scales is resolved. In other words, every eddy, from the largest (integral scale) to the smallest (the so-called Kolmogorov scale), is computed. In theory, DNS is the best approach because it makes the fewest simplifications and provides a complete description of the flow. However, the required computing power increases rapidly with Reynolds number. As Pope (2000) indicates, increasing the Reynolds number by a factor of 10 causes

the time needed for a simulation to increase by a factor of 1000. Therefore, even with present day computers, DNS is limited to simple flows at moderate Reynolds numbers.

In practical applications, simplifications need to be made and models are devised to approximate a solution. Within the wind energy community, numerical simulations are mainly performed using two different approaches: linearized models and Computational Fluid Dynamics (CFD) codes. In the first case, the analysis of the flow is performed using linear approximations to the Navier-Stokes equations (Palma *et al.*, 2008), sometimes together with the potential flow hypothesis, where the flow is assumed to be irrotational, which is clearly not the case of highly turbulent flows where the vorticity effects are important. Conversely, CFD codes attempt to solve the Navier-Stokes equations, using a proper model of turbulence to, for example, calculate the effect of the velocity fluctuations on the main flow or to filter some turbulence scales that are then resolved while modelling the effect of the remaining scales. Of these two assumptions, the former is the principle of the Reynolds-Averaged Navier-Stokes (RANS) approach, which makes use of the *Reynolds decomposition* to divide the velocity field into the time-average velocity $\langle U \rangle$ and the velocity fluctuation u' around the mean (as in eq. 1.2). Once this principle is applied to the Navier-Stokes equations, the turbulent motions are not explicitly calculated, though their effect on the mean flow is quantified through the *Reynolds stresses* $\langle u_i u_j \rangle$. Hence, a model is required to relate $\langle u_i u_j \rangle$ to mean flow quantities and *close* the set of equations. Amongst the varied techniques, there are two primary approaches for estimating the Reynolds stresses (Wilcox, 1994):

- Turbulent viscosity models, where the Boussinesq hypothesis is applied to relate the Reynolds stresses with a turbulent viscosity ν_t (in an analogous way to the construction of ν_{SGS}, shown in Sec. 2.1). In turn, ν_t can be obtained from algebraic relations (e.g., the mixing-length model) or from turbulence quantities such as the turbulent kinetic energy and its dissipation rate, for which modelled transport equations are solved.

- Reynolds stress models, where a modelled transport equation for $\overline{u_i u_j}$ is solved.

In its original conception, the RANS model provides the steady-state solution of a flow field. If transient solutions are needed, the Unsteady-RANS (URANS) model can be used. As computational power increases, RANS models have become the one of the standard methods for calculations in the wind industry. A fair amount or research in wakes and wind turbine aerodynamics has been done using RANS models (see reviews by Crespo *et al.*, 1999; Vermeer *et al.*, 2003; Réthoré, 2009). In the present study, however, it is desired to investigate turbulence features in the inflow and wake that cannot be reproduced if transient turbulence structures are not resolved (such as their characteristic eddy lengthscale or the redistribution of energy along the fluctuation scales).

To address the problem of simulating the turbulence and other non-linear phenomena in wind turbine wakes, the Large-Eddy Simulation (LES) model is used. In contrast with RANS, LES solves the velocity field above some eddy scale in the flow, whereas the effect of the smaller scales on the main flow is only modelled. The selection of LES allows to treat the problem of the inherent unsteady features in the flow of wind turbine wakes, that should be simulated to provide a description of wake turbulence features. Evidently, the increase in detail for the flow description is accompanied by an escalation in computational expense, which is the main reason as to why RANS (or linear models) still dominate the wind industry. As diverse authors have pointed out (Jiménez *et al.*, 2007; Calaf *et al.*, 2010), LES has the potential to provide with more accurate solutions of the flow if the scales of the resolved turbulence are small enough to properly represent the effects of eddies and flow fluctuations. This is the model employed in this work, described in Sec. 2.1.

Another alternative of modelling turbulent flows in CFD is to combine two models to take advantage of the computational economy of the one technique in combination with the accuracy and detailed description of another one to solve a particular setting. This is the *hybrid model* approach and in the wind energy field, the use of RANS and LES is a typical combination (Bechmann, 2006). When modelling wind turbines in the ABL flow, RANS is used close to the wall to model the effect of the small fluctuations that would otherwise require an increasing refinement in LES or the use of wall models (Kim and Patel, 2000). In this way, LES can be

used in higher regions where the interaction of the flow with the turbine requires the concentration of the computational expense. A particular issue in hybrid technique is the treatment of the interfaces between models. This is due to the absence of transient turbulence structures arising from the RANS region towards that covered by the LES. For interfaces at fixed locations, different approaches are used to solve this issue, often making use of forcing terms in the momentum equation to incite turbulence fluctuations (Davidson and Dahlström, 2005). Another alternative is to establish a transition between regions based on dynamic characteristics of the flow and the grid employed. This is the principle of the Detached-Eddy Simulation (DES) model (Spalart *et al.*, 1997), where RANS are used close to the boundary and the switch to LES is made according to the definition of a lengthscale, which is a function of the local grid refinement and the eddy estimated size. This model was originally designed to resolve flows around airfoils (to improve the estimations in regions of detached flow) so different adjustments need to be made for its use in ABL flows (Bautista, 2015). Different issues discovered in the original version have been addressed by subsequent modifications to the base model formulation: the Delayed DES (DDES) and the Improved DDES (IDDES).

1.6 Approaches in the generation of inflow turbulence

A fundamental element in the study of turbulence in wakes is the representation of the physical characteristics of the inflow. In RANS computations, this aspect does not represent an issue of the same complexity as in LES, since it is common to adjust the energy level of turbulence at the inlet, based on the desired level of turbulence intensity (e.g. Sumner *et al.*, 2013). In LES, this level should arise from the magnitude of the velocity fluctuations and principally from the resolved ones. Therefore, a velocity field representing the kinetic energy level as well and (coherent) turbulence structures of the desired values needs to be produced. In ABL flow, the inlet turbulence field should also be suitable to the particular conditions of the terrain or atmospheric stability. In that case, an appropriate alternative (albeit a computationally-expensive one) is to make use of a *precursor* simulation where turbulence "adapts" (from an initial field) or arises from the topographic and (potentially) atmospheric conditions. However,

in unbounded flows as in the present cases of study, a precursor simulation cannot be used and turbulence should be created by purely *synthetic* methods. Different techniques have been devised for this purpose. Without making an exhaustive review, the most relevant methods within the wind energy research are mentioned below.

The simplest method consists in the generation of random velocity values (white noise), which results in a uniform distribution of energies along the different frequencies, contrary to the real turbulence. Moreover, in the absence of other perturbations, this type of fluctuations will vanish when introduced in the CFD solver due to the lack of spatial or temporal coherence (Tabor and Baba-Ahmadi, 2010). To reproduce the real conditions of turbulence (such as two-point statistics), methods based on stochastic techniques are employed. One of the commonly used methods is the one created by Veers (1988), which in an expanded version by Kelley (1992) based on observations of ABL spectra, has been implemented in the *Turbsim* code (National Renewable Energy Laboratory, USA). Lee *et al.* (1992) developed a model with a prescribed spectra and overlapping of random modes whose amplitude is derived from the spectral tensor, which is used to generate spatially evolving isotropic turbulence (using the Taylor hypothesis of frozen turbulence). Another technique is the digital filter used by Klein *et al.* (2003), where a random velocity field is manipulated to reproduced prescribed one- and second-order statistics, with a locally defined autocorrelation function obtained from the turbulence scales. This principle is also applied by Lund *et al.* (1998) and Veloudis *et al.* (2007). The method of Mann (1998) employs a modelled spectral tensor to create a turbulence field employing Fourier methods. Notably, this model can simulate the vertical anisotropy of ABL turbulence by means of rapid distortion theory, to simulate the effect of shear on the spectral tensor. This model has provided good results simulating homogeneous (e.g. Gilling and Sørensen, 2011) and ABL turbulence in neutral (e.g. Nilsson *et al.*, 2015) and non-neutral (Chougule *et al.*, 2015) conditions. Due to this and its wide application in wind energy research, an implementation of this model has been used for this study. This is described in Sec. 2.5.1.

A slightly different approach in the generation and implementation of stochastic generators has been recently introduced by Muller (2014). There, an inverse wavelet transform capable of

generating anisotropic flows is employed, with the particularity that a motion compensation is applied to simulate the local variations in the convection velocity[2]. The synthetic field is then applied not only at the inlet, but also at the sides of the domain allowing to represent turbulence scales larger than the domain dimensions. Muller (2014) employed this methodology in the reproduction of wake meandering measurements, obtaining a good comparison with LES computations[3].

1.7 Previous work

A brief summary of earlier work performed within the field of study is presented below. This also serves to provide the background for the motivation of this work.

Studies of turbulence in the wake of wind turbines are often made with the aim at analyzing the influence of the flow and rotor models in the reproduction of the characteristics of the wakes. Investigations with various rotor models in the ABL have been made either with the goal of improving the production efficiency of a cluster of turbines (e.g. Crespo *et al.*, 1999; Calaf *et al.*, 2010; Churchfield *et al.*, 2012; Nilsson *et al.*, 2015) or aim at comparing the characteristics of wakes with respect to measurements of real or downscaled turbines (Troldborg, 2008; Ivanell, 2009; Chamorro and Porté-Agel, 2009; Porté-Agel *et al.*, 2011). It has been shown that uniformly loaded disks (Aubrun *et al.*, 2013; Jiménez *et al.*, 2008; Porté-Agel *et al.*, 2011), such as the simplest conception of the AD, can be used to simulate the main flow characteristics of the far region of rotor wakes. Conversely, a noticeable improvement in field representation of the near wake can be obtained when the rotor model comprises an airfoil description of the blades as well and rotation (Porté-Agel *et al.*, 2011). More sophisticated models such as the AL enhance the reproduction of the wake field near the rotor, albeit at a higher computational demand (Troldborg *et al.*, 2015).

[2]In other models, when Taylor hypothesis is applied, it is assumed that velocity fluctuations are transported at the same convection velocity when introduced in the computational domain, which is clearly not the case at different heights in the ABL.

[3]Interestingly, Muller (2014) mentions a few turbulence generation methods employed in the creation of special effects in theatrical films, where a method based on the wavelet principle has been also applied (Kim *et al.*, 2008).

When studying the turbulence characteristics in the wake, a simpler setting of the inflow contributes to facilitate the analysis of the turbulence field arising purely from the interaction of the flow with the rotor. For that reason, investigations are performed in decaying isotropic turbulence produced in a wind tunnel, which greatly reduces the physical modelling in the absence of turbulence production, otherwise present in the ABL. This setup has been intensively studied as the most feasible approach to the Homogeneous Isotropic Turbulence (HIT). Moreover, it is an often used benchmark to test LES implementations (Goodfriend *et al.*, 2013; Wachtor *et al.*, 2013). Amongst the available measurements sets, perhaps the best known is the one produced by Comte-Bellot and Corrsin (1966) and Comte-Bellot and Corrsin (1971), which has set a standard for the subsequent investigations on this topic. In those works, reports of turbulence kinetic energy and its dissipation, correlation functions and energy spectra are given at various stations downstream of the turbulence grid. Later work (e.g. Mydlarski and Warhaft, 1996, 1998) has extended these investigations to a wide range of Re flows ($30 \leq \mathrm{Re}_\lambda \leq 731$) with the use of active grids, verifying the applicability of assumptions made regarding the behaviour of HIT while also describing the distinguishing features between weak ($\mathrm{Re}_\lambda \sim 50$) and strong turbulence ($\mathrm{Re}_\lambda \geq 200$). Other works (e.g. Kang *et al.*, 2003) have combined experiments with reproductions of their measured quantities via LES computations, also comparing the performance of different Sub-Grid Scale (SGS) closures.

Following the principle of an AD, an experimental investigation of the wakes produced by a porous disk has been performed by Aubrun *et al.* (2007) and Espana *et al.* (2011), where the disk is made of metallic meshes representing different solidities (defined as the ratio of the surface obstructed by the grid with respect to the disk frontal area). Moreover, experimental studies have been performed of wakes produced by a decaying isotropic turbulence inflow (Espana *et al.*, 2012; Thacker *et al.*, 2010). Recent work by Sumner *et al.* (2013) has focused on the reproduction of the measurements in the wake of the porous disks as well as the properties of the employed inflow using RANS models. Although good results are obtained, the experimental study represents an opportunity to perform comparisons with numerical simulations

that allow a greater detail in the reproduction of the turbulence characteristics. Therefore, the reproduction of these experiments employing LES computations seems appealing.

To reproduce the inflow properties measured experimentally, it is necessary first to model the flow of decaying turbulence produced in a wind tunnel. In this regard, works like those of Tabor and Baba-Ahmadi (2010), Lund *et al.* (1998) and Klein *et al.* (2003) have been dedicated to investigate different methodologies to produce adequate inlet conditions. Amongst the different techniques, the model developed by Mann (Mann, 1998; Peña *et al.*, 2013) to create a synthetic turbulence field has been continuously used in recent years to create inflow conditions for wake simulations in ABL (e.g. Troldborg, 2008; Ivanell, 2009; Keck *et al.*, 2014; Nilsson, 2015) as well as in HIT (e.g. Bechmann, 2006; Gilling and Sørensen, 2011; Troldborg *et al.*, 2015). In these works, it has been proved that the model of Mann is capable of producing turbulence fields with the same second order statistics than real turbulence (see also Gilling, 2009). This algorithm permits to create synthetic turbulence fields by prescribing two parameters, the turbulence lengthscale and (albeit indirectly) the turbulence intensity for HIT. An anisotropy factor is also used when the algorithm is applied to create boundary layer fields. The transition and evolution of turbulence characteristics when synthetic fields are introduced in LES domains, specially of integral lengthscales, has been previously studied by Gilling and Sørensen (2011) for HIT and recently by Keck *et al.* (2014) and Nilsen *et al.* (2014) (using the turbulence generation method of Klein *et al.* 2003) in sheared flows. Likewise, the impact of using different SGS models in LES computations of turbulent wakes has been assessed by Sarlak *et al.* (2015b), where it was found that while different subgrid viscosities are computed in each model, differences in the AL-modelled rotor performance are negligible. Some works have been also dedicated to comparisons between OpenFOAM and EllipSys3D in the context of wake turbulence. In particular, Sarlak *et al.* (2014) shows results that point towards a greater preservation of the turbulence structures (and an earlier breakup of the wake) in OpenFOAM compared to EllipSys3D.

CHAPTER 2

APPLIED TECHNIQUES FOR THE MODELLING OF WIND TURBINE WAKES

The study of turbulence in the wakes of HAWTs is made by performing numerical simulations to represent all the significant features that take part in the interaction of the wind and rotor. As in most physical models, the impracticality of resolving the Navier-Stokes equations for all scales of fluctuation as well as the full rotor geometry, including the boundary layer developed around the blades, constrains the calculations to the adoption of models. Certainly, the level of sophistication in each model to represent the ensuing physical phenomena depends largely on the elements under study and the availability of computational resources. Therefore, according to the objectives of our investigation, in this Chapter we introduce the different techniques used in the solution of the wind flow, the representation of the rotor, the modelling of turbulence and the platform where these methods are implemented: OpenFOAM. Lastly, note that since the AD and turbulence models constitute a new implementation on this computational platform, a validation procedure accompanies its presentation.

The first element to be introduced here concerns the modelling of the turbulent flow. In investigations that imply to carry out computations of large domains where wind turbines are immersed in the ABL, RANS models offer a cost-effective alternative by simplifying the amount of resources needed to obtain estimations such as power production and turbine setting optimization. Conversely, when the studies are addressed primarily towards turbulence phenomena, the mean quantities yielded by RANS impose a serious limit with respect to the features that can be studied when transient phenomena are wished to be reproduced. Indeed, the use of sonic anemometers or Large Detection and Ranging (LIDAR) for wind turbines in the ABL, as well as Hot-Wire Anemometry (HWA) or Particle Image Velocimetry (PIV) in wind-tunnel experiments, affirm the interest of counting with measurements that can be used to obtain information (second and higher-order statistics) about the features of turbulence beyond the offerings of RANS models. Hence, turbulence models that permit the portraying of these characteristics are desired.

2.1 LES modelling

Large-Eddy Simulations represent the compromise between the importance of resolving the velocity fluctuations that exert the largest influence in the flow dynamics while recognizing that modelling is still needed, due to the prohibitiveness of resolving a vast range of turbulence scales. Thus, in the LES approach, the large eddies (the energy-containing flow structures) are resolved, whereas the effects of the smaller eddies are only modelled. The separation of scales is achieved through a spatial filtering process which in general terms, is obtained by a convolution of the instantaneous field with a certain filter type. Although different *explicit* filter functions exist (Gaussian, box, cutoff for spectral space, etc., see Sagaut (2006) for a review), the most commonly used in wind energy applications is the *implicit* filtering, where the filter width is associated to the grid resolution. The filtering process (indicated by $\overline{\cdot}$) applied to the Navier-Stokes eqs. (1.5) yields the LES momentum equation (Leonard, 1974):

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) + \frac{1}{\rho}\tau_{ij}^{\text{SGS}}\right] + f_i \tag{2.1}$$

where $\tau_{ij}^{\text{SGS}} \equiv -\rho(\overline{u_i u_j} - \overline{u}_i\overline{u}_j)$ is the SGS stress tensor. The term f_i represents a source term acting as an external force[1] (Bechmann, 2006). Likewise, the filtered continuity equation is $\partial\overline{u}_i/\partial x_i = 0$. The term τ_{ij}^{SGS} needs to determined, so considering the decomposition of the velocity field into the filtered (or resolved) component $\overline{u}_i$ and a residual (or subgrid scale, SGS) component u_{SGS},

$$u_i = \overline{u}_i + u_{\text{SGS}}, \tag{2.2}$$

then the nonlinear term $\overline{u_i u_j}$ in τ_{ij}^{SGS} can in turn be decomposed into a combination of these terms. Diverse decompositions have been suggested for the SGS stress tensor, such the one by Leonard (1974) and the more general decomposition of Germano (1986). However, when a simulation is performed, the information of the SGS scales is lacking and additional information is needed, so assumptions need to be made. Precisely, the evaluation of τ_{ij}^{SGS} comprises a closure problem analogous to that presented in RANS. In practical situations, additional in-

[1] In fact, $f_i = F_i/\rho$ is a body force, a force per unit volume, through which the force measure in Newtons F_i is introduced in the computation.

formation regarding the resolved scales as well as new variables created from diverse flow properties are used to solve it. The choice of techniques used to model this term constitutes the SGS model.

One of the alternatives, of analogous use in RANS modelling, is to assume the notion from Boussinesq (1897), to relate the SGS stresses to a product of the strains of the flow and an eddy viscosity ν_{SGS} that accounts for the effects of the subgrid motions, in an analogy to the molecular mechanisms of viscous diffusion. Hence, the SGS stress tensor is written as (Pope, 2000),

$$\tau_{ij}^{\text{SGS}} = 2\rho\nu_{\text{SGS}}\overline{S}_{ij} + \frac{1}{3}\tau_{kk}^{\text{SGS}}\delta_{ij}\,, \tag{2.3}$$

where

$$\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial\,\overline{u}_i}{\partial x_j} + \frac{\partial\,\overline{u}_j}{\partial x_i}\right) \tag{2.4}$$

is the filtered rate-of-strain tensor.

Models using the above assumptions are known as turbulent viscosity models. Most of them evaluate the turbulent viscosity as $\nu_{\text{SGS}} \sim l \times u$ and they essentially differentiate from each other in the choice of velocity scale u since the length scale l is linked to the filter size (Bailly and Comte-Bellot, 2003). Notably, the latter premise should be valid as long as the filter is within the inertial subrange (Pope, 2000). Under this principle, the LES momentum equation becomes (Bechmann, 2006),

$$\frac{\partial\,\overline{u}_i}{\partial t} + \frac{\partial\,\overline{u}_i\,\overline{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial\,\overline{p_m}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[(\nu + \nu_{\text{SGS}})\left(\frac{\partial\,\overline{u}_i}{\partial x_j} + \frac{\partial\,\overline{u}_j}{\partial x_i}\right)\right] + f_i, \tag{2.5}$$

where p_m denotes the modified pressure, which includes the isotropic part of the SGS stress tensor $\overline{p}_m = \overline{p} - \tau_{kk}^{\text{SGS}}\delta_{ij}/3$ into the static pressure.

The main purpose of the SGS model is then to dissipate turbulent kinetic energy, since this is precisely exercised by the smallest fluctuations whose effect is only modelled. Turbulent viscosity models, in particular, only model the forward cascade process (i.e. the transfer of energy from large to small scales) so are purely dissipative. The opposed effect, the backward

energy transfer, it is very week in comparison and thence rarely modelled in wind energy applications. In fully developed isotropic turbulence, it is proportional to k^4 (Sagaut, 2006).

In the spectral space, where turbulent motions of lengthscale ℓ correspond to wavenumber $\kappa = 2\pi/\ell$, the separation between the modelled and resolved fluctuations is set by the cutoff wavenumber $\kappa_c \equiv \pi/\Delta$. This has to be set within the inertial subrange to effectively separate the two scales. Since the idea of scale separation between energy containing and dissipation ranges is based in the Kolmogorov hypothesis, the statement over the existence of an inertial subrange that divides the two regions assumes high Reynolds numbers (Pope, 2000). Nevertheless, LES models are often used despite this number not been necessarily high[2].

According to the classification made by Sagaut (2006), three types of viscosity models can be distinguished according to the hypothesis made to compute ν_{SGS}, those based on: *a)* the resolved scales, *b)* the energy at cutoff (where the energy is obtained from the highest resolved frequency) and *c)* the subgrid scales. Unlike the first two, the third type does not require to make assumptions over the characteristics of the resolved scales and in principle it should provide a better description of the SGS motions. However, the latter techniques are comparatively more computationally demanding. In this work, a model belonging to the first category is employed (Smagorinsky) for our OpenFOAM computations. Yet, the SGS model used in EllipSys3D in Chapters 3 and 4 (Mixed scale model) is a combination of (*a*) and (*b*). A description of only these two models is made in the present work. A description of a collection of SGS models can be found in the work Sagaut (2006).

2.1.1 The Smagorinsky model

The simplest and arguably one of the most widely used of the subgrid viscosity models was proposed by Smagorinsky (1963), which assumes a constant equilibrium between the production of k, its transfer rate through the cutoff and and its dissipation, so there is no accumulation of energy at any frequency and the shape of the energy spectrum remains constant. The subgrid

[2] According to Celik *et al.* (2005), for high Reynolds number $\text{Re}_\lambda \simeq 155$.

viscosity is expressed as:

$$\nu_{\text{SGS}} = (C_s\Delta)^2 \sqrt{2\overline{S}_{ij}\overline{S}_{ij}} \ , \tag{2.6}$$

where C_s is the Smagorinsky constant, Δ is the filter width and their product defines the Smagorinsky lengthscale $l_s = C_s\Delta$. Although C_s is in theory also calculated from the model assumptions, its value is usually calibrated for the intended purpose, in order to regulate the transfer of energy to the residual motions (or dynamically adjusted in a variety of forms). The value used in this work is $C_s = 0.168$ which comes from adjustments made to reproduce decaying-HIT (Muller, 2014). Likewise, the filter width is calculated from the local cell length as $\Delta = (\Delta_x\Delta_y\Delta_z)^{1/3}$.

Disadvantages associated to the Smagorinsky model occur for the most part in bounded flows, e.g. overdissipation close to walls (Porté-Agel *et al.*, 2000; Pope, 2000). In such cases, C_s is often lowered to reduce the energy transfer to subgrid scales (alternatively, a damping function is also used). As only homogeneous flows are used throughout this work, this particular problem should not be a concern.

2.1.2 Ta Phuoc mixed-scale SGS model

For the LES computations performed with EllipSys3D, the mixed-scale model developed by Ta Phuoc (Sagaut, 2006) is used. In this technique, the SGS viscosity is obtained from information related to the resolved as well as the unresolved scales, by making an extrapolation of the former into the subgrid range. The subgrid viscosity is defined as,

$$\nu_{\text{SGS}} = C_m \left| \overline{S}_{ij} \right|^{\alpha_m} \left(\overline{k}_c{}^2\right)^{\frac{1-\alpha_m}{2}} \Delta^{1+\alpha_m}, \tag{2.7}$$

where $C_m = 0.01$ and $\alpha_m = 0.5$ are model constants and k_c is the turbulent kinetic energy, evaluated as,

$$\overline{k}_c{}^2 = \frac{1}{2}\left(\overline{u}_i\right)^t \left(\overline{u}_i\right)^t, \tag{2.8}$$

where the test field

$$(\overline{u}_i)^t = \overline{u} - \widetilde{u}, \tag{2.9}$$

defines the highest frequency part of the resolved velocity field. This is obtained using a second–test—filter, defined as $\widetilde{\Delta} = 2\Delta$ where Δ is also defined as $\Delta = (\Delta_x \Delta_y \Delta_z)^{1/3}$.

Sagaut (2006) interprets this model in two ways: firstly, as a model based on the subgrid scales, through the scale similarity hypothesis[3] of Bardina *et al.* (1980), thence $k_c \simeq k_{\text{SGS}}$. Secondly, the model is alternatively seen as based on the energy at cutoff.

Results from computations using the two above describe SGS models are compared in Chapters 3 and 4. As mentioned in Chapter 1, the use of different techniques obeys to the employment of what we consider the *best practises* in each code to obtained the desired results. First, the Smagorinsky model is used in OpenFOAM as it is perhaps the simplest and most widely used SGS model. Consequently, its principal deficiencies are reasonably well identified. In this particular, we make note that the main flaws of this model such as overdissipation of k in the presence of mean shear (Porté-Agel *et al.*, 2000) or departures from the similarity theory (Mason and Thomson, 1992; Chow *et al.*, 2005), are related to its behaviour near the wall and therefore absent from the problems treated here. For these reason, the Smagorinsky model seems a safe choice for OpenFOAM. On the other hand, the Ta Phuoc model is one of the most commonly used SGS models in EllipSys3D when LES computations are used (e.g. Ivanell *et al.*, 2010; Troldborg *et al.*, 2010; Machefaux *et al.*, 2013; Eriksson *et al.*, 2014; Keck *et al.*, 2014; Sarmast *et al.*, 2014; Troldborg *et al.*, 2014; Ivanell *et al.*, 2015; Nilsson *et al.*, 2015; Sørensen *et al.*, 2015). Unlike OpenFOAM, EllipSys3D is a platform exclusively developed for investigations on rotor interaction with turbulent wind flows. It is remarked that the objective in computations of Chapters 3 and 4 is first to observe how the results from OpenFOAM compare to the experimental results and the analytic predictions while using the results of EllipSys3D as a reference. Consequently, it is not a primary goal to perform a direct comparison of the performance between the two codes.

[3] Also from Sagaut (2006), it is assumed that the statistical structure of tensors constructed on the basis of the subgrid scales is similar to that of those evaluated on the basis of the smallest resolved scales.

When using different SGS models, computations can potencially yield different results in the development of the wake. As both SGS models are viscosity based, a difference in the computation of ν_{SGS} can have a clear effect in the solution; for instance, in the wake development. Precisely, it may be presupposed that the shear layer around the AD will break at different downstream positions for distinct values of ν_{SGS}. This would have a general effect over the wake velocity deficit and its recovery. As also mentioned in Chapter 1, these assumptions are supported by the work of Sarlak *et al.* (2015a) and Sarlak *et al.* (2015b) (albeit using the AL technique to model the rotor).

2.2 Numerical methods

Different approaches can be followed to compute the solution of the LES equations, each in turn subjected to procedures where diverse techniques can be applied. To better explain this, we distinguish between four elements involved in the numerical calculation. Firstly, the discretization of the LES equations can be made based on three techniques: finite-elements, finite-differences and finite-volume methods. From this list, the last technique is favoured in various CFD applications for its convenience in problems consisting of complex geometries[4]. Secondly, the resulting algebraic, non-linear equations might require an iterative solution technique where a "guessed" solution is used in the process of linearizing the equation and later, improving the solution at every step until convergence is achieved. For this part, the algorithms SIMPLE or PISO are used. Thirdly, these algorithms require methods of solution for the matrices implicated in the linearizing process, for which diverse strategies can be applied. Lastly, the finite-volume method specifically requires the estimation of certain quantities at the position of the faces of the grid cells, from those available at the cell centres. These are calculated by an interpolation procedure, where a collection of methods exists to choose from. Complete details about the numerical methods mentioned here can be found in Ferziger and Perić (2002) and Versteeg and (2007).

[4]This advantage is shared with the finite-element method, although with the drawback of using (generally) unstructured grids which reduces the efficiency of solution methods (Ferziger and Perić, 2002).

2.2.1 Numerical platform: OpenFOAM

The numerical procedure outlined above, based on the finite-volume technique, is executed by the Open-source Field Operation and Manipulation, or *OpenFOAM*®, the computational platform used in this work (Weller *et al.*, 1998). Only a general description of this code is provided, complete details about the platform as well as the source code and documentation are provided by The OpenFOAM Foundation[5], that distributes the code under a GNU general public license. Other comments about the functioning of OpenFOAM are also given by Churchfield *et al.* (2010) and Bautista (2015). The simulations of this work were performed using the version 2.1.0, except for the computations shown in Sec. 2.3.1.1 which were partly carried out using version 1.6 .

OpenFOAM is an open-source numerical platform that employs an unstructured, collocated, finite-volume approach. Instead of being solely a CFD solver, OpenFOAM is rather a versatile computational framework where a large collection of C++ libraries can be used to create an ad-hoc solver and boundary conditions. The executables created from the libraries are known as *applications*. According to their function, the applications are classified in two types: *solvers* and *utilities*. The former are used to solve a variety of problems, principally CFD, while data manipulation is performed with the later. A series of applications are available and ready to use in the standard distribution of OpenFOAM. Each of these applications can be modified to better suit the needs of the problem in question. The incompressible solvers, boundary conditions, turbulence model, etc., used in our simulations are based on versions already implemented in OpenFOAM.

2.2.2 The finite-volume method

Being a finite-volume code, OpenFOAM divides the domain into discrete control volumes (or cells) around the nodes, with boundaries (i.e. faces) located midway between contiguous centres. The technique is based on evaluating different quantities through integration over

[5] http://www.openfoam.org (last visited on Nov 9th, 2015).

the control volumes. Furthermore, the divergence theorem is used to calculate these as surface instead of volume integrals. In turn, as detailed by Ferziger and Perić (2002), the exact solution of the surface integrals requires a prior knowledge of the value of the quantity over the surface, which is only known at the centres. For this (following the description of Ferziger and Perić), the so-called midpoint rule is used to approximate the integral value as the product of the integrand f_e at the cell face centre (assuming equal to the mean surface value) and the face area. Thus, for the location e and the surface S_e,

$$F_e = \int_{S_e} f \mathrm{d}S = \langle f_e \rangle \, S_e \approx f_e S_e. \tag{2.10}$$

With the above, it is now only needed to estimate the value of f_e at the cell face. For such calculation, various methods of interpolation can be employed. The assumption of eq. (2.10) entails a critical consequence: the midpoint approximation carries an intrinsic second-order accuracy. This sets a limit to higher-order interpolation methods that are used to calculate centre-to-face values, as their accuracy is restricted by the above assumption[6]. Analogous to the previous expression, volume integrals can be replaced by the product of the mean value of the integrand q and the volume V, but the former can be further approximated by the value of the quantity at the cell centre q_p, this is,

$$Q_p = \int_V q \mathrm{d}V = \langle q \rangle \, V \approx q_p \Delta V. \tag{2.11}$$

But unlike eq. (2.10), the evaluation of this expression is made at the cell centres, precluding the need of interpolation. The approximation is exact if q is constant or varies linearly within the cell, else, the error is second-order.

[6]Ferziger and Perić (2002) discuss the utilization of higher-order approximations of the surface integrals, for which the evaluation of the flux in more than one location is needed (e.g. the fourth-order Simpson's rule). However, to the knowledge of this author, this is not implemented within the standard distribution of OpenFOAM.

2.2.3 Discretization schemes

When selecting adequate interpolation methods for the cell face values, the linear interpolation, also called central-difference (second-order accurate, where *n*th-order is defined in terms of the truncation error), between the two nearest nodes comes as a straightforward choice. However, the demands on the grid refinement are comparatively higher with respect to other schemes, which in turn depend on the relative strength of the convection and viscosity (including ν_{SGS}) in the flow. This is commonly characterized by the Péclet number Pe, which represents the ratio of the convective mass flux per unit area F_e and the diffusion conductance at cell faces D_e,

$$\mathrm{Pe} = \frac{F_e}{D_e} = \frac{\rho u}{\Gamma/\Delta x} = \frac{u}{\nu_{eff}/\Delta x}, \tag{2.12}$$

where Γ is the diffusion coefficient, which for incompressible cases is equivalent to ν or rather $\nu_{eff} = \nu + \nu_{\text{SGS}}$ in LES. Versteeg and (2007) as well as Ferziger and Perić (2002) point out that the linear scheme can be stable and accurate only if $\mathrm{Pe} < 2$, which results in a very high demand of refinement in the grid[7]. When the cell size does not comply with this requirements, an oscillatory behaviour around the real solution may appear in collocated grids[8], due to the pressure-velocity decoupling. To correct this undesired behaviour, Rhie and Chow (1983) introduced a technique that modifies the calculation of the pressure at the cell faces, which is also implemented[9] in OpenFOAM (Churchfield *et al.*, 2010). However, as shown by Réthoré (2009), the presence of a momentum source, such as in the technique used to model the AD, can still produce a solution where wiggles appear in spite of the application of the Rhie-Chow correction (later illustrated in Figs. 2.2 and 2.5). In such case, a spatial smearing of the momentum source may be employed to alleviate the problem, as shown in Sec. 2.3.1. Still, the use of other interpolation schemes can contribute to relieve the apparition of oscillations.

[7] As an example, take $u = 1$ m/s and assuming $\nu_{eff} \sim 1 \times 10^{-5}$, the condition $\mathrm{Pe} = 2$ is fulfilled if $\Delta x = 2 \times 10^{-5}$ m, which is indeed too small considering that rotor radii of HAWT are in the range of tens of metres. This number is still small for the domain size presented in Chapters 3 and 4 where disk radii are equal to 5 cm.

[8] A detailed description about the origin of this feature in collocated grid solvers, such as OpenFOAM, can be found in the work of Réthoré (2009).

[9] Although the technique is not explicitly implemented in OpenFOAM, a correction is applied which is equivalent in its effect to the original Rhie-Chow correction (Kärrholm, 2008).

Unlike the linear interpolation, the upwind scheme takes into consideration the direction of the flow by setting the value at the cell face equal to that of the upstream node. It is underlined by Versteeg and (2007) that although oscillations do not occur in the solutions (do to its unconditional boundedness), when the flow is not aligned with the grid lines *false diffusion* (i.e. numerical diffusion) arises. Thus, rapid variations in the variables are smeared if the grid is not refined to increase accuracy (which is only first-order), suppressing the possible advantages. The Quadratic Upstream Interpolation for Convective Kinetics (QUICK, Leonard, 1979) improves the approximation at the cell faces by making use of quadratic profiles between the cell centres in question and the upstream node. Although this scheme is third-order accurate, it is limited to second-order under the midpoint approximation. The numerical diffusion is reduced and solutions on coarse grids are often largely more accurate than those using upwind or central/upwind schemes (Versteeg and , 2007). However, the method is only conditionally stable and small under/over-shoots in the solutions might appear.

Alternatively, a hybrid scheme can be used, where two models are combined depending on the local conditions. A common approach is to use the central scheme for small Pe numbers while the upwind scheme is used otherwise. In OpenFOAM, in particular, the scheme *filteredLinear* consists of a dynamic blend of these schemes where, depending on the velocity flux and the magnitude of the velocity gradients at the cell faces, an amount of up to 20% upwind is used in combination with the linear interpolation. In this way, the upwind part is employed only in regions of steep velocity gradients while the flow maintains second-order accuracy elsewhere.

Although the discussion about the interpolation schemes is intended to be made in general terms (this is, for any quantity that may require interpolation), the hybrid scheme description appeals to the velocity flux which reveals the mayor criterion when choosing interpolation schemes. This is, when interpolations cell centre/face are needed, most of the terms in the LES equations can be interpolated using the linear scheme without compromising the outcome of the calculations, unlike the case of the velocity flux as the stability and accuracy of the computation depend largely on the interpolation scheme applied in the evaluation of this term. In our simulations, the interpolation required for the velocity flux calculation utilizes the QUICK

scheme in computations of Chapter 5 while the filteredLinear is used in Chapters 3 and 4. Essentially all the rest of the interpolation schemes are set to linear except for the time advancement, which uses the implicit backward scheme. The schemes used in every quantity in an OpenFOAM computation are set through the dictionary *fvSchemes*. There, terms are separated into categories according to its type, for instance, gradient or divergence (i.e. convective) terms. In Appendix II we include the two instances of this library used in this work, one for computations of Chapter 5 and another for Chapters 3 and 4. All interpolation schemes implemented in OpenFOAM can be consulted in the available documentation (User Guide, The OpenFOAM Foundation, 2016).

The solution of the coupled pressure-velocity equations is approached using the Pressure-Implicit Split-Operator (PISO) algorithm (Issa, 1986). This method uses one predictor step and two corrector steps to solve the discretized flow equations and although the option of adding more corrector steps might increase the accuracy in one order, the midpoint approximation sets the threshold of spatial accuracy to second-order. Complete details regarding this technique can be found in Versteeg and (2007). The choice of PISO algorithm (or, alternatively, SIMPLE) is made in OpenFOAM through the dictionary *fvSolution*. In this file, the solution techniques for the linear, discretized equations resulting from the PISO or SIMPLE are also selected, along with tolerances, number of corrector steps and other parameters available to the chosen technique. Two copies of this file, used for the simulations presented in Chapter 5 and in Chapters 3 and 4, are presented also in the Appendix II. All techniques available for selection in *fvSolution* can be consulted within the User Guide (The OpenFOAM Foundation, 2016).

The swak4Foam library

As with many other open-source platforms, OpenFOAM takes advantage of the collaboration efforts from its users to increase its capabilities. In diverse instances, users have developed new tools (pre/post-processing utilities, solvers, etc.) for particular purposes that are later shared with the community. This has lead to either the production of utilities or libraries that can

be individually used along the standard version of OpenFOAM, or the development of entire software forks[10]. An example of the former is the case of *swak4Foam*[11] (acronym of SWiss Army Knife for Foam), a library created by Bernhard Gschaider, that has been used in some of the computations performed for this work. Amongst its different uses, swak4Foam allows to create and to modify fields and boundary conditions by means of expressions that, depending on the purpose, can become more practical than developing applications from scratch. This library permits to implement a range of manipulations that would be otherwise very complex to achieve solely by use of C++. The practicality of this library is further increased by its capability of handling C++ and python code in combination with its native expressions, all this during run-time or pre/post-processing.

2.2.4 Some comments about EllipSys3D

LES computations of Chapters 3 and 4 are also performed with the CFD code EllipSys3D. It should be noted that the pre/post-processing work, developed concurrently to the one used in OpenFOAM, was conceived by this author and adapted for its use in EllipSys3D with the help of Simon-Philippe Breton from the Department of Earth Sciences, Uppsala University. Furthermore, the simulations on EllipSys3D were performed by Simon-Philippe Breton. Only a limited, general description of this platform is provided here. A description of EllipSys3D and more details about the numerical techniques employed within can be found in Troldborg (2008), Ivanell (2009) and Réthoré (2009).

EllipSys3D code is a general purpose 3D solver, originally developed by J. Michelsen and N. Sørensen (Michelsen, 1992, 1994; Sørensen, 1995) at Risø National Laboratory and the Technical University of Denmark. As OpenFOAM, EllipSys3D is formulated in a finite-volume and a collocated arrangement of variables. Likewise, the Rhie-Chow correction is also implemented. The interpolation scheme for the convective terms employs a blend of QUICK (10%) and a fourth order central-difference scheme (90%), while it uses a second-order central-scheme for

[10]The best example of an OpenFOAM fork is the *extended project*, see: http://www.extend-project.de and http://sourceforge.net/projects/openfoam-extend (last visited on Nov 17th, 2015)

[11]https://openfoamwiki.net/index.php/Contrib/swak4Foam (last visited on Nov 17th, 2015)

the remaining terms. This blended scheme is implemented, as in OpenFOAM, to avoid the apparition of wiggles in the velocity field while limiting numerical diffusion. The pressure correction equation is based on the SIMPLE algorithm and the time derivative is solved using a second-order iterative time-stepping method.

2.3 Rotor modelling

We provide a description of the techniques used to model the rotor of a horizontal axis wind turbine. Two models are implemented in our work, both based on the actuator disk model: the *uniformly loaded* AD and the BEM-based AD *with rotation* where tabulated airfoil data is used to compute lift and drag based on local flow characteristics. Note that in the latter, the name is only a convention as it is wake that rotates (not the disk), as a result of the introduction of a tangential force component.

2.3.1 The actuator disk model

The rotor of a horizontal-axis wind turbine is modelled in the computations as an actuator disk (Sørensen and Myken, 1992), where the effect of the blades on the wind flow is reproduced by forces distributed over a disk. The area of this disk corresponds to the surface swept by the blades which, for the incoming wind, is seen as a porous region. As the actual geometry of the blades is not reproduced, the load of the turbine is taken as an integrated quantity in the azimuthal direction. In its simplest conception, it is assumed that the forces over the AD point only in the axial direction and are distributed uniformly over the disk, acting as a momentum sink in the momentum equation. If U_0 is the inflow velocity, the thrust force is calculated as

$$F_x = -\frac{1}{2}\rho U_0^2 C_T A_D, \tag{2.13}$$

where A_D is the area of the disk and C_T is the thrust coefficient. In turn, F_x is added to the momentum equation as a body force. Consequently, in the implementation of the AD into the

LES solver (i.e. the discretized version of eq. (2.5)), $f_x = F_x/\rho V_D$ where $V_D = A_D\Delta_x$ is the volume[12] of the AD, with Δ_x the cell length in the axial direction.

The introduction of the forces represents an abrupt discontinuity in the flow field, so large velocity gradients occur in the vicinity of the AD and spatial oscillations (wiggles) on the velocity field may appear due to the pressure-velocity decoupling inherent to collocated grids (Sec. 2.2.3). To avoid this effect, different approaches can be followed, such as the use of a staggered grid or the introduction of special treatments for the interpolation of p/U variables between cell centres and faces. For instance, Réthoré (2009) implemented a modification of the algorithm of Rhie and Chow (1983) that yields a pressure discretization where no wiggles emerge. Conversely, in this work we adopt the more common approach of distributing the forces that comprise the AD in the axisymmetric direction (e.g. Troldborg, 2008; Ivanell, 2009). This is done by taking the convolution of the forces f_x with a Gaussian distribution

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{2.14}$$

In this manner, the value of the standard deviation σ (i.e. the distribution width) will define the thickness of the disk. The force distribution is defined between the limits $[-3\sigma, 3\sigma]$ so that it contains 99.7% of magnitude of the forces computed for the original—one cell thick—disk.

2.3.1.1 Validation of the actuator disk implementation

To validate our implementation of the uniformly loaded AD technique in OpenFOAM, we follow a procedure previously used by Réthoré and Sørensen (2008), where the simulated velocity and pressure are compared in two test cases where an analytical solution is known. Specifically, an incompressible, inviscid flow is computed across an infinite strip and an AD, both with very light loads ($C_T \ll 1$)[13]. These calculations are performed in a steady state with a RANS solver for laminar, inviscid flow and the SIMPLE algorithm. While wiggles

[12] Clearly, when the AD does not have a fully circular contour (e.g., when cubic cells are used), the area and volume occupied by the corresponding cells should be considered.

[13] Although not employed in this work, validation procedures above this condition are discussed in page 46).

are observed in the results of both cases, an example of the smoth solution yielded by the convolution with $g(x)$ is shown only for the AD.

Infinite strip

For the first validation case, simulations are made in a box of dimensions $\mathrm{L}_x \times \mathrm{L}_y \times \mathrm{L}_z = 512D \times 0.125D \times 512D$ with a strip of diameter $D = 1$ m. A scheme of the domain containing the strip in the mid x-direction is shown in Figure 2.1. The grid contains $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_z = 1000 \times 4 \times 1000$ cells. The domain size and grid are the result of a sensitivity study where the convergence of the results of p in the transversal direction was sought after. In was found that the employed parameters would result in a variation of less than $1\,\%$ of the value of p when using fewer points or smaller domains. Cells are stretched in the streamwise direction from the position of the strip towards the inlet and outlet (both with the same expansion ratio). In the spanwise direction, cell spacing is maintained constant within the strip and from its edges, cells are stretched in such a way that expansions are equal in both directions, this is, $\Delta x_{max}/\Delta x_{min} = \Delta z_{max}/\Delta z_{min} = 80$. To simulate the infinite strip in the vertical direction, the top and bottom faces are set to symmetry planes while the sides are set to zero gradient (Neumann type). A streamwise velocity of $U_0 = 1$ m/s is fixed at the inlet, while the outlet is set to zero gradient as well as $p = 0$. The uniform force over the strip is calculated using eq. (2.13) with $C_T = 0.01$.

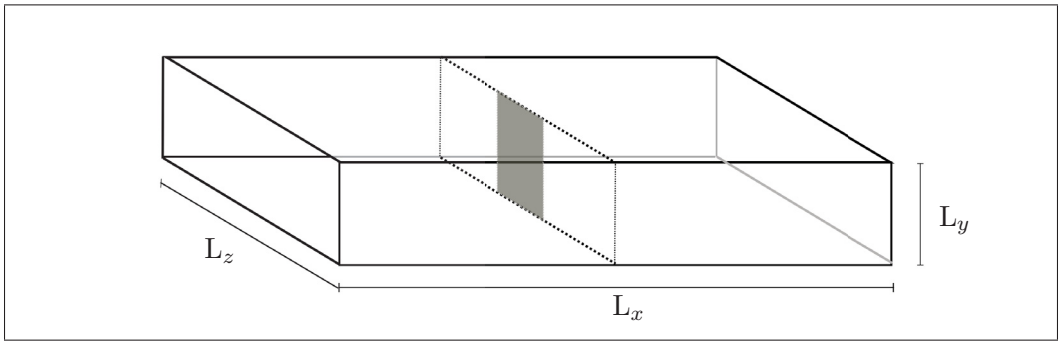

Figure 2.1 Infinite actuator strip validation setup. The shaded region corresponds to the surface.

The analytical solution for a lightly loaded strip, derived by Madsen (1988) are:

$$p(x, y, \Delta p, D) = \frac{\Delta p}{2\pi}\left[\tan^{-1}\left(\frac{D/2 - y}{x}\right) + \tan^{-1}\left(\frac{D/2 + y}{x}\right)\right], \tag{2.15}$$

$$U(x, y, \Delta p, D) = U_\infty - \frac{p(x, y, \Delta p, D)}{\rho U_\infty} - \underbrace{\frac{\Delta p}{\rho U_\infty}}_{\text{only in the wake}}. \tag{2.16}$$

The validation consists in comparing the analytic predictions of p and U at the centre-line in the streamwise direction as well at $1D$ behind the strip in the spanwise direction with the simulation results. There, the uniform load on the strip is computed with eq. (2.13) with no spreading of forces, so the actuator surface is one-cell in thickness. In addition, the strip is also simulated through a *pressure jump*, where instead of adding a momentum source in the cell centres, a pressure discontinuity Δp is imposed over the cell faces along the strip area. The use of this technique results in an actuator surface of infinitesimal thickness that avoids the apparition of wiggles. The pressure difference is computed as:

$$\Delta p = -\frac{F_x}{A_D} = \frac{1}{2}\rho C_T U_0^2. \tag{2.17}$$

The Figure 2.2 shows the results of the computations. There, it can be seen that wiggles appear in the vicinity of the strip when the momentum source technique is used without any distribution of forces. Outside this region, the results of this simulation as well as that performed with the pressure discontinuity fit very well the analytic predictions. Only a slight difference can be appreciated in the spanwise distribution of pressure behind the strip for the momentum source technique, very small compared to the magnitude of Δp. This difference is also observed by Réthoré and Sørensen (2008).

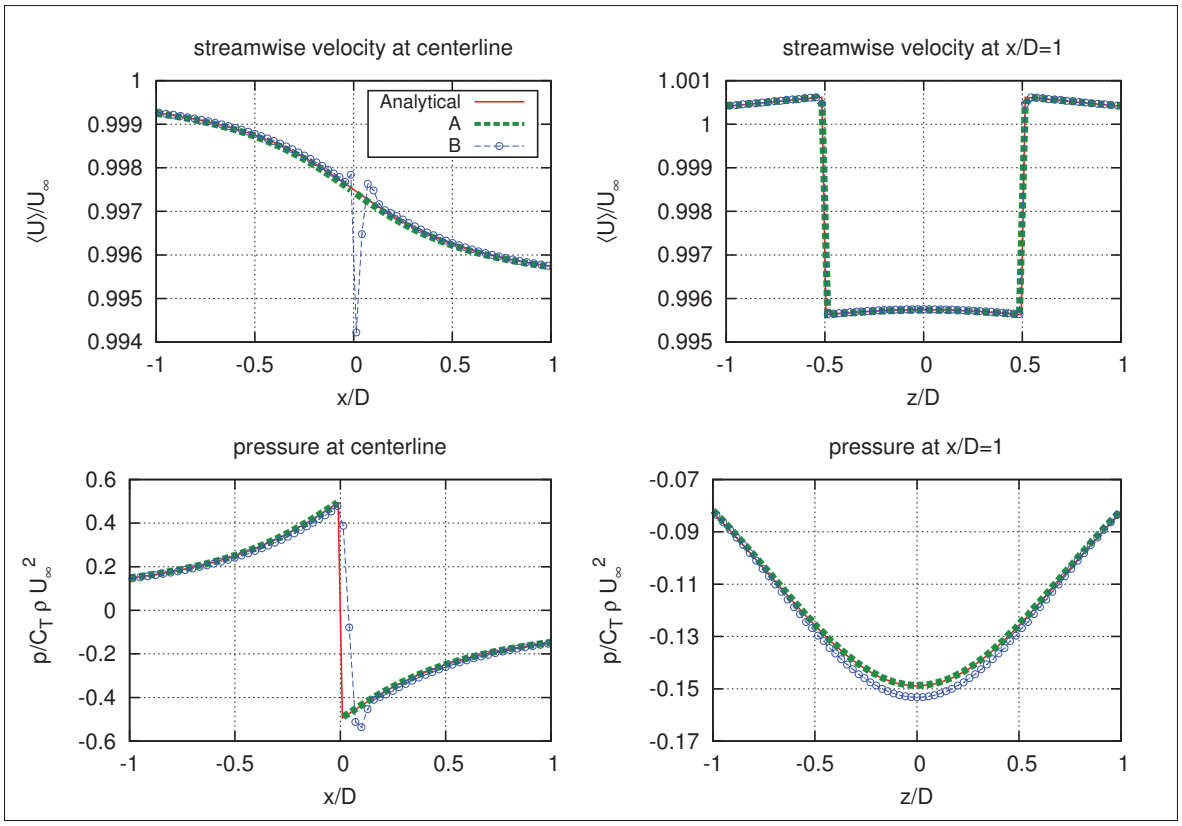

Figure 2.2 Actuator Strip validation. The results of the (A) pressure jump and (B) momentum source technique are compared to the analytic predictions of eqs. (2.15) and (2.16).

Actuator disk

For the validation of the actuator disk we employ a similar procedure than for the infinite strip. The setup of this case is shown in Figure 2.3. The domain consists of a box of $\mathrm{L}_x \times \mathrm{L}_y \times \mathrm{L}_z = 28\,D \times 20\,D \times 20\,D$ where D is the AD diameter. Boundary conditions of all sides are set to zero gradient. The form of the cells in and around is modified to obtain a fully circular shape for the AD. This is exemplified in Figure 2.4 where a cross section (plane $y - z$) of this type of mesh is shown. There, the innermost circle constitutes the AD. The AD and surrounding area, also circular in shape, cover an area equivalent to $2D$ in diameter and contains 72 uniform cells along the y and z axes. As seen in the figure, the surrounding area to the AD contains cells laid in a polar configuration, as opposed to those elsewhere in the grid, so they should be counted

accordingly in the total number of cells[14]. Inside the polar region and AD, no cell stretching is used although cells are distorted towards the edges of the AD. Outside these areas, the cells are stretched in a ratio $\Delta y_{max}/\Delta y_{min} = \Delta z_{max}/\Delta z_{min} = 10$ (not shown in the example of Figure 2.4). In the longitudinal direction, cells are also stretched in a ratio of $\Delta x_{max}/\Delta x_{min} = 8$. The size of the domain as well as the number of points in the streamwise and transversal directions are determined with a similar principle as with the infinite strip. Thus, the employed parameters yield a variation of less than 1% in the transversal pressure. Notably, a good match is found for this parameter with domain dimensions appreciably smaller than in the infinite strip computation.

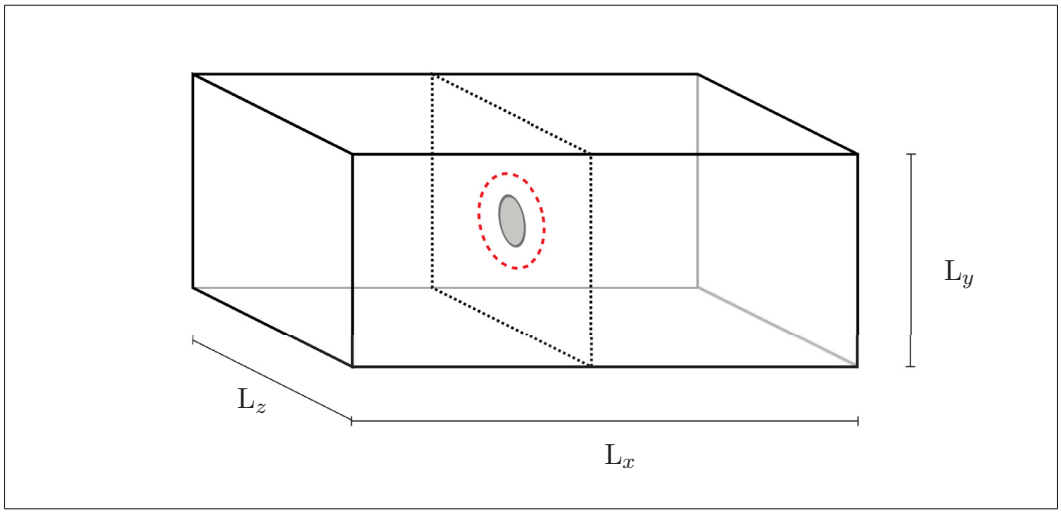

Figure 2.3 Actuator disk validation setup. The shaded circular region on the mid x-direction corresponds to the AD while the dashed perimeter around it contains cells laid in a polar configuration.

The analytic solution for a lightly loaded propeller in polar coordinates are given by Koning (1963):

$$p(x, r, \Delta p, \theta, D) = \frac{\Delta p}{4\pi} \int_0^{D/2} \int_0^{2\pi} \frac{r' x dr' d\theta'}{\left[r'^2 + r^2 + x^2 - 2r'r\cos(\theta' - \theta)\right]^{3/2}}, \tag{2.18}$$

[14]In our computations, the number of cells outside such polar region was $160 \times 94 \times 94$, including the AD. In the polar region, the cells are counted as $160 \times 18 \times 18 \cdot 4$ (18 rings with $18 \cdot 4$ cells in the azimuthal direction).

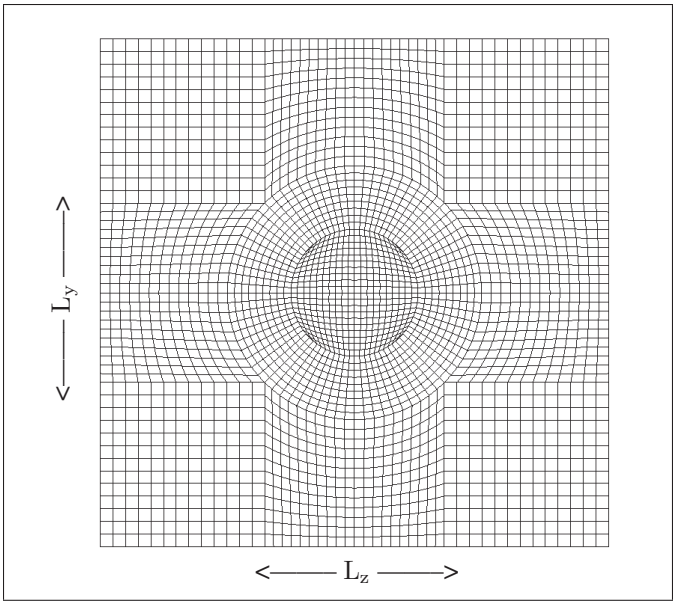

Figure 2.4 Cross sectional plane of the domain used for the validation of the AD implementation. This figure does not show that cells outside the two concentrical circular regions are stretched towards the boundaries, as in the computations.

and

$$U(x, r, \Delta p, \theta, D) = U_0 - \frac{p(x, r, \Delta p, \theta, D)}{\rho U_0} - \underbrace{\frac{\Delta p}{\rho U_0}}_{\text{only in the wake}}, \tag{2.19}$$

which can be assumed equivalent for an AD. Koning provides an approximation to eq. (2.18) at the centreline ($r = 0$) in the following form:

$$r = 0,\ x < 0 \qquad p = \frac{\Delta p}{2}\left(-1 - \frac{x}{\sqrt{(D/2)^2 + x^2}}\right), \tag{2.20}$$

$$r = 0,\ x > 0 \qquad p = \frac{\Delta p}{2}\left(+1 - \frac{x}{\sqrt{(D/2)^2 + x^2}}\right). \tag{2.21}$$

Koning also derives expressions for the velocity from eq. (2.18) in Cartesian directions. For this, he distinguishes the regions from inside the slipstream (the wake envelope[15]) where $(y^2 + z^2) < D^2/4$):

$$U = \frac{\Delta p}{2\rho U_0}\left(1 + \frac{x}{\sqrt{(D/2)^2 + x^2}}\right), \tag{2.22}$$

$$V = -\frac{\Delta p}{4\rho U_0}\left[\frac{R^2 y}{((D/2)^2 + x^2)^{3/2}}\right], \tag{2.23}$$

$$W = -\frac{\Delta p}{4\rho U_0}\left[\frac{R^2 z}{((D/2)^2 + x^2)^{3/2}}\right] \tag{2.24}$$

and outside the slipstream ($(y^2 + z^2) > D^2/4$):

$$U = -\frac{\Delta p}{4\rho U_0}\left[\frac{R^2 x}{(x^2 + y^2 + z^2)^{3/2}}\right], \tag{2.25}$$

$$V = -\frac{\Delta p}{4\rho U_0}\left[\frac{R^2 y}{(x^2 + y^2 + z^2)^{3/2}}\right], \tag{2.26}$$

$$W = -\frac{\Delta p}{4\rho U_0}\left[\frac{R^2 z}{(x^2 + y^2 + z^2)^{3/2}}\right], \tag{2.27}$$

where the subindex in U indicates the direction where the velocity is sampled. In this way, a validation analogous to the one made for the infinite strip (i.e. centreline p, U and spanwise U behind disk) can be done in this case using eqs. (2.20)-(2.27) while only the spanwise distribution of pressure needs to be evaluated numerically from the integral in eq. (2.18).

The Figure 2.5 shows the comparison of our results with the analytic solutions. Unlike the case of the infinite strip, this time the results obtained with a Gaussianly-distributed momentum

[15]These expressions are valid for laminar flow where there is not wake expansion.

source (i.e. the convolution of eqs. (2.13) and (2.14)) are also shown. Except for the solutions of the non-distributed momentum source at centreline (which produces wiggles around the AD), our computations match very well the analytic predictions. Notably, the results demonstrate that the Gaussian distribution of the momentum source prevent the wiggles from appearing, yielding instead a smooth solution for p and U across the AD. This is a crucial feature in transient simulations that employ the rotating AD technique, to be described in the next section, since the local value of the velocity vector is required to calculate the aerodynamic forces over the AD.

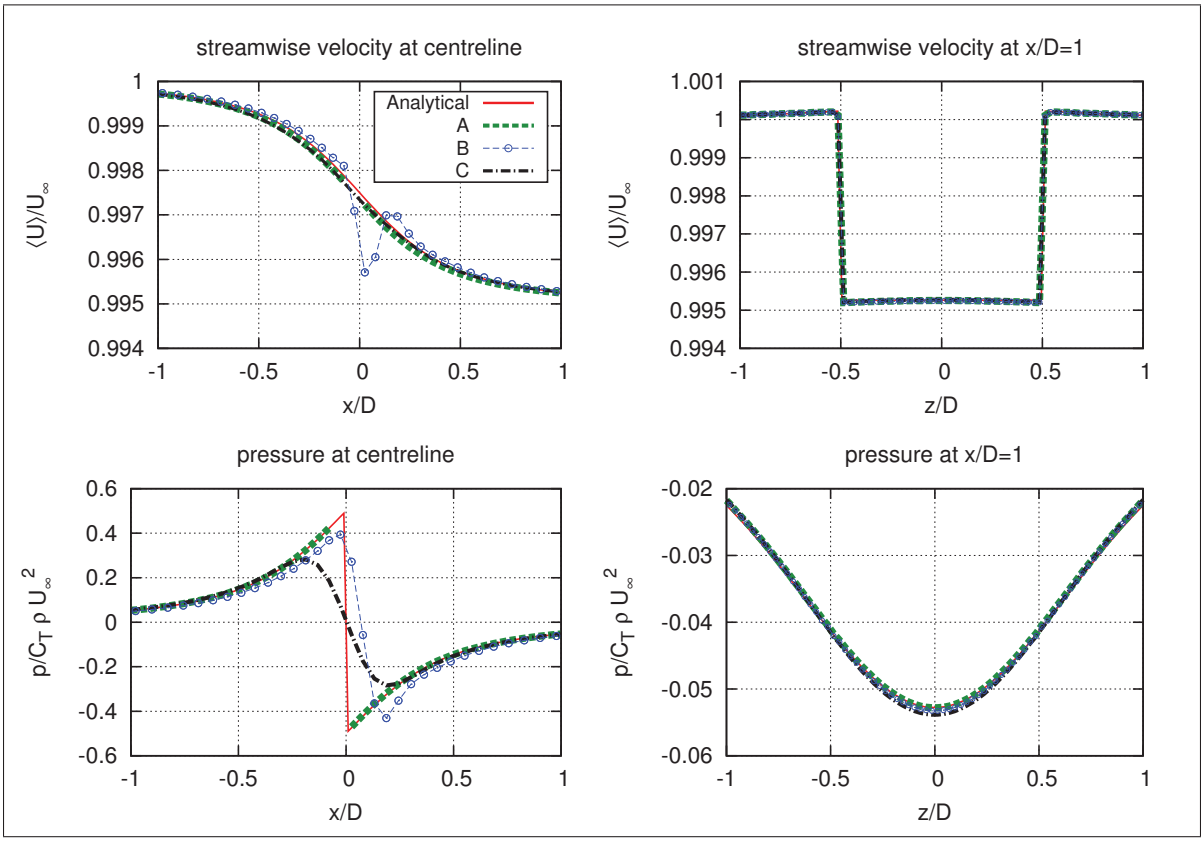

Figure 2.5 Actuator Disk validation. The results of the (A) pressure jump and (B) momentum source and (C) momentum source with a Gaussianly distribution are compared to the analytic predictions of eqs. (2.18)-(2.27).

The validation process can be extended outside the restriction of lightly loaded disks. For that, the solutions provided by the models of Conway (1995) and Conway (1998) for different

load distributions can be used (albeit considerably more complex in comparison), as recently done by Réthoré *et al.* (2014). Additionally, analytical forms for the axisymmetrical expansion of the wake can be compared to simulation results, such a study of self-similarity[16]. These forms have been discussed by Johansson *et al.* (2003) and corroborated experimentally by Johansson and George (2006) using a solid disk in high Re flows. As the validation shown here (focused on uniformly loaded, porous disk) provides satisfying results, it was decided to carry on with comparisons with experimental data as well as with the results from computations with EllipSys3D, which has been previously tested by various authors in a number of similar works (e.g. Ivanell, 2009; Réthoré, 2009; Troldborg, 2008; Troldborg *et al.*, 2015; Keck *et al.*, 2014). These comparisons are shown in Chapter 4.

2.3.2 Rotating actuator disk

The previous implementation of the AD distributes the thrust uniformly over the area of the disk. When real rotors are modelled, the omission of rotational effects and a representative distribution of the actual loads deprive the model from offering better results. Thus, without largely increasing the level of sophistication of the model, which would be the case if the actuator line or a fully resolved rotor geometry were used, these effects can be accounted for while still using the AD technique (see Troldborg *et al.* (2015) for a comparison between these rotor models) . For this, the BEM theory can be be combined with a flow solver to produce a more realistic model that includes the influence of the airfoil on the incoming flow.

The Figure 2.6 presents a scheme of the modelled airfoil with the different angles, forces and velocities in question. There, Ω is the angular velocity of the rotor, that itself has B number of blades and is divided in sections having with local chord c and pitch angle θ_p. The relative velocity U_{rel} is the vector sum of the wind velocity at the rotor and the velocity due to the blade rotation. $\alpha_a = \varphi - \theta_p$ is the local angle of attack, where φ is the angle between U_{rel} and the rotor plane. U_x and U_θ are the axial and tangential components of the wind velocity,

[16] A wake is said to become self-similar when the mean velocity profiles *collapse* when normalized by the velocity deficit at centreline and a lengthscale based on the wake width. This occurs for far downstream regions of the wake $x > 10D$, see Johansson *et al.* (2003).

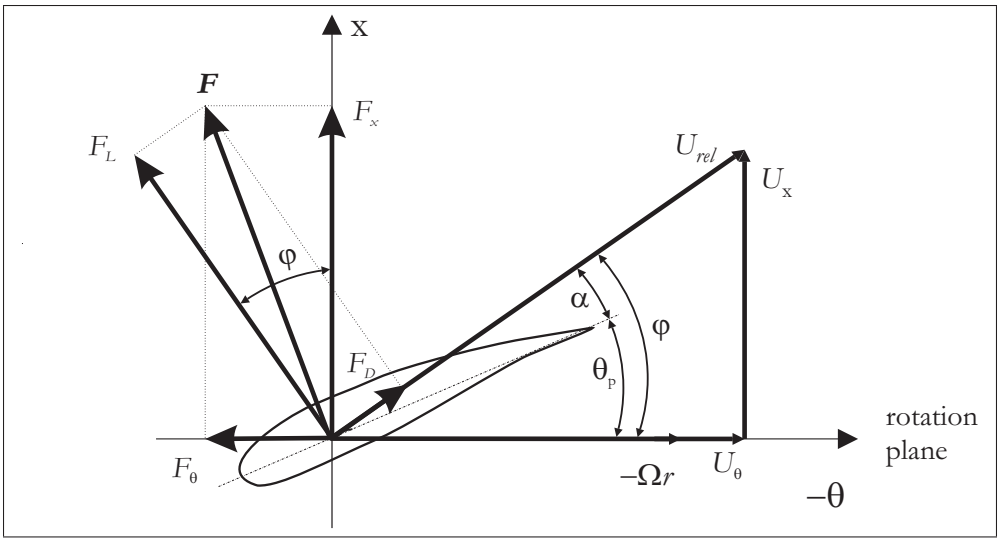

Figure 2.6 Geometry and forces in an airfoil section of the blade.

respectively. Also from this Figure 2.6, it can be deduced that:

$$U_{rel} = \sqrt{U_x^2 + (\Omega r + U_\theta)^2}, \tag{2.28}$$

$$\varphi = \tan^{-1}\left(\frac{U_x}{\Omega r - U_\theta}\right). \tag{2.29}$$

The lift and drag forces, depicted in Figure 2.6 as F_L and F_D, are calculated from projections of the vector $\boldsymbol{F}$, the force induced by the turbine. Denoting the directions where these forces act with the unitary vectors $\mathbf{e}_L$ and $\mathbf{e}_D$, the forces per unit area due to the blades are calculated from

$$\mathrm{d}\boldsymbol{F} = \frac{1}{2}\rho c U_{rel}^2 \frac{B}{2\pi r}(C_L\mathbf{e}_L + C_D\mathbf{e}_D)\mathrm{d}A, \tag{2.30}$$

for an annular area segment $dA = 2\pi r dr$ within the disk area swept by the blades. The lift and drag coefficients C_L, C_D are obtained from tabulated airfoil data, where values are listed as a function of α_a for a given Re value.

A critical step in the implementation of this model consists in the evaluation of the time-averaged force $\mathrm{d}\boldsymbol{F}$ during one period of rotation on the corresponding control volume. In an LES computation, the demands of a uniform filter in the regions of both the AD and wake

causes that the preferred shape of the control volume cells does not coincide with that of the annular segments dA. In this regard, Ammara (1998) and Ammara *et al.* (2002) proved that the time-averaged force is in fact independent from the shape of the control volume. Hence, the factor $B/2\pi r$ in eq. (2.30) corresponds to the equivalent force during one rotation of the B number of blades. This feature allows this expression to be used irrespective of the choice of domain discretization, which is highly desirable when Cartesian grids are used (or other mesh restrictions are considered).

The projection of $\mathrm{d}\boldsymbol{F}$ on the longitudinal axis as well as at the rotor plane results in:

$$F_x = F_L \cos\varphi + F_D \sin\varphi \tag{2.31}$$

and

$$F_\theta = F_L \sin\varphi - F_D \cos\varphi, \tag{2.32}$$

which correspond to the axial (thrust) and tangential forces, respectively. Evaluating lift and drag forces with eq. (2.30) yields:

$$\mathrm{d}F_x = \frac{1}{2}\rho c U_{rel}^2 \frac{B}{2\pi r} \left(C_L \cos\varphi + C_D \sin\varphi\right) \mathrm{d}A \tag{2.33}$$

and

$$\mathrm{d}F_\theta = \frac{1}{2}\rho c U_{rel}^2 \frac{B}{2\pi r} \left(C_L \sin\varphi - C_D \cos\varphi\right) \mathrm{d}A. \tag{2.34}$$

A tip correction factor is introduced to account for the lift losses due to the flow of air around the blade tip (caused by the pressure difference). This factor is computed as (Hansen, 2003):

$$\vartheta = \frac{2}{\pi} \cos^{-1}\left[\exp\left(-\frac{B}{2}\frac{R-r}{r\sin\varphi}\right)\right]. \tag{2.35}$$

The correction is applied by replacing C_L by C_L/ϑ in the eqs. (2.33) and (2.34) following the example of Masson *et al.* (2001) and Ammara *et al.* (2002). Hence, the tip-corrected thrust and

torque forces are calculated as:

$$\mathrm{d}F_x = \frac{1}{2}\rho c U_{rel}^2 \frac{B}{2\pi r}\left(\frac{C_L}{\vartheta}\cos\varphi + C_D \sin\varphi\right)\mathrm{d}A \tag{2.36}$$

and

$$\mathrm{d}F_\theta = \frac{1}{2}\rho c U_{rel}^2 \frac{B}{2\pi r}\left(\frac{C_L}{\vartheta}\sin\varphi - C_D \cos\varphi\right)\mathrm{d}A. \tag{2.37}$$

As these equations provide the force per unit area, the body force inserted in the LES solver is computed as $f_i = \mathrm{d}F_i/\rho\Delta x$, where Δx is the length of the cell in the axial direction. As in the previous case of the uniformly loaded AD, the single-point force is Gaussianly distributed over the longitudinal direction to avoid the undesired oscillations in the velocity field.

To compute the total torque and thrust of the rotor, $\mathrm{d}F_i$ from eqs. (2.33) and (2.34) is integrated over the surface of the disk,

$$F_x = \int_{A_D} \mathrm{d}F_x \tag{2.38}$$

and

$$Q_{aero} = \int_{A_D} r\mathrm{d}F_\theta, \tag{2.39}$$

where the so called aerodynamic torque is written as Q_{aero} to differentiate this quantity from Q_{gen} , defined in the next section.

Therefore, the power output is calculated as:

$$P = \Omega Q_{aero}. \tag{2.40}$$

Make note that eqs. (2.36) and (2.37) are only used to correct the effect of the disk forces on the flow, but not in the evaluation of the total aerodynamic load and torque.

Besides the correction for the tip forces, the conception used here does not consider a special treatment for the cells at the edge of the AD to approximate a circular contour nor it uses a

smearing function other than in the axial direction, in contrast to other implementations of the AD with rotation (e.g. Ivanell, 2009).

2.3.2.1 Rotational control method

The simulation of a rotor that responds to variations of inflow velocities requires the addition of a technique to regulate the rotational speed. In HAWTs this can be achieved two types of systems:

- With a *generator-torque controller* where the turbine rotation is regulated in function of the incoming wind speed.
- Using a *blade-pitch controller* to vary the pitch angle of the blades to reduce lift as a method to regulate the generator speed.

The use of either system will depend on the rated operation point, or *rated power*. In general, the turbine operates with the generator-torque controller below rated power, to maximize the energy production by maintaining a constant tip-speed ratio $\Lambda = \Omega R/U_0$. At rated power, occurring at higher inflow velocities, the pitch controller is used to reduce the lift force at the blades and adjust the rotational velocity.

The procedure is regulated by means of tabulated data of the generator torque as a function of the rotor angular velocity (computed from the local wind speed). A so-called generator curve provides the information about the optimal relation between these two quantities, in agreement to specifications of the manufacturer. In this way, the rotational speed is the result of the wind velocity computed at the disk and the equilibrium between the aerodynamic torque and the generator torque. The methodology to create the generator curve can be found in Jonkman *et al.* (2009), where in function of the rotational velocity, three main regions of operation are distinguished: 1) the startup region, where the rotor accelerates but no power is extracted, 2) the control region, where Λ is kept constant to optimize power production and 3) the pitch-control

region where generator power is maintained constant. Only the generator-torque controller of region 2, henceforth called simply the *controller*, is implemented in this work.

The control system applied here is based on the work presented by Breton *et al.* (2012), also described in Nilsson (2015). Following the latter, the starting point consists in considering the computed torque Q_{aero} from eq. (2.39). Then, observing that in a rotating, rigid body, the net torque is proportional to the angular acceleration of the object, with the proportionality factor being the moment of inertia, we obtain

$$Q_{aero} - Q_{gen} = (I_{rot} + I_{gen}) \frac{\Delta\Omega}{\Delta t}, \tag{2.41}$$

where the rotor and generator moments of inertia can be combined and to yield the drivetrain moment of inertial $I_d = I_{rot} + I_{gen}$. In this way, the method comprises the following steps:

a. Calculate $Q_{aero}(\Omega(t))$

b. Determine the corresponding value Q_{gen} for the given $\Omega(t)$ in the generator curve

c. The difference in angular velocity between the current $\Omega(t)$ and the one dictated by the generator curve is then computed as:

$$\frac{\Delta\Omega}{\Delta t} = \frac{Q_{aero} - Q_{gen}}{I_d} \tag{2.42}$$

d. If Δt is taken the computational time-step, the angular velocity that the rotor should follow at the next time step $(t + \Delta t)$ is

$$\Omega(t + \Delta t) = \Omega(t) + \Delta\Omega \tag{2.43}$$

As Q_{aero} is calculated with the local velocities, the rotational response of the rotor is effectively modelled following realistic conditions. This completes the presentation of the model of the rotating AD. The implementation made in OpenFOAM of this method is validated by

comparing the performance of the modelled rotor with values reported by the designer. These results are shown in Chapter 5.

2.4 Homogeneous isotropic turbulence

The large part of the flow computations in this work (Chapters 3 and 4) concern the simulation of homogeneous turbulence. We present here some definitions that will be used in our study[17].

Turbulence is characterized by random processes, but also by the apparition of coherent structures. In homogeneous turbulence, where the statistical properties are invariant under spatial displacements, the two-point correlation function (a covariance tensor), defined as

$$R_{ij}(\mathbf{r},t) \equiv \left\langle u_i'(\mathbf{x}+\mathbf{r},t)u_j'(\mathbf{x},t)\right\rangle , \tag{2.44}$$

provides a fundamental description of the spatial structure of turbulence. From here, the interaction of the two velocity components can be traced by the correlation coefficient

$$\mathcal{R}_{ij}(\mathbf{r},t) \equiv \frac{\left\langle u_i'(\mathbf{x},t)u_j'(\mathbf{x}+\mathbf{r},t)\right\rangle}{\sqrt{{u_i'}^2(\mathbf{x},t)}\sqrt{{u_j'}^2(\mathbf{x}+\mathbf{r},t)}} . \tag{2.45}$$

The characteristic size of the largest eddies is identified as the distance L required to nullify the correlation function. With this assumption, the integral lengthscale

$$L_{ij}^{(d)} = \int_0^\infty \mathcal{R}_{ij}(\boldsymbol{e}_d r,t)dr \tag{2.46}$$

in the direction d is defined. From all scales defined by this expression, those most commonly used are the longitudinal integral lengthscale $L_1 = L_{11}^{(1)}$ as well as the transversal one $L_2 = L_{22}^{(1)}$, both to be used later in this work. Similarly, the Taylor lengthscale (or micro-scale) is defined by the osculating parabola to the correlation function eq. (2.45). In this manner, it can be shown that the longitudinal ($\lambda_1 = \lambda_{11}^{(1)}$) and transverse ($\lambda_2 = \lambda_{22}^{(1)}$) Taylor lengthscales are

[17] These definitions are based on those provided by Bailly and Comte-Bellot (2003) and Pope (2000).

given by

$$\frac{1}{\lambda_1^2} = \frac{1}{2\left\langle {u_1'}^2 \right\rangle} \left\langle \left(\frac{\partial u_1'}{\partial x_1} \right)^2 \right\rangle \tag{2.47a}$$

$$\text{and} \quad \frac{1}{\lambda_2^2} = \frac{1}{2\left\langle {u_1'}^2 \right\rangle} \left\langle \left(\frac{\partial u_1'}{\partial x_2} \right)^2 \right\rangle, \tag{2.47b}$$

respectively. If isotropy is assumed (or at least between the 1 & 2 directions) the equivalences $L_{22}^{(1)} = L_{11}^{(2)}$ and $\lambda_{22}^{(1)} = \lambda_{11}^{(2)}$ are also valid. In incompressible isotropic turbulence, it is found that the longitudinal and transversal components for each scale are related through the expressions

$$L_1 = 2L_2 \tag{2.48a}$$

$$\text{and} \quad \lambda_1 = \sqrt{2}\lambda_2 \ . \tag{2.48b}$$

In the absence of shear, the Taylor hypothesis of frozen turbulence can be adopted comfortably. This is, it is assumed that the turbulence field does not change as it is convected by the mean wind at $\langle U \rangle$, which yields the equivalence between the spatial and temporal correlations. In this way, correlations can be made from the time series of each velocity component. In particular, the autocorrelation will provide the integral time scales $\mathcal{T}_{11}$ and $\mathcal{T}_{22}$ from where the integral lengthscales can be computed by means of $L_1 = \langle U \rangle \mathcal{T}_{11}$ and $L_2 = \langle U \rangle \mathcal{T}_{22}$. Likewise, the longitudinal Taylor lengthscale can be calculated from the expression

$$\frac{1}{\lambda_1^2} = \frac{\langle U \rangle^{-2}}{2\left\langle {u_1'}^2 \right\rangle} \left\langle \left(\frac{\partial u_1'}{\partial t} \right)^2 \right\rangle \tag{2.49}$$

as seen in Jiménez, Javier (Ed.) (1997)[18]. The determination of the Taylor scale is specially useful to typify the flow, as it allows to define a Reynolds number $\mathrm{Re}_\lambda = u_{rms}\lambda_2/\nu$ without ambiguity (using the shorthand $u_{rms}^2 = \left\langle {u_1'}^2 \right\rangle$ for the root-mean-square –r.m.s.– velocity). Moreover, in isotropic turbulence, λ_2 is related to the amount of dissipation of the turbulent

[18]Note that our eq. (2.49) differs from the one presented in that report (third eq. in page 10) by a factor of $\sqrt{2}$.

kinetic energy,

$$\varepsilon = \frac{15\nu \left\langle {u_1'}^2 \right\rangle}{\lambda_2^2} = \frac{30\nu \left\langle {u_1'}^2 \right\rangle}{\lambda_1^2} \quad . \tag{2.50}$$

The lengthscale corresponding to the dissipative structures of turbulence is defined as

$$\eta \equiv \left(\frac{\nu^3}{\varepsilon} \right)^{1/4} \quad . \tag{2.51}$$

The derivation of the turbulence energy spectrum $E(\kappa)$ is given by the integration of the spectrum tensor, which is in turn defined as the Fourier transform of the correlation function. However, it is greatly more practical to compute the one-dimensional spectra E_{11} and E_{22}, which can be calculated from the Fourier transform of the corresponding correlation function (eq. 2.44), this is:

$$E_{ij}(\kappa_1) \equiv \frac{1}{\pi} \int_{-\infty}^{\infty} R_{ij}(\boldsymbol{e}_1 r_1) e^{-i\kappa_1 r_1} dr_1 \, . \tag{2.52}$$

Make note that the use of the Taylor hypothesis permits a change of variable between the frequency spectra $E_{ij}(f)$ computed from a time series of one-point velocity to $E_{ij}(\kappa_1)$ by means of $\kappa_1 = 2\pi f / \langle U \rangle$.

In the analysis of the HIT, the model of the energy spectrum suggested by von von Kármán (1948) is particularly useful,

$$E(\kappa) = \alpha \varepsilon^{2/3} L^{5/3} \frac{L^4 \kappa^4}{(1 + L^2 \kappa^2)^{17/6}} \tag{2.53}$$

where L is the lengthscale characterizing the large eddies $L \equiv k^{3/2}/\varepsilon$, α the Kolmogorov constant and ε the viscous dissipation. From this expression, the one-point, one-sided spectra are derived as,

$$F_1(\kappa_1) = \frac{18}{55} \alpha \varepsilon^{2/3} L^{5/3} \frac{1}{(1 + L^2 \kappa_1^2)^{5/6}} \tag{2.54}$$

for the longitudinal spectrum and

$$F_i(\kappa_1) = \frac{6}{110} \alpha \varepsilon^{2/3} L^{5/3} \frac{3 + 8L^2 \kappa_1^2}{(1 + L^2 \kappa_1^2)^{11/6}} \tag{2.55}$$

for the transversal spectra $i = 2, 3$ (Mann, 1998).

As indicated by Mann (1994), L can be characterized by the maximum of $\kappa_1 E_i(\kappa_1)$. Furthermore, noting that the wavenumber at maximum of $\kappa_1 E_i(\kappa_1)$ is $1/L_{max,i}$, Mann estimates that

$$L_{max,1} = \left(\frac{2}{3}\right)^{1/2} L \approx 0.816L \tag{2.56}$$

and

$$L_{max,i} = \frac{2}{\left(6 + 3\sqrt{5}\right)^{1/2}} L \approx 0.561L \quad \text{for} \quad i = 2, 3 \ . \tag{2.57}$$

If the lengthscales on the left of the two previous equations are identified with the integral lengthscales L_1 and L_2, the above expressions provide a useful link between L_1 obtained from the velocity correlations and the von Kármán model. Especially when the spectrum of an experimental or numerical velocity field is fitted to that model. Pope (2000), indicates that this relation depends on the Re number, going from approximately the value of eq. (2.56) for $\mathrm{Re}_\lambda \sim 30$ and approaching asymptotically to 0.43 for $\mathrm{Re}_\lambda \sim 10000$ (Celik *et al.*, 2005 uses 0.55).

2.4.1 Decaying turbulence

When grid turbulence is used to approximate the theoretical case of decaying isotropic turbulence, the characteristics observed at different positions downstream from the grid correspond to the time evolution of isotropic turbulence with zero mean velocity. Thus, a decay during the interval Δt is approximated by that occurring within Δx in a wind tunnel. In this manner, the turbulence kinetic energy decay has been observed to follow the expression

$$\frac{k}{\langle U \rangle^2} = c_A \left(\frac{x - x_0}{M}\right)^{-n} \ , \tag{2.58}$$

where M is the turbulence grid size, c_A (also written as $1/A$) is a fitting parameter, n is the decay exponent and x_0 a virtual origin. Eq. (2.58) is commonly employed to track the streamwise turbulence intensity decay, replacing k for $\langle u_1'^2 \rangle$. While Bailly and Comte-Bellot (2003)

mentions $1.1 \leq n \leq 1.3$ and $c_A \simeq 1/30$, Kang *et al.* (2003) report to have observed $n = 1.25$ with $x_0 = 0$, whereas Pope (2000) mentions $1.15 \leq n \leq 1.45$ remarking that c_A varies greatly depending on the geometry of the grid and Re_λ. The decay of dissipation of k can be deduced from the previous expression as

$$\varepsilon = -\frac{dk}{dt} = -\langle U \rangle \frac{dk}{dx} = n c_A \frac{\langle U \rangle^3}{M} \left(\frac{x}{M}\right)^{-n-1} \tag{2.59}$$

with the reported value of $c_A = 1.8$ (Kang *et al.*, 2003). Bailly and Comte-Bellot (2003) recount that the dissipation can also be quantified in terms of the integral lengthscale, this is,

$$\varepsilon = \frac{\left\langle u_1'^2 \right\rangle^{3/2}}{L_1} = \frac{\left\langle u_1'^2 \right\rangle}{L_1 / \sqrt{\left\langle u_1'^2 \right\rangle}}, \tag{2.60}$$

where the denominator of the right hand side corresponds to the characteristic time of turbulence extinction. In turn, this time also corresponds to the time correlation in a frame of reference convected at $\langle U \rangle$. The fulfilment of this equation can be considered an indication of the fully-development of turbulence, in the sense that ε can be calculated from the large scales (Mydlarski and Warhaft, 1996). Make note that they reported that eq.(2.60) is found to be $\varepsilon = 0.9 \left\langle u_1'^2 \right\rangle^{3/2} / L_1$ for a range of flows with $50 \leq R_\lambda \leq 473$ (the study is later extended in Mydlarski and Warhaft (1998) to $30 \leq R_\lambda \leq 731$). Such relation is also used in Kang *et al.* (2003).

Bailly and Comte-Bellot (2003) indicate that the integral lengthscale evolves downstream according to

$$L_2 \simeq c_{B_1} M \left[\frac{x - x_0}{M}\right]^{n_1} \tag{2.61}$$

and for the Taylor lengthscale

$$\lambda_2 \simeq c_{B_2} M \left[\frac{x - x_0}{M}\right]^{n_2}, \tag{2.62}$$

with the values $c_{B_1} = 0.06$, $n_1 = 0.35$, $c_{B_2} = 0.02$ and $n_2 = 0.5$, making note of the non-similarity in the growth of these scales.

2.5 Modelling of Turbulence

To produce the HIT field used as inflow in our computations we make use of the algorithm introduced by Mann (1994), further discussed in Mann (1998) and Peña *et al.* (2013). This technique has been widely used for the generation of inflow turbulence in uniform, non-sheared flows (e.g. Bechmann, 2006; Troldborg, 2008; Gilling and Sørensen, 2011; Troldborg *et al.*, 2015), as well as for the generation of inflow turbulence in ABL computations (e.g. Troldborg, 2008; Ivanell, 2009; Peña *et al.*, 2010; Nilsson, 2015). This model is recommended by the International Electrotechnical Commission (IEC, 2005) for the reproduction of inflow conditions aimed at computing loads on wind turbines. Due to the relative complexity of the model, we consider appropriate to provide a description of what we consider to be the essential points the model and the algorithm implementation. This is also relevant as the components of the model shown here are taken from the three works of Mann and are presented here together. Subsequently, the results from two instances of the algorithm (ABL and homogeneous flow) are validated.

The underlying idea of the technique of Mann is the modelling of the velocity-spectrum tensor Φ_{ij} of a neutral atmospheric surface layer turbulence. Initially, the conditions of the turbulence field (i.e. second-order statistics) are given by the von Kármán tensor, with energy spectrum equal to that of eq. (2.53). From there, the model calculates the evolution of the velocity field employing a linearized version of the Navier-Stokes equations by making use of the Rapid Distortion Theory (RDT), which gives an equation for the stretching of the spectral tensor, having assumed a linear shear profile caused by wind. Since the stretching of the eddies would continue indefinitely under this assumption, the concept of eddie life time $\tau(\kappa)$ helps to model the eventual breaking of the eddies under the shear. Note that in the following equations we maintain the notation used in the works of Mann, where the vectorial notation (e.g. $\boldsymbol{\kappa}$ de-

notes the wavenumber vector with components $(\kappa_1, \kappa_2, \kappa_3)$) is used in combination with the Einstein notation.

2.5.1 Description of the Mann model

As mentioned above, second-order statistics of turbulence can be derived from its covariace tensor (eq. 2.44) or alternatively, from its Fourier transform, which corresponds to the spectral tensor Φ_{ij}. While the non-periodic, statistically-stationary, stochastic velocity field $\boldsymbol{u}(\boldsymbol{x})$ does not have a direct Fourier transform, it does have a spectral representation given in terms of the generalized stochastic Fourier-Stieltjes integral:

$$\boldsymbol{u}(\boldsymbol{x}) = \int_{-\infty}^{+\infty} \mathrm{e}^{i\boldsymbol{\kappa}\cdot\boldsymbol{x}} \mathrm{d}\boldsymbol{Z}(\boldsymbol{\kappa}), \tag{2.63}$$

where $\boldsymbol{Z}(\boldsymbol{\kappa})$ is a complex random function, whose spectrum yields the spectral tensor[19]

$$\Phi(x)_{ij}\mathrm{d}\kappa_1\mathrm{d}\kappa_2\mathrm{d}\kappa_3 = \left\langle \mathrm{d}Z_i^*(\boldsymbol{\kappa})\mathrm{d}Z_j(\boldsymbol{\kappa})\right\rangle, \tag{2.64}$$

where " * " denotes the complex conjugate.

The model relies on the RDT to simulate the effect of a linear shear on the eddies in an otherwise homogeneous field. RDT is applied under the condition that the magnitude of the mean velocity gradients is much larger than the turbulence rates ($\mathcal{S}\tau = \mathcal{S}k/\varepsilon >> 1$[20]) for the theory to be used with the energy-containing motions (Bailly and Comte-Bellot, 2003; Pope, 2000). In this way, the linearized Navier-Stokes equations of an incompressible flow are obtained by means RDT, which in turn leads to the basic rapid distortion equation of shear flow,

$$\frac{\mathrm{D}\mathrm{d}Z_i(\boldsymbol{\kappa},t)}{\mathrm{D}t} = \frac{\mathrm{d}U}{\mathrm{d}z}\left[-\delta_{i1} + 2\frac{\kappa_i\kappa_1}{\kappa^2}\right]\mathrm{d}Z_3(\boldsymbol{\kappa},t). \tag{2.65}$$

[19] Besides the work of Mann, see Sec. E.3 of Pope (2000) for a similar derivation.

[20] Here $\mathcal{S} = \left(2\bar{S}_{ij}\bar{S}_{ij}\right)$ where $\bar{S}_{ij}$ is the mean rate-of-strain tensor $\bar{S}_{ij} = \frac{1}{2}\frac{\partial\langle U_i\rangle}{\partial x_j} + \frac{\partial\langle U_j\rangle}{\partial x_i}$ not to be confused with the filtered rate-of-strain $\overline{S}_{ij}$ of eq. (2.4) used in LES.

At $t = 0$, the wavenumber number vector is given $\boldsymbol{\kappa}_0 = (\kappa_1, \kappa_2, \kappa_{3,0})$, its development in time follows

$$\boldsymbol{\kappa}(t) = \left(\kappa_1, \kappa_2, \kappa_{3,0} - \kappa_1 t \frac{\mathrm{d}U}{\mathrm{d}z}\right), \tag{2.66}$$

where $\mathrm{d}U/\mathrm{d}z$ is constant if shear is linear. If a non-dimensional time β is used, we write $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \kappa_3)$ with $\kappa_3 = \kappa_{3,0} - \beta k_1$. Mann postulates that the eddies are stretched over a time proportional to their life time as $\tau(\kappa) \propto \varepsilon^{-1/3}\kappa^{-2/3}$ at least along the inertial subrange (eddies with wavevector magnitude $\kappa = |\boldsymbol{\kappa}|$). Under this assumption, Mann redefines the non-dimensional time as

$$\beta \equiv \frac{\mathrm{d}U}{\mathrm{d}z}\tau = \Gamma \frac{\mathrm{d}U}{\mathrm{d}z}\left(\kappa L\right)^{-2/3}, \tag{2.67}$$

where L is the turbulence scale of eq. (2.53) and Γ is a parameter that models the effect of anisotropy in the field due to shear[21]. Considering this above, Mann writes the solution to eq. (2.65) as

$$\mathrm{d}Z_i(\kappa, \beta) = \begin{bmatrix} 1 & 0 & \zeta_1 \\ 0 & 1 & \zeta_2 \\ 1 & 0 & \kappa_0^2/\kappa^2 \end{bmatrix} \mathrm{d}Z_i^{iso}(\boldsymbol{\kappa}_0), \tag{2.68}$$

where

$$\zeta_1 = \left[C_1 - \frac{\kappa_2}{\kappa_1}C_2\right], \ \zeta_2 = \left[\frac{\kappa_2}{\kappa_1}C_1 + C_2\right], \tag{2.69}$$

with

$$\begin{aligned} C_1 &= \tfrac{\beta\kappa_1^2(\kappa_0^2 - 2\kappa_{3,0}^2 + \beta\kappa_1\kappa_{3,0})}{\kappa^2(\kappa_1^2+\kappa_2^2)}, \\ C_2 &= \tfrac{\kappa_2\kappa_0^2}{(\kappa_1^2+\kappa_2^2)^{3/2}} \arctan\left[\tfrac{\beta\kappa_1(\kappa_1^2+\kappa_2^2)^{1/2}}{\kappa_0^2 - \kappa_{3,0}\kappa_1\beta}\right], \end{aligned} \tag{2.70}$$

while $\mathrm{d}\boldsymbol{Z}^{iso}(\boldsymbol{\kappa}_0, \beta)$ is determined from the isotropic von Kármán tensor

$$\Phi_{ij}\frac{E(\kappa)}{4\pi\kappa^4}(\delta_{ij}\kappa^2 - \kappa_i\kappa_j) \tag{2.71}$$

with $E(\kappa)$ given by eq. (2.53).

[21] Mann derives an better approximation of τ for scales beyond the inertial range. However, in the implementation used in this work eq. (2.67) is used.

The actual simulation of the velocity field for a domain (known as turbulence box) of dimensions $\mathrm{L_{B,1}} \times \mathrm{L_{B,2}} \times \mathrm{L_{B,3}}$ and $\mathrm{N}_1 \times \mathrm{N}_2 \times \mathrm{N}_3$ points is performed by approximating the solution of eq. (2.63) with a Fourier series

$$u_i(\boldsymbol{x}) = \sum_k \mathrm{e}^{i\boldsymbol{k}\cdot\boldsymbol{x}} C_{ij}(\boldsymbol{\kappa}) n_j(\boldsymbol{\kappa}), \tag{2.72}$$

where the sum is performed over the wave vectors $\kappa_i = m2\pi/\mathrm{L_{B,}}_i$ along $-\mathrm{N}_i/2 \leq m \leq \mathrm{N}_i/2$, $n_j(\boldsymbol{\kappa})$ are independen random complex variables with unit variance and C_{ij} are coefficients that Mann estimated to be

$$C_{ij}(\boldsymbol{\kappa}) = \frac{(2\pi)^{3/2}}{\sqrt{\mathrm{Vol}}} A_{ij}(\boldsymbol{\kappa}), \tag{2.73}$$

where "Vol" is the domain volume. A_{ij} is computed from the inversion of the spectral tensor since $A^*_{ij}A_{ij} = \Phi_{ij}$. By comparing to eq. (2.71) it can be deduced that for $\mathrm{d}Z_i^{iso}$,

$$A(\boldsymbol{\kappa}) = \sqrt{\frac{E(\kappa)}{4\pi\kappa^4}} \begin{pmatrix} 0 & \kappa_3 & -\kappa_2 \\ -\kappa_3 & 0 & \kappa_1 \\ \kappa_2 & -\kappa_1 & 0 \end{pmatrix}. \tag{2.74}$$

The above equation is the last component needed to close the algorithm: $\mathrm{d}Z_i^{iso}(\boldsymbol{\kappa}_0)$ is calculated using eq. (2.73) and eq. (2.74) while the effect of shearing is accounted by the matrix multiplication in eq. (2.68) to obtain $\mathrm{d}Z_i(\boldsymbol{\kappa})$. The ensuing product is multiplied with n_i, which has to be created from a random generator (with a unit variance and a Gaussian distribution). Finally, a FFT of the result yields the desired $\boldsymbol{u(x)}$ of eq. (2.72).

Three issues about the resulting turbulence are pointed out in Mann (1998):

a. If the dimensions of the domain are not much larger than L, C_{ij} cannot be estimated with eq. (2.73). This problem is solved by *a)* a different expression of C_{ij} (provided also by Mann) or *b)* assuring that any side length of the domain is at least $\mathrm{L_{B,}}_i \gtrsim 8L$ (which always occurs in the simulations of this work).

b. The simulated velocity field is periodic in all directions. This produces undesired effects, such as the growth of the coherence for separations larger than $\mathrm{L}_{\mathrm{B},i}/2$. The solution proposed by Mann is to use a larger spatial window, achived by doubling the crosswise dimensions of the domain and using only the box $\mathrm{L}_{\mathrm{B},1} \times \mathrm{L}_{\mathrm{B},2}/2 \times \mathrm{L}_{\mathrm{B},3}/2$ for the desired purpose. Such approach is followed in this work.

c. Aliasing is presented in the spectrum of the turbulence field. This is due to the unavoidable averaging of velocities at high wavenumbers over the volumes $\Delta\mathrm{L}_{\mathrm{B},i} = \mathrm{L}_{\mathrm{B},i}/\mathrm{N}_i$ of the discretized domain. To alleviate this problem, Mann (1998) provides a different expression to eq. (2.73) that increases the spectral density at high wavenumbers.

In addition, it should be noticed that although the algorithm of Mann is in principle capable of generating incompressible turbulence, this is not achived in discretized domains, for the same reasons stated in c. Gilling (2009) included a correction for this in his implementation of the Mann technique. The model implementation produced for the present work does not include this correction, and neither the one suggested for point c above. This is justified by the fact that zero divergence is enforced by the LES solver once the turbulence enters the computational domain. Furthermore, the turbulence created with this implementation has similar second-order statistics than those computed from the turbulence created with the generator used in EllipSys3D, which consists also of an implementation of the Mann model[22], used for the comparisons shown in Chapters 3 and 4. The spectral comparison of turbulence created with these generators is discussed in the next section.

The calculations performed by Mann result in a model with three adjustable parameters: 1) the factor $\alpha\varepsilon^{2/3}$ and 2) the turbulence scale L (both adjusted through the von Kármán energy spectrum) that control the intensity of the fluctuations and the size of the eddies, respectively and 3) the anisotropy factor Γ, that controls the effect of the linear shear to model the boundary flow. Mann (1998) estimated the values of these parameters by making a least-squares fit of the spectral tensor to the analytic one-point spectral forms deduced from diverse measurements of

[22] As the turbulence in EllipSys3D is created without access to the source code of the generator (called *windsimu*), it was not possible to verify if any of the corrections mentioned here were implemented.

ABL turbulence. The agreement is in general good, although differences can be seen particularly in the wavenumber or intensity of the maxima (specially for the spectra of the crosswise velocity components).

2.5.2 Validation of implementation for ABL and homogeneous turbulence

The model of Mann described above has been implemented in this work, based on the publicly available code developed by Perrone (2015). To validate our implementation we proceed in two parts: the first validation is performed for a generated ABL turbulence field, while a second one is carried out for a generated homogeneous-isotropic field. Since the turbulence generated with our implementation is used for the OpenFOAM/EllipSys3D comparisons of Chapters 3 and 4, the results of the validation are also compared with those computed from the generator used in EllipSys3D to evaluate the consistency of the inflow conditions.

The first validation is based on a procedure used by Mann (1998), where one-point velocity spectra computed from a generated ABL turbulence field are compared to the analytical forms of the spectra estimated by Kaimal (Kaimal and Finnigan, 1994) from experimental data of a neutrally stable atmosphere over flat terrain. The expressions are:

$$\frac{\kappa_1 F_1(\kappa_1)}{u_*^2} = \frac{52.5\kappa_1 y}{(1+33\kappa_1 y)^{5/3}}, \tag{2.75}$$

$$\frac{\kappa_1 F_2(\kappa_1)}{u_*^2} = \frac{8.5\kappa_1 y}{(1+9.5\kappa_1 y)^{5/3}} \tag{2.76}$$

and

$$\frac{\kappa_1 F_3(\kappa_1)}{u_*^2} = \frac{1.05\kappa_1 y}{1+5.3(\kappa_1 y)^{5/3}} \tag{2.77}$$

where u_* is the friction velocity and y is the height of the measurement. Mann (1998) estimated that the parameters of the generator that would reproduce the Kaimal spectra were

$$\begin{aligned} \Gamma &= 3.9 \\ L &= 0.59y \\ \alpha\varepsilon^{2/3} &= 3.2\frac{u_*^2}{y^{2/3}}. \end{aligned} \tag{2.78}$$

Turbulence is generated using these parameters as well as $u_* = 1.78$ m/s and a roughness length of $y_0 = 0.0054$ m with the logarithmic mean velocity profile $U(z) = \frac{u_*}{\kappa^*}\ln\left(\frac{y}{y_0}\right)$, where κ^* is the von Kármán constant taken as 0.40. The domain used consists of $\mathrm{L}_{\mathrm{B},x} \times \mathrm{L}_{\mathrm{B},y} \times \mathrm{L}_{\mathrm{B},z} = 1600\,\mathrm{m} \times 400\,\mathrm{m} \times 400\,\mathrm{m}$ containing $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_y = 1024 \times 256 \times 256$ points. Note that for the transversal size of the domain $\mathrm{L}_{\mathrm{B},(y,z)} \simeq 17L$, fulfiling the condition stated before regarding this ratio (a. of the list in page 61). An example of the generated velocity field is shown in Figure 2.7, where the ABL turbulence is shown next to the homogeneous field created for the second validation.

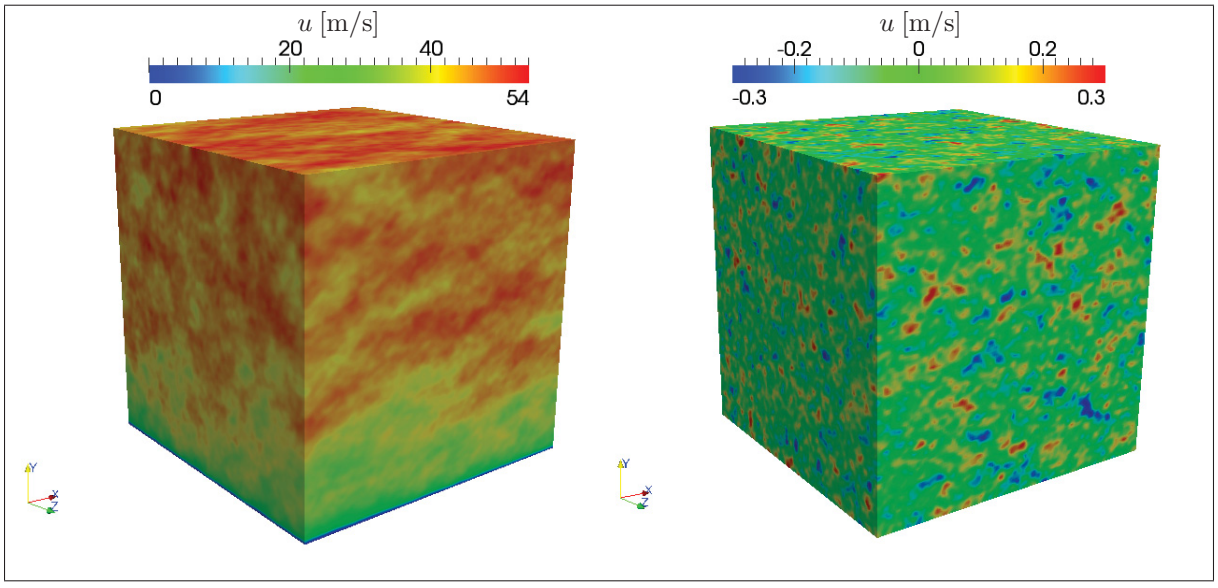

Figure 2.7 Turbulence velocity fields created for the validation procedures. *Left:* ABL flow. *Right:* homogeneous flow (256^3 points).

One-point spectra of every velocity component are computed in the streamwise direction at a height of $y = 40$ m[23] for every $z-$position in the spanwise plane and later averaged. Results are shown in Figure 2.8, compared to the eqs. (2.75)-(2.77) as well as the one-point spectra obtained from a field created with generator used in EllipSys3D using the same parameters. The comparison displays a good match between our results from the Mann model and the analytic expressions of the Kaimal spectrum. Yet, the maxima of the spectra are slightly off for the v, w components, as well as the intensity of the latter, both features can be also observed in results obtained by Mann (1998) for the same comparison Likewise, the aliasing effect observed in the curves is due to the domain discretization and the resulting absence of fluctuations at high wavenumbers, as previously discussed. Notably, the spectra obtained from our implementation of the Mann model resembles very well those obtained with the generator of EllipSys3D, which assures the consistency of the turbulence fields to be used as inflow in the comparisons of OpenFOAM/EllipSys3D.

The second validation consists in the comparison of the one-point spectra from a homogeneous-isotropic turbulence field. For this, we follow a procedure analogous to that used by Bechmann (2006), where a turbulence field is created to reproduce the one-point velocity spectrum obtained from the experiments of Comte-Bellot and Corrsin (1971) of decaying isotropic turbulence produced by a grid in a wind tunnel. Moreover, these results are also compared to the analytic expression for the one-point longitudinal spectrum eq. (2.54) and the transversal spectrum eq. (2.55). By comparing with the spectrum reported from the experiments, Bechmann found the input parameters of the Mann algorithm that produce the best fit with the spectrum of the computed velocity field. These are:

$$\begin{aligned} \alpha &= 1.7 \\ \varepsilon &= 0.3 \text{ m}^2/\text{s}^3 \\ L &= 0.03 \text{ m}, \end{aligned} \tag{2.79}$$

[23]The values are taken from the closest available position to this height in the domain, as no interpolation is used.

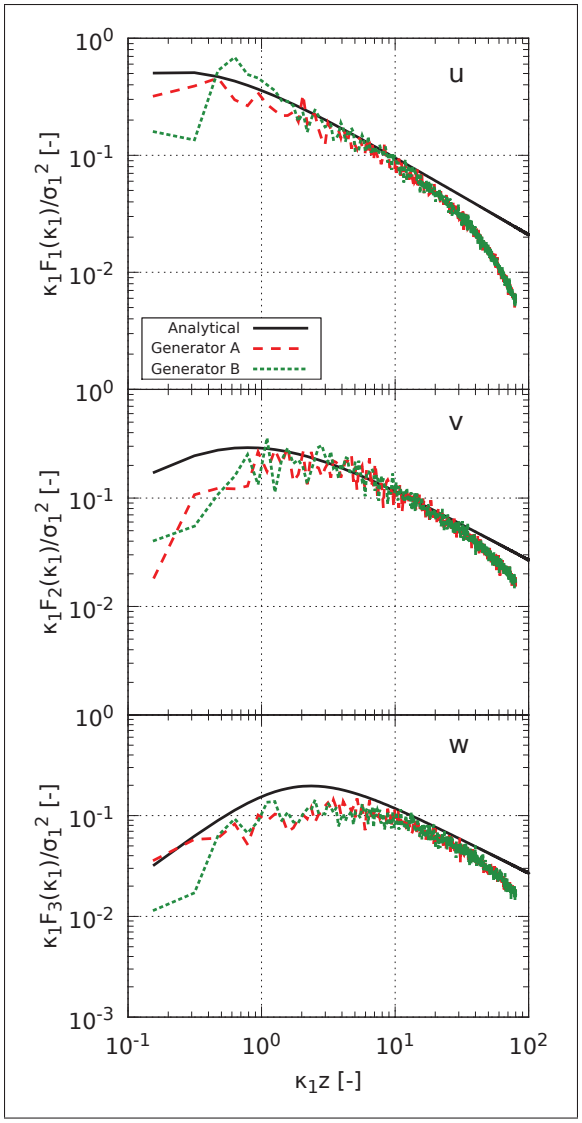

Figure 2.8 Comparison of the one-point spectra obtained from ABL turbulence generated with the Mann model with the Kaimal spectra. Generator A refers to the results obtained from the Mann model implementation produced for this work while Generator B corresponds to results from the generator used in EllipSys3D.

for the measured spectrum at the position $U_0 t/M = 42$ (relative to the grid), where $U_0 = 10$ m/s is the inlet velocity and $M = 0.0508$ m is the grid spacing. In addition, Bechmann calculated the total turbulence kinetic energy k_{tot} of the experiment by integrating the longitudinal spectrum over all the wavenumber range. This value is used to compare with the one obtained from the simulated turbulence field (Table 2.1).

Using the input values of eq. (2.79), we compute a turbulent velocity field with our implementation of the Mann algorithm (with Γ set to zero to simulate non-sheared turbulence). As in the study of Bechmann, the dimensions of the computational domain are $\mathrm{L}_{\mathrm{B},x} \times \mathrm{L}_{\mathrm{B},y} \times \mathrm{L}_{\mathrm{B},z} = 1\,\mathrm{m} \times 1\,\mathrm{m} \times 1\,\mathrm{m}$. This dimensions are chosen according to the extension of the largest, most energetic eddies, with $\kappa_1 = 10\ \mathrm{m}^{-1}$ which corresponds to a lengthscale of $L = 2\pi/\kappa_{max} = 0.63$ m. Three different grids have been used, with a number of cells equal to 64, 128 and 256 per side, with a corresponding cutoff wave number (assuming the Nyquist theorem) of $\kappa_c = \pi/\Delta = 201.06\ \mathrm{m}^{-1}$, $402.12\ \mathrm{m}^{-1}$ and $804.25\ \mathrm{m}^{-1}$, respectively.

In Figure 2.9 we can observe the comparison of the longitudinal and transversal velocity spectra obtained from the turbulence generator compared to that of the experiments (only available for the $u-$spectrum) and to the analytical expressions. The spectra shown comprise those obtained from the grids with 64^3, 128^3 and 256^3 points, as indicated. Each curve represents the average of all the spectra obtained in the longitudinal direction. We can observe that in every case, the modelled velocity field reproduces well the spectral decay obtained from the measurements, although separating from the analytic expressions and experimental results as it reaches the cutoff wavenumber, as expected. We should make note that the Reynolds number used in Comte-Bellot and Corrsin (1971) is not sufficiently high ($\mathrm{Re}_\lambda = 72$) to allow the appearance of an extended inertial range, so the cutoff wavenumber of the experiment cannot clearly be established.

In Table 2.1 some characteristics of the simulated turbulence are compared to the experimental results of the work of Comte-Bellot and Corrsin. The results shown for every grid represent domain-averaged statistics. We can see that while the r.m.s. values remain practically un-

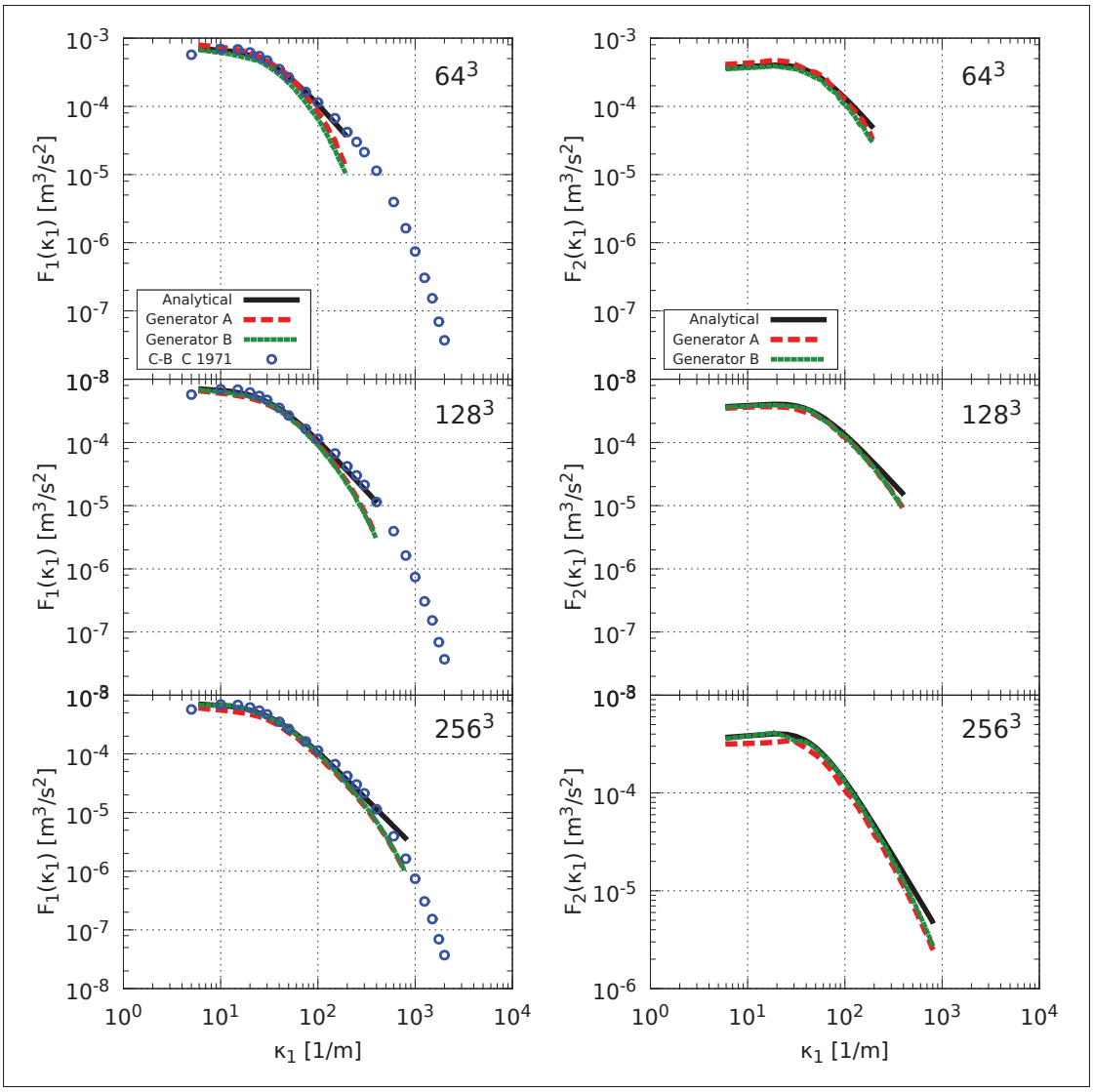

Figure 2.9 Comparison of the one-point spectra obtained from turbulence simulated with the Mann model with the von Kárman spectra. Generator A refers to the results obtained from the Mann model implementation produced for this work while Generator B corresponds to results from the generator used in EllipSys3D. *Left:* longitudinal spectra (that compares also with the spectrum obtained from the experiments of Comte-Bellot and Corrsin, 1971). *Right:* transversal spectra. Computations performed over domains with 64^3 points (*top*), 128^3 points (*middle*) and 256^3 points (*bottom*).

changed for the three cases, they are smaller than the one reported from the experiments (this can be improved by increasing the value of ε in 2.79). The turbulent kinetic energy shown in the third column is calculated considering v_{rms} and w_{rms}. The assumption of isotropy is veri-

fied in our computations as we observe very little variation among the r.m.s. of each velocity component. For example, for the case of 256^3, the averaged r.m.s. values in every direction are $u_{rms} = 0.1949$, $v_{rms} = 0.1923$ and $w_{rms} = 0.1924$. A similar comparison is observed for the 64^3 and 128^3 boxes.

Table 2.1 Comparison of the r.m.s., turbulence kinetic energy and integral lengthscale computed from the synthetic turbulence field, using three box resolutions, with the experiments of Comte-Bellot and Corrsin (1971).

	u_{rms} [m/s]	k_{tot} [m^2/s^2]	$L_1(\kappa_1)$ [m]
64^3	0.196	0.0576	0.0308
128^3	0.195	0.0572	0.0267
256^3	0.195	0.0570	0.0240
Exp.	0.222	0.0687	0.0240

In the last columns of Table 2.1 we compare the longitudinal integral lengthscale (eq. 2.46). This is calculated from the first zero-crossing of the autocorrelation curve of u for each line in the x-direction, which is later volume averaged. Unlike the case of u_{rms}, we observe an improvement in the comparison with the experimental value for higher grid refinements. The fact that a match is found only in the case of the finest grid is an obvious indication of the extent of refinement to model the eddies. In effect, for the grids used, and taking the integral lengthscale reported in the experiments as a reference, a resolution of $L_1/\Delta \sim 1.5$, 3 and 6 (cells per L_1) is being used for each case. Although one cannot conclude solely from this result that the finest resolution is needed to model the eddies, we take into account the fact that L_1 is slightly overestimated for two coarsest resolutions when modelling the turbulence field for our LES computations. Additionally, it is important to note that the relationship $L_1 = 0.816L$ of eq. (2.56) holds for the integral lengthscale measured experimentally, using the assumed input value for the turbulence lengthscale of $L = 0.03$ m employed in the Mann algorithm.

The turbulence generated with our implementation has been proved capable of reproducing the expected spectral behaviour of ABL and isotropic turbulence, concluding the validation process of the model implementation used to generate the inflow conditions of our LES computations. Although only HIT turbulence is to be used for that purpose, ABL turbulence was also tested for sake of completeness. The estimation of the adequate parameters of the inflow turbulence for the LES will be discussed in the remaining Chapters of this work.

2.6 Adequate resolution of LES

We conclude the Chapter with a discussion of the resolution in LES simulations. These arguments are relevant in the situation when the size of the turbulence eddies, measured through the integral lengthscale L_i, is small enough to consider if they are adequatly represented by the grid. This applies to the simulations of the last two Chapters of this work.

The ability to estimate the characteristics of the turbulence field (such as the integral and Taylor scales) will depend in the accuracy to determine the two-point correlations (eq. 2.45) of the velocity field. This in turn, depends on the accuracy of the flow solution which as in any LES computation, relies on two factors: 1) the precision of the SGS model to estimate and to represent the effect of the dissipative scales and 2) a mesh refinement capable of reproducing the range of fluctuations from the largest scales down to the cutoff filter scale (providing this is set appropriately). A third factor is comprised by the method to generate turbulence and its capability to render the desired turbulence features.

Unlike RANS models, the accuracy of the LES model is inherently subjected to the grid used for the computations (there is no mesh independent solutions in LES). Freitag and Klein (2006) affirm that in fact, LES with an implicit filter does not represent the solution to a set of differential equations because the SGS models depends on the grid. Some works (Geurts and Fröhlich, 2002; Celik *et al.*, 2005) have suggested mechanisms to assess the accuracy of the LES computation through the estimation of a "quality" parameter. Similarly, others (Klein, 2005; Freitag and Klein, 2006) have presented procedures that attempt to make a distinction between

the model error ϵ_m, due to the SGS part, and the numerical error ϵ_n, due to discretization schemes. Most of these works comprise the execution of various computations, with different mesh refinements, much in the spirit of the Richardson extrapolation methods.

In this work we follow a more straightforward approach, as it is out of its scope to explicitly evaluate the error in the LES computations. Instead, as the lengthscale parameters of the flow we wish to model are known, we utilize a grid refinement that in principle should be enough to reproduce the flow characteristics. Specifically, we attempt to reproduce the turbulence structure by means of representing L_1, assuming its value is well resolved and accurate. If L_1 is represented in the flow, then the scales of fluctuation that are predominant in the dynamics of the flow will be resolved. If this argument is accepted for now, the question of resolution is reduced to determine the adequate number of cells to represent L_1. Clearly, to opt for such criterion carries the disadvantage of cutting short the energy cascade, which can affect the accuracy of the lengthscale reproduction. The consequences of this choice are studied in this work.

As for the adequate resolution of the integral scales, some insight is provided by Pope (2000) as he showed that for a high-Re HIT, if a sharp cutoff filter $\kappa_c = \pi/\Delta_f$ is used, a filter width of $\Delta_f \approx 1.16 L_1$ yields 80% of k to be within the resolved fluctuations of the LES. The mentioning of this figure in Pope (2004) was interpreted by diverse authors (see Davidson, 2009) as a suggestion of a criterion to determine a well-resolved LES. Assuming this, Davidson (2009) has remarked that neither this value nor the often reported observation of the -5/3 slope in the scaling range of the energy spectra are reliable estimators for the quality of the LES, providing examples using channel flow computations. Instead, he recommended a verification through the comparison of two-point correlations and a resolution of at least 8 cells for the largest scales. The latter assertion is supported by Celik *et al.* (2005) as they calculated that the integral lengthscale should be resolved using 8 cells (taking the average of the required resolutions for sharp cutoff and Gaussian filters $L/\Delta \approx 12, 17$, respectively, and assuming $L_1 \simeq 0.55L$). In the same work, it is also estimated that the adequate resolution in terms of the Kolmogorov lengthscale should be $\Delta/\eta \approx 25$ for high Re ($\mathrm{Re}_\lambda \simeq 155$) and $\Delta/\eta \approx 9$ for low Re ($\mathrm{Re}_\lambda \simeq 78$).

Spalart (2001) mentions that the wavelength threshold of resolved eddies is "perhaps" $\ell = 5\Delta$ although no calculations are provided. This value was used by Gilling and Sørensen (2011) as a measure of minimum resolution for the convection of eddies.

For relatively large domains ($\mathrm{L}_x \gg L_1$) it would be very computationally demanding to follow the above mentioned requirement of $L_1/\Delta = 8$ to simulate a decaying isotropic turbulent field. Precisely, as the lengthscale increases downstream from the turbulence grid (eq. 2.61), if one wishes to simulate the evolution of the turbulence field from such position, the value of L_1 at that grid has to be appreciably smaller than the one wished to be reproduced at some downstream location. Because of these constrains, the determination of the proper resolution is posed simply in terms of what is physically realizable. This is, considering the representation of the lengthscale in the wavenumber space, to find the minimum number of cells to represent a wavelength ℓ. Being aware of its limitations, the effectivity to reproduce the turbulence structures under these conditions will be evaluated and compared with the measurements. Indeed, as pointed by Fletcher (1991), even though wavelengths can be represented in the discretized space down to the minimum resolution of $\ell = 2\Delta$, the accuracy to estimate the amplitude of their derivatives diminishes at low resolution (although this can be slightly alleviated by the use of higher order—than central—schemes, as he indicates). A similar observation was also made by Spalart (2001).

CHAPTER 3

INFLOW GENERATION AND ASSESSMENT OF DECAYING TURBULENCE CHARACTERISTICS

The generation of the turbulence field used as inflow for the wake simulations is treated in this chapter. Based on the generation method presented in Chapter 2, we present the procedure devised to introduce a turbulence field that attains the desired characteristics at a given position in the computational domain. In this way, measurements obtained from decaying-HIT created in a wind tunnel are reproduced with LES in OpenFOAM. Similarly, the experimental values are also reproduced with EllipSys3D, which results are used as a benchmark for comparison. Diverse turbulence features are computed with the goal of assessing the capability of reproducing the characteristics of a turbulent flow, particularly, with respect to the limited resolution of the turbulence lengthscales. The flows discussed in this chapter are used as inflow in the wake simulations examined in the next Chapter.

3.1 Experimental setup and measurement campaigns

The experimental data used in this work were obtained at the Eiffel-type wind tunnel of the Prisme laboratory of the University of Orléans. These come from two separate experimental campaings. Most of the data employed come from the first campaing while data of the second one are only used to complement some parts where measurements from the first one cannot be used.

For the first campaing, experiments are credited to G. Espana and S. Aubrun. Complete details about the experimental setup, the measurement techniques as well as the characteristics of the flows generated by this wind tunnel can be found in Espana (2009) and Espana *et al.* (2012). The second campaing is described by Thacker *et al.* (2010). Only an overview of the procedure and the available data is provided here.

The test section of the wind tunnel has a width and a height of 0.5 m and a length of 2 m. Two different grids were used to generate turbulence at the entrance of the test section, resulting in two different turbulence intensities. At a distance of $x = 0.5$ m from that grid, the reported reference values of streamwise turbulence intensity and integral lengthscale (measured at the centreline) were $\mathrm{TI} = 3\%$ and $L_1 = 0.01$ m as well as $\mathrm{TI} = 12\%$ and $L_1 = 0.03$ m. These cases are identified henceforward as Ti3 and Ti12, respectively. The streamwise position where the values are reported is referred to as the *target position* x_D.

The measurement campaigns include experiments performed with wind turbine models located also at $x = 0.5$ m downstream from the turbulence grids. These consisted in disks made of a metallic mesh to simulate the effect of the AD model (a porous surface) on the flow. Two disks were used, each with a diameter of $D = 0.1$ m but made with a different wire to produce different induction factors. The thrust coefficient C_T of each disk is calculated following the procedure presented by Aubrun *et al.* (2007) and revisited by Sumner *et al.* (2013), based on the measurement of the velocity deficit in the wake. In total, the measurement campaign comprises six experimental cases, summarized in Table 3.1. The reproduction of the measurements made in the wakes of the disk models with OpenFOAM and EllipSys3D is the subject of Chapter 4. Yet, some measurements made outside the wake of such experiments are used to complement the cases without the disks, as described below. Make note that because of its practicality and to maintain the consistency with Chapter 4, longitudinal distances from the turbulence grid (or from the inlet in the LES) are given in diameters D of the disks.

Table 3.1 Reference parameters of flow and disks used in the experiments.

TI [%]	$\boldsymbol{L_1}$ **[m]**	**Case**
3	0.01	No-disk
		$C_T = 0.42$
		$C_T = 0.62$
12	0.03	No-disk
		$C_T = 0.45$
		$C_T = 0.71$

The data used in this work were obtained using two different techniques:. Firstly, with the aim of obtaining time-series of the flow velocity, a Hot-Wire Anemometry (HWA) probe was located along vertical lines at $x = 3D$, $4D$ and $6D$ from the disk center (the origin of the reference system $x, y, z = 0, 0, 0$ is set there). The probe moved along each vertical line between $0 \leq y/D \leq 1.5$ registering data in steps of $0.1D$, with extra steps at $y/D = 0.35$ and $y/D = 0.65$. Additionally, steps of $0.02D$ were used between $0.4 \leq y/D \leq 0.6$. A scheme of the measuring locations with respect to the experimental arrangement is shown in Figure 3.1. At each probe position, data was acquired with a sampling frequency of $f_{acq} = 2$ kHz during about 1 min. A low-pass filter was also used, with a cut-off frequency fixed at $f_c = 1$ kHz. The reference velocity during the measurements was $U_\infty = 3$ m/s. Of the measurements made with this technique, only the database corresponding to the cases of Ti12 is used in our comparisons as the sampling rate was assessed to be too low in the Ti3 case. Due to this, HWA measurements from Thacker *et al.* (2010)—identified above as the second experimental campaing—are used to complement the experimental data for the comparison of the Ti3 case. These were made using the same experimental setup as the other HWA measurements, with $\mathrm{TI} \simeq 3\%$ also at the target position. However, the mean inflow velocity was set to 20 m/s so the Reynolds number is noticeably higher, leading to higher dissipation occurring at smaller scales, so these last features will not be compared with our LES results.

Secondly, a Laser-Doppler-Anemometer (LDA) was used to simultaneously measure two components of velocity (u, v). Measurements without the disks were made only at $x = 0$ for the Ti3 and at $x = 1D$ for the Ti12 case. The recording positions were aligned in the vertical direction. Measurements were performed in steps of $0.1D$ between $-1.5 \leq y/D \leq 1.5$ with extra steps at $y/D = \pm 0.45, \pm 0.55, \pm 1.1, \pm 1.3$ and ± 1.5 for the Ti12 cases. For the Ti3 cases, the positions in the vertical direction where data is available vary slightly, but most of them are made in steps of $0.1D$ between $-1.0 \leq y/D \leq 1.0$ with extra steps of $0.02D$ between $0.4 \leq y/D \leq 0.6$. Measurements behind the disks are made along the vertical directions (at the same y/D stations) at $x = 2D$, $4D$, $6D$, $8D$ and $10D$ from the disk center. To supplement the single longitudinal recording set available for each of the no-disk cases, measurements

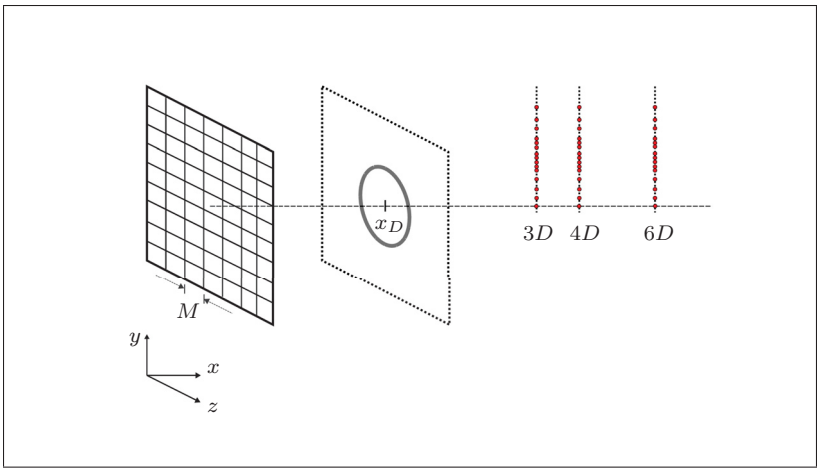

Figure 3.1 Representation of the measurement positions of the hot-wire. The turbulence is generated by a grid of spacing M. The reported values of TI are measured at $x = 0.5$ m from such grid, where the ADs are subsequently located. This position is referred to as x_D. Time-series of the velocity are recorded at various positions along vertical lines at $3D$, $4D$ and $6D$.

made outside the wake of the lowest C_T disks are considered. For that purpose, the values employed correspond to the average of the two farthest recordings from the disk axis: $y = \pm 1D$ for Ti3 and $y = \pm 1.5D$ for Ti12. Measurements were made using a non-uniform sampling frequency, with an average of 1 KHz during 90 s. The reference velocity was $U_\infty = 6$ m/s for the cases Ti12 and $U_\infty = 10$ m/s for Ti3. As it was shown by Comte-Bellot and Corrsin (1966) and later work, various estimations in grid generated turbulence can be considered Reynolds independent (but no observations such as the scaling region of the spectrum, as shown by Mydlarski and Warhaft, 1996). Therefore, non-dimensional results of mean velocities and r.m.s. statistics obtained with the LDA technique will be used despite the differences in reference velocity.

3.2 Numerical setup

In this section we provide a description of the setup employed for the simulations in each platform. It is recalled that as it is not within the scope of this work to provide a comprehensive

comparison of the numerical performance of these two codes, no modifications have been made with the aim of approaching the implementations of each platform.

3.2.1 Computational domain and grid resolution

The dimensions of the computational domain are set to imitate those of the measuring region in the wind tunnel. Due to the differences between the codes regarding the procedure to introduce the turbulence into the computational domain (Sec. 3.2.2), the lengths of the domain vary slightly. The domain and grid sizes of the LES computations as well as of the synthetic velocity field, identified as *turbulence box* are listed in Table 3.2. In OpenFOAM, the dimensions of the computational domain are set equal to those of the measuring region in the wind tunnel, while in EllipSys3D the domain length is slightly longer. The extra length comes from the fact that turbulence is introduced downstream from the inlet, due to the technique implemented in this code. With this, the longitudinal extension—measured from the plane where turbulence is introduced to the outlet—is the same in both codes. As in the experiments, the origin of the coordinate system for the computations is at the center of the spanwise $(y-z)$ plane, at $5D$ from the inlet in OpenFOAM and at $7.5D$ from the inlet used in EllipSys3D. Likewise, this position is labeled x_D. The reason to imitate the dimensions of the experiment, in particular in the crosswise directions, is to reproduce the potential effects of blockage on the wake development. A small blockage of 1.3% in average has been reported for measurements in this wind tunnel (Sumner *et al.*, 2013).

Due to the choices of domain size, a domain independence procedure is not performed. The election of the grid, on the other hand, is closely related to the adequate resolution of the turbulence scales. Consequently, the grid size is determined by the optimum number of cells per L or rather, L_1. Unlike the ABL, where L_1 is typically of two to three times the diameter of the rotor, the turbulence grids used in the wind tunnel produce turbulence with an eddy size approximately ten to three times smaller—at x_D—than the diameter of the AD. Evidently, this imposes a strict demand for the cell resolution, particularly for the turbulence box as the turbulence scale there $L_{1,\mathrm{B}}$ should be even smaller to account for its increase along the flow

Table 3.2 Main parameters of the computational domains of LES and synthetic field (turbulence box). Dimensions of computational domains are given as $\mathrm{L}_x \times \mathrm{L}_y \times \mathrm{L}_z$ with grids containing $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_z$ cells. Synthetic field domains are given as $\mathrm{L}_{\mathrm{B},x} \times \mathrm{L}_{\mathrm{B},y} \times \mathrm{L}_{\mathrm{B},z}$ containing $\mathrm{N}_{\mathrm{B},x} \times \mathrm{N}_{\mathrm{B},y} \times \mathrm{N}_{\mathrm{B},z}$ cells. Lengths measured in $D = 0.1$ m.

		OpenFOAM	**EllipSys3D**
LES domain size		$20D \times 5D \times 5D$	$22.5D \times 5D \times 5D$
Layout		Uniform region $20D \times 3.6D \times 3.6D$	All uniform
Case Ti3	LES domain grid	$1000 \times 208 \times 208$ cells	$1152 \times 256 \times 256$ cells
	Turbulence box	$40D \times 5D \times 5D$	
	Box grid	$2048 \times 256 \times 256$ cells	
Case Ti12	LES domain grid	$500 \times 104 \times 104$ cells	$576 \times 128 \times 128$ cells
	Turbulence box	$80D \times 5D \times 5D$	
	Box grid	$2048 \times 128 \times 128$ cells	

direction (eq. (2.61)). The subject of establishing the adequate resolution is discussed in Sec. 2.6.

On account to these limitations, the determination of the cell resolution of the turbulence boxes is based on what is physically realizable. Although a higher resolution would be preferred, we are restricted by the total number of cells in those domains. Precisely, while the lateral dimensions of the turbulence box are chosen to be equal to those of the computational domain, the length is determined by the recycling period of the box into the computational domain (considering that the synthetic field moves across the LES domain at a constant velocity). Furthermore, a box with twice the length of the desired lateral dimensions must be generated due to the nature of the algorithm: as indicated by Mann (1998), the simulated velocity field is periodic in all directions, so it is recommended to create a turbulence box with cross-flow dimensions twice as big as the desired size and only use one quarter of the simulated field (see b. in page 61). Consequently, the optimum size found for the turbulence boxes is a compromise between: 1) a minimum resolution of $L_1/\Delta = 2$ cells, 2) a minimum length equivalent to at least 2 longitudinal flow-times, abbreviated as LFT (1 LFT is defined as $\mathrm{L}_x/\langle U_\infty \rangle$) and 3) the total number of cells in the box. Considering these arguments, two turbulence boxes were created using the Mann implementation for each TI case (the boxes used in OpenFOAM and

EllipSys3D share the same grid and dimensions). The mesh of the turbulence box for the Ti3 case is finer than in the Ti12 since the sought after integral lengthscale is three times smaller ($L_1 = 0.01$ m vs. 0.03 m at x_D). The parameters of these boxes are listed in Table 3.2. Note that the dimensions of the boxes are set to multiples of 2^n ($n \in \mathbb{N}^+$) due to the Fourier techniques used in the generation algorithm. As the grid size limit of the OpenFOAM installation in our cluster has been found to be $\sim 180 \times 10^6$ cells[1], it is easy to see that a larger mesh than the one used for the Ti3 case (e.g. $2048 \times 512 \times 512$) would have surpassed this ceiling. The turbulence generator has been implemented outside the OpenFOAM framework so when the turbulence is originally generated, with twice as many points in the lateral directions, the cell number is not restricted by this limit. The meshes used for each case are different so as to reduce the computational time while fulfilling the minimum resolution requirement for the two different L_1 values. Since the mesh of the Ti12 case is coarser, it was possible to increase the length of the domain, allowing for a smaller recycling rate of the turbulence box. The values found for the integral lengthscale in the turbulence box $L_{1,\mathrm{B}}$ that produce the desired L_1 at the target position will be shown in the results (Sec. 3.3).

In each TI case, the mesh of the computational domain is set according to the resolution used for the corresponding turbulence box, as it is preferred that the cell size beween the domain and box are approximately equal. If the mesh was coarser, the small fluctuations would be filtered out when introducing the turbulence, rendering the refinement of the turbulence box unnecessary. Conversely, a finer mesh would be advantageous only until small fluctuations developed downstream in the flow. Yet, this would come at a higher computational expense while the priority is to reproduce the large scales. Nonetheless, finer cells in the synthetic turbulence domain would lessen the need of velocity interpolations between available planes at the inlet or turbulence plane, minimizing the loss of TI. A short study about the influence of mesh refinement of the synthetic turbulence and compuational domain on the reproduction of turbulence characteristics in the LES is shown in the Appendix I.

[1]This limit seems to be associated to the floating-point precision used to store the grid data, but its origin has not been exactly determined.

Due to the nature of the mesh generators used for each code, the domains have been defined with some differences. But these exist well outside the measuring regions and they should not represent a source of discrepancy for the comparisons. In the LES domain of OpenFOAM, unlike the turbulence box, the grid is not completely uniform across the domain. Instead, only a central region of $20D \times 3.6D \times 3.6D$ of uniform (cubic) cells is defined. This region is needed to assure a consistent filtering for the SGS scales, as implicit filtering is used in the LES. In addition, the uniformly distributed cells should comprise all the positions of measurement, which includes those made up to $y = 1.5D$ from the centreline. Outside the uniform grid region, the cells are stretched towards the lateral boundaries with an aspect ratio $\Delta z_{max}/\Delta z_{min} = \Delta y_{max}/\Delta y_{min} = 4$, where $\Delta z_{min} = \Delta y_{min} = \Delta x = \Delta$ is the cell side length in the uniform region. This central region has approximately the same cell size as in the corresponding turbulence boxes. The slight differences arise from the purpose of accommodating an integer number of cells along the diameter of the AD in OpenFOAM. The central region of each mesh consists of uniform cells with a side length Δ of 0.002 m (Ti3) and 0.004 m (Ti12). Hence, the cell resolution of the integral scales L_1/Δ at the measured location of $x = 0.5$ m from the inlet corresponds to 5 and 7.5 cells, respectively. On the other hand, the speed of the solver in EllipSys3D is greatly reduced if the number of cells in each direction of the domain does not consist of a number of 2^n cells. In light of this constrain, a fully uniform grid is employed in this code, with a cell size identical to the one of the turbulence boxes. As a result, the cell size used in each code are only approximately equal. The cell side lengths used in OpenFOAM (within the uniform region) are 2.4% larger than those of EllipSys3D, in both TI cases.

3.2.2 Generation of turbulent inflow, introduction into the computational domain and boundary conditions

Following the methodology presented in Sec. 2.5, we describe here the procedure to create the synthetic turbulence field to be used as an inflow in the LES computations. In the homogeneous case, the use of the Mann method requires to adjust two parameters to produce the turbulence with the demanded characteristics: the lengthscale L and TI. The latter is normally controlled

by means of varying ε (or $\alpha\varepsilon^{2/3}$) in the energy spectrum eq. (2.53) until the desired TI is achieved. As it is not straightforward to give an exact relationship between ε and the generated TI, a more practical procedure is followed in this work. As suggested by Larsen (2013), instead of changing ε, a scaling factor $SF = \sqrt{\frac{\sigma^2_{target}}{\sigma^2}}$, is used, where σ^2_{target} is the desired average variance of the turbulence and σ^2 is the variance of the turbulence field in each direction. In this way, the desired TI can be obtained by multiplying SF by each velocity component of the turbulence box. It is expected that when the HIT field is convected at a uniform velocity, the TI will decay monotonically in the streamwise direction. To estimate the turbulence intensity value that the box ($\mathrm{TI_B}$) should have in order to attain the desired TI at the given position, we can use eq. (2.58). However, since this equation does not provide an exact value (only approximate values of A and n are given in the literature), $\mathrm{TI_B}$ cannot be calculated with precision. Furthermore, as it will be seen in the results, the averaged value of TI measured next to the position where turbulence is introduced in the computational domain does not correspond to $\mathrm{TI_B}$. In consequence, some testing was necessary to find the right value. Likewise, the value of $L_{1,\mathrm{B}}$ can in principle be predicted using eq. (2.61) but for the same reasons, tests were necessary to determine its magnitude. The values found for $\mathrm{TI_B}$ and $L_{1,\mathrm{B}}$ are presented in the results section.

Boundary conditions are set to replicate the conditions in the wind tunnel. Thus, slip conditions are used for the lateral boundaries, whereas the outlet is set using a Neumann boundary condition. In OpenFOAM, the synthetic turbulent flow is introduced in the domain at the inlet using the procedure described below, whereas in Ellipsys3D, a constant velocity is set at the inlet so the fluctuations are introduced at a plane downstream of the inlet. As mentioned before, when the disks are introduced, these are located at $x = 0.5$ m from the inlet in OpenFOAM and at $x = 0.75$ m in Ellipsys3D, at the center of the $y - z$ plane. Assuming the Taylor hypothesis of frozen turbulence for the spatial and temporal correlations, the streamwise axis of the turbulence box is assumed equivalent to time.

Different approaches are used to introduce the turbulence into the computational domain in each code:

In OpenFOAM, cross-sectional planes of the turbulence box are taken for every available longitudinal position and their velocity values are mapped onto the inlet of the computational domain. As the crosswise locations of the cell centres of the synthetic turbulence do not exactly correspond to those of the computational domain, linear interpolations are used to evaluate the velocity values at the required positions. Different Courant-Friedrichs-Lewy (CFL) conditions are used in each case. For the Ti3, CFL ≈ 0.8 while for the Ti12, CFL ≈ 0.5. These are the maximum CFL values over the whole computational domain, which are attained next to the inlet, where the velocity fluctuation is the highest. In comparison, their domain-averaged values are ≈ 0.3 and ≈ 0.1, respectively. The time-steps used in the computations are $\Delta t = 2 \times 10^{-4}$ s and $\Delta t = 1.2 \times 10^{-4}$ s for each case. Linear interpolations are used in the streamwise direction (i.e. between planes of the turbulence box) to compute the required velocity values at the given time.

In EllipSys3D, instead of introducing the turbulence velocity field directly at the inlet, the turbulence is introduced at a plane downstream from it, called Turbulence Plane (TP). Moreover, the procedure comprises the computation of forces (momentum sources) that will induce the fluctuations of the pre-generated turbulence into a uniform flow set at the inlet. Such forces are gaussianly distributed in the x-direction along the domain. The corresponding standard deviation is $\sigma = 0.2D/\sqrt{2}$ so the extension of the TP is constant in absolute dimensions, independently of the cell size. The longitudinal thickness of the TP is equivalent to 21.72Δ in the Ti3 case and 10.86Δ for Ti12. A complete description about the procedure is provided by Troldborg (2008). The time steps used in each computation are equal to those set in in OpenFOAM. As in OpenFOAM, linear interpolations are used to calculate the required turbulence velocity values between the available planes.

While it would have been desirable to follow the TP approach of EllipSys3D also in OpenFOAM, this would have required a good amount of time spent on its implementation. Therefore, for convenience, in OpenFOAM we applied the method to introduce the turbulence at the inlet as it required considerably less time to achieve. Then, the comparison of the results obtained in this way with those of EllipSys3D becomes relevant to illustrate the benefits of a

more sophisticated approach, for this particular purpose. Evidently, the use of different methods will impact the values of the turbulence characteristics in the vicinity of the region where it is introduced. Nontheless, it is remarked that the main interest is in reproducing the values of TI and L_1 at the target position x_D with each code.

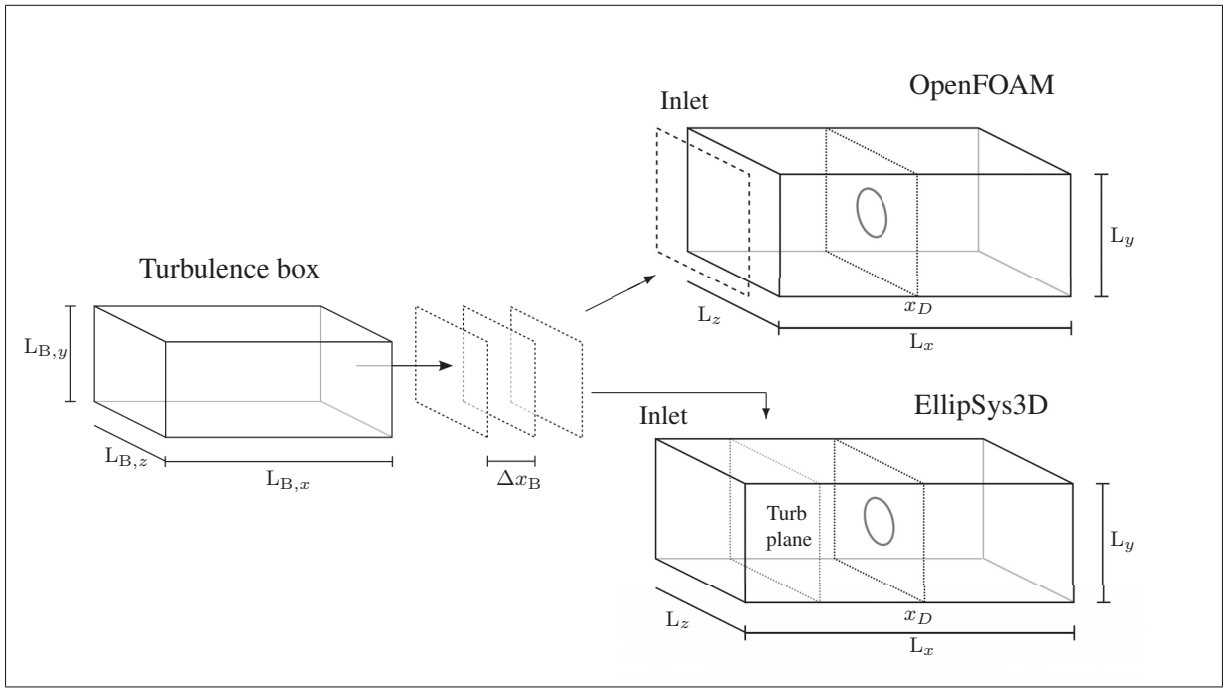

Figure 3.2 Introduction of synthetic turbulence field in OpenFOAM and EllipSys3D.

The introduction of the synthetic turbulence into the computational domain of each code is illustrated in Figure 3.2. There, crosswise planes of the synthetic turbulence field (turbulence box) are introduced into the inlet in OpenFOAM or at the TP in EllipSys3D (only the fluctuating part of the velocity, for the latter), where the velocity at every Δt is interpolated from the available planes (separated by Δx_{B}). The turbulence velocity is introduced in OpenFOAM while only the fluctuating part around $\langle U \rangle$ is introduced at the TP in EllipSys3D, where a constant inflow is fixed at the inlet. The separation between the inlet and x_D is $5D$ in OpenFOAM and $7.5D$ in EllipSys3D. The dimensions of each domain are indicated in Table 3.2, along with the mesh resolution used in each computation.

In both codes, simulations are let run initially for 5 LFT to allow the stabilization of the flow. After this, measurements are made during a time equivalent to 20 LFT, which is equal to approximately 13.33 s in real time. Since the turbulence boxes, defined in Table 3.2 are only enough to supply an inflow during 2 LFT (Ti3) and 4 LFT (Ti12), the boxes are recycled for the duration of the computations. Velocity data are sampled at every time-step, resulting in a higher sampling frequency than the one used in the experiments, although it is made during a shorter period (13.33 s compared to ~ 60 s). The length of the computations is chosen as to maximize the bandwidth of the data employed for the calculations of spectra and correlations.

3.2.3 Layout of probes to store velocity series

With the aim at exploring the streamwise evolution of the flow characteristics, a series of probes to store the velocity series data are distributed within the domain. The relative position of these probes over the cross-section of the domain is shown in Figure 3.3. There, the closest probe to the plane centre represents the closest cell centre to the position $y = z = 0$ (in turn occupied by a cell face). This arrangement is repeated at 25 streamwise positions that extend from the inlet to the outlet in OpenFOAM. A similar setup is used for EllipSys3D, but extending from the TP to the outlet. In OpenFOAM, the first set of probes is located at $x = -0.498$ m and then, they are located at every 0.01 m between $-0.49\,\mathrm{m} \leq x \leq 0.45\,\mathrm{m}$ and every 0.1 m between $-0.4\,\mathrm{m} \leq x \leq 1.4\,\mathrm{m}$. In case the location of the probe does not coincide with a cell centre, the value at the closest one is registered. In this way, all results of longitudinal evolution that follow in this work are presented as the mean taken from the points distributed over each crosswise plane along the streamwise lines.

Similarly, probes were also located laterally, at approximately the same position of the hot-wires in the experiment (provided in Sec. 3.1). As in the longitudinal case, the positions of measurement could not coincide due to the limitations of the cell size and the consequent location of the cell centres (no spatial interpolations are used and only data stored at cell centres is employed). The main objective of these probes was to perform a correlation analysis between the velocity series (Sec. 3.3.7). For this purpose, a group of 11 probes were located at the centre

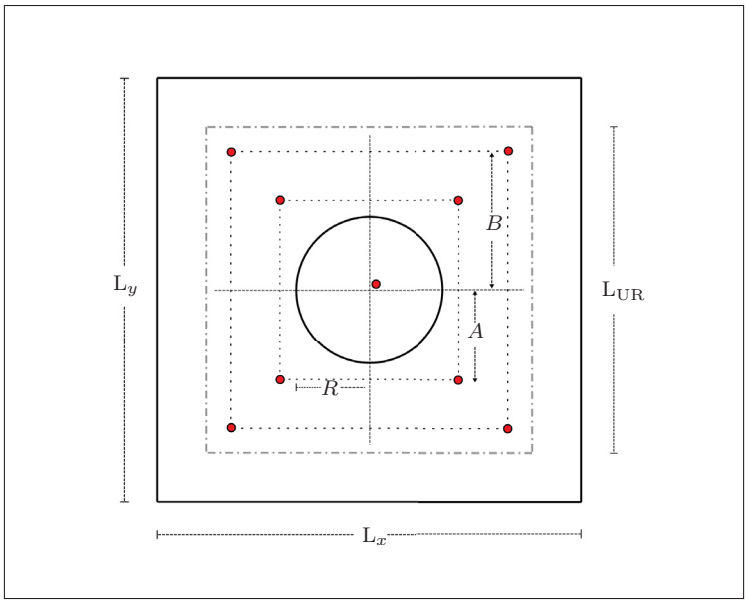

Figure 3.3 Locations of probes over a cross-sectional plane of the computational domain, represented by the small circles. L_{UR} is the side length of the uniform region used in OpenFOAM (see Table 3.2) whereas $A = 0.07$ m and $B = 0.15$ m. The location of the AD of radius R is also shown.

of consecutive cells. In OpenFOAM, the positions were between $0.041\,\mathrm{m} \leq y \leq 0.061\,\mathrm{m}$ for the Ti3 case and $0.03\,\mathrm{m} \leq y \leq 0.07\,\mathrm{m}$ for the Ti12 case. For EllipSys3D, positions were between $0.04\,\mathrm{m} \leq y \leq 0.06\,\mathrm{m}$ for the Ti3 case and $0.041\,\mathrm{m} \leq y \leq 0.08\,\mathrm{m}$ for the Ti12 case. Unlike the HWA measurements, the probes of the simulations register velocities simultaneously at all locations, allowing the computation of spatial and two-point correlations (but cannot be compared with experimental results). Make note that when quantities computed from these probes appear in comparisons of longitudinal development, the values shown will correspond to the averages of the corresponding quantities along every vertical line.

3.2.4 Estimation of integral lengthscales

The integral lengthscales shown in this work are deduced from the autocorrelation curves of u and v in the longitudinal direction in the synthetic turbulence or from their time-series in

the LES. In this way, L_1 and L_2 are obtained by making use of eq. (2.46). The method used to compute L_i consists of approximating the autocorrelation curve by a sum of six decaying exponentials. A similar procedure has been also used by Espana (2009) and Thacker *et al.* (2010), based in a technique first suggested by Kaimal and Finnigan (1994). Although other techniques exist that do not directly employ velocity correlations, such as to extrapolate the one-dimensional spectra to $\kappa = 0$ (using the relation between the spectrum and the Fourier transform of the autocovariance), all lengthscales shown here are calculated through the autocorrelations $\mathcal{R}_{11}(x,t)$ and $\mathcal{R}_{22}(x,t)$ only. While this method provides L_i for the synthetic turbulence, in the LES the autocorrelations provide a time-scale that can be in translated into a spatial one only under the assumption of the Taylor hypothesis (here, the integral time-scale obtained from the autocorrelations is multiplied by the average streamwise velocity at the point where the data are registered, which can be slightly different from U_∞). Three methods were examined for this calculation: 1) the first zero crossing of the $\mathcal{R}$ curve, 2) the value of the abscissa where $\mathcal{R} = 1/\,\mathrm{e}$ and 3) the approximation of $\mathcal{R}$ with a sum of decaying exponentials. The latest technique is used as it avoids the uncertainty of determining the crossing of the oscillating function $\mathcal{R}$ around zero as well as approximating better the expected asymptotic behaviour of a theoretical autocorrelation sampled to infinity, $\lim_{x\to\infty} \mathcal{R}_{ii}(x,t) = 0$, yielded by the exponentials. When comparing the results from computing L_i with these three approaches, it was observed that methods 1 and 2 could not provide a smooth solution in all cases, yielding large variations in L_i between nearby streamwise locations. Yet, when solutions of methods 1 and 2 do not show these problems, it was observed that method 2 would yield values slightly below method 3, whereas in method 1 they were approximately the same to method 3. The advantages of the selected method become more important when it is employed to calculate the autocorrelation of time-series data from the LES, as results cannot be averaged over the whole volume, like is done in the case of the turbulence boxes.

3.3 Results and discussion

3.3.1 Synthetic turbulence field

The first step of the investigation consists in the calibration of the parameters associated to the turbulence box. This is, to find $\mathrm{TI_B}$ and $L_{1,\mathrm{B}}$ such that when synthetic turbulence is introduced in the computational domain, the desired target values are attained at x_D. Since both TI and L_1 evolve differently in each code for the same turbulence box, different $\mathrm{TI_B}$ and $L_{1,\mathrm{B}}$ had to be determined for each platform. In OpenFOAM, these values were found for the target position at $5D$ from the inlet (where turbulence is introduced) in both TI cases. In EllipSys3D, it was opted to follow two different strategies:

a. To produce a turbulence field with $\mathrm{TI_B}$ and $L_{1,\mathrm{B}}$ values that would produce the sought after values at 5D downstream from the turbulence plane. This principle, equal to that used in OpenFOAM, was applied to the Ti3 case.

b. To generate a turbulence field with the corresponding input values to those employed in the OpenFOAM Ti12 case. Since the decay in EllipSys3D is stronger than the one yielded by OpenFOAM, the turbulence plane is brought closer to the target location so the desired TI and L_1 values are attained after such distance. The adequate separation between x_D and the TP was found to be $1.925D$. This procedure was used for the Ti12.

Figure 3.4 shows the relative positions of the inlet and TP and x_D in each case and for each code. The parameters of the synthetic turbulence for all boxes are shown in Table 3.3. These were computed longitudinally and averaged over the whole volume. We immediately notice the high TI values that were necessary to reproduce the evolution of the turbulence intensities reported by the experiments. In consequence, the followed approach can seem crude, on account of the Taylor approximation. However, the results reproduce, for the most part, the longitudinal evolution of turbulence predicted also by the analytical relations found in the liter-

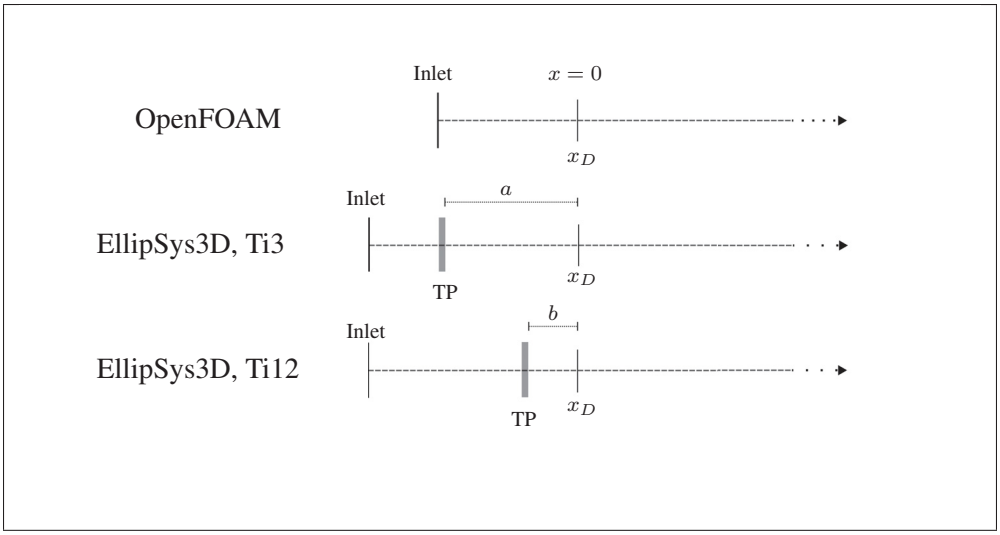

Figure 3.4 Comparison of the inlet and turbulence plane positions relative to x_D in OpenFOAM and EllipSys3D. One setup is used for OpenFOAM while in EllipSys3D the position of the TP is changed in each case. In this way, $a = 5D$ and $b = 1.925D$. Likewise, x_D is at $5D$ from the inlet in OpenFOAM and at $7.5D$ from the inlet in EllipSys3D. Note that in the employed coordinate system, x_D is located at $x = 0$.

ature (in the database used there are no measurements available in the vicinity of the grid used to generate turbulence or closer than $5D$).

Table 3.3 Main characteristics of the synthetic turbulence fields used in the LES.

		$\mathbf{TI_B}$ [%]	$\boldsymbol{L_{1,B} \times 10^{-3}}$ [m]	$\boldsymbol{k/\frac{1}{2}U_\infty^2}$ [-]
Case Ti3	OpenFOAM	35.0	5.82	0.37
	EllipSys3D	48.0	3.83	0.69
Case Ti12	OpenFOAM	60.2	15.3	1.08
	EllipSys3D	60.2	15.6	1.08

Table 3.3 presents the main properties of the generated synthetic turbulence fields to be used in each LES simulation. According to these results, we can see that the condition of a minimum of 2 cells per L_1 is barely fulfilled. In effect, for the case Ti3, $L_1/\Delta \simeq 3$ for OpenFOAM but $L_1/\Delta \simeq 2$ for EllipSys3D. The resolution of eddies is somewhat improved in the case of

Ti12, where $L_1/\Delta \simeq 4$ in both codes. These resolutions, that seem a priori too coarse, are the result of a series of compromises that have been previously explained. As it will be seen later, these values are enough to supply an integral lengthscale to procure the desired magnitudes at x_D. Nilsen *et al.* (2014) used a similar resolution for the synthetic inflow in LES computations of the wake of a rectangular channel, obtaining good comparison with experimental results related to the flow structures.

3.3.2 Statistics convergence

The simulations were run for an initial period of 5 LFT before data is stored to compute statistics. This period is shown in Figures 3.5 and 3.6, where we can see the evolution of the resolved and SGS parts of k in OpenFOAM and EllipSys3D. While k_{res} is obtained according to eq. (1.4), the subgrid parts are calculated in each code according to the different SGS models: in OpenFOAM $k_{\text{SGS}} = C_S\Delta^2 \left|\overline{S_{ij}}\right|^2$ while it is equal to eq. (2.8) in EllipSys3D. Having computed these values at every cell within an averaging region, the curves represent the volume-averaged value of these quantities at every time step. The averaging region is slightly different in each code: it corresponds exactly to the uniform region defined in OpenFOAM while in Ellipsys3D it has the same lateral dimensions but is longitudinally smaller, extending from $1D$ to $14D$. Since the main objective of this calculation is only to prove the convergence of second-order statistics, the disparity does not represent a problem. It can be seen in Figure 3.5 that after 5 LFT, convergence has been essentially achieved in OpenFOAM. The same occurs for EllipSys3D in Figure 3.6, although larger oscillations of k_{res} can be observed. A larger time is included in this figure so the periodic character of the oscillations can be noticed. Also in the same figure, note that k develops later in the Ti3 case compared to TI12 due to the larger separation of the turbulence plane from the beginning of the averaging volume.

Also from the Figures 3.5 and 3.6, we can deduce the average level of turbulence modelled by the SGS scheme in the LES simulation. This is, having $k_{tot} = k_{\text{SGS}} + k_{res}$, we observe that at 5 LFT, in OpenFOAM $k_{\text{SGS}}/k_{tot} \simeq 0.16$ for Ti3 and $k_{\text{SGS}}/k_{tot} \simeq 0.11$ for Ti12, while for EllipSys3D, we get $k_{\text{SGS}}/k_{tot} \simeq 0.08$ for Ti3 and $k_{\text{SGS}}/k_{tot} \simeq 0.22$ for Ti12. These results

support the assumption of a well resolved LES (at least 80% of k within the resolved scales) except for the Ti12 case in EllipSys3D (although by a small margin) where SGS scales are the most active. The difference in the results of each code arises from the stronger k decay in EllipSys3D, which is accentuated by the fact that the averaging volume used there does not include the region near the turbulence plane where k is the highest. These disparities originate also from the different handling of the SGS/resolved scales in each model with respect to the resolution L_1/Δ. This point will be elaborated later on, when other results are presented (power spectral density of velocity series are shown in Sec. 3.3.12).

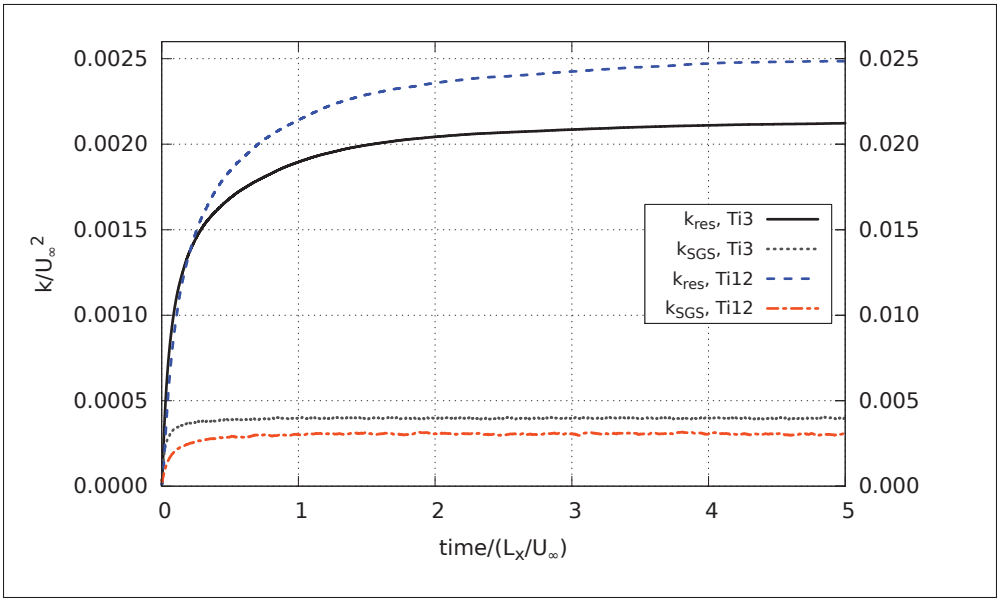

Figure 3.5 Evolution of the resolved and SGS parts of k during the first 5 LFT in OpenFOAM. The scale of the curves for the Ti3 case is on the left and for Ti12 on the right.

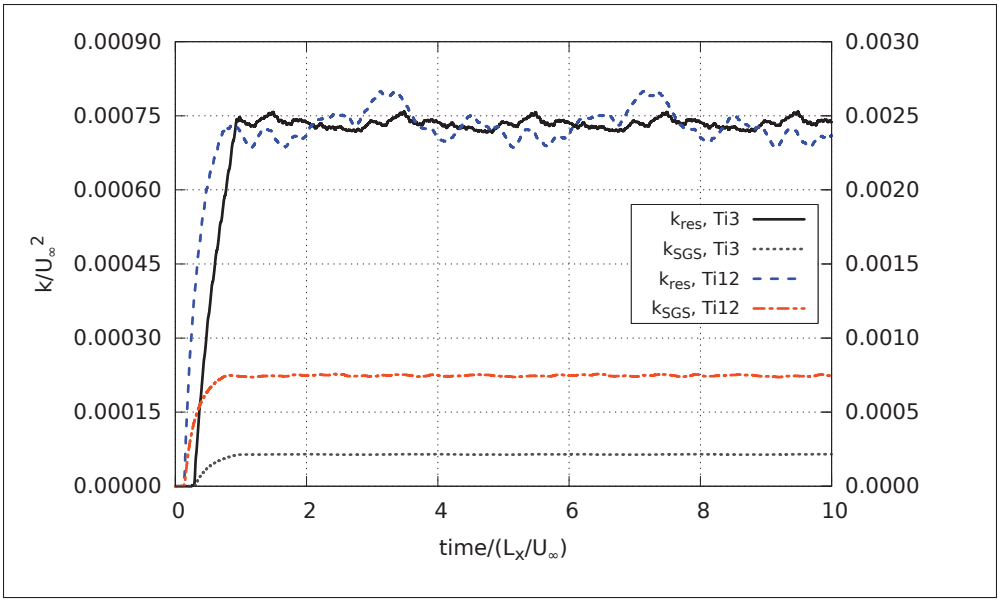

Figure 3.6 Evolution of the resolved and SGS parts of k during the first 10 LFT in EllipSys3D. The scale of the curves for the Ti3 case is on the left and for Ti12 on the right.

3.3.3 Turbulence decay

The longitudinal decay of turbulence intensity obtained from the LES is compared with the experimental results[2]. Due to the fact that time-series data is only available at 3 streamwise positions (and only for the Ti12 case), LDA measurements are used. Yet, as noted in Sec. 3.1, LDA measurements without disks were made at only one longitudinal position, so data from outside the wake envelope are used to supplement it. While the TI shown for the no-disk correspond to the mean of all recordings made in the y-direction, each of the remainder points in the curve represents the average of the measured TI values at $y = \pm 1$D with disk $C_T = 0.42$ (Ti3 case) and $y = \pm 1.5$D with disk $C_T = 0.45$ (Ti12 case).

The Figure 3.7 shows the TI decay of case Ti3 computed by the LES with each code compared to the measurements. The computed TI from the LES with OpenFOAM at x_D is 2.85% while EllipSys3D gives 3.37%. Make note that although the experimental databases are reported

[2]Previous work has been done that shows the capability of OpenFOAM to reproduce the decay of HIT, see Bautista (2015).

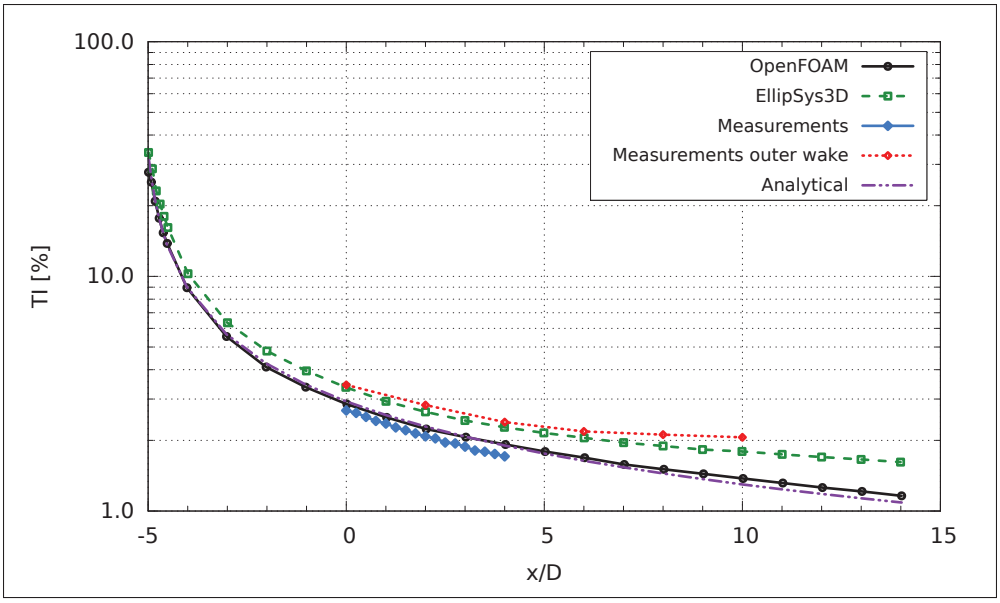

Figure 3.7 Comparison of the TI decay for the Ti3 case.

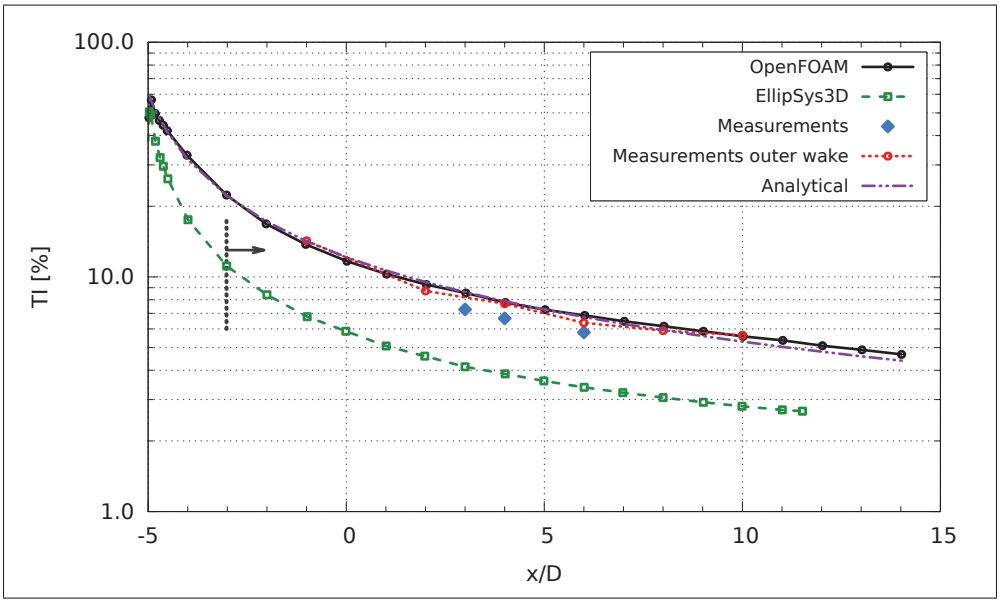

Figure 3.8 Comparison of the TI decay for the Ti12 case. The origin of the curve of EllipSys3D is at $x = -1.925D$ in the simulations but it has been shifted to $x = -5D$ in the figure to facilitate the comparison. The short vertical dotted line indicates the position where the curve attains the desired target TI value (exactly 11.13%).

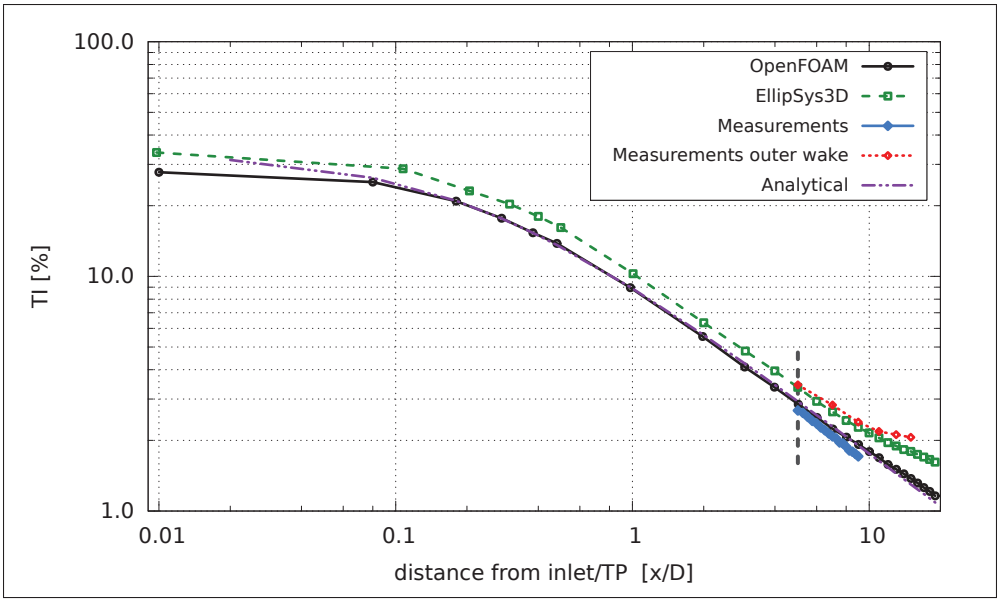

Figure 3.9 Comparison of the TI decay for the Ti3 case in log-log scale. The vertical long-dashed line indicates the target position x_D.

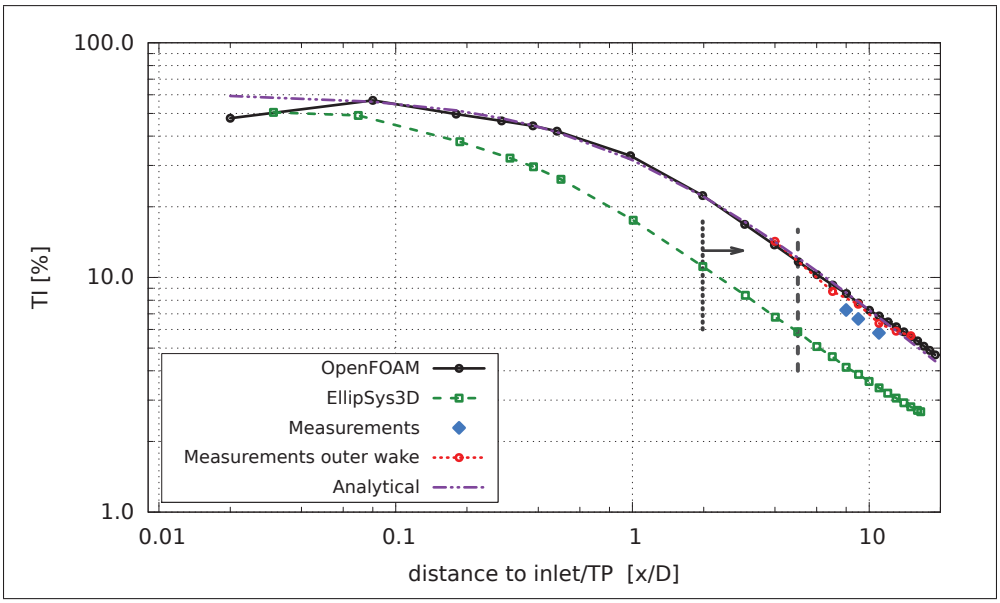

Figure 3.10 Comparison of the TI decay for the Ti12 case in log-log scale. The vertical long-dashed line indicates the target position x_D. See also notes on caption of Figure 3.8.

as having a TI of 3% and 12% at x_D, the actual values are slightly different. Since various turbulence boxes with different $\mathrm{TI_B}$ had to be tested to produce the desired $\sim 3\%$, these num-

bers were considered satisfactory for the comparisons of this work (as a reference, a box with $\mathrm{TI_B} = 30\%$ yields $\mathrm{TI} = 2.6\%$ in OpenFOAM—with the same $L_{1,\mathrm{B}}$). As it can be seen in Table 3.3, the $\mathrm{TI_B}$ value used for the LES in EllipSys3D is 18% higher than the one used in OpenFOAM, which indicates a stronger decay in the former code. Moreover, we can observe a large difference between the $\mathrm{TI_B}$ values of the synthetic boxes and those measured next to the inlet in OpenFOAM or at the turbulence plane in EllipSys3D: $\mathrm{TI_B} = 35.0\%$ vs $\mathrm{TI} = 27.8\%$ (OpenFOAM) and $\mathrm{TI_B} = 48.0\%$ vs $\mathrm{TI} = 33.7\%$ (EllipSys3D). The Ti3 case is also supplemented by data obtained[3] from the measurement campaign of Thacker *et al.* (2010) acquired with the same technique and experimental setup but with an inflow velocity of $U_\infty = 20$ m/s. The comparison with this data is supported by the fact that the decay rate (given by exponent n in eq. (2.58)) will vary only slightly for different Re_λ (Bailly and Comte-Bellot, 2003). In fact, Mohamed and LaRue (1990) conclude that n should be independent of Re_λ (after having studied flows of $28 \leq \mathrm{Re}_\lambda \leq 100$) but the the decay coefficient is in turn dependent on Re_λ. Considering that c_A in the turbulence decay equation only determines the TI and the fact that the results of Thacker *et al.* (2010) were obtained also at $\mathrm{TI} \simeq 3\%$ and generated with a grid of equal spacing, the measurement should provide information of a turbulence decay equivalent to that of our Ti3 case. While the decay predicted by OpenFOAM fits well the HWA measurements, the one drawn by EllipSys3D fits better those obtained through LDA (labeled as outer wake in the Figure). This is due to the fact that the TI used during the LDA measurements seems to have been higher than 3%, while the HWA is lower.

The decay of Ti12 is presented in Figure 3.8. For this figure, the origin of the curve of EllipSys3D is shifted to $x = -5D$ to better appreciate the variation in decay between codes. Most of the figures of longitudinal evolution will be presented in this way. We observe there that for the LES performed with EllipSys3D, the predicted decay is much stronger than in OpenFOAM, despite having used a synthetic field with the same turbulence characteristics. At a distance of $5D$ from the inlet, OpenFOAM has $\mathrm{TI} = 11.7\%$ while at the same distance from the turbulence plane EllipSys3D predicts $\mathrm{TI} = 5.9\%$. Furthermore, having measured $\mathrm{TI_B} \simeq 60\%$, the LES yields a TI of 56.8% next to the inlet with OpenFOAM and 50.5% at

[3] S. Aubrun, personal communication.

the turbulence plane in EllipSys3D. Thus, as in the Ti3 case, the TI predicted at the position where turbulence is introduced in the computational domain is lower in EllipSys3D than in OpenFOAM. Yet, contrary to what is observed in the previous Ti3 case, the TI read by the first longitudinal probes in OpenFOAM ($\mathrm{TI} = 47.7\%$) is lower than the second one (56.8%). After this, the TI decays monotonically as in all other simulations. The LDA measurements made outside the wake compare very well with the OpenFOAM results while the values obtained with HWA are slightly below, but they seemingly display the same decay rate. The dotted vertical line indicates the position where the curve coincides with x_D when the turbulence plane is at its original location of $x = -3.075D$.

Also in Figures 3.7 and 3.8, the analytical form of the TI decay (eq. (2.58)) is fitted to the results of OpenFOAM. In most of the results reported in the literature a fit is produced setting $x_0 = 0$ in the decay equation, which neglects the agreement close to the grid or the place where turbulence originates, since the stations where measurements or calculations are reported are generally far from such region. However, as the complete evolution of TI is monitored, a better fit is obtained by setting x_0 to a position different from where the turbulence is introduced (in particular, to an upstream location). This way, the fit of eq. (2.58) in the Ti3 case gives $A = 1/9.85$, $n = 1.519$ and $x_0 = -0.021$ m and in the Ti12 case, $A = 1/11.43$, $n = 1.661$ and $x_0 = -0.08453$ m (x_0 is measured with respect to inlet). If the fit is made using $x_0 = 0$, the parameters of eq. (2.58) are closer to those reported in the literature (see Sec. 2.4.1): $A = 1/24.11$ and $n = 1.281$ for the Ti3 case and $A = 1/28.49$ and $n = 1.15$ for Ti12. Yet, the predicted curve yields a much higher TI at the inlet than the one given with $x_0 \neq 0$, this is, $\sim 60\%$ for Ti3 and $\sim 100\%$ for Ti12. The mesh spacings used for the fits are $M = 0.0225$ m for Ti3 and $M = 0.20$ m for Ti12. Figures 3.9 and 3.10 represent the same data of the turbulence decay, where the only difference with Figures 3.7 and 3.8 is that the x-axis is presented in a logarithmic scale (which represents the distance from the inlet or TP). In this way, the power law decay (of slope $-n$) of the TI can be better appreciated. In the case of Ti3, Figure 3.9 shows that the rate of decay is equally predicted by each code, the only difference being the slightly larger TI in EllipSys3D which is conserved throughout the domain. In the Ti12 case,

Figure 3.10 shows that the rate of decay predicted by each code is different for a distance of approximately $2D$ after the inlet/TP. After that, the decay rate is equally predicted in both platforms.

One reason for the variation of TI between the synthetic field and the point of its introduction in the LES domain is the interpolations performed at every time step between the available streamwise planes. Specifically, due to the high TI the local velocities at such point are much larger than their average U_∞ so the turbulence box is introduced at very small time-steps to keep the CFL condition. Therefore, as more intermediate values of velocity are needed between those of the available planes, more interpolations need to be performed. This, in turn, raises the abatement of the fluctuations magnitude. In Table 3.4 we compare r.m.s. of the fluctuation velocity for every component, obtained at the place where turbulence is introduced in the domain with each code. The r.m.s. values of every turbulence box are included also as a reference. We can see that the technique utilized in OpenFOAM shows smaller losses compared to EllipSys3D at this location, also displaying more homogeneity between components. These differences are explained considering (as pointed at by Gilling and Sørensen, 2011) that a pressure change is induced by the TP which causes that the changes in the streamwise velocity component will build up slowly, attaining their maximum value until after some distance. Conversely, the fluctuations in the transverse components should attain the desired value immediatly after the TP.

When comparing the same techniques employed here to introduce synthetic HIT into a LES with EllipSys3D, Gilling and Sørensen (2011) observed that when turbulence is imposed at the inlet, the r.m.s of fluctuations next to the inlet was slightly higher ($\sim 5\%$) than in the synthetic field while the turbulence plane method produced values 30% lower. Moreover, while they observed losses smaller than 10% in the transverse components when turbulence is imposed at the inlet, the strong decay seen in the streamwise component is reduced to $\sim 10\%$ for the transverse component, when the turbulence plane is used. On the contrary, our simulations show a stronger decay in the transversal directions than streamwisely with the use of either technique. Yet, the calculations of Gilling and Sørensen (2011) were performed at considerably

lower TI (1.2%), with a larger longitudinal refinement of the turbulence box (compared to their LES domain) and with a L_1 value of half the domain height, resolved by $L_1/\Delta = 16$. In addition, make note that in our simulations, the losses in the case with higher TI in the synthetic field are smaller than in the case with lower TI, as shown in Table 3.4.

Table 3.4 Comparison of the r.m.s. velocities obtained from the synthetic turbulence field with the values computed by the LES of each code at the point where turbulence is introduced in the computational domain.

		u_{rms} [m/s]	v_{rms} [m/s]	w_{rms} [m/s]
OpenFOAM Ti3	Box	1.05	1.05	1.05
	LES	0.84	0.75	0.71
EllipSys3D Ti3	Box	1.44	1.44	1.44
	LES	1.01	0.65	0.74
OpenFOAM Ti12	Box	1.80	1.80	1.80
	LES	1.48	1.42	1.44
EllipSys3D Ti12	Box	1.80	1.80	1.80
	LES	1.52	1.15	1.18

The exact reason for the stronger decay in EllipSys3D compared to the results of OpenFOAM is difficult to determine with absolute certainty and it is not within the objectives of this work. Conversely, only a hypothesis of the origin of this difference is presented. This is done in Sec. 3.3.6 after the dissipation of turbulence energy produced in each code is compared.

3.3.4 Homogeneity

With the aim at observing the homogeneity of the velocity in the computed flow, in Figures 3.11 and 3.12 we compare the r.m.s. values obtained along vertical profiles at $z = 0$ for different downstream distances. In the Ti3 case, LDA measurements at $x/D = 0$ are also included for comparison. These data show a disparity between the values measured for the streamwise and vertical components. When comparing the results of OpenFOAM of Figure 3.11 with those of EllipSys3D in Figure 3.12, we notice that the former shows almost no variation among the

magnitudes of the r.m.s velocity components, while the latter exhibits a small difference, with the magnitude of the spanwise component somewhat larger than the rest.

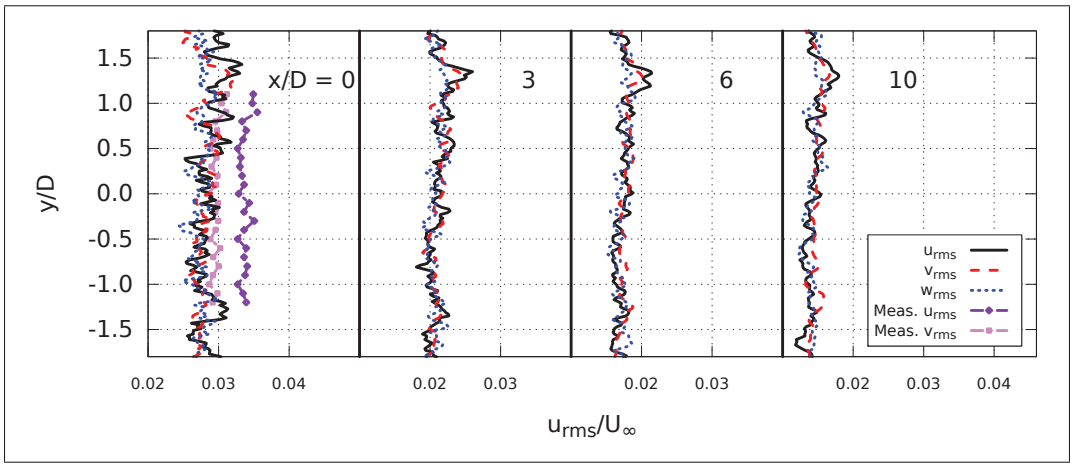

Figure 3.11 Vertical profiles of r.m.s. velocities obtained with OpenFOAM for the case Ti3.

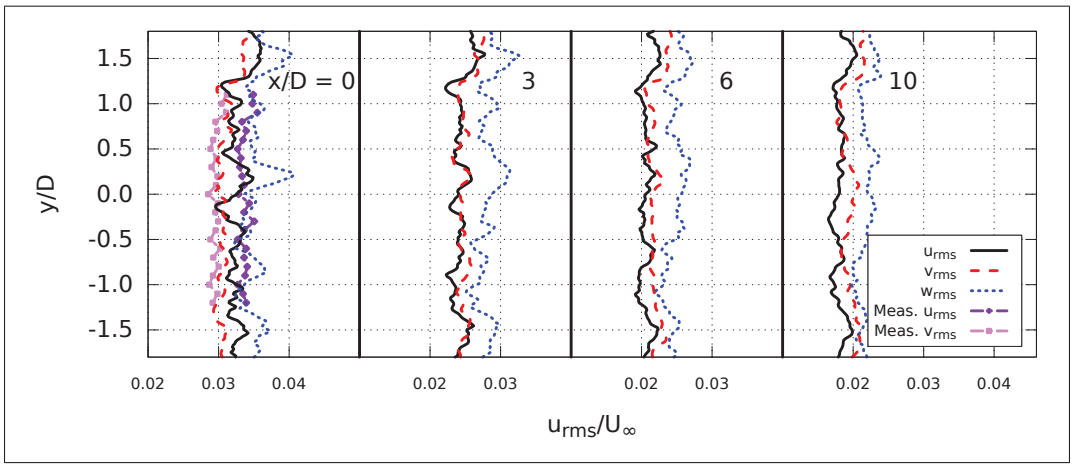

Figure 3.12 Vertical profiles of r.m.s. velocities obtained with EllipSys3D for the case Ti3.

The profiles for the Ti12 cases can be compared from Figures 3.13 and 3.14. Unlike other results of longitudinal evolution of EllipSys3D, these figures show curves with the turbulence plane at its original position of $x = -3.075D$. This is done to compare the variation of r.m.s velocity components between codes at the target position and downstream from it. The

measurements correspond to the HWA data at $x = 3D$ and $6D$. While the homogeneity prevails in OpenFOAM, the results of EllipSys3D show stronger fluctuations for the streamwise component, specially for the first longitudinal position (in agreement to Table 3.4). Later, as the u_{rms} decay appears to be more pronounced than in the transversal components, the r.m.s velocities become more homogeneous. Still, for the last three profiles, the vertical component falls behind the other two. Except for the large u_{rms} at $x = 0$ in Figure 3.14, there is no apparent relation between homogeneity (or the lack thereof) in the flow and the level of TI.

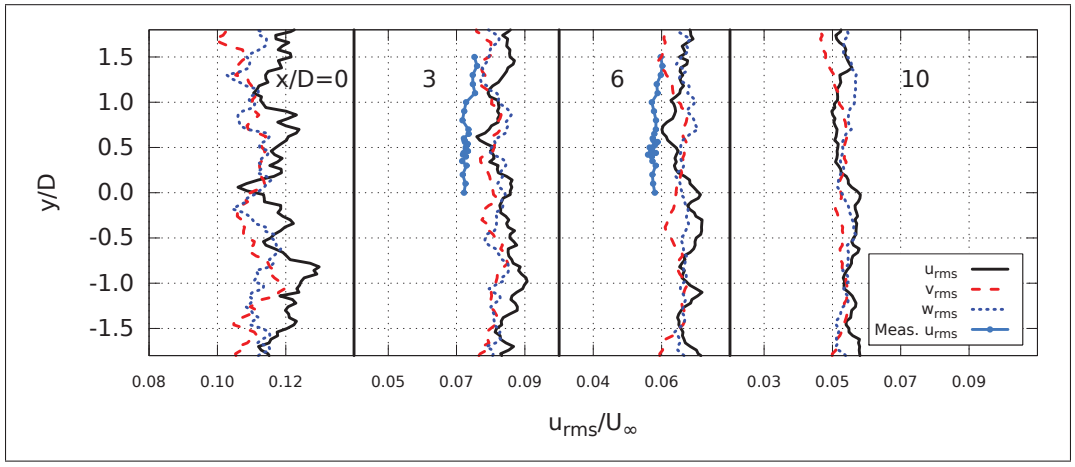

Figure 3.13 Vertical profiles of r.m.s. velocities obtained with OpenFOAM for the case Ti12.

3.3.5 Longitudinal evolution of turbulence kinetic energy

To study the LES modelling in the decaying-HIT, we show the longitudinal evolution of the resolved part of the turbulence kinetic energy k_{res} with respect to the total amount $k_{tot} = k_{\text{SGS}} + k_{res}$ in Figure 3.15, for both TI cases. Clearly, most of the total computed value k_{tot} is composed by the resolved part k_{res} in all cases. When comparing the results bewteen codes, the largest difference is observed for the Ti12 case, where the contribution of the subgrid part appears to be larger in EllipSys3D, since the one in OpenFOAM is reduced to about 10% quite rapidly. This in turn coincides to what is observed in Figures 3.5 and 3.6. In the Ti12 case, thanks to the better resolution of the velocity fluctuations, the subgrid parts

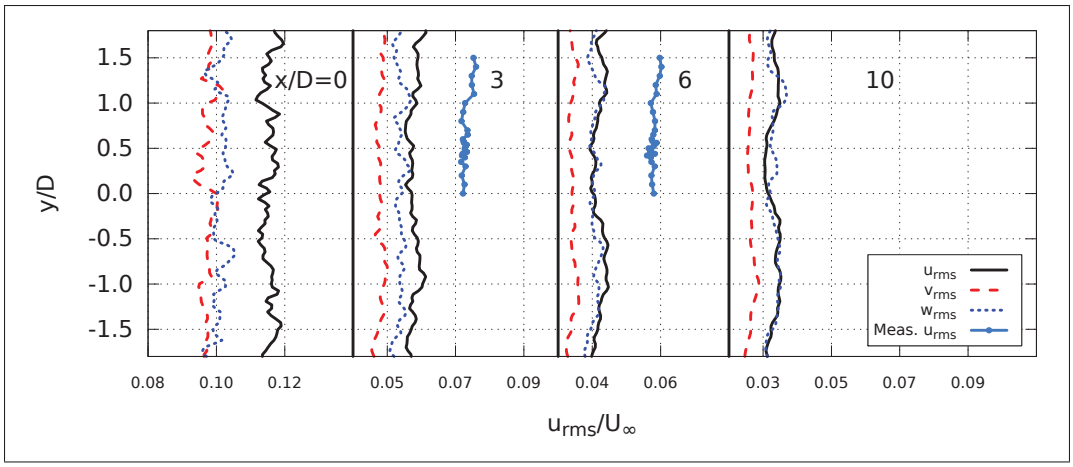

Figure 3.14 Vertical profiles of r.m.s. velocities obtained with EllipSys3D for the case Ti12.

are consistently smaller with respect to the resolved parts in OpenFOAM. Yet, in EllipSys3D the subgrid contribution remains appreciably larger, and does not seem fall to the $\sim 10\%$ seen in OpenFOAM. It is remarked that in the Ti3 case the subgrid contribution cannot be much bigger due to lack of small fluctuations, on account of the limited resolution of the synthetic turbulence field.

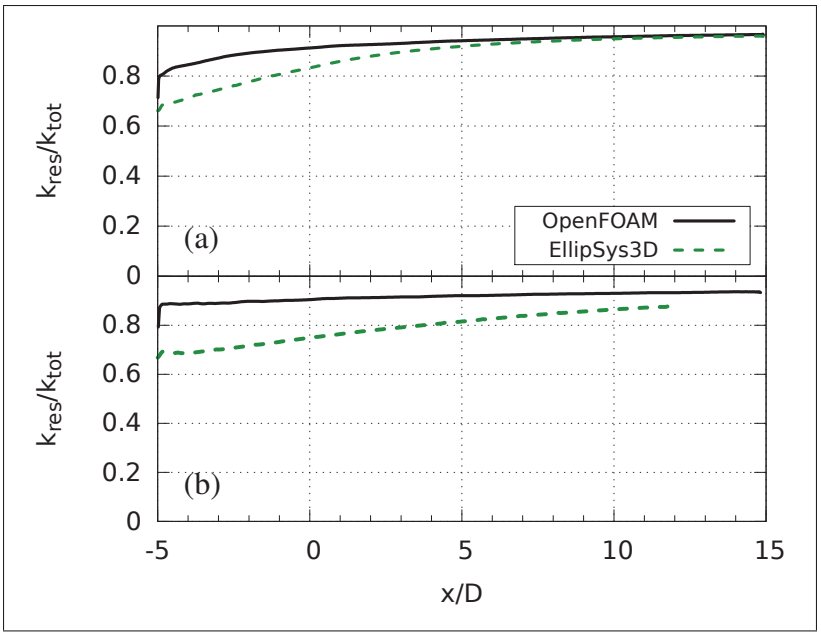

Figure 3.15 Longitudinal evolution of k_{res} in the LES. (a) Ti3 and (b) Ti12.

3.3.6 Longitudinal evolution of turbulence dissipation

Although the previous results support the notion of a well resolved turbulent flow (spectra are shown in Sec. 3.3.12), yet more limited in the case of EllipSys3D, there seems to be a difference in how each code handles the turbulence dissipation due to the disparate decay. In Figure 3.16 we compare the ratio of the subgrid dissipation ε_{SGS} to the total value $\varepsilon_{tot} = \varepsilon_{\text{SGS}} + \varepsilon_{res}$ for each code. For the case Ti3, we obtain essentially identical results for both codes. At the beginning of decay, the majority of dissipation occurs in the subgrid parts, with the resolved parts accounting for approximately only 10%. But the ε_{SGS} contribution rapidly decreases, crossing the 50% at $\sim 2.8D$. The falling trend continues for all the length of the domain. Conversely, differences in the modelling arise in the Ti12 case, despite being equal on both codes at the beginning (above $\sim 90\%$), the subgrid contribution drops faster in EllipSys3D, reaching 50% after about $10D$. Meanwhile, in OpenFOAM the contribution ε_{SGS} only falls to $\sim 60\%$ at the end of the domain.

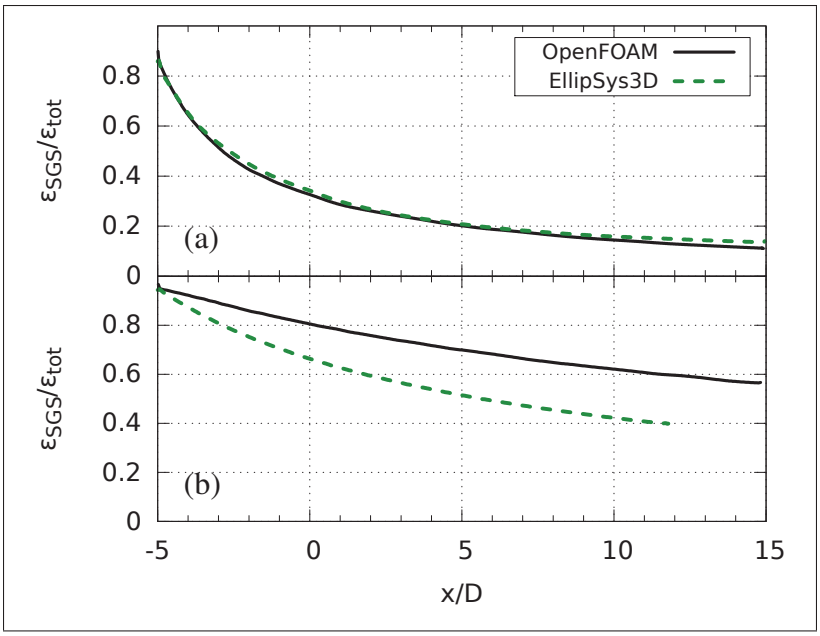

Figure 3.16 Longitudinal evolution of ε_{SGS} in the LES. (a) Ti3 and (b) Ti12.

The values of ε_{tot} obtained with each code are compared in Figures 3.17 and 3.18, next to the estimated value of the dissipation obtained from different methods. The difference in the dissipation calculated in each LES is small, but noticeable, in the Ti3 case. In contrast, for the high turbulence case we first notice that the total dissipation computed by OpenFOAM is considerably larger than the one from EllipSys3D. Moreover, the results from OpenFOAM compare very well to the measured dissipation (calculated with eq. (2.50) and eq. (2.49)). The larger ε_{tot} value in EllipSys3D helps to explain the smaller subgrid contribution seen in Figure 3.16 for EllipSys3D in the Ti12 case. This is, as the small scales disappear faster than in OpenFOAM, the dissipation is caused in a larger part by the large scale eddies. Considering that in both codes the turbulence development initiates with similar TI values (Figure 3.8 and Table 3.4) but in EllipSys3D it shows a stronger decay, it seems that in the latter code there is a source of dissipation that is not accounted either by ε_{res} or ε_{SGS}. In both codes, the SGS part dominates the contribution to ε_{tot} (at the onset of the decay) and although the dissipation rate is larger in EllipSys3D than in OpenFOAM, its magnitude is still smaller. While this suggests that the different SGS models could produce a disparity in dissipation, it does not completely

explain the larger turbulent energy decay observed in EllipSys3D. In fact, previous studies have demonstrated that the Smagorinsky model produces the highest dissipation when compared directly with a dynamic mixed-lengthscale model such as the one of Ta Phuoc (Sarlak *et al.*, 2015b). In turn, this can be an indication that the numerical dissipation is larger in EllipSys3D than in OpenFOAM. After considering different elements that can cause the stronger TI decay in EllipSys3D compared to OpenFOAM, three factors have been identified:

I) **Non-incompressibility of the synthetic turbulence.** For isotropic turbulence of a continous field, the divergence should be zero. But in a discrete field, the Mann algorithm can yield fields that are not divergence free, as shown by Gilling (2009) (as mentioned in page 62). When such field is introduced in the LES domain, divergence free conditions will be enforced by the continuity equation, but how this is done exactly depends on the approach followed by the solver in each code. In particular, the multigrid approach included in EllipSys3D is in principle more effective and faster[4] in producing incompressibility, as it accelerates the convergence of the pressure correction equation (Michelsen, 1994; Sørensen, 1995; Versteeg and , 2007). Under this assumption, the rapid adjustments in the velocity field in EllipSys3D can lead to a larger loss of turbulence energy than in OpenFOAM, where the same multigrid technique is not used.

II) **Interpolation schemes in LES.** From the different numerical schemes utilized in each code, the ones that can produce the largest disparities in numerical dissipation are the interpolation schemes employed for the convection terms. Although in both codes the central interpolation scheme is mostly used, a different amount of upwind portion is used. While EllipSys3D uses a fixed amount of 10% of the QUICK scheme, OpenFOAM employs a varying portion of the upwind scheme that can go from zero to 20% (see Sections 2.2.3 and 2.2.4). However, the upwind parts in OpenFOAM are used only to remove staggering caused by the unphysical oscillations from the velocity/pressure decoupling which can occur, for instance, in the presence of objects inside the flow

[4]This means that divergence-free conditions are attained for shorter longitudinal distances from the plane where turbulence is introduced (compared to LES solvers that do not follow this approach).

such as the AD. Therefore, the upwind part should remain low in the computation of free flow. It should be noted that this implementation is built in within the code and the exact ratio of the upwind part cannot be easily outputted

III) **Techniques to introduce synthetic turbulence.** In EllipSys3D, turbulence is introduced by means of body forces in the TP (which are also Gaussian-distributed in the longitudinal direction) instead of imposing the velocity field directly, as it is done in OpenFOAM. When referring to the reasons to use the first technique, Troldborg (2008) argues that its use can help to avoid the problems associated to the lack of incompressibility of the synthetic field. Therefore, the calculations performed in this process can lead to a loss of turbulence energy.

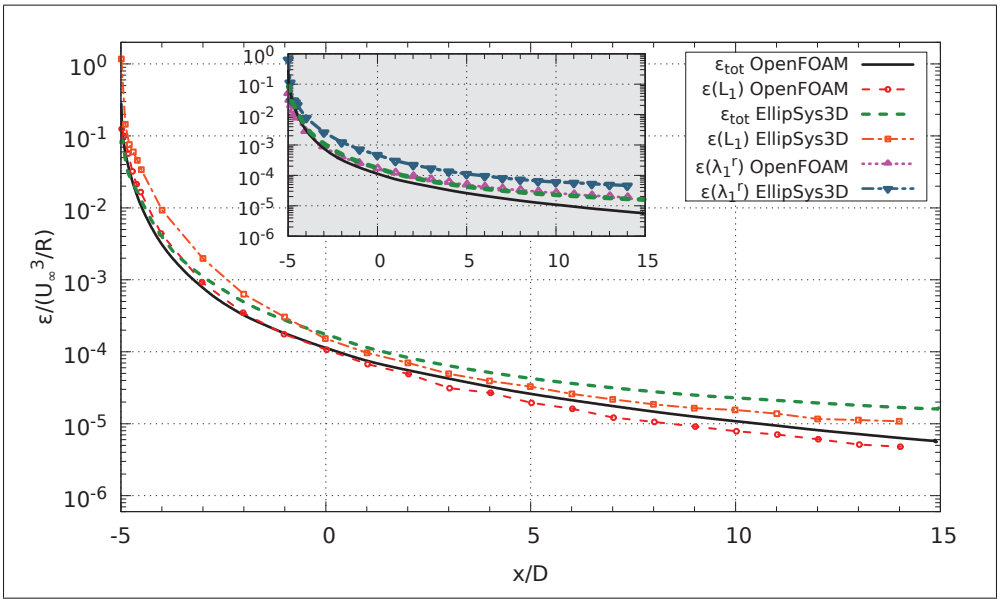

Figure 3.17 Longitudinal evolution of ε for the case Ti3.

Make note that while (I) and (II) influence the loss of turbulence energy in the LES computation, (III) could only conduce to the computation of lower turbulence energy values before the LES simulation is performed. To determine the exact reason for the different turbulence decay obtained in each code is out of the scope of this work, so we offer only a list of plausible

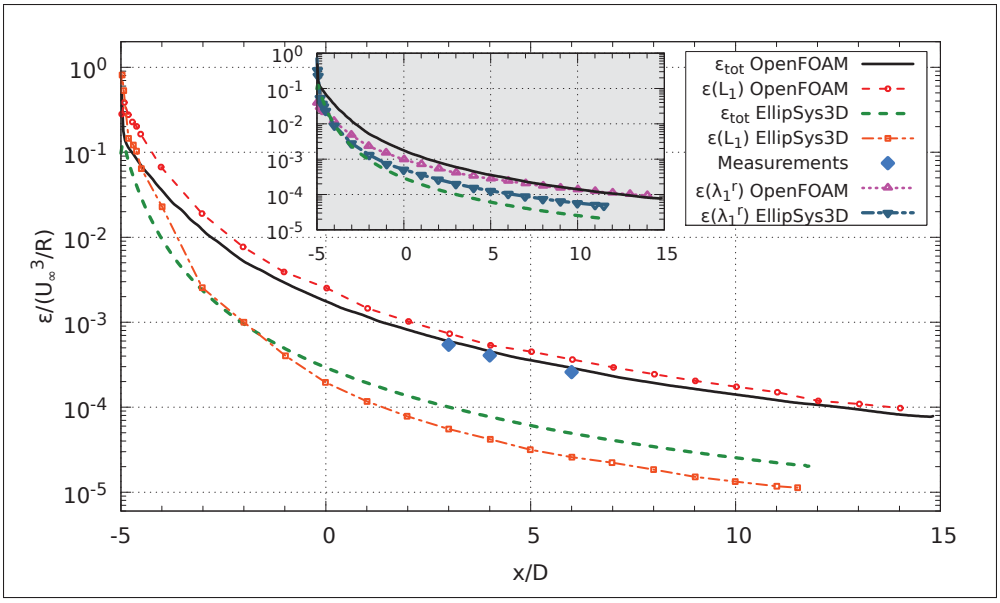

Figure 3.18 Longitudinal evolution of ε for the case Ti12.

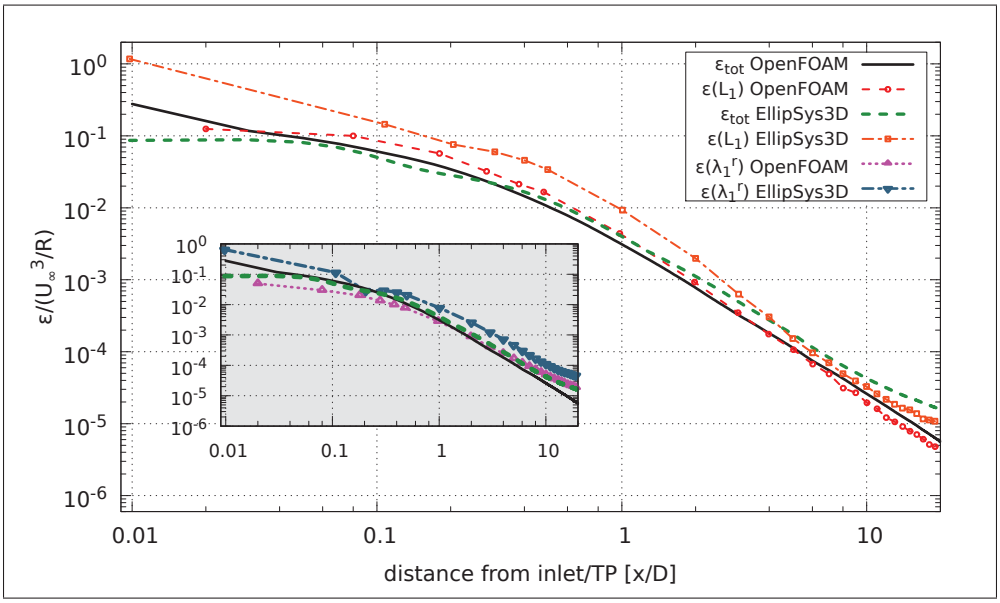

Figure 3.19 Longitudinal evolution of ε for the case Ti3 in log-log scale..

reasons. Conversely, to fulfil our objectives, it was sufficient to establish the setup in each code that allows to reproduce the desired turbulence features at the target position (Sec. 3.2).

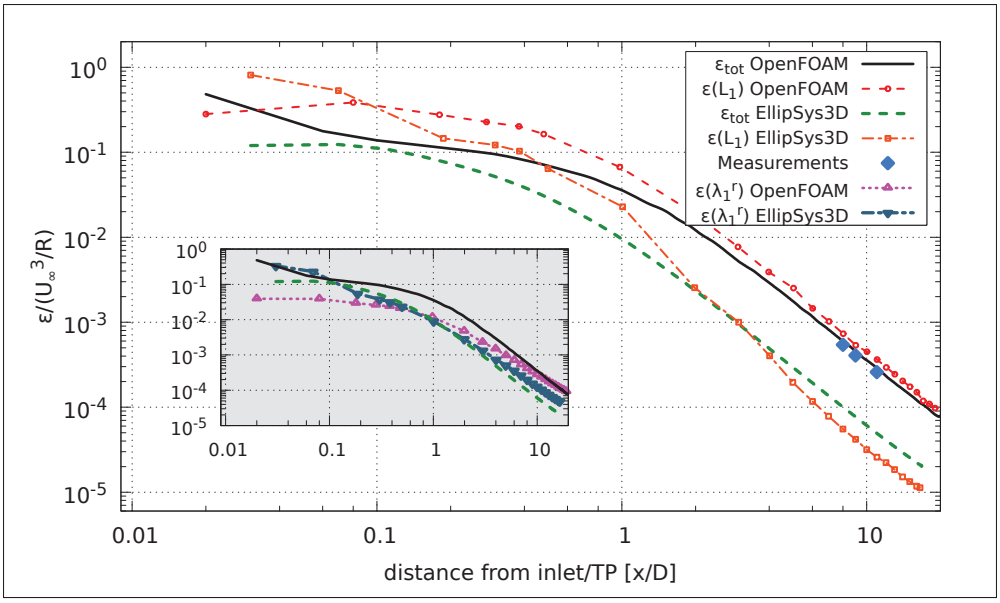

Figure 3.20 Longitudinal evolution of ε for the case Ti12 in log-log scale.

Continuing with the analysis of turbulence dissipation, the results deliver valuable information about the resolution of turbulece lengthscales, as different scenarios result in each case. Despite the fact that the mesh of the Ti3 case has the smallest cells, the Ti12 is the case with the best resolution (in terms of L_1). This assertion is confirmed by the comparison with the experimental results. Figure 3.16 – (b) shows that in those conditions, most of the dissipation occurs in the SGS part. The contribution of the resolved part increases only to the extent that the smallest fluctuations (modelled by the SGS part) dissipate and the viscous dissipation of the resolved eddies becomes more important. But under this assumption, the behaviour seen in Figure 3.16 – (a) seem to contradict the previous results that pointed towards a well resolved LES. This can be explained by the location of the SGS filter: while the results of the Ti12 support the hypothesis of a good position of the filter (in the inertial range), in the Ti3 case the location seems to be too close to the energy containing range (determined by $1/L_1$) with the consequence of a very short turbulence cascade. Moreover, due to the coarse resolution of L_1 also in the synthetic field of case Ti3, no smaller eddies really exist, at least next to the inlet or turbulence plane. As a result, most of the dissipation in the flow occurs due to resolved fluctuations. Still, Figure 3.16 shows that up to $\sim 2.8D$, ε_{SGS} dominates the dissipation process.

This could be explained through the adjustment of the scantly resolved flow into the computational domain, i.e. the adjustment of the turbulence field to the divergence-free conditions imposed by the LES. Precisely, although the Mann algorithm is in principle capable of generating incompressible fields, in reality, the resulting turbulence will not be divergence-free for discrete representations of the velocity field. As indicated by Gilling and Sørensen (2011), this is because the continuous field contains harmonics with very high frequencies that cannot be accurately represented in the discrete domain. Thus, as the resolution of the turbulence box is coarser with respect to the turbulence scales, the adjustment of the synthetic field to the incompressibility conditions enforced by the LES solver will be more noticeable. To solve this problem, a correction in the Mann algorithm has been proposed by Gilling (2009).

Next to the above, within Celik *et al.* (2005), L. Davison argues that a coarse mesh yields values of the rate of strain $\overline{S}_{ij}$ that are too small, which in turn limits the value of the resolved rate-of-strain (eq. (2.4)) and in consequence, the total dissipation $\varepsilon_{tot} = 2(\nu_{\text{SGS}} + \nu)\overline{S}_{ij}\overline{S}_{ij}$. Although this reasoning was used for a problem of wall bounded flows, in our case it could contribute to explain the somewhat larger drop of ε_{tot} in the Ti3 case compared to what is seen in Ti12.

In a study of grid generated turbulence over an extensive range of Re_λ, Mydlarski and Warhaft (1996) affirm that the adherence of the flow to $\varepsilon = 0.9\left\langle u'^2\right\rangle^{3/2}/L_1$ (see eq. (2.60)) implies the existence of a fully developed cascade, where the dissipation can be inferred from the large scales. The prediction of eq. (2.60) is also shown in Figures 3.17 and 3.18 as $\varepsilon(L_1)$. The expression fits reasonably well to results of both OpenFOAM and EllipSys3D in the Ti3 case. Clearly, the prediction of this equations depends strongly on the accuracy to obtain L_1, which will be discussed later. For the Ti12 case, while the fit is relatively good for OpenFOAM (it overpredicts ε_{tot}, particularly near the inlet), the expression compares similarly to the results of EllipSys3D until about $x = -4D$, predicting a lower dissipation afterwards which is in fact closer to ε_{SGS}. This is due to the rate of growth of L_1 predicted by EllipSys3D, which is larger than in OpenFOAM in the Ti12 case (Sec. 3.3.9).

In the insert frames in Figures 3.17 and 3.18, a comparison is also made between ε_{tot} and the dissipation estimated from the resolved scales of the streamwise velocity through eqs. (2.50) and (2.49), denoted $\varepsilon(\lambda_1^r)$. As the latter expression is valid for isotropic flows, the comparison represents an assessment of the degree of anisotropy in the fluctuations. The result show that the predictions of $\varepsilon(\lambda_1^r)$ in both cases provide a better match with ε_{tot} early in the flow development and it separates further downstream. The only exception is in the Ti12 case for the OpenFOAM results, where the opposite occurs. This could be due to the very large difference between the SGS and resolved contributions to the dissipation at the beginning of the turbulence decay. Figures 3.19 and 3.20 represent the same data of comparisons of the computed dissipation but in a log-log scale. This is done to compare the decaying rate of the different curves. It can be seen that in each case, the decay rate predicted by each code is very similar. In the Ti12 case we can observe that the decay of the dissipation in EllipSys3D starts earlier than in OpenFOAM, although the slope of the curves resembles each other for the most part. In the OpenFOAM results in both figures we can appreciate a higher slope of decay at the beginning of the ε_{tot} curves, but this only represents the decay along the first two cells of in the domain.

3.3.7 One- and two-point correlations

3.3.7.1 Comparison of one-point correlations of LES and measurements

The study of correlations in the flow represents another way to study the capability of the code to reproduce the structures in the flow and to evaluate the limitations of the resolution. First, the normalized, time autocorrelation of the velocity series, $\mathcal{R}_{11}(x,t)$, registered at $x = 3D$ in the centreline of the domain are compared in Figures 3.21 and 3.22 for the Ti3 and Ti12 cases, respectively. In each curve, the marks represent a time step of the simulation. The insert figures show a closer comparison of the early correlation decay. For the Ti3 case, we can see that the curves commence to deviate from each other after $t \approx 0.01$ s. From there, the curves reveal low velocity fluctuations or drifts. In the case Ti12 we can compare the results of the codes with the autocorrelation of the velocity sampled in the wind tunnel. We can appreciate in the

insert frame the difference in sampling rate between the experiment and the LES. It can also be seen that despite the fact that EllipSys3D predicts a closer correlation to the experiments for the first steps, OpenFOAM provides a better match of the entire correlation curve (i.e. until $\mathcal{R}_{11}(x,t)$ approaches to zero). Unlike the OpenFOAM results or the measurements, the curve drawn by EllipSys3D does not observe a smooth decay. Instead, a noticeable change in the slope occurs between 0.005 s $< t < 0.01$ s which suggests a disparate change in the evolution of the turbulence structures. After this, the curve greatly decreases the decay rate, changing little from below the 0.2 value for the rest of the figure. The one-point correlations for the positions $4D$ and $6D$ compare in a very similar way to those described here (at $3D$) for both TI cases.

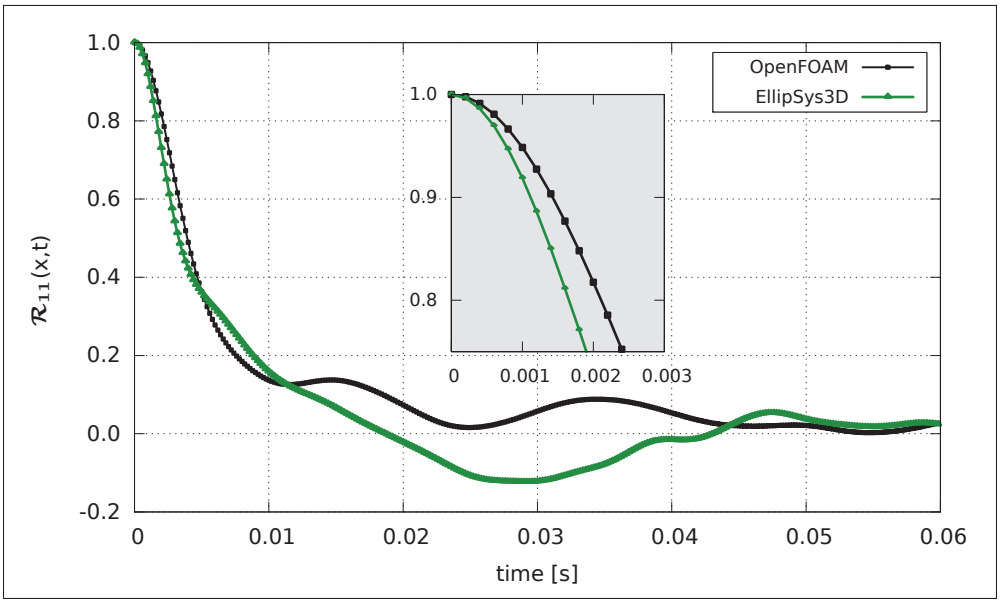

Figure 3.21 Comparison of autocorrelation in time at $x = 3D$, case Ti3. Each mark in the curves represent a time step.

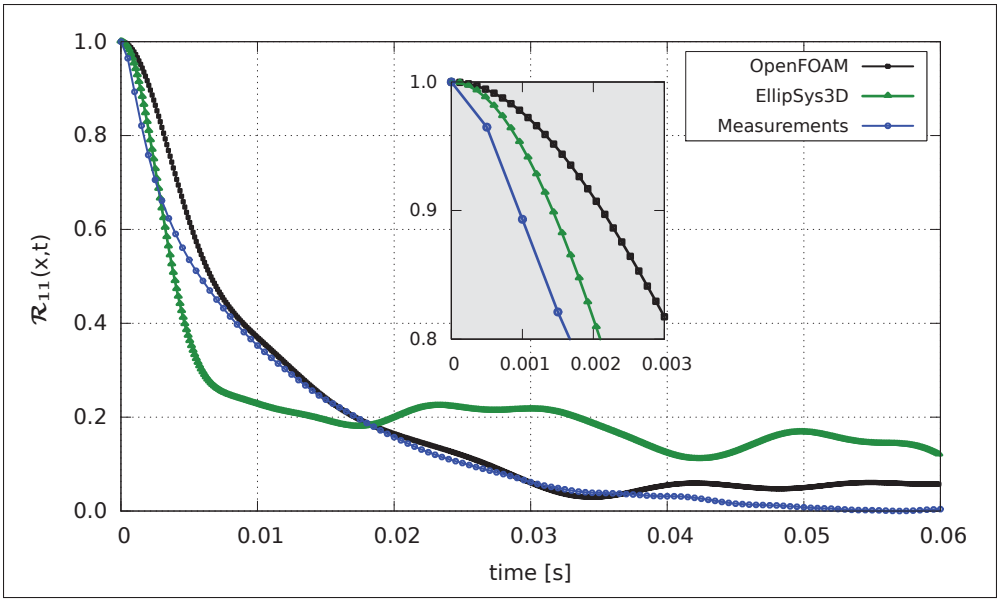

Figure 3.22 Time autocorrelation in time at $x = 3D$, case Ti12. Each mark in the curves represent a time step.

3.3.8 Space-time correlations

A better picture of the spatial structure of turbulence is provided by the space-time correlations. For this, we look into the preservation of the turbulence structures in the vertical direction, where the flow is assumed to be isotropic and homogeneous (Sec. 3.3.4). The space-time correlations are obtained as the maximum value of the normalized two-point time correlation at successive cell separations, from $\Delta y = 0$ to $\Delta y = 10$. The computations are made starting from y_0, which in each case corresponds to the distance of the closest cell centre above $y = 0.04$ m from the centreline. The procedure is repeated at $x = -2.5D$, $3D$, $4D$ and $6D$.

In Figures 3.23 and 3.24 we compare the space-time correlations $\mathcal{R}_{11}(y_0 + \Delta y, t)$ obtained from OpenFOAM and Ellipsys3D, respectively, for the case Ti3. In the results of OpenFOAM, at $x = -2.5D$ we see a decrease in the correlation to a value close to zero within the first five cells, then the value increases but it finally falls again to close to zero after ten cells. This could be caused by the apparition of a large structure whose effect is also accentuated by its repeated passage due to the recycling of the turbulence box. In the next $x-$positions, we observe an

increase in the correlation values due to the growth of the integral lengthscale. Even for the maximum separation $\Delta y = 10$ cells, the correlations at $3D$, $4D$ and $6D$ have not fallen yet to zero. For the results of EllipSys3D, the comparison of the correlations indicates a good resolution for last three positions, with little growth of the large scales. For $x = -2.5D$, the correlations decay more rapidly, to about 70% within the first two cells (in OpenFOAM this value is attained until the third cell).

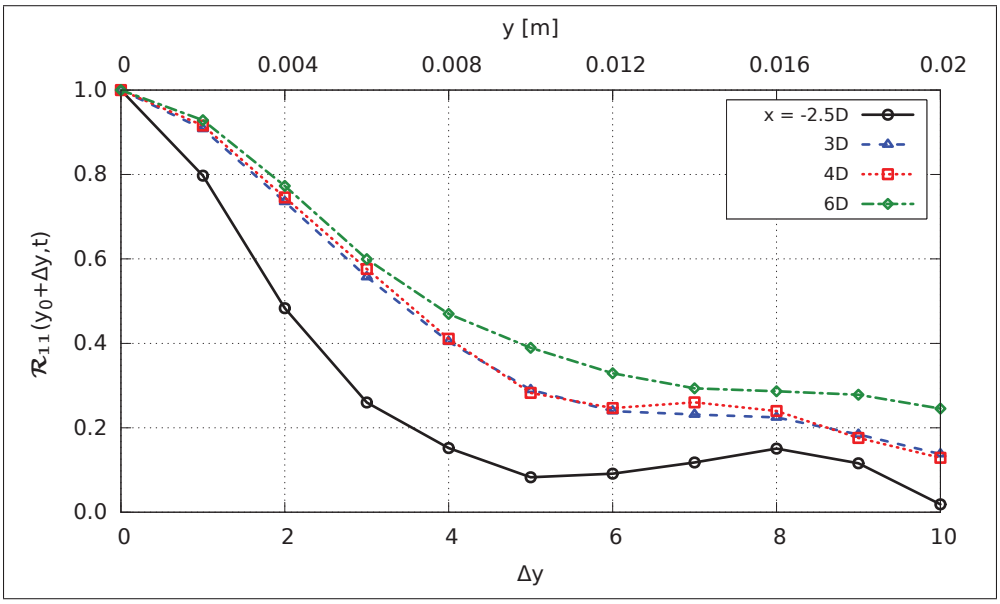

Figure 3.23 Space-time correlations computed in the vertical direction, at four streamwise positions. Results obtained with OpenFOAM for the Ti3 case. For each curve, the markers represent the maximum of the two-point time correlation of the streamwise velocity between probes separated by an integer number of cell widths Δy, indicated by the $x-$axis at the bottom (the top $x-$axis indicates the equivalent value in metres).

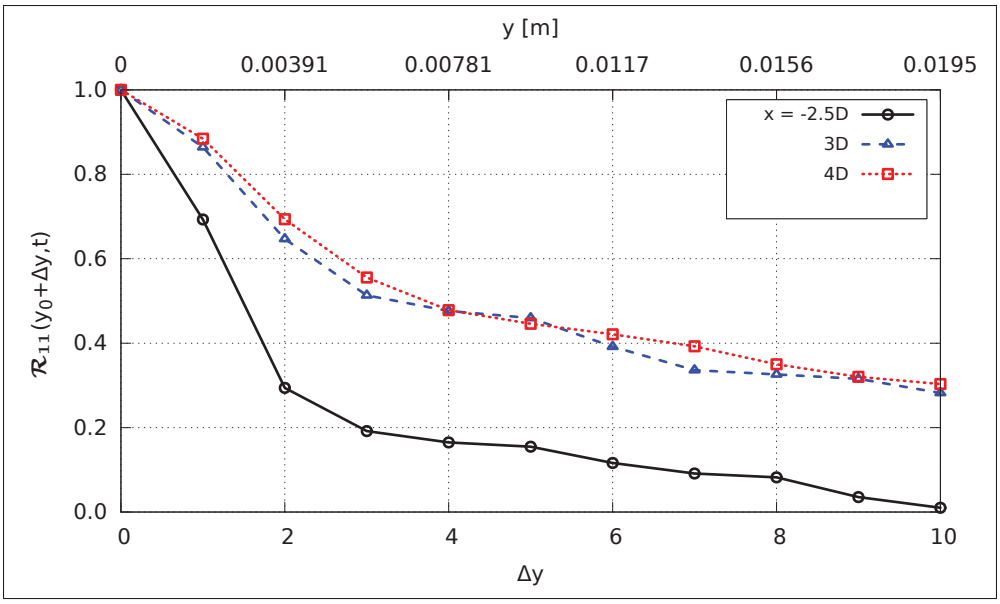

Figure 3.24 Space-time correlations at different streamwise positions obtained with EllipSys3D, case Ti3.

The space-time correlations for the Ti12 case are shown in the Figures 3.25 and 3.26. For OpenFOAM, there is little variation between the correlation curves. For all the $x-$positions, the value of $\mathcal{R}_{11}(y_0 + \Delta y, t)$ approaches to zero at $\Delta y = 10$ cells. This behaviour indicates a slow growth of the integral lengthscales. In comparison, for the results of EllipSys3D, the curves at $x = 3D$ and $4D$ maintain their resemblance but some variations are observed for the curve at 6D. In agreement with what has been observed for the Ti3 cases when comparing the one-point correlations, the curves decay slightly faster in EllipSys3D than for OpenFOAM. Also, they attain a zero value just earlier than their OpenFOAM counterparts. Unfortunately, data at $x = -2.5D$ was not available for this comparison.

In all correlation curves of the Ti12 case there is a fair number of cell points before the correlation falls to zero. This is an indication that the resolution is adequate to resolve the largest fluctuations in the flow. Whether the lengthscale of such fluctuations does correspond to that of the experiment will be seen later when the measured L_1 is compared. Yet, these results support our previous examinations about the adequate mesh resolution employed in the LES to obtain the desired flow characteristic at the target position. Conversely, the faster decay in

the correlation values of the most upstream positions observed in Figure 3.23 (OpenFOAM) and specially in Figure 3.24 (EllipSys3D) for the case Ti3 could be taken as an indication of the need of mesh refinement. This is indeed the case in wall flow simulations. In an analysis of the space-time correlations of shear-flow for different mesh refinements, Davidson (2009) concluded that the largest scales should be resolved with at least 8 cells. He observed that for coarser meshes, the non-linear processes of generating turbulence cannot be sustained and the correlations provide wrong estimates of the largest scales in the flow. Evidently, this does not apply to the purely dissipative flow studied here. Other works (Nilsen *et al.*, 2014), have used a similar resolution (L_1/Δ) as in our Ti3 case obtaining a good agreement between LES and experimental results of two-point correlations in wall flows. Seemingly, the need of refinement is higher in Ellipsys3D due to the faster decay in velocity correlations.

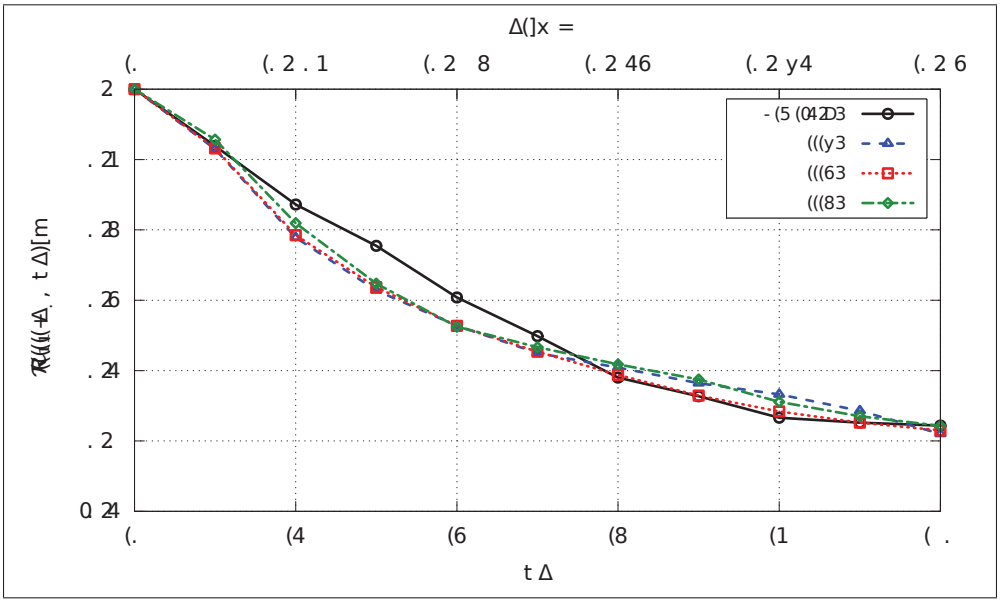

Figure 3.25 Space-time correlations at different streamwise positions obtained with OpenFOAM, case Ti12.

For the Ti12 case, OpenFOAM results in Figure 3.25 show that in all curves there is a fair number of cell points before the correlation falls to zero. In fact, the zero crossing occurs beyond the 10 points in the curve of every x-position. This is an indication that the resolution is adequate to resolve the largest fluctuations in the flow. Also for OpenFOAM, there appears

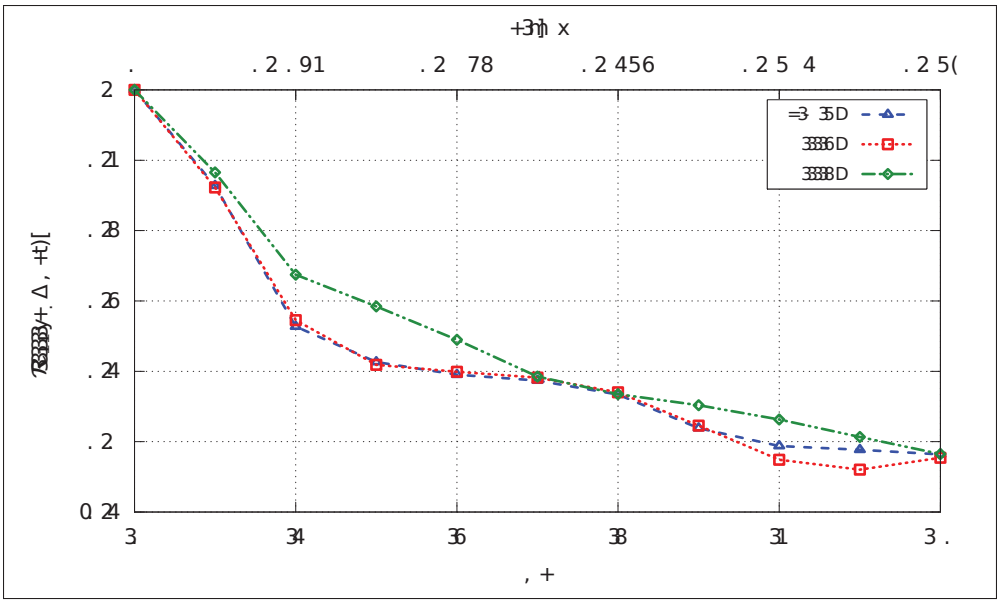

Figure 3.26 Space-time correlations at different streamwise positions obtained with EllipSys3D, case Ti12.

to be a velocity drift or fluctuation starting at about $\Delta y = 5$ cells which slightly increases the correlation value for the velocities at $x = -2.5D$ and is unnoticeable for the rest of the x-positions. In comparison, results of EllipSys3D show that the zero crossing takes place before for all the x-positions shown in the Figure 3.26 (data for the correlations at $x = -2.5D$ was not available). However, note that the change in the correlation slope of the curves indicates a difference in the structure of turbulence in comparison with OpenFOAM.

3.3.9 Longitudinal evolution of the integral lengthscale

The longitudinal evolution of the integral lengthscales is presented in Figures 3.27 (Ti3) and 3.28 (Ti12). In the abscence of measurements for the Ti3 case, computations are compared with the experimental results of L_1 from Thacker *et al.* (2010). In that work, L_1 is obtained from the autocorrelations of data sampled using HWA, employing a method analogous to the one use here (method 3 in Sec. 3.2.4). In Figure 3.27 we can notice that the values of L_1 at the point where turbulence is introduced into the domain do not exactly correspond to those defined in the turbulence box (Table 3.3), in analogy to what is observed for the r.m.s values.

The prediction from each code is larger than $L_{1,\mathrm{B}}$, although more so in EllipSys3D. Conversely, in Figure 3.28 the variation with respect to $L_{1,\mathrm{B}}$ is smaller in EllipSys3D than in OpenFOAM. Gilling and Sørensen (2011) predict a decay in the correlation value from the synthetic box when this is imposed at the inlet or introduced with a turbulence plane, albeit slightly larger for the former technique. In Figure 3.27 we also notice that for the results of OpenFOAM, there is a slight variation of both L_1 and L_2 right just outside the inlet, that after about $1D$ seems to stabilize. This is an indication of the adjustment of the lengthscales of the synthetic turbulence to the LES conditions in the computational domain, next to the resolution restrictions. The behaviour is similar for the lengthscales in EllipSys3D, albeit with a larger variation (specially for L_2) likely due to the increased restriction in the refinement (i.e. $L_{1,\mathrm{B}}/\Delta$) used there. A significant result is that $L_1 \neq 2L_2$ in both codes, which does not support the hypothesis of isotropy in the flow. The sought after value of $L_1 = 0.01$ m at x_D is approximately attained in both codes. The comparison with the measurements seems good considering that the experimental L_1 is not equal to the target value, but the slope drawn by the measurements seems consistent with our results. A fit of eq. (2.61) is also made for L_2 obtained with OpenFOAM. The least-squares fit method applied yields $B_1 = 0.089$, $n_1 = 0.392$ and an origin set at $x_0 = -0.188$ m (upstream) from the inlet. Therefore, according to reference values provided along eq. (2.61), OpenFOAM overestimates the growth of L_2. On the other hand, when a fit is applied to L_1 from OpenFOAM, we get $B_1 = 0.118$, $n_1 = 0.411$ and $x_0 = -0.293$ m, which are slightly closer to the reference values (taking $B_1/2$ and using $L_1 = 2L_2$ in eq. (2.61)). For the measurements, a fit from Thacker *et al.* (2010) produced $B_1 = 0.128$ and $n_1 = 0.375$, for an origin set at $x_0 = 0$. Note that in the results of Comte-Bellot and Corrsin (1971), the analytical fit to the lengthscales is reported with $x_0 = 3.5M$ upstream of the turbulence grid.

For the Ti12, in Figure 3.28 we can notice that the initial fluctuation in the development of L_1 and L_2 is barely noticeable in all cases except for L_1 from EllipSys3D. Yet, in this example, the difference in the results of OpenFOAM and EllipSys3D is more pronounced for L_1. In the Ti3 case a smaller $L_{1,\mathrm{B}}$ had to be used for EllipSys3D while for the present case both codes share a turbulence box with almost equal parameters. Thus, EllipSys3D predicts a larger growth of the

lengthscales than OpenFOAM. Still, this growth occurs at a different rate for the longitudinal and vertical components of the lengthscale since the slopes of the curves L_1 and L_2 computed from EllipSys3D are also appreciably different until about $x = 1D$. In the Ti3 case, the growth of L_1 and L_2 in EllipSys3D diverges less, but there seems to be also an inhomogeneity in the lengthscales before $x = 1D$. In Figure 3.28 we also observe that the value of L_1 at the target position in the EllipSys3D curve matches the desired magnitude of 0.03 m. Measurements are available for the Ti12 case, showing a fair agreement with the results of OpenFOAM although they suggest a larger growth rate. Precisely, the fit of L_2 from eq. (2.61) to OpenFOAM results yields $B_1 = 0.064$ and $n_1 = 0.254$ with $x_0 = -0.068$ m. Also from OpenFOAM, the fit to L_1 produces $B_1 = 0.11$ and $n_1 = 0.342$ with $x_0 = -0.18$ m. Compared to the reference values of eq. (2.61), these parameters indicate that OpenFOAM slightly underestimates the growth of the lengthscales. Nonetheless, we can observe that the isotropy condition of the scales is improved in the results of this code with respect to the Ti3 case since $L_1 \simeq 2L_2$, presumably due to the increased resolution in relation to the eddy size.

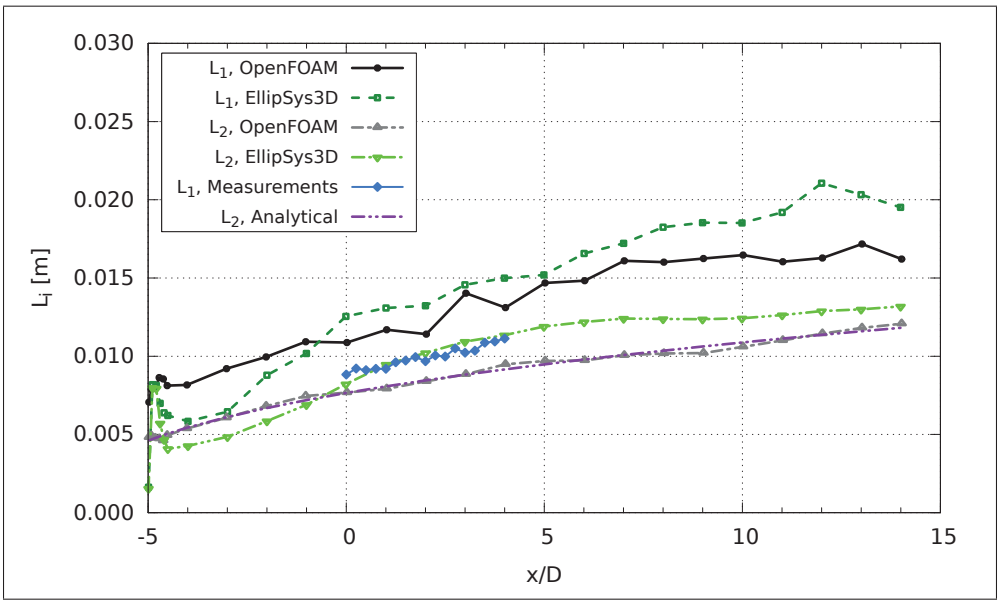

Figure 3.27 Longitudinal evolution of L_1 and L_2 for the Ti3 case.

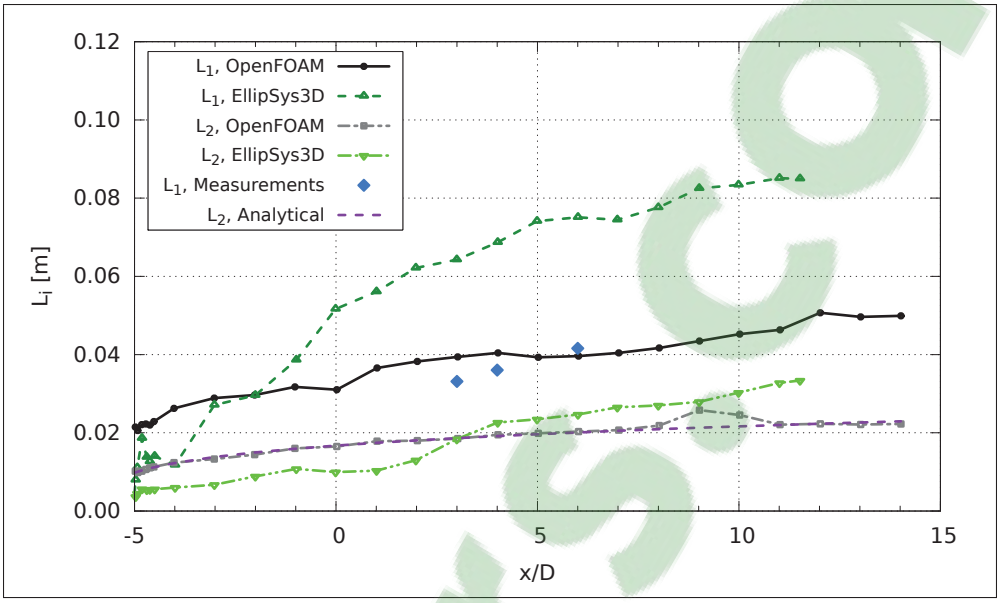

Figure 3.28 Longitudinal evolution of L_1 and L_2 for the Ti12 case.

With the results for the integral lengthscale we show that despite the resolution restrictions, they evolve in a way that the values of L_1 are fairly close to the target magnitudes. Still, the curves reveal a disparate growth (i.e. non-smooth curves) with larger variations than what is observed experimentally (results of Thacker in Figure 3.27). Another factor to consider in the representation of the turbulence scales by the LES is provided by Spalart (2001), where it is mentioned that although an eddy can be resolved with minimum resolutions (he suggests $\ell = 5\Delta$), it cannot be very accurate due to the lack of energy cascade with smaller fluctuations and that in consequence, the resulting eddies will be under the influence of eddy viscosity which depends on the performance of the differencing scheme for short waves.

3.3.10 Taylor lengthscale evolution

In Figures 3.29 and 3.30 we observe a comparison between the longitudinal Taylor lengthscale for Ti3 and Ti12, respectively. λ_1 is obtained using eq. (2.50) with the dissipation equal to $\varepsilon_{tot} = \varepsilon_{\text{SGS}} + \varepsilon_{res}$ from the LES calculation. The Taylor lengthscale of the measurements (Ti12 only) is computed using eq.(2.49) which is also applied to the—resolved—velocity field of the LES computation to obtain λ_1^r. In addition, the Taylor lengthscale can also be estimated from

the macro scale by combining eqs. (2.50) and (2.60), the result is included in the figures as $\lambda_1(L_1)$. The measurements shown in the Ti3 case are obtained from Thacker *et al.* (2010). The analytical expression in the comparison is derived from eqs. (2.48b) and (2.62) plotted with the reference values mentioned for the latter one.

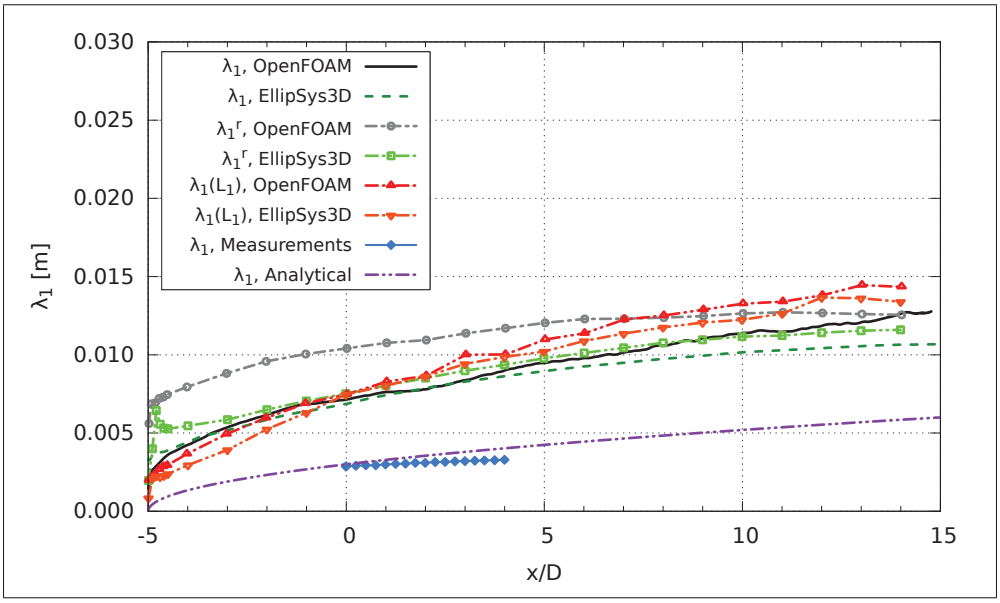

Figure 3.29 Longitudinal evolution of λ_1, Ti3 case.

Figure 3.29 displays a good comparison between the curves drawn by OpenFOAM and EllipSys3D except for λ_1^r. In the case of λ_1, computed from the dissipation, the agreement is somewhat lost towards the end of the domain while the opposite trend is observed for the comparison of λ_1^r. Since λ_1 is estimated from the total dissipation, its agreement is analogous to that shown in Figure 3.17. On the contrary, looking at the results of λ_1^r, computed directly from the resolved velocity field, we can see that the lengthscale is overestimated with respect to the other methods due to the shortage of the intermediate scale fluctuations, in the resolved part, that define the Taylor lengthscale (the shorter the turbulence cascade is, the farther λ_1^r is from the actual λ_1). This effect is more pronounced next to the plane where synthetic turbulence is introduced, where eddies are smaller, and it decreases further downstream. However, this observation alone does not explain why the results from EllipSys3D offer a better match com-

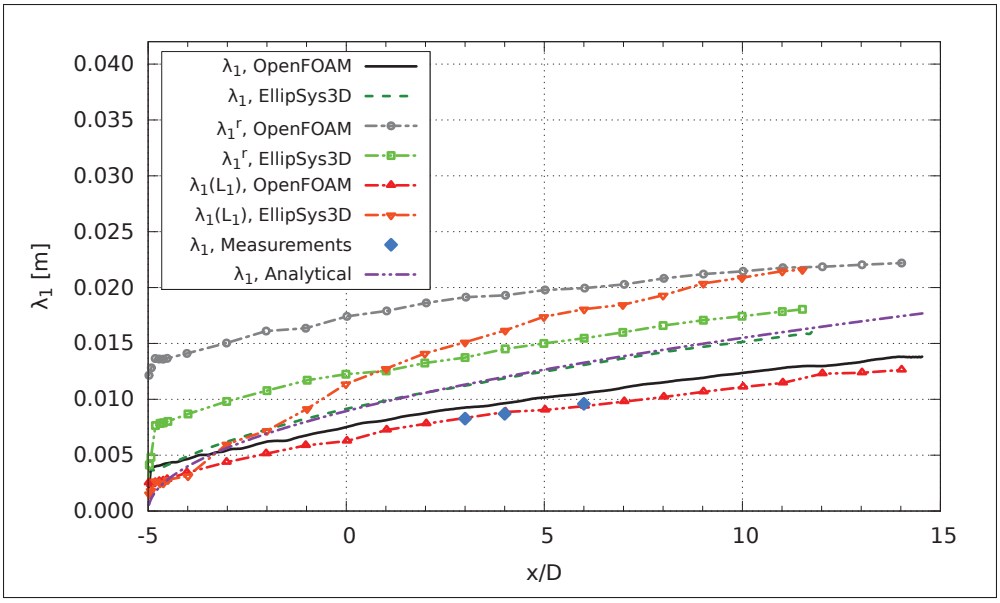

Figure 3.30 Longitudinal evolution of the λ_1, Ti12 case.

pared to λ_1. The information provided by the different correlations help to explain this. As seen in the example provided by Figures 3.21 and 3.22, the autocorrelation curves from EllipSys3D decay faster than in OpenFOAM during the first time steps. Thus, since the Taylor lengthscale is defined as the osculating parabola to these correlations (of the form $y = 1 - \frac{x^2}{\lambda_1^2}$), it is easy to see that the prediction of EllipSys3D will be smaller due to the quicker decay, regardless of how each correlation curve evolves later on. A better result is obtained for $\lambda_1(L_1)$ as the curves obtained from each code compare well to each other, similar to the comparison of L_1 in Figure 3.27. Irrespective of the method used to estimate the Taylor lengthscale, the values obtained are above the measurements, which also agrees with the analytical form. While this can be a consequence of the lack of small scale fluctuations in the synthetic field, it can also be a result of higher Re_λ used in the experiments of Thacker *et al.* (2010) –see below–. This is because λ_i is intermediate in size between the dissipative scales η and L_i, and as η decreases in size with Re_λ (Pope, 2000), λ_i does so accordingly.

For the Ti12 case in Figure 3.30, the comparison between the results of both codes becomes disparate. The main reason is the discrepancy in the total dissipation and its constituents $\varepsilon_{\mathrm{SGS}}$ and ε_{res} in each code, as argued before for Figure 3.18. First, comparing the curves of λ_1

obtained from both codes we see that the results of EllipSys3D are slightly larger throughout, as a result of the larger $\left\langle {u_1'}^2 \right\rangle / \varepsilon$ ratio. Later, the Taylor scale λ_1^r is again above λ_1 due to the higher activity of the SGS parts, although less so for EllipSys3D due to the stronger decay of TI and velocity correlations (see insert in Figure 3.22). As for $\lambda_1(L_1)$, the results reflect the difference in L_1 between codes seen in Figure 3.28, although the curve obtained with OpenFOAM gets noticeably closer to the corresponding λ_1. Interestingly, the OpenFOAM results of $\lambda_1(L_1)$ have a perfect match with the experimental results while the curve of λ_1 matches the analytical prediction very well for the results of EllipSys3D (note that no fit has been made here). The good comparison of the measurements with λ_1 from OpenFOAM is consistent with the same comparison for the dissipation results seen in Figure 3.18.

On the other hand, the two flows reproduced in this work have been identified by means of their respective Re_λ. For each flow, the values used to identify the flows (at the target position x_D) are obtained from: a) for the Ti3 case, the value of λ_1 from Thacker *et al.* (2010) where $\lambda_1 = 2.8 \times 10^{-3}$ m (as λ_1 obtained at a considerably higher U_∞ is used, this Re is given as an approximation) and b) for Ti12, the value of $\lambda_1(L_1)$ at x_D from OpenFOAM, $\lambda_1 = 6.5 \times 10^{-3}$, as the curve fits very well the experimental results (and no measurements are available at x_D). Then, using eq. (2.48b), we obtain the Taylor-based Reynolds numbers $Re_\lambda = u_{rms}\lambda_2/\nu$ of ~ 74 for Ti3 and ~ 113 for Ti12.

3.3.11 Kolmogorov lengthscales

Further down along the fluctuation scales, opposite to the—energy containing—integral lengthscales and below the Taylor lengthscales, we find the Kolmogorov lengthscales, where viscous dissipation occurs. The computation of these scales from the LES is useful as an indirect measurement of the adequacy of the filter location and resolution, as shown below. In Table 3.5 we can see the computed values of the Kolmogorov scales at $3D$ from two methods: $\eta^{[1]}$ is calculated from eq. (2.51) with the dissipation equal to ε_{tot} extracted from the computations. Conversely, $\eta^{[2]}$ is calculated directly from the velocity series, this is, the sampled velocity in the experiments or the resolved velocity $\overline{u}$ in the LES. In this way, $\eta^{[2]}$ is obtained also from

eq.(2.51) but with eq. (2.50) for ε, where λ_1 is calculated from eq. (2.49). Clearly, $\eta^{[2]}$ from the simulations is not the Kolmogorov scale since a turbulence model is being used, but we included only as a reference. As for the measurements, $\eta^{[2]}$ from the data of Thacker *et al.* (2010) cannot be compared with our results since its value scales with Re_λ (Pope, 2000) and such database was obtained at a much higher mean velocity, so the dissipating scales are smaller (we calculated $\eta^{[2]} = 1.51 \times 10^{-4}$ m).

For the Ti3 case, the values of $\eta^{[1]}$ compare well between each code whereas the comparison of $\eta^{[2]}$ shows a larger difference, in agreement with the dispair predictions of λ_1^r as seen in Figure 3.29. It is also seen that $\eta^{[1]} > \eta^{[2]}$, which is due to the fact that the dissipation calculated from the resolved velocity series (eq. (2.50)) predicts a larger dissipation than ε_{tot} computed by the LES, a consequence of the coarse resolution[5]. In the Ti12, the difference between the scales $\eta^{[1]}$ predicted by the codes increases. The comparison with measurements reveals that values predicted by OpenFOAM are closer to the experiment. For $\eta^{[2]}$, the results of both codes are closer to the measured value owing to the better estimation of the dissipation from the time-series, because of the improved mesh resolution (L_1/Δ).

Table 3.5 Comparison of Kolmogorov lengthscale $x = 3D$. $\eta^{[1]}$ is estimated using eq. (2.51) with ε_{tot} computed from the LES while $\eta^{[2]}$ is computed using the resolved velocity series.

		$\boldsymbol{\eta^{[1]} \times 10^{-4}}$ [m]	$\boldsymbol{\eta^{[2]} \times 10^{-4}}$ [m]
Case Ti3	OpenFOAM	6.11	5.34
	EllipSys3D	5.51	4.31
Case Ti12	OpenFOAM	3.15	3.46
	EllipSys3D	4.92	4.14
	Measurements		3.13

In an ideal LES computation, where the filter is set in the inertial range, the resolved fluctuations should be considerably larger than the Kolmogorov scale. Given the filter size of

[5]Considering that the dissipation computed from the time-series is $\varepsilon_{\text{TS}} = \frac{15\nu}{\langle U \rangle^2} \left\langle \left(\frac{\partial \overline{u}_1'}{\partial t} \right)^2 \right\rangle$ (from combining eqs. (2.50) and (2.49)), that $\varepsilon_{tot} = 2(\nu + \nu_{\text{SGS}})(\overline{S}_{ij}\overline{S}_{ij})$ and also that $\nu_{\text{SGS}} > \nu$, if $\varepsilon_{\text{TS}} > \varepsilon_{tot}$, then the coarse mesh favours the overprediction of the (temporal) gradient $\frac{\partial \overline{u}_1'}{\partial t}$ over those of the (spacial) $\overline{S}_{ij}$.

$\Delta \simeq 2 \times 10^{-3}$ m (Ti3) and $\Delta \simeq 4 \times 10^{-3}$ m (Ti12), the SGS filter sizes of the LES are ~ 13.3 and ~ 12.8 times the Kolmogorov lengthscales, respectively (assuming, just for comparison, $\eta^{[2]} = 1.51 \times 10^{-4}$ m for the Ti3 case). Following the rationale for a well resolved LES with implicit filtering from Celik *et al.* (2005), where the ratio of the filter size to η is determined by

$$\frac{\Delta}{\eta} = \frac{Re_{L_1}^{3/4}}{8}, \tag{3.1}$$

with $Re_{L_1} = u_{rms} L_1 / \nu$, the adequate resolutions for our problem (at the target location) would be $\Delta/\eta \sim 2.7$ and $\Delta/\eta \sim 17.6$, demonstrating that our resolution for Ti3 is too coarse but that of Ti12 is more than acceptable, confirming our previous remarks. However, the derivation of eq. (3.1) is based on the assumption that $k_{res}/k_{tot} = 0.8$ suffices to test a well resolved LES, which is inconclusive, as it has been seen in this work. Yet, the dissipation process does not necessarily occurs at scales equal to η, but often at larger scales (Comte-Bellot and Corrsin, 1971; Pope, 2000).

3.3.12 Spectra

To investigate the distribution of turbulence energy along the fluctuating velocity scales, we compute the spectra of the streamwise velocity series; specifically, the Power Spectral Density (PSD). To reduce the noise in the spectral curves, the time-series of each register are divided into eight non-overlapping blocks with an equal number of samples. Then, the PSD of all blocks are averaged to produce the curve at each longitudinal position. As the spectra are calculated from data at a fixed location (sampled in time), the Taylor hypothesis is applied to transform the frequency spectra into a wavenumber spectra using $\kappa_1 = 2\pi f / \langle U \rangle$ where f is f_{acq} for measurements or $f = 1/\Delta t$ for the LES. In this way, it is possible to compare also with the PSD from the synthetic turbulence, which is calculated as the volume average of the spectra computed in the longitudinal direction.

3.3.12.1 Evolution of spectra next to inlet and turbulence plane

PSD are used first to analyze the evolution of the energy distribution next to the inlet. By comparing the results of each code, we can also observe the differences in the spectra caused by the use of distinct techniques to introduce the turbulence. Figure 3.31 shows the spectra for 5 longitudinal positions within the first $1D$ downstream of to the inlet and the spectra of the turbulence box, in OpenFOAM. Besides the turbulent decay, a gradual readjustment of the energy distribution can be seen, where the highest wavenumbers loose energy at a higher rate due to the lack of refinement to reproduce the smallest scales of the synthetic field in the LES. The results for EllipSys3D are shown in Figure 3.32 where the effect of the readjustment of the fluctuations in the flow is evident, seemingly due to the technique employed where turbulence fluctuations are added to a uniform, non-turbulent inflow, as opposed to the introduction of the turbulence field at the inlet used in OpenFOAM. At the position $-4D$ the energy distribution is shown to have stabilized. Comparing the curve at this latest location with that of OpenFOAM, also at $-4D$, we can observe that the energy containing region of the spectra from EllipSys3D extends slightly more towards the high wavenumbers. This is consistent with the indication that the flow in EllipSys3D reaches smaller Kolmogorov scales in the Ti3 case, as seen in Table 3.5. Note that the straight dotted line indicates the characteristic $-5/3$ slope of the inertial range, this is included in these and all the subsequent images of spectra. Also, make note that the maximum wavenumbers yielded by the mesh in each case are $\kappa_c = \pi/\Delta \approx 1571$ m^{-1} (Ti3) and 785 m^{-1} (Ti12) which are easily identified in the figures as since they correspond to the maximum wavenumber of the synthetic fields.

The spectra for the Ti12 case are shown in Figures 3.33 and 3.34, for each code. Unlike the results for OpenFOAM in the Ti3 case, Figure 3.33 does not show a constant decay of energies downstream of the inlet. On the contrary, the energy shown by the spectra in the LES increases with respect to that of the synthetic turbulence. At the position $-4D$, the energy level in the spectra is about the same as in the turbulence box. The loss of energy at high wavenumbers is noticeable but lower than in the Ti3 case. For the results in EllipSys3D, a readjustment of the energy content is evident. The spectrum takes its expected shape, without oscillations, at

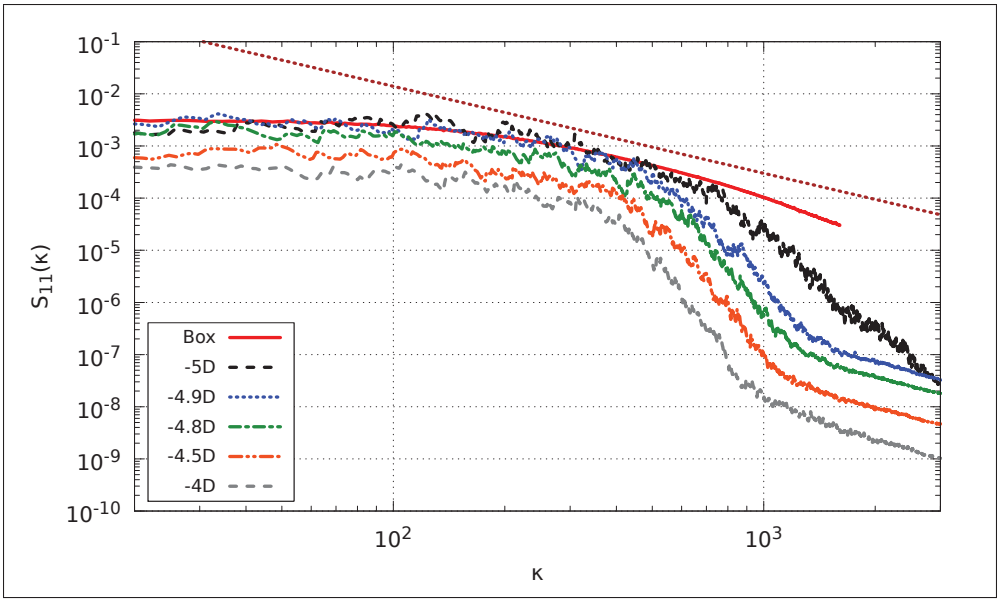

Figure 3.31 Longitudinal evolution of spectra next to inlet in OpenFOAM, Ti3 case. The dotted straight line marks the -5/3 slope of the inertial range.

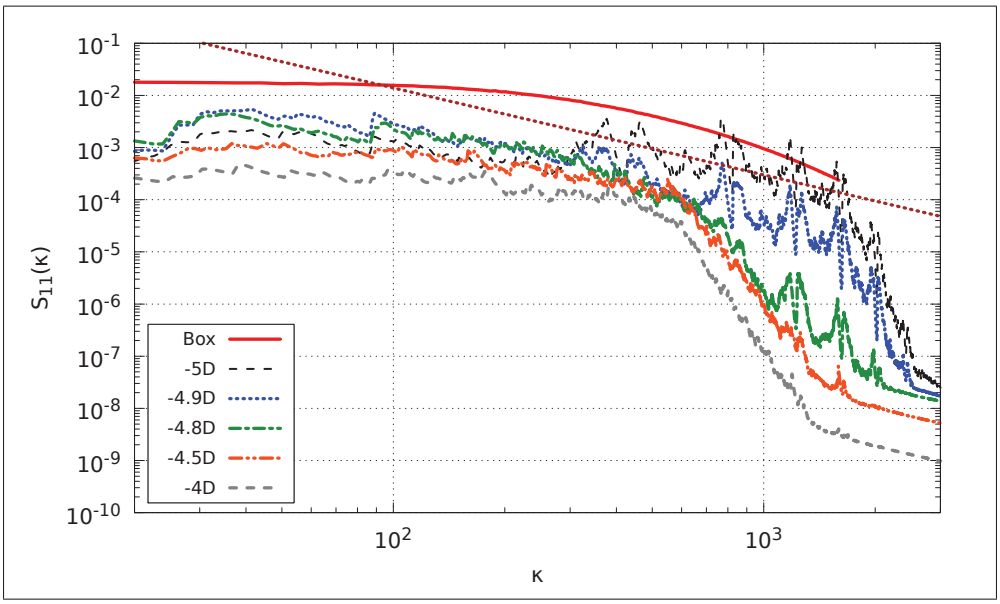

Figure 3.32 Longitudinal evolution of spectra next to turbulence plane in EllipSys3D, Ti3 case.

$-4.5D$. Other figures showing the longitudinal evolution of the spectra for further downstream positions are shown in Sec. 4.2.7, where they are compared with the spectra behind the disks.

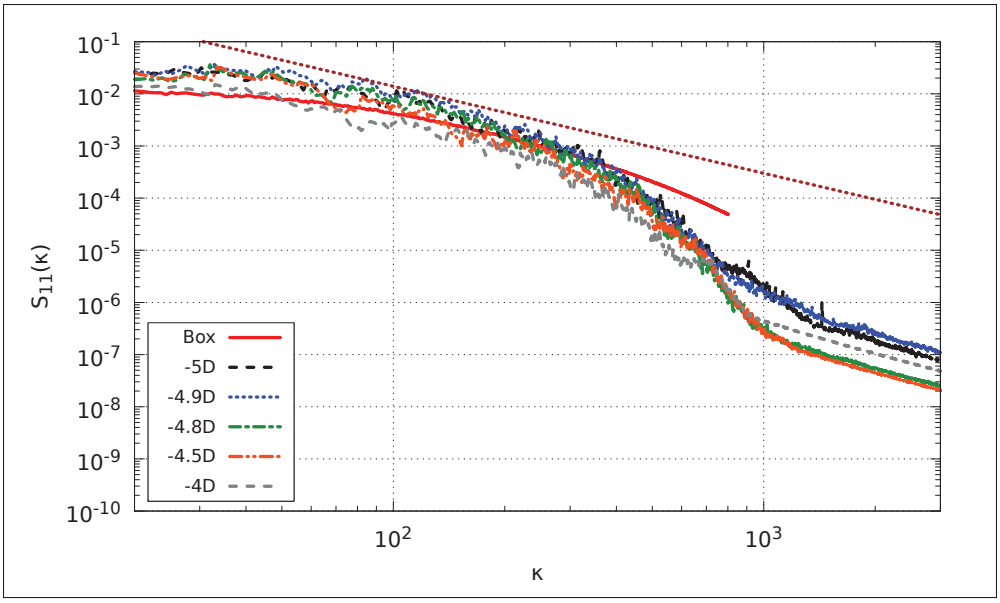

Figure 3.33 Longitudinal evolution of spectra next to inlet in OpenFOAM, Ti12 case.

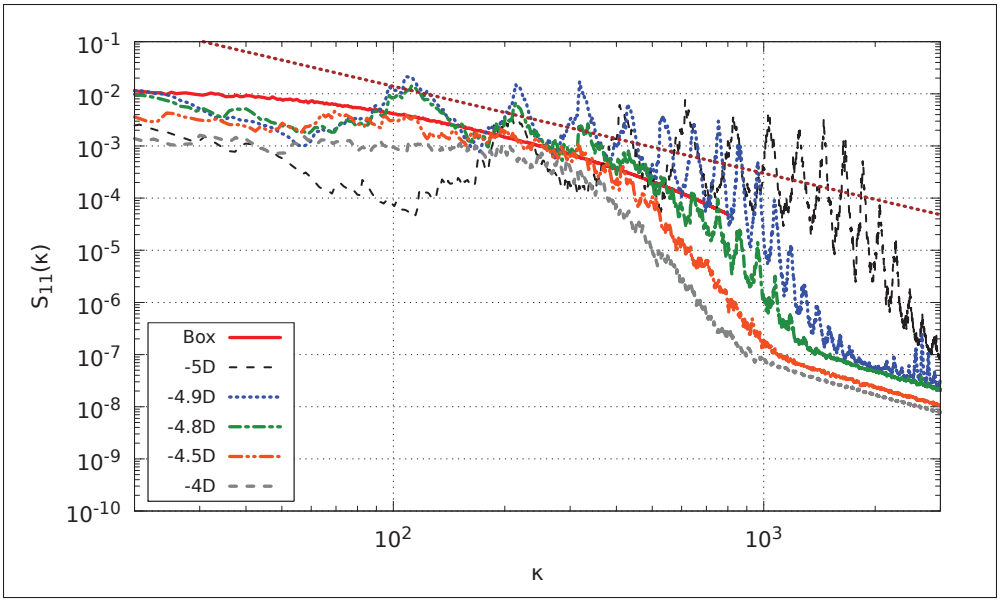

Figure 3.34 Longitudinal evolution of spectra next to turbulence plane in EllipSys3D, Ti12 case.

3.3.12.2 Spectra in the dissipation and energy containing regions

We focus now on the energy distribution shown by the spectra at a particular location. For this, spectra are presented with two different normalizations, one to accentuate the dissipative range and another to highlight the energy containing range (as seen in Pope, 2000). The first scheme is applied to results shown in Figure 3.35 which presents a comparison of the PSD registered by OpenFOAM and EllipSys3D at $x = 3D$, for the Ti3 case. The spectra are also compared to the analytical form eq. (2.54) (using L_1 from OpenFOAM and eq. (2.56)) and to the spectra of the synthetic turbulence. The curves from both codes match very well up to the dissipation region. There, the Figure shows that the peak of dissipation in EllipSys3D occurs at a higher wavenumber than that of OpenFOAM, consistent with our previous observations regarding the dissipation of the resolved flow. Because of its higher TI, the spectrum of the synthetic field is above the LES results (only turbulence boxes from OpenFOAM are used for comparison).

The same comparison is made in Figure 3.36 for the results of case Ti12, including also the spectra computed from the measurements. The comparison with the experimental results reveals that the SGS filter is well placed, within the inertial range, leaving most of the dissipation to be carried by the subgrid model. Meanwhile, for the resolved scales of the LES, the difference in the wavenumbers where dissipation reaches its maximum is reduced with respect to the Ti3 case (due to the higher resolution), but still larger for EllipSys3D. This feature validates the previous assessment regarding the dissipation of the resolved field in EllipSys3D: it is larger in magnitude and it also extends to smaller scales (where the slightly smaller cell size can be a contributing factor). In the Ti3 case, the disparity in the dissipation peaks is due to the fact that lack of mesh refinement hinders the apparition of an extended turbulence cascade, which in turn increases the impact of the differencing scheme for the transport of the small fluctuations (Spalart, 2001). This effect is reduced in the case of Ti12 because of the improved resolution of L_1/Δ. Yet, the stronger TI decay in Ellipsys3D near the turbulence plane suggests that the numerical dissipation is higher than in OpenFOAM despite both having a similar resolution ratio L_1/Δ.

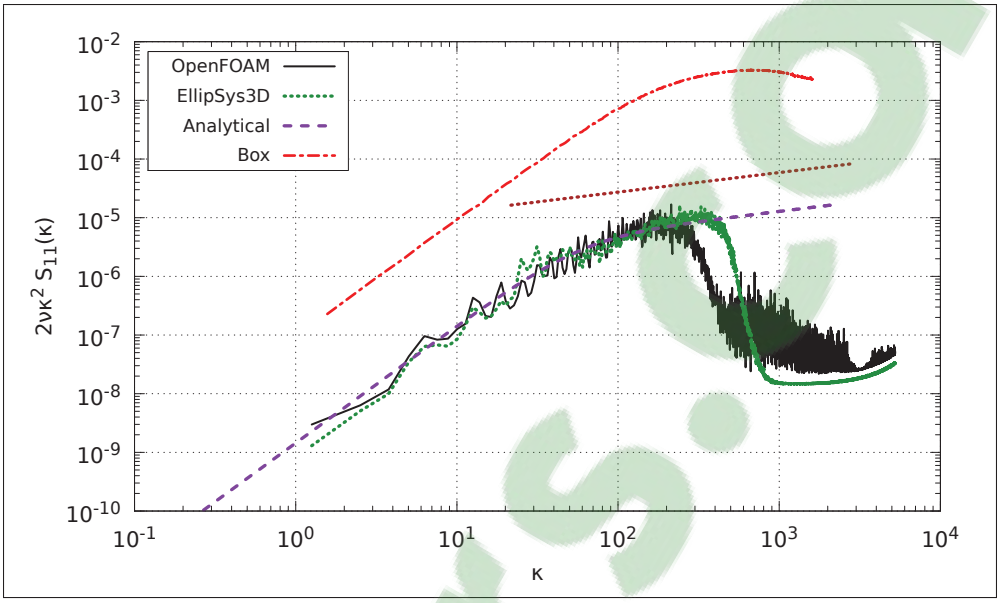

Figure 3.35 Power spectral density spectrum normalized to emphasize the dissipation range, Ti3 case. The Box spectrum corresponds to the one used in OpenFOAM. Also, the analytic spectrum follows eq. (2.54) with parameters extracted from the OpenFOAM results.

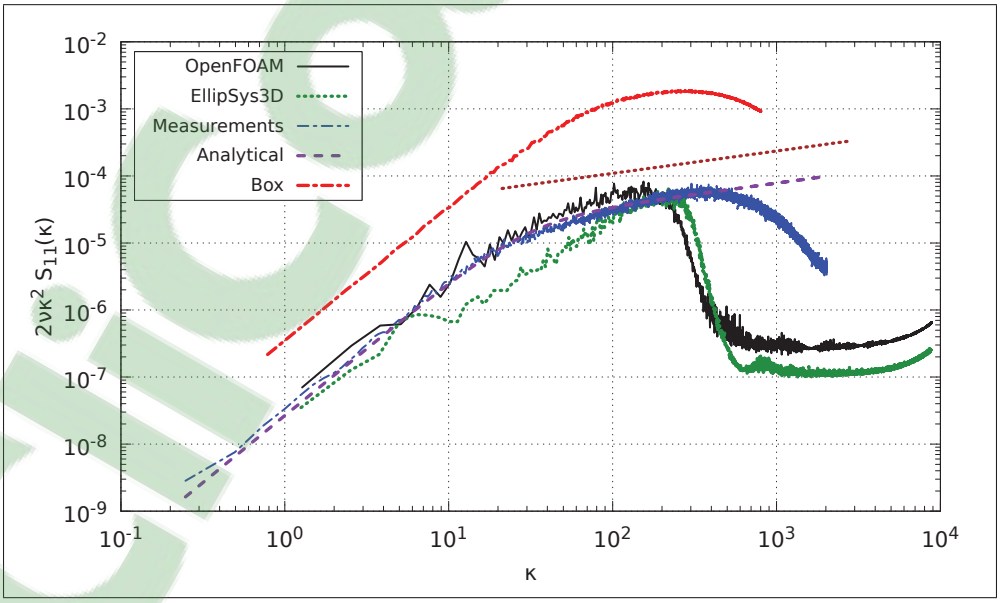

Figure 3.36 Power spectral density spectrum normalized to emphasize the dissipation range, Ti12 case. The analytic spectrum follows eq. (2.54) with parameters extracted from the measurements.

Figures 3.37 and 3.38 show the spectra normalized by the streamwise turbulent kinetic energy k_1 and L_1. In this way, the spectral curves are level to the energy containing range, allowing to compare the distribution of energy along the fluctuation scales. For the case Ti3 in Figure 3.37 we notice the lack of a clear inertial range, something expected due to the very low Re_λ. The lack of this feature was observed by Mydlarski and Warhaft (1996) for flows with $R_\lambda \sim 50$. These results contrast with the distinct scaling range seen in Figure 3.38 (also discernable in Figure 3.36), although the slope of the curve of OpenFOAM in this region is somewhat closer to the analytical and experimental results than the prediction of EllipSys3D. In Figure 3.38 we also notice a displacement of EllipSys3D results to higher wavenumbers, with respect to Figure 3.36. This is due to the appreciably larger integral lengthscales predicted by this code, as seen in Figure 3.28.

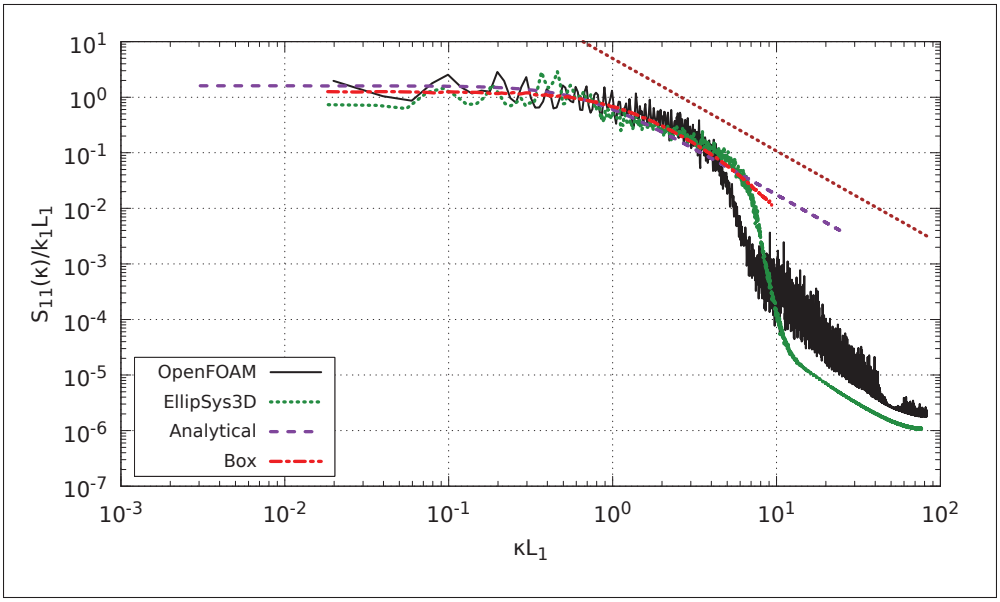

Figure 3.37 Power spectral density spectrum normalized to level out the energy containing scales, Ti3 case.

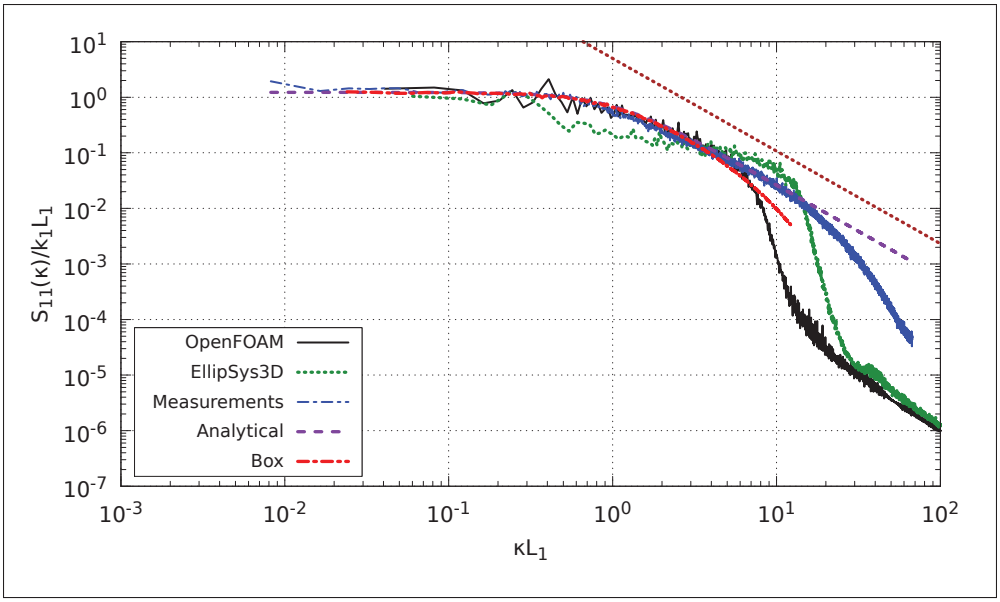

Figure 3.38 Power spectral density spectrum normalized to level out the energy containing scales, Ti12 case.

3.4 Summary and conclusions

A methodology was developed and implemented with the goal of replicating the inflow characteristics for a subsequent computation of wakes. Specifically, Large-Eddy Simulations (LES) were performed to reproduce the reference parameters of two instances of a flow of decaying isotropic turbulence created in a wind tunnel. In each case, the flow had streamwise turbulence intensities (TI) of approximately 3% and 12% and corresponding longitudinal integral scales (L_1) of 0.01 m and 0.03 m, measured at 0.5 m from the turbulence grid. While the Mann algorithm is used to create synthetic turbulence, the LES simulations have been carried out employing OpenFOAM, with the addition of EllipSys3D for the purpose of comparison. The numeric schemes used in each code have not been modified to resemble each other, so each platform is used a more typical, distinctive setting. Indeed, while OpenFOAM employs the more common Smagorinsky SGS model, EllipSys3D uses a setup that has been employed in various works on wake simulation and production in wind parks, including the use of a mixed-scale SGS model. In this way, the applied procedure was to reproduce the reference

flow parameters separately for each code, while the results of the evolution of turbulence characteristics are later compared.

Turbulence structures measured in the experiments were much smaller than the volume of the computational domains. For this reason, the cell resolution with respect to the integral scales was very restricted in the synthetic field as well as in the LES, particularly in the region where turbulence is introduced. Moreover, due the approach employed, synthetic turbulence fields imposed in the LES domains contained very high turbulence intensities. In consequence, the assumption of the Taylor hypothesis is admittedly crude. Despite these limitations, the turbulence characteristics of the experimental flows could be reproduced with both codes at the reference positions. It was also shown that in OpenFOAM the employed methodology yields results in agreement with the predictions of grid turbulence. Still, noticeable differences in the evolution of turbulence parameters computed in each code were encountered. In consequence, distinct strategies had to be employed to achieve the desired turbulence characteristics. This was in part expected as different SGS models as well as numerical strategies and implementations are used in each program. In particular, it was found that the TI decay computed by the LES solver in EllipSys3D was stronger that the one in OpenFOAM. A discussion was presented about the probable reasons that cause this difference.

A study of the evolution of turbulence characteristics was presented, comprising the longitudinal development of large to intermediate fluctuating scales (integral and Taylor scales). For the integral lengthscales, it was found that values computed in Ellipsys3D fluctuate more after turbulence is introduced, while also attaining larger values in comparison to the predictions of OpenFOAM. The comparison of Taylor scales brought about small differences between the results of each code, but only in one case the results did compare well with the measured quantities. This is due to the very limited cell resolution of fluctuating scales in the low TI simulations (where lengthscales are the smallest), so the turbulence cascade is cut short limiting the apparition of structures below the macro scale. This in turn hints towards disparities in the performance of the interpolation schemes in each code (likely those used for the velocity convection, discussed in Sec. 3.3.6). It is also argued that in the absence of a *very active*

subgrid model (due to the lack of small scales), the influence of the numerical dissipation increases, specially with respect of the accuracy of the representation of the large scales. These observations are supported by the results of the instance of the flow with a better resolution of turbulence lengthscales (case Ti12). In those computations, results show a good agreement exists between the integral and Taylor scales, as well as between the estimation of the dissipation of the LES and the value extracted from the measurements. It is also found that in those cases, each code presents a noticeably distinct handling of the numerical dissipation. Resolution of the large scales is also studied by means of one- and two-point correlations, where it can be seen that although resolution does not largely varies in each code, differences in the shape of the correlation curves indicate some disparities in the development of the turbulence structures. These observations complemented by the analysis of spectra at different locations in the two codes.

CHAPTER 4

STUDY OF WAKE TURBULENCE CHARACTERISTICS

The methodology to produce turbulence inflows is used next to a rotor model to reproduce wake turbulence fields. Specifically, the two instances of the decaying, homogeneous flow described in Chapter 3 are used as an inflow to reproduce wind tunnel measurements made along the wakes produced by porous disks with two different solidities. Simulations are performed with LES and the uniformly loaded AD implementation in OpenFOAM. Making use of an analogous approach, computations are also carried out with EllipSys3D, a reference numerical platform for wake simulations. Additionally, results from previous work made with RANS are included. General characteristics of the wake, like the velocity field and the turbulence kinetic energy are evaluated. More importantly, features such as the turbulence dissipation and the effect of shear on the integral lengthscales are assessed. Likewise, changes in the LES modelling in both codes along the wake with respect to the freestream flow also studied.

4.1 Model description

The experimental data used in this Chapter were collected in the campaigns described in the previous Chapter, Sec. 3.1. Averaged quantities at $x = 2D$, $4D$, $6D$, $8D$ and $10D$ from the disk centre were obtained with LDA while time-series obtained by HWA at $x = 3D$, $4D$ and $6D$ are used to compute other turbulence features. Different streamwise velocities were used while employing the different measurement techniques. Based on this velocity (U_∞) and D, the Reynolds number used for HWA is $\mathrm{Re}_D \approx 20400$, whereas for LDA $\mathrm{Re}_D \approx 40800$ (Ti12) and 68000 (Ti3). The main properties of the porous disks used in the experiments are listed in Table 3.1. These disks are modelled using the AD technique (for a uniformly distributed thrust) described in Sec. 2.3.1. As mentioned there, the forces that comprise the AD are distributed in the streamwise direction using the convolution with a Gaussian distribution (eq. 2.14) to avoid the oscillations that otherwise appear in the pressure and velocity fields. The value of σ is defined differently in each code, causing the thickness of the disk to be slightly different:

- *In OpenFOAM* $\sigma = 2\Delta x$ so the disk thickness is equal to $12\Delta x$ for all cases. Therefore, the magnitude of the thickness will change according to the cell length.
- *In EllipSys3D* the distribution is done using $\sigma = 0.1D/\sqrt{2}$ so the thickness is constant in absolute dimensions, regardless of the cell length. In the Ti3 case the disk is formed by $21.72\Delta x$ while for Ti12 the value is $10.86\Delta x$.

As in the free-flow case, measuring probes to record time-series data in the LES are located in the longitudinal direction, distributed over the cross-section of the computational domain. In the wake simulations, measuring positions are added to those described in Sec. 3.2.3, particularly over the region covered by the AD. Figure 4.1 shows the locations of these probes over the cross-section of the domain. The distribution of probes is repeated at the same $x-$positions defined in the previous chapter.

In a study by Sumner *et al.* (2013), RANS computations were performed to reproduce the same LDA measurements used in our study. In their work, a RANS turbulence model, labeled as "Sumner and Masson", based on modifications to the $k-\varepsilon$ model of El Kasmi and Masson (2008) is proposed. While the latter model attempts to correct the well known overestimation of turbulent stresses (Réthoré, 2009) by introducing a dissipative term proportional to the turbulence production in the ε–equation, Sumner and Masson pursue the same objective by neglecting some terms of turbulence production also in the vicinity of the disk (the cylindrical volume centred at the AD, extending $\pm 0.25D$ in the axial direction), obtaining a good comparison for the velocity deficit and k along the wake of the disks. We include the results obtained with this model along with our computations as they serve as a reference element of the capabilities of an industry standard to reproduce the evolution of turbulence features in the wake. Note that since the simulations of Sumner *et al.* (2013) were made for only half of the wake, we show their results (velocity deficit, k and ε) duplicated—mirrored—in the vertical direction.

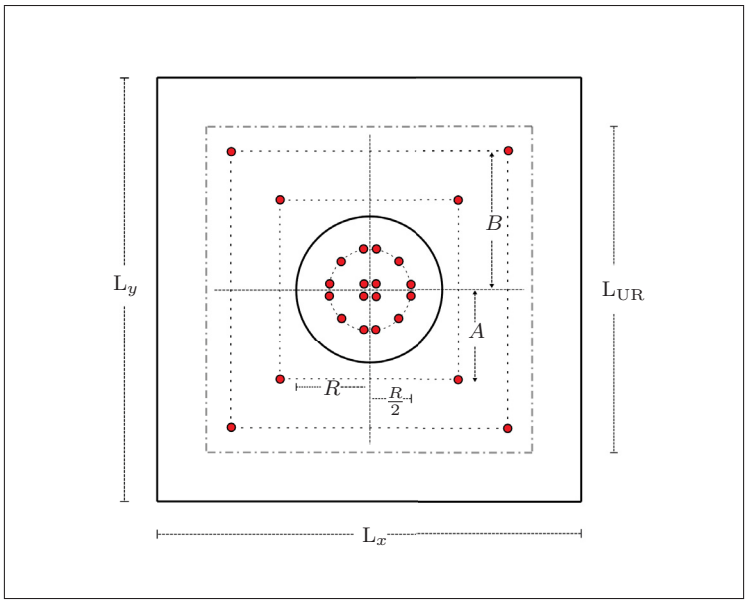

Figure 4.1 Locations of probes over a cross-sectional plane of the computational domain, represented by the small circles. L_{UR} is the side length of the uniform region used in OpenFOAM (see Table 3.2) whereas $A = 0.07$ m and $B = 0.15$ m. The location of the AD of radius R is also shown in the figure. The four circles around the middle correspond to the centremost cell centres

4.2 Results and discussion

We present the results of our computations of different quantities focused on the turbulence characteristics along the wakes produced by the different inflows and disk thrusts. A visualization of each of these wakes is presented at the end of this Chapter by means of planes representing velocity and vorticity fields in the streamwise and vertical directions.

4.2.1 Velocity deficit

The first comparison is made from the results of the streamwise velocity deficit along the vertical direction at different longitudinal positions. The results are normalized by the freestream velocity at $y = 1.5D$. In the Figures 4.2 and 4.3 we see the results for the high and low solidity

disks under the inflow Ti3, $C_T = 0.42$ and $C_T = 0.62$ respectively. The agreement to the experimental results is very good in both codes, with the larger difference observed around the shear layer from the disk edges, specially for the disk with higher thrust. In that case (Figure 4.3), EllipSys3D offers a slightly better match in such region, although the last position indicates that it predicts an anticipated wake recovery (this is discussed in the next section). This feature can also be appreciated in the results of Sec. 4.2.8.

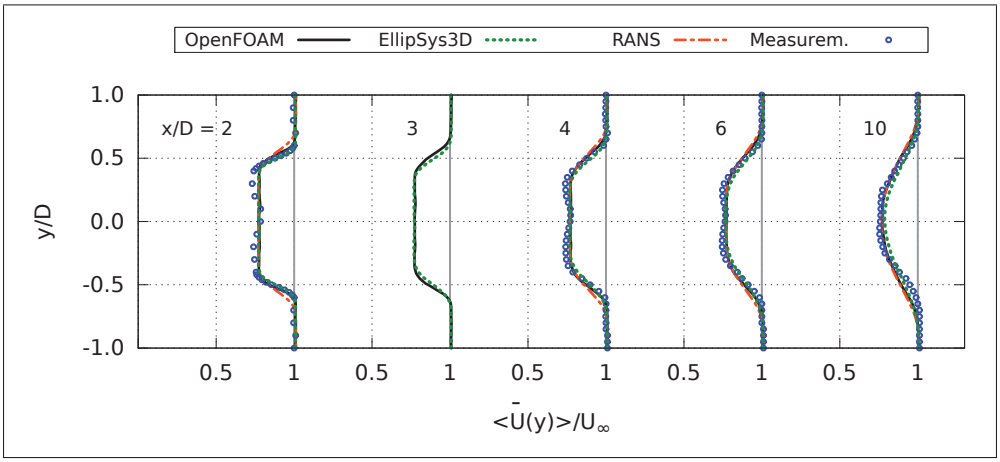

Figure 4.2 Vertical profiles of velocity deficit behind the disk $C_T = 0.42$, Ti3 case.

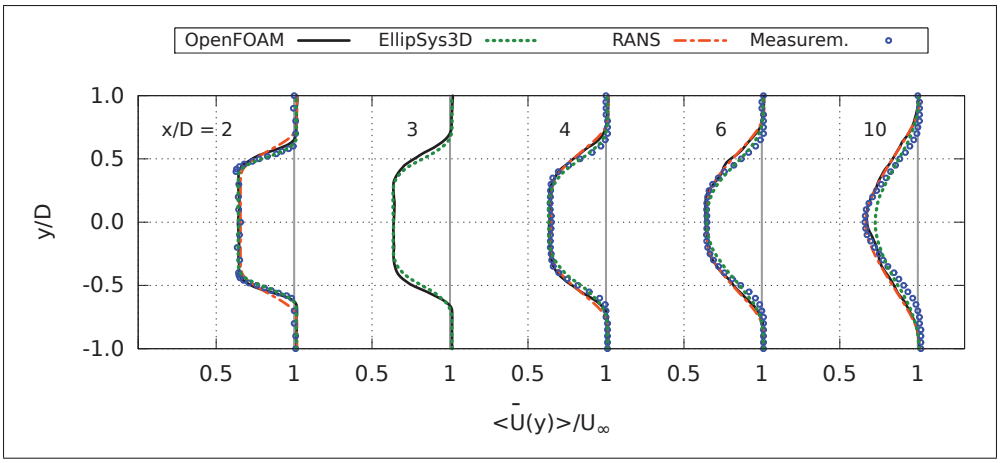

Figure 4.3 Vertical profiles of velocity deficit behind the disk $C_T = 0.62$, Ti3 case.

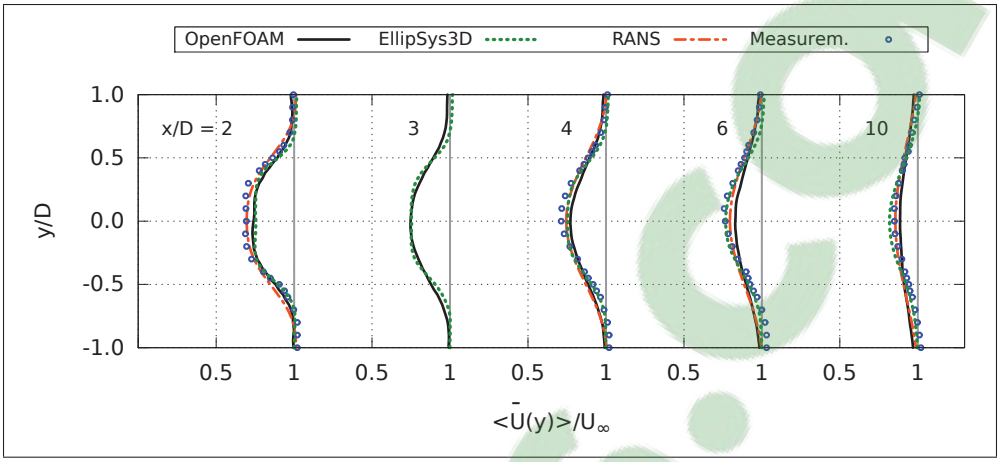

Figure 4.4 Vertical profiles of velocity deficit behind the disk $C_T = 0.45$, Ti12 case.

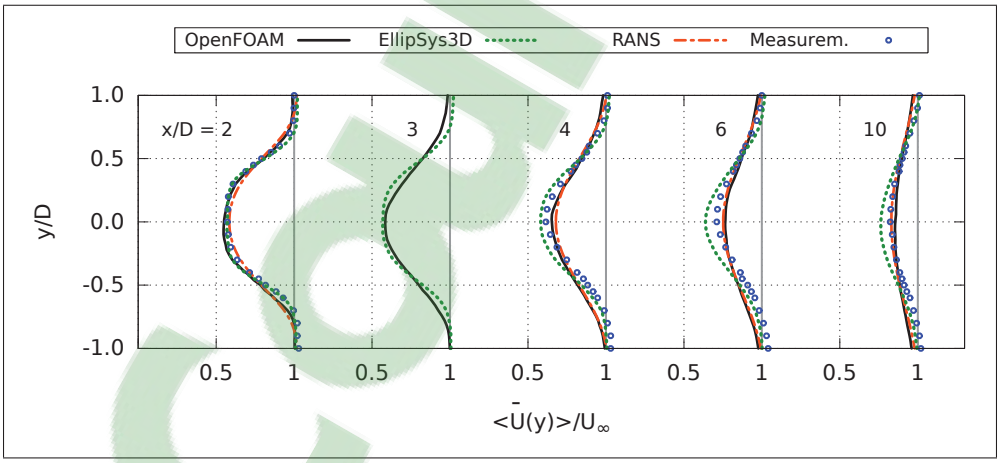

Figure 4.5 Vertical profiles of velocity deficit behind the disk $C_T = 0.71$, Ti12 case.

In the case of the Ti12 inflow, Figures 4.4 and 4.5 show a minor reduction in the agreement of the OpenFOAM results with the measurements, with the largest differences observed also in the shear layer region. Meanwhile, the prediction of EllipSys3D is marginally better for the disk $C_T = 0.45$. For the disk $C_T = 0.71$, the predictions of each code commence to differ when moving further into the far wake, specially close to the centreline, where the recovery indicated by OpenFOAM occurs slightly faster than in EllipSys3D. At $x = 4D$ and $6D$ the measurements fall mostly in between the result of each LES computation, whereas at the last position ($x =$

$10D$), OpenFOAM results compare better to the measurements by a small margin. Remarkably, the results of RANS are almost identical to those of OpenFOAM. As previously noted by Sumner *et al.* (2013), the blockage effect was observed to be more evident in these cases as the normalized velocity outside the wake is higher than the inflow reference value.

4.2.2 Turbulence kinetic energy in the wake

It is assumed that the wake created by the disks augments the turbulence level with respect to the ambience value. Having studied the evolution of the TI and k in the decaying-HIT, we investigate now how the computations of the added turbulence compare to the experimental results within the wake. Figures 4.6 and 4.7 show the profiles of k (this is, $k_{tot} = k_{\text{SGS}} + k_{res}$ for the LES) at different downstream positions along the wake, when the inflow of the case Ti3 is used. There, we observe that the results from OpenFOAM match quite well the measured turbulence levels. This is seen behind both disks except perhaps for the last longitudinal positions with the highest thrust AD. Yet, we notice that except for the nearest position to the disk, both LES predict a higher difusion of shear turbulence in the crosswise direction, an effect that is increased with the disk thrust. The results from EllipSys3D predict a higher turbulence level, which does not seem to arise from inflow turbulence since it is only marginally higher in this code compared to OpenFOAM, as seen in Figure 3.7 (where the difference in TI is about 0.58% at x_D and 0.42% at $x = 10D$). Instead, the higher levels in EllipSys3D seem to be directly caused by the added turbulence in the wake, since the difference between codes increases with the thrust of the disk and the levels of k outside the wake (i.e. $y = \pm 1.0$) are very similar (recall that EllipSys3D fit the decay outside the wake very well as shown in Figure 3.7). In the simulation with disk $C_T = 0.42$, the difference in turbulence energy with respect to the measurements and OpenFOAM seem to increase when moving away from the disk. The wake seems to reach a full turbulent state also faster in EllipSys3D, as k increases towards the centreline at a higher rate. It is difficult at this point to identify with clarity the origin of the higher turbulence arising at the shear layer in EllipSys3D. Although a noticeable difference has

been observed in the numerical dissipation between the two codes, this does not seem to be the cause of the difference in the estimation of k.

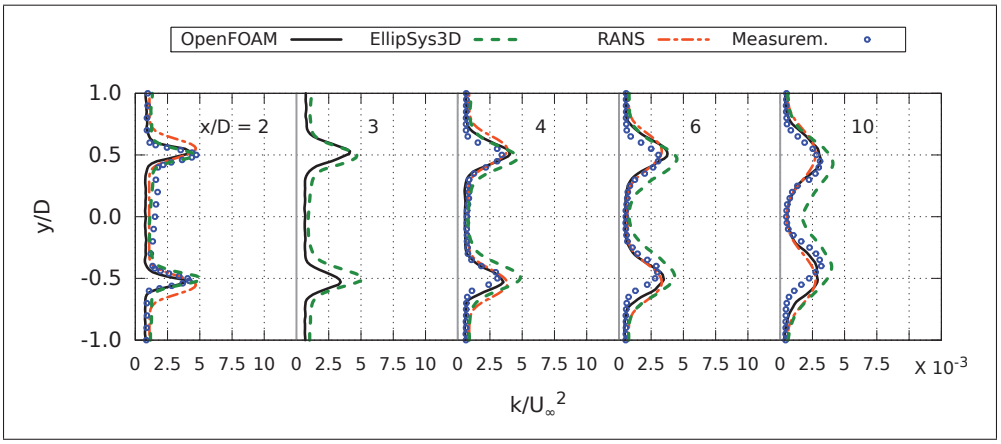

Figure 4.6 Vertical profiles of k behind the disk $C_T = 0.42$, Ti3 case.

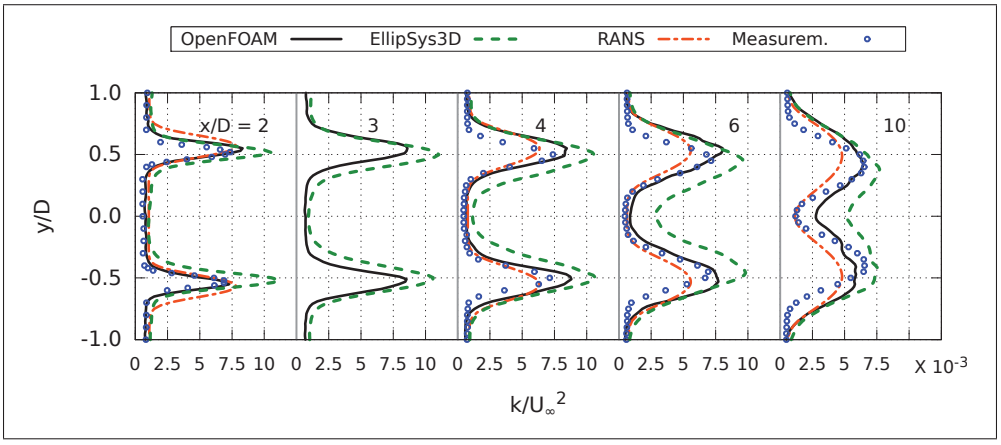

Figure 4.7 Vertical profiles of k behind the disk $C_T = 0.62$, Ti3 case.

For the disks in the Ti12 case, Figures 4.8 and 4.9, the profiles obtained with OpenFOAM compare mostly well with the experimental data, although the simulations from this code overestimate k near the disk. On the other hand, EllipSys3D matches the measurements just behind the disk ($x = 2D$), but falls short in the predicted k for the other positions. At the same location, OpenFOAM overestimates the turbulence. In these two figures, we observe that the shear layer originating at the edges of the disk is mixing faster with the ambience turbulence

compared to the Ti3 inflow. Indeed, the effect of shear prevails deeper into the wake in the LES with the highest thrust disk, whereas it is mixed faster into the ambience turbulence when the thrust is lower. The turbulence level in the wake is lower in EllipSys3D likely due to the lower level of ambience turbulence level compared to OpenFOAM. Downstream of the target position $x = 0$ where $\mathrm{TI} \simeq 12\%$ in both codes, the difference between the values of EllipSys3D and those yield by the measurements and OpenFOAM increases rapidly. This is illustrated in Figure 4.10, which shows the TI decay without the turbines (the local level of turbulence at the downstream position can be identified faster here than in Figure 3.8 of the previous Chapter, where the origin of the curve of EllipSys3D is displayed shifted at $x = -5D$, see Sec. 3.3.3 for details). Moreover, although in Figure 4.10 the free-flow simulation with OpenFOAM seems to adjust very well to the measured TI decay, the results in Figures 4.8 and 4.9 contradict this comparison, as the computed level of k at $y = \pm 1.0$ is higher than in the measurements, except for the farthest positions. As for the RANS computations, overall comments are presented within the discussion of results of turbulence dissipation.

When comparing the decay of k in the wake with that of the velocity deficit, we notice that the former is slower than the latter. Interestingly, this is consistent with various studies in the ABL (Vermeer *et al.*, 2003) where the same behaviour is observed. In a comparison between LES computations of a wake created by an actuator line with a homogeneous, non-turbulent inflow with OpenFOAM and EllipSys3D (with SGS Smagorinsky in both cases[1]), Sarlak *et al.* (2014) observes that EllipSys3D predicts a slower wake recovery as well as a lower k_{res} far in the wake ($x > 10D$) than OpenFOAM. In those simulations, for the solution of the convective terms EllipSys3D uses the $90\%/10\%$ blend of central and QUICK schemes, respectively, while OpenFOAM uses a purely central scheme. It is worth to notice that in that work, the OpenFOAM simulations were repeated using a blended interpolation scheme analogous to the one applied in EllipSys3D, without observing a large difference nor a trend compared to the results of the central scheme. Moreover, when the same comparison is made with the rotor positioned in the wake of two other—aligned—rotors (to simulate a turbulent inflow), the trends reported

[1] Besides this, the PISO algorithm was used in both codes. Yet, other differences are found with regard of the airfoil data interpolation along the blades. See reference for more information.

in the laminar case are reduced or reversed, but in this case the differences between the results of each code could be considered negligible. Furthermore, EllipSys3D seems to predict a more stable vortex sheet than in OpenFOAM, as in the latter the wake destabilizes much earlier ($x \sim 7.5D$ vs. $x \sim 17.5D$), which could be due in part to the different methods for the interpolation of airfoil data along the blade. We present a similar comparison in Sec. 4.2.8, where vorticity contours from each code are shown. However, this behaviour is not observed in our results.

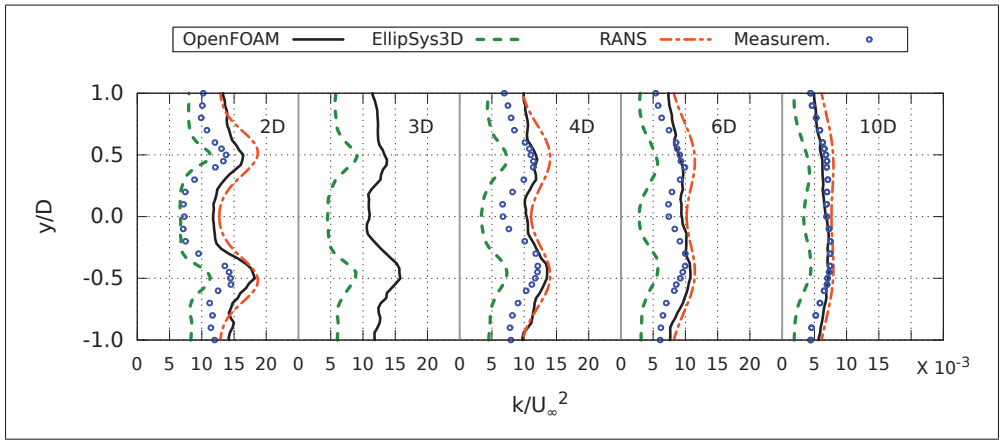

Figure 4.8 Vertical profiles of k behind the disk $C_T = 0.45$, Ti12 case.

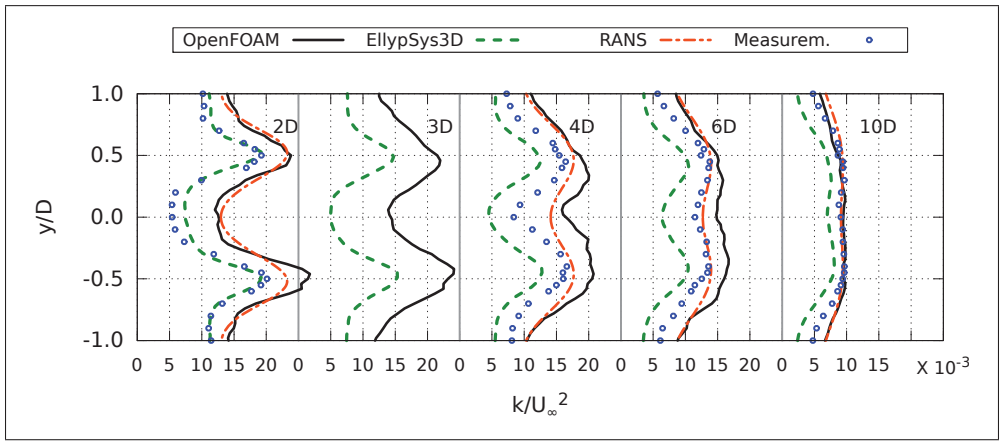

Figure 4.9 Vertical profiles of k behind the disk $C_T = 0.71$, Ti12 case.

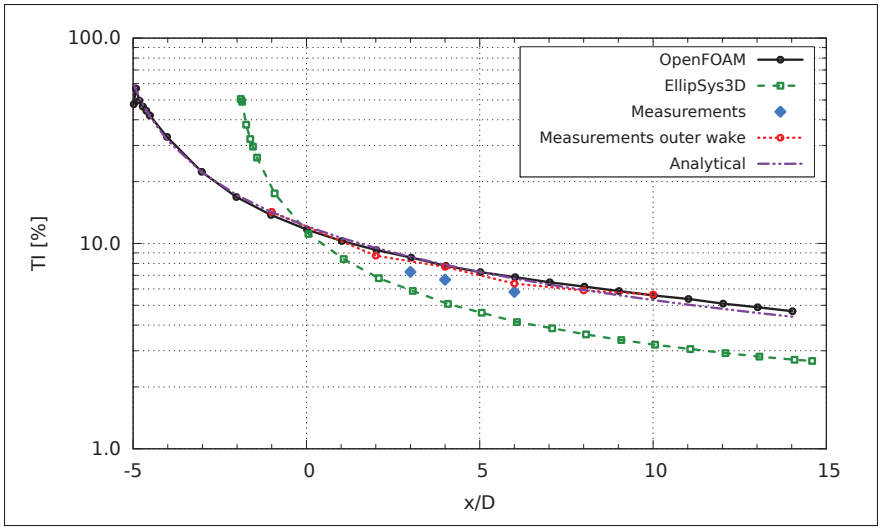

Figure 4.10 TI decay for the Ti12 case without disks.

It is also noticed that some inhomogeneities appear in the results of both codes, very appreciable in the simulations of OpenFOAM with the Ti12 inflow. Although this feature could evidence the need of creating synthetic turbulence that would cover longer simulation periods, we also notice that the profiles in EllipSys3D look in general smoother. Therefore, these fluctuations seem to arise from a more enduring footprint of the turbulence structures of the inflow turbulence in OpenFOAM. This can be observed in the vorticity contours of the corresponding wakes (Figures 4.39 and 4.43 at the end of this Chapter), where it is certainly difficult to discern the outline of the shear structures from those of the ambience turbulence, unlike the case of EllipSys3D.

4.2.3 Turbulence dissipation in the wakes

The profiles of turbulence kinetic energy dissipation in the wakes with the Ti3 inflow are compared in Figures 4.11 and 4.12. There, the dissipation corresponds to $\varepsilon_{tot} = \varepsilon_{res} + \varepsilon_{\text{SGS}}$ in the LES computations. Remarkably, very little difference is observed in the dissipation computed by each code, unlike the previous results for k. Even with the RANS model differences are small, as the curves differ only at $x = 2D$ where it predicts a higher dissipation within the shear layer. In light of the difference noticed in the computation of k for the Ti3 inflow, this

match between the results of the two LES rules out a potential explanation based of a different dissipation within the wake.

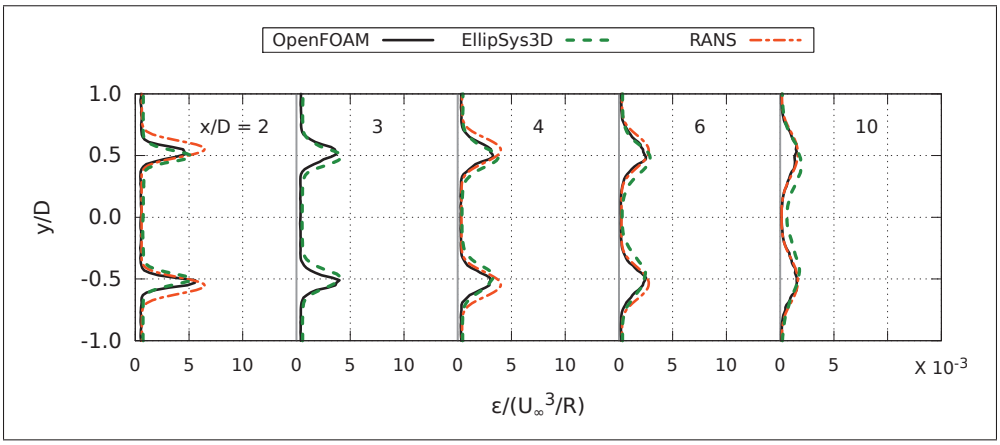

Figure 4.11 Vertical profiles of ε behind the disk $C_T = 0.42$, Ti3 case.

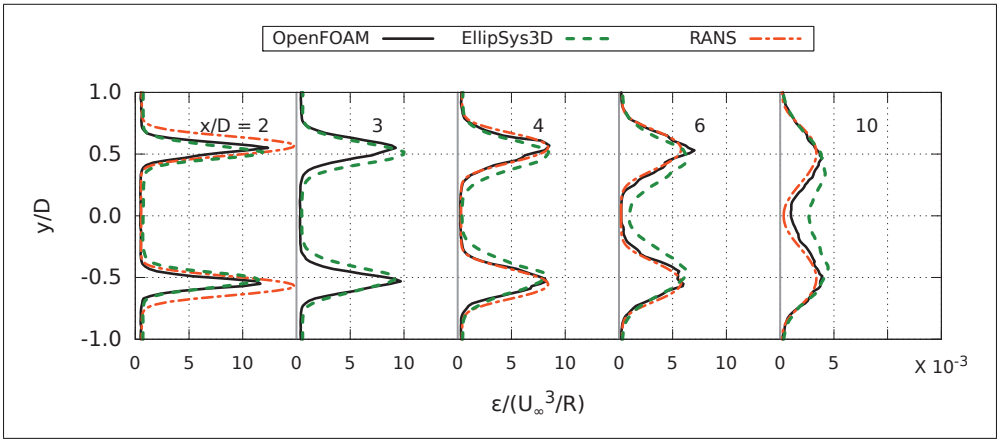

Figure 4.12 Vertical profiles of ε behind the disk $C_T = 0.62$, Ti3 case.

For the results with the Ti12 inflow, the experimental dissipation has been computed using eqs. (2.50) and (2.49), which assume isotropic conditions. Note that, unlike the previous figures where LDA measurements were shown, the experimental data employed in these comparisons (as well as in all following figures) was obtained from HWA. We see that for the disk $C_T = 0.45$, OpenFOAM predictions compare well with measured values. For the disk $C_T = 0.71$, we see that the measurements reveal a large increase of dissipation within the

shear layer, compared to the data of computations with the lower thrust AD. Furthermore, at least within the three longitudinal positions available, dissipation in the shear layer is more or less maintained. Meanwhile, the computations in OpenFOAM display a somewhat stronger mixing of turbulence since from $x = 4D$, dissipation becomes more uniform and less predominant at the shear layer. In EllipSys3D, this trait seems slower, yet the overall level is smaller than in OpenFOAM. This is in fact expected, due to its lower levels of wake turbulence in EllipSys3D as seen in Figures 4.8 and 4.9.

The RANS computations with the modified $k - \varepsilon$ model of Sumner and Masson have been previously shown capable of reproducing the turbulence level in the wake. In our comparison, we see that for the Ti3 inflow the agreement is very good for the disk $C_T = 0.42$ while it falls somewhat behind in the far wake of $C_T = 0.62$. However, we notice that in both these cases the agreement in the computed dissipation of RANS and OpenFOAM is very good except for $x = 2D$. Interestingly, it is the vicinity of the disk where the $k - \varepsilon$ is often corrected by adding dissipative terms to the ε equation to overcome the miscalculated turbulence stresses (Réthoré, 2009). The results with the Ti12 show the opposite picture with regard of the estimation of k, as the agreement with measurements becomes better for farther distances from the disk. For the closest position, the turbulence level is overestimated (as it is in OpenFOAM) despite the drop of the turbulence production terms near the disk ($x = 2D$ is outside this region). Dissipation seems overestimated in the case of $C_T = 0.45$ when comparing to the measurements. This is less certain for the higher thrust disk, where at $x = 4D$ the peak value of dissipation seems equal to the measured one, but much smaller in the case of $x = 6D$. Notably, ε from RANS is always higher than any LES in the wakes of the Ti12 inflow. Previous work (Réthoré, 2009) has shown that in the ABL, the $k - \varepsilon$ model overestimates the dissipation around the disk when comparing with LES. This has been observed to occur even upstream of the disk, where ε has been seen to increase unlike computations of LES, where this value does not grow until $0.5D$ downstream from the rotor.

To complete the comments regarding the RANS/$k - \varepsilon$ simulations, it should be remarked that Sumner *et al.* (2013) showed that results of U and k in the wake with various turbulence

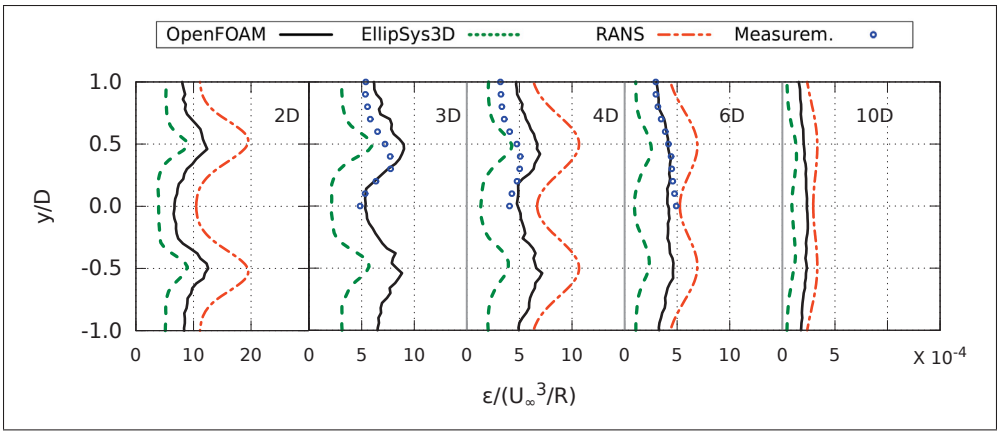

Figure 4.13 Vertical profiles of ε behind the disk $C_T = 0.45$, Ti12. The scale for the curves at $x = 2D$ has been doubled to accommodate the larger values.

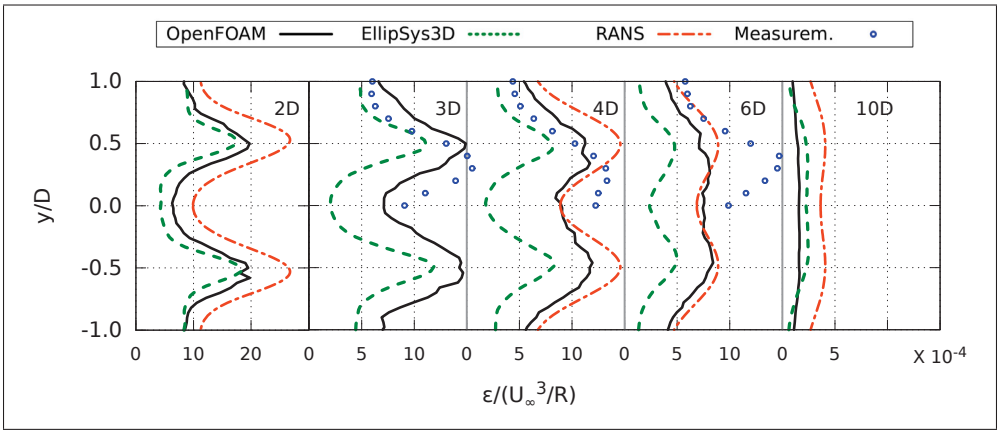

Figure 4.14 Vertical profiles of ε behind the disk $C_T = 0.71$, Ti12.

closures[2] compare, in essence, equally well to the measurements, with no apparent advantage of their proposed correction to the $k-\varepsilon$ model (interestingly, ε yielded by the different closures was not compared). The fact that all models compare well to measurements contradicts the otherwise inadequate results obtained in simulations of wakes in the ABL flow. It is argued in that work that this is due to the relative decrease of the modelled turbulent viscosity ν_t in the reproduction of wind-tunnel wakes with homogeneous inflow with respect to its proportion in

[2]Besides the proposed Sumner and Masson model, results are compared to the standard $k-\varepsilon$, the Renormalization Group (RNG) as well as the El Kasmi and Masson model (El Kasmi and Masson, 2008)

the modelling in atmospheric conditions. In those conditions, previous work by Réthoré (2009) has successfully proved the advantages of LES to estimate the velocity deficit and turbulence levels in the wake.

4.2.4 LES modelling in the wake

4.2.4.1 Resolved and modelled turbulence kinetic energy

The previous results for k and ε indicate that OpenFOAM and EllipSys3D are able to predict with relative accuracy not only the velocity deficit in the wake, but also the level of turbulence and its dissipation in the case where the TI in the inflow is low ($\sim 3\%$). For the high TI inflow ($\sim 12\%$), the prediction becomes more imprecise according to the comparison with the experimental data, despite the good results obtained for the simulation of the free flow. In the absence of disks, we show that in the two codes and for both TI values, k occurs for the most part in the resolved scales. In the Ti3 case the situation varies, as the resolved dissipation increases fast after a short distance from the inlet/TP, while for the Ti12 case it remains mostly modelled in OpenFOAM and the resolved part turns more prominent towards the outlet in EllipSys3D (Figures 3.15 and 3.16). It is therefore interesting to evaluate what occurs in the wake in this respect.

In Figures 4.15 and 4.16 we compare the fraction of the turbulence kinetic energy that is resolved by the LES with respect to the total, k_{res}/k_{tot}. Note that, as we are not restricted by the experimental data available for these comparisons, we show profiles at different longitudinal positions from other figures. The first is at $x = 1D$ instead of $2D$ to study the modelling closer to the disks, while the rest are chosen in increments of $3D$ starting at $x = 3D$. We observe that for both disks, it is only for that position that the difference between the modelling in each code is noticeable, with OpenFOAM resolving slightly more fluctuations (as opposed to SGS modelling) than EllipSys3D. The difference is particularly apparent in the shear layer, marked in OpenFOAM by an increase in the SGS modelling, which is in turn barely noticeable in EllipSys3D. For the rest of the wake the LES modelling is remarkably similar in both codes, with

more than 90% of k occurring in the resolved scales. The existence of a defined shear layer with higher levels of turbulence (as seen in Figures 4.6 and 4.6) does not appear to influence the modelling of the flow inside and around the wake with this inflow, at least beyond $x = 3D$. Due to fact that subgrid increases only at $x = 1D$ and that the resolved proportion is very similar to the no-disk computation elsewhere, it can be deduced that shear from the wake envelope creates turbulence at smaller lengthscales than the ambience turbulence but only very near the disk. However, these scales do not endure further in the wake, prevailing those of the inflow instead. This discussion will be resumed later on.

Vertical profiles of k_{res}/k_{tot} are shown in Figures 4.17 and 4.18 for each disk, using the Ti12 inflow. Some differences are immediately apparent with respect to the lower TI inflow. In OpenFOAM the added turbulence does not seem to modify the ratio of the resolved part in the LES. The only difference with respect to the Ti3 inflow is the absence of an increase in the SGS part within the shear layer at the closest position to the AD. This is related to the larger level of ambience turbulence, as seen in Sec. 4.2.2 where this and the added turbulence by the shear layer are compared. On the other hand, in EllipSys3D the SGS modelling decreases as a function of the distance to the disk, varying from about 30% at $x = 1D$ to close to 10% at $x = 12D$, for both disks, matching the trend seen in the free flow (Figure 3.15). Unlike the case of the Ti3 inflow, we also observe that for some positions, the ratio k_{res}/k_{tot} in EllipSys3D is larger close to the center of the wake. Still, in every position and for both disks, the resolved part of k_{tot} is lower than in OpenFOAM.

4.2.4.2 Resolved and modelled turbulence dissipation

The study of the LES modelling in the wakes is complemented with an analysis of the ratio of subgrid dissipation with respect to the total value $\varepsilon_{\text{SGS}}/\varepsilon_{tot}$ along the wake. For the Ti3 inflow, the results for each disk are presented in Figures 4.19 and 4.20. We notice that both LES predict, to the same extent, an appreciable increment in subgrid dissipation within the shear layer. Furthermore, unlike the modelling of k, this increase persists longitudinally even as far as when the wake appears to reach a full-turbulent state, i.e. at $x = 12D$ with disk $C_T = 0.62$.

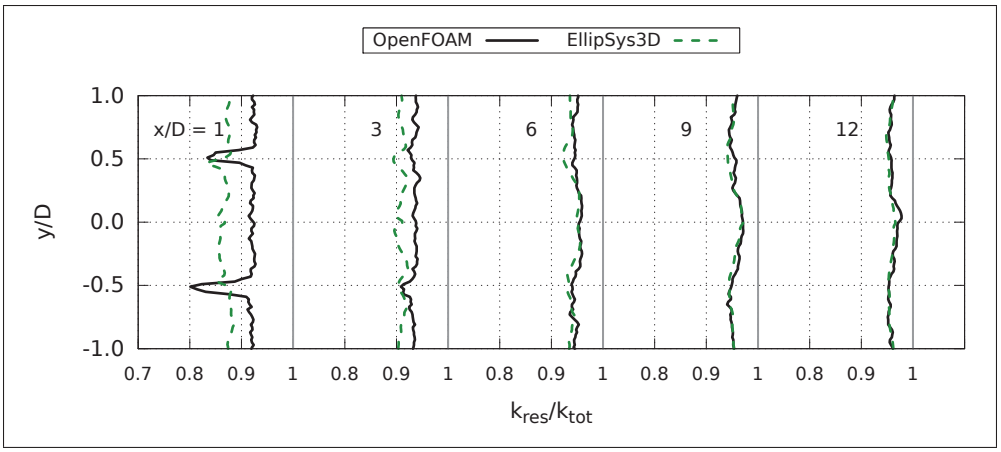

Figure 4.15 Vertical profiles of k_{res}/k_{tot} behind the disk $C_T = 0.42$, Ti3 case.

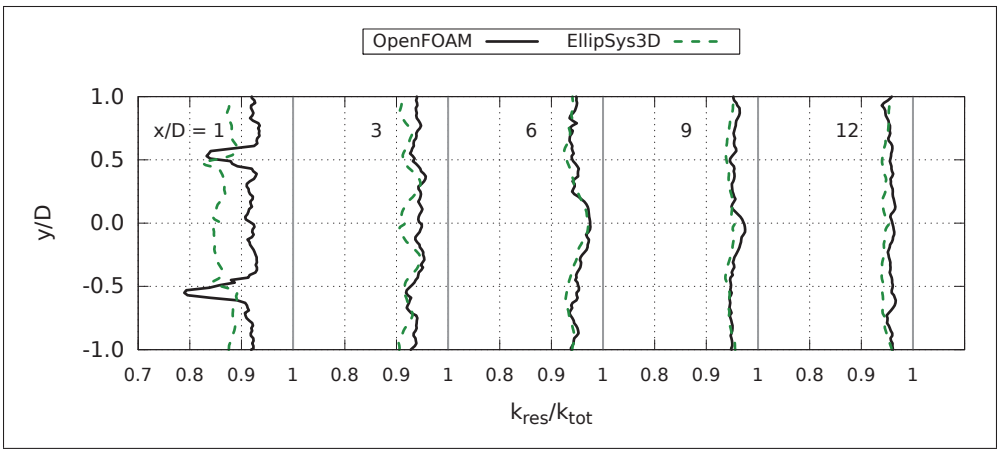

Figure 4.16 Vertical profiles of k_{res}/k_{tot} behind the disk $C_T = 0.62$, Ti3 case.

This is consistent with the hypothesis that small-scale turbulence is created from the shear at the disk edge. Although not seen to noticeably increase the proportion of k_{res}/k_{SGS} beyond the vicinity of the disk, we see in our computations that this small-scale turbulence becomes the main carrier of dissipation in the wake. The subgrid dissipation part is also larger with higher thrust, yet by a small margin. Make note that in the absence of disks (Figure 3.16 – (a)), most of the dissipation comes from the resolved fluctuations

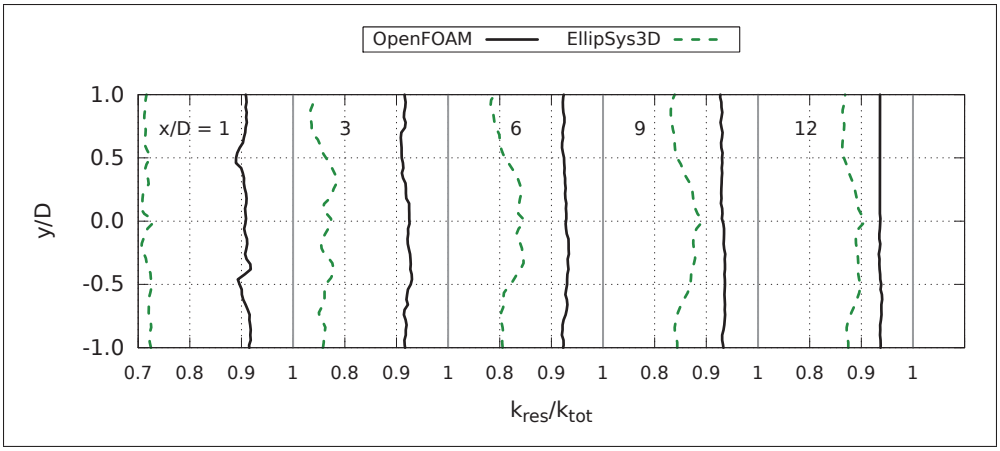

Figure 4.17 Vertical profiles of k_{res}/k_{tot} behind the disk $C_T = 0.45$, Ti12.

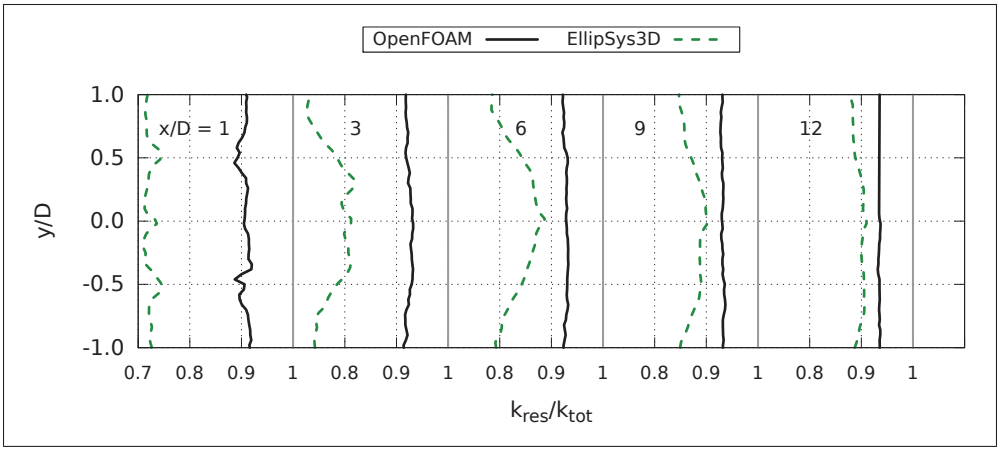

Figure 4.18 Vertical profiles of k_{res}/k_{tot} behind the disk $C_T = 0.71$, Ti12.

The comparison between codes is different when the Ti12 inflow is used. We can see in Figures 4.21 and 4.22 that when the inflow turbulence raises (which comprises better resolved lengthscales), the increment of subgrid dissipation in the region of the wake envelope is largely absent. As a result, the modelling ratio seen in the no-disk LES is essentially conserved in both codes. In that computation (Figure 3.16 – (b)), the subgrid part of the LES is smaller in EllipSys3D than in OpenFOAM except only for x_D. This difference appears to be more or less conserved outside the wake. In EllipSys3D, there is a minor increase of modelled dissipation in the shear layer. Conversely, in OpenFOAM, the presence of the wake seems to have little

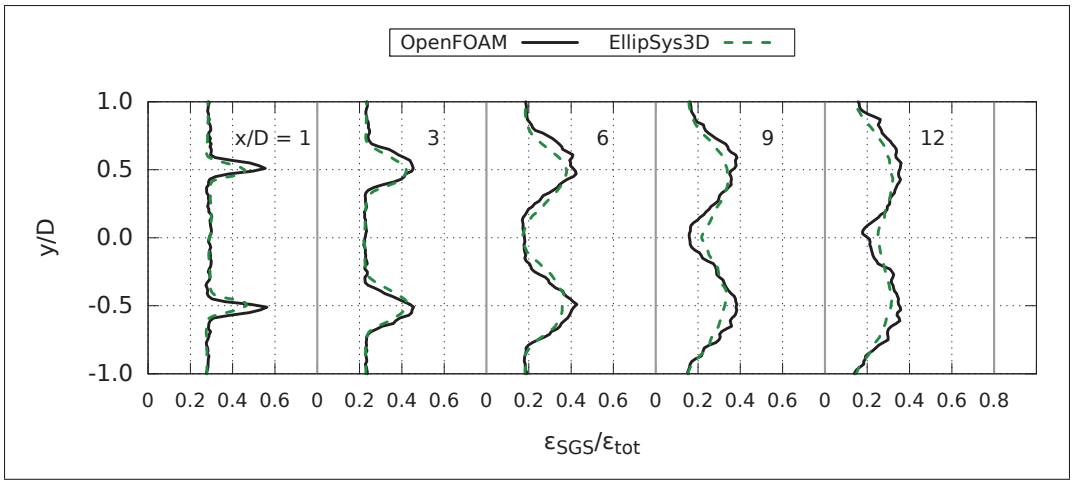

Figure 4.19 Vertical profiles of $\varepsilon_{\text{SGS}}/\varepsilon_{tot}$ behind the disk $C_T = 0.42$, Ti3 case.

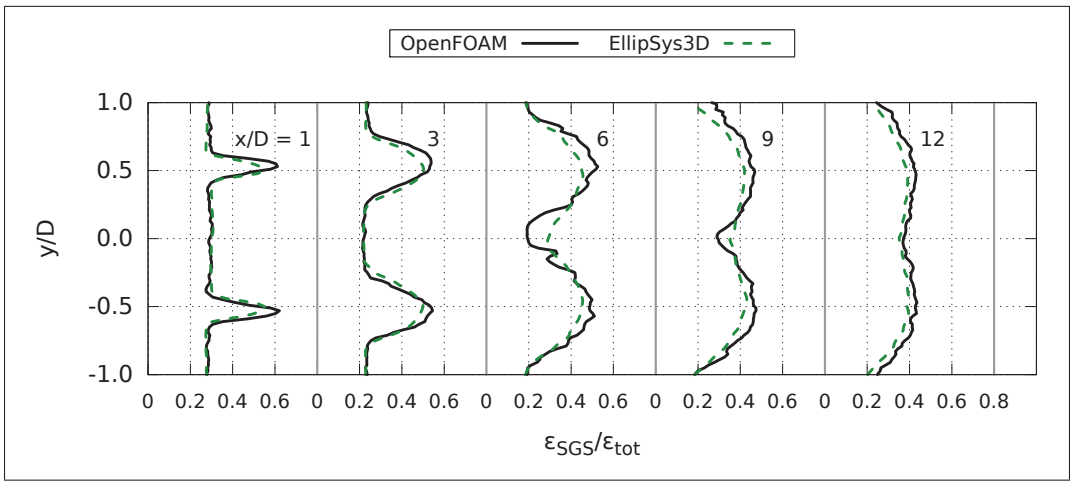

Figure 4.20 Vertical profiles of $\varepsilon_{\text{SGS}}/\varepsilon_{tot}$ behind the disk $C_T = 0.62$, Ti3 case.

influence in how the dissipation is modelled, except perhaps only for the closest position to the disk.

From the results of k_{res}/k_{tot} and $\varepsilon_{\text{SGS}}/\varepsilon_{tot}$ we can observe that the LES modelling in the wake is largely determined by the ambience turbulence. In the case of k_{res}, the changes occur only for the closest position of the wake ($x = 1D$) for the low TI inflow (more so for OpenFOAM). Similarly, the resolved part increases slightly within the wake for the high TI inflow, but only

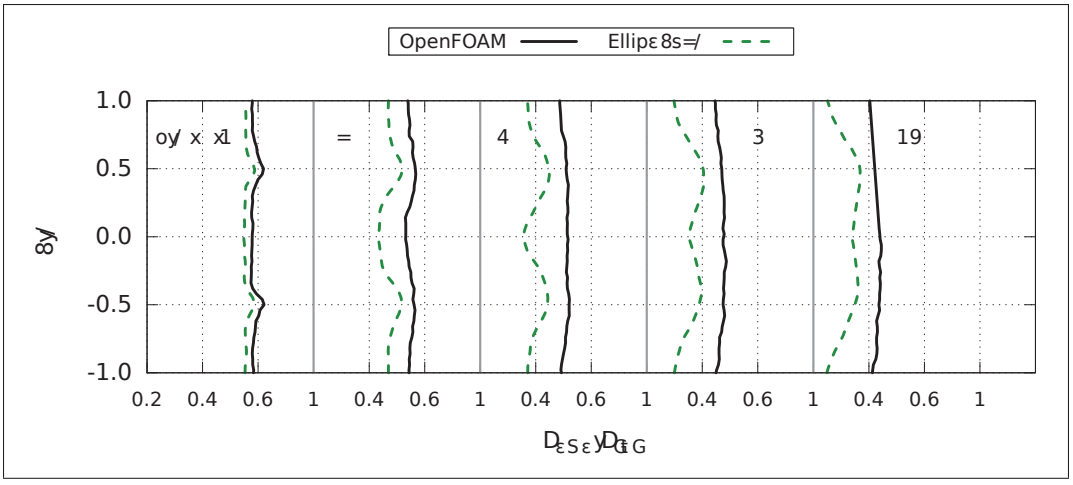

Figure 4.21 Vertical profiles of $\varepsilon_{\text{SGS}}/\varepsilon_{tot}$ behind the disk $C_T = 0.45$, Ti12.

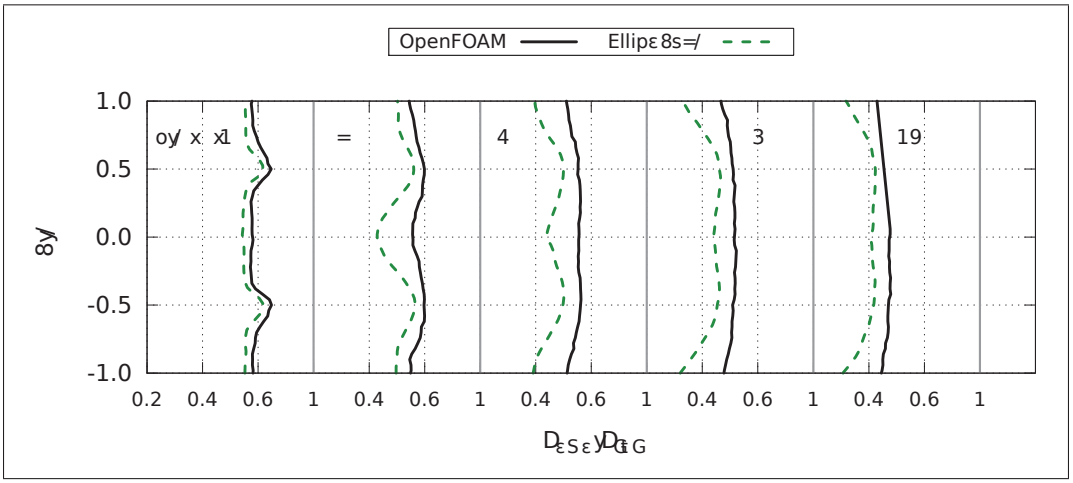

Figure 4.22 Vertical profiles of $\varepsilon_{\text{SGS}}/\varepsilon_{tot}$ behind the disk $C_T = 0.71$, Ti12.

for some positions ($3D \leq x \leq 12D$) in EllipSys3D. As for ε_{SGS}, the effect of the shear layer is more obvious, but it is greatly reduced with the increase of inflow TI. It should also be considered that due to the limited resolution of turbulence lengthscales in the Ti3 flow (missing in the synthetic flow as well), the increase in subgrid dissipation is produced at scales that seem absent in the incoming flow.

We also notice that while the overall level of k and ε increase in the shear layer with disk thrust (as well as producing an earlier break-up of the wake), its effect on the LES modelling of the

wake is rather small. Considering that each code employs a different SGS model, our results show consistency with previous work. For instance, Sarlak *et al.* (2015b) performed a comparison of the wake characteristics in the wake of two rotors modelled using the actuator line method in LES performed using Ellipsys3D, with a decaying-HIT inflow ($\mathrm{TI} = 0.24\%$). They found that above a given cell resolution (70 cells per rotor diameter), k_{res} varies only slightly when using different SGS models (including Smagorinsky, dynamic Smagorinsky, mixed-scale model). However, the ν_{SGS} predicted by each model is noticeably different, with the value computed with Smagorinsky being the highest. Despite this, it was found that there is a negligible correlation between its value and the predicted k_{res} in the wake. Using an equivalent setup to that work but with an inflow turbulence of $\mathrm{TI} = 14\%$, Sarlak *et al.* (2015a) also found that despite ν_{SGS} obtained by different SGS models is appreciably different along the wake, the mean velocities are not affected by such modelling. In our investigation, we find that next to the negligible influence in the velocity deficit, it is rather difficult to identify differences in k and ε obtained along the wake that can be directly attributable to the different SGS models in each code. As it was deduced from the comparisons made for the no-disk computations, the interpolation schemes for the convection are more likely to be the cause for the differences in the TI decay, that in turn establishes the level of ambience turbulence and the resolved/modelling ratios of the LES in the wake simulations. Since the latter is mainly determined by the resolution of the integral lengthscales, we now investigate the changes in the development of L_i due to the presence of the disks.

4.2.5 Integral lengthscale across the wake

We investigate now the changes in the evolution of L_1 caused by the shear and the resulting increase in turbulence levels along the wake. The computation of L_1 is performed as described in Sec. 3.2.4, which involves the assumption of the Taylor hypothesis to transform the computed time-scales into lengthscales. Evidently, this supposition becomes more difficult to accept when shear is present in the flow. However, previous work has reported satisfactory results in wake studies that support the continuing applicability of the hypothesis. For instance, Thacker

et al. (2010) has compared the lateral distribution of L_1 behind the wake produced by a porous disk (in a similar setting to this work) computed from HWA with the one obtained from PIV. They did not find a difference in the results obtained from either technique, despite the fact that HWA uses the local mean velocity to calculate the lengthscale, compared to the direct spatial measurement offered by PIV. Making the same assumption, we study the longitudinal distribution of the lengthscales in the AD computations.

In Figures 4.23 and 4.24 we compare the longitudinal distribution of L_1 for the different disks in OpenFOAM and EllipSys3D, for each inflow turbulence level. In every plot, the lengthscale values are shown for three different positions: along the center, mid-radius (i.e. $R/2$) and $2R$. Data for each of these locations is obtained according to the probes distribution shown in Figure 4.1. This is, at each $x-$position, the reported value at *centre* is given by the average value of the results of the four central probes. Likewise, $R/2$ is the mean obtained from the 12 probes located at such position from the center while $2R$ corresponds to the mean of the four probes at that distance from the center[3]. The results from the decaying-HIT (no-disk) are also included. The mid-radius position has been chosen to investigate changes in the lengthscale inside the wake envelope (L_1 at the shear layer will be shown later).

In the case of the Ti3 inflow in 4.23, we first note that in both codes and for every disk, there is little difference between the results at $2R$ and the no-disk cases. Then, we see that the effects of the disks are slightly different in each code. In OpenFOAM, a small increase in L_1 right behind the AD is seen with either disk for the values at the centre and $R/2$, followed by an oscillatory pattern. Next, for the furthermost $x-$positions, there is an increase in L_1 (at least for the most part), with the notable exception of the values at centre for $C_T = 0.42$. On the other hand, for the results of EllipSys3D we observe no increase immediately behind the disks. For the lowest thrust AD, little changes in the lengthscales are observed between all curves (only for $R/2$, somewhat larger values are obtained towards the outlet). The largest thrust does cause more variations in the results, with the curve at $R/2$ stably growing in value from about $x = 4D$. Also, an increase can be observed for the curve at the centre, despite the oscillations seen from

[3]The distance of each probe to the centreline is $2R$ assuming that $A = 0.07$ m $\simeq \sqrt{2}R$ in Figure 4.1

approximately the same $x-$position. Therefore, we see that in both codes a constant increase is obtained for the position $R/2$ for the final part of the domain. The same effect is seen also for the curve at the centre, although with more oscillations and only for the $C_T = 0.62$ disk.

For the case with the Ti12 inflow shown in Figure 4.24, experimental data is available. As HWA measurements were made at only three longitudinal positions behind the disk, the resulting three points points available for comparison with each curve cause that a trend can be hardly established. Yet, it is observed that for the low thrust disk, values at centre and $R/2$ tend to increase in a rate similar to the measurements of the no-disk case, but with lower values (see Figure 3.28). These magnitudes are similar for the points at $2R$. In the high thrust disk, the $R/2$ and $2R$ curves seem to maintain the value measured without the disk at x_D, i.e. $L_1 \simeq 0.03$ m, while the points from the center are mostly below that.

For the computations, we notice in Figure 4.24 that the resemblance between the curves obtained outside the wake at $2R$ and the no-disk case is mostly maintained in Ellipsys3D, but not in OpenFOAM. For the latter code, L_1 increases behind the wake in comparison with the no-disk case. Next, the growth observed immediately behind both disks (for curves at centre and $R/2$), previously seen for the Ti3 inflow in OpenFOAM, also appears. This feature is, in comparison to the Ti3 inflow case, larger with the low thrust disk and smaller in the high thrust case. After this, both curves at centre and $R/2$ decrease to a value similar to (or below) the no-disk case. A similar feature is absent in EllipSys3D results. Instead, the largest scales are essentially provided by the no-disk case. Precisely, just like in OpenFOAM far from the disks, the curves from centre and $R/2$ also fall below the no-disk case.

From the analysis of our computations, we can conclude that:

- For the Ti3 inflow, the effect of the disk in OpenFOAM is to increase L_1 for $R/2$ *very* far in the wake ($x \gtrsim 6D$). This effect is seen also for the curve at centre but only for the hight thrust, so it seems related to the turbulence mixing due to shear

- Also for the Ti3 inflow, EllipSys3D predicts less changes in L_1 when the disk is introduced, compared to OpenFOAM. At $R/2$ there is an increase again far from the wake (but less evident than in OpenFOAM). Yet, as in OpenFOAM, the curve at centre also seem to show a growth of L_1 far from the disk for the largest thrust.

- For the Ti12 inflow, as it has been deduced before, the wake characteristics are dominated by the ambience turbulence, specially so for the low thrust disk. Moreover, the predicted behaviour of L_1 due to the disk are also distinct. It is observed in OpenFOAM that L_1 increases in the near wake (more evident in the low thrust disk) followed by a contraction. L_1 outside the wake envelope grows more than in the no-disk case.

- For the Ti12 inflow, EllipSys3D predicts a decrease in L_1 behind the disk, seemingly more so for the values at the centreline than for $R/2$.

4.2.6 Profiles of L_1 behind disks

To study the effect of the shear layer and its turbulence production on the longitudinal length-scale, we compare profiles of L_1 obtained from each code at the positions where HWA data for the Ti12 inflow is available, this is $3D$, $4D$ and $6D$. In Figure 4.25 we see the values of L_1 computed from the LES in each code with the Ti3 inflow, from $y = 0$ to $y = 1.5$. We notice first that the magnitudes of the lengthscales are similar in both codes. However, there is not a clear influence of the shear layer in the size of the turbulence scales. In EllipSys3D the profiles remain with very little variation across the wake. It is only in the results from OpenFOAM that, within approximately the shear region, larger lengthscales can be discerned amongst the variations in the profile. Indeed, for OpenFOAM, the maximum values of L_1 at each $x-$position are at around $y = 0.5D$ in the wake of the disk $C_T = 0.42$. This is consistent with the previous results with regard of the location of the shear layer along the wake (e.g. k and ε). Conversely, for the other disk the maxima of L_1 suggest a wake that expands to about $y = 0.75D$ at $x = 6D$ which seems slightly larger than what the previous computations indicate with respect to the position of the wake envelope.

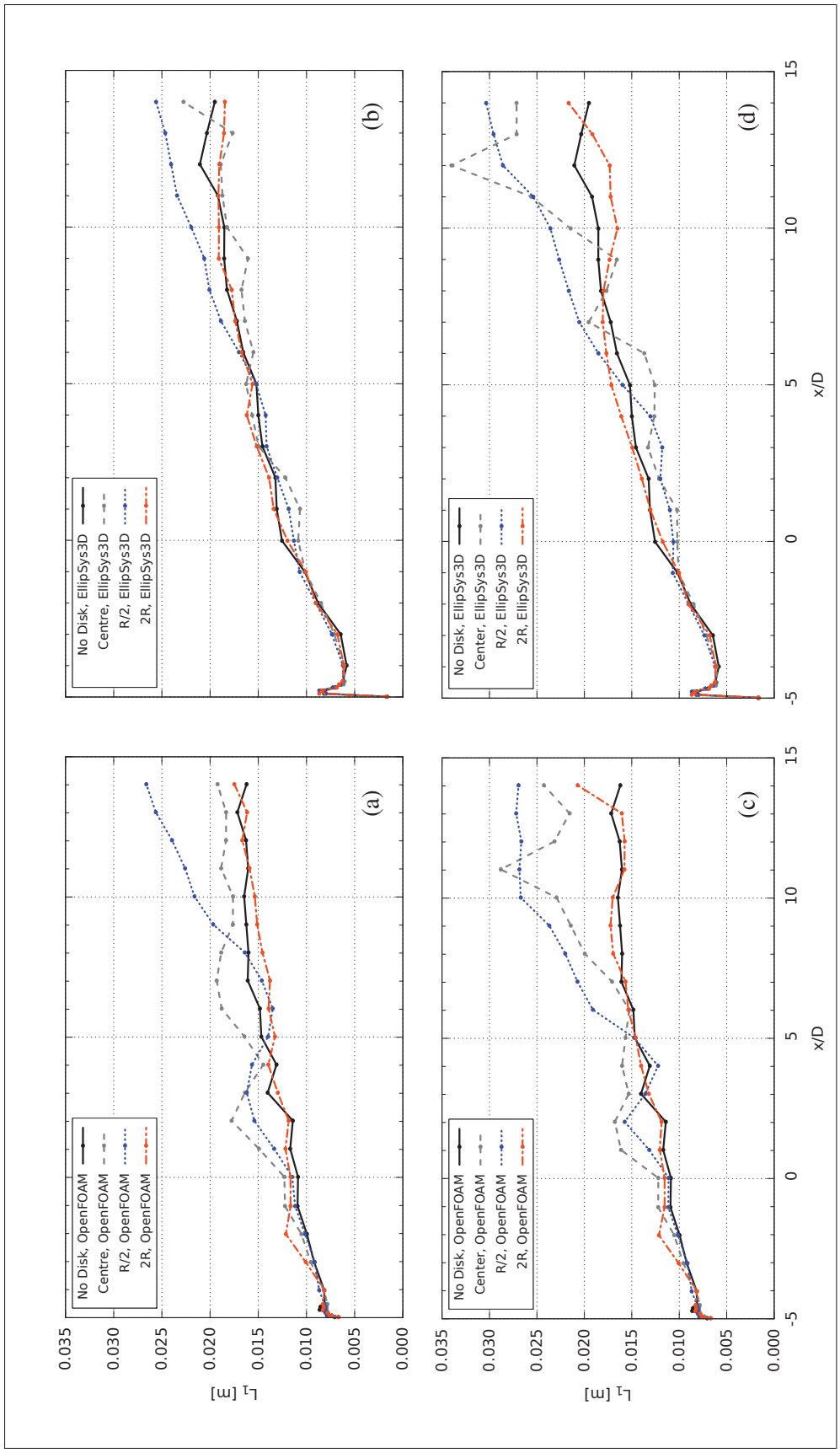

Figure 4.23 Longitudinal evolution of L_1 for the inflow Ti3. Results with disk $C_T = 0.42$ in the top row with: (a) OpenFOAM and (b) EllipSys3D. In bottom row, results with disk $C_T = 0.62$: (c) OpenFOAM and (d) EllipSys3D.

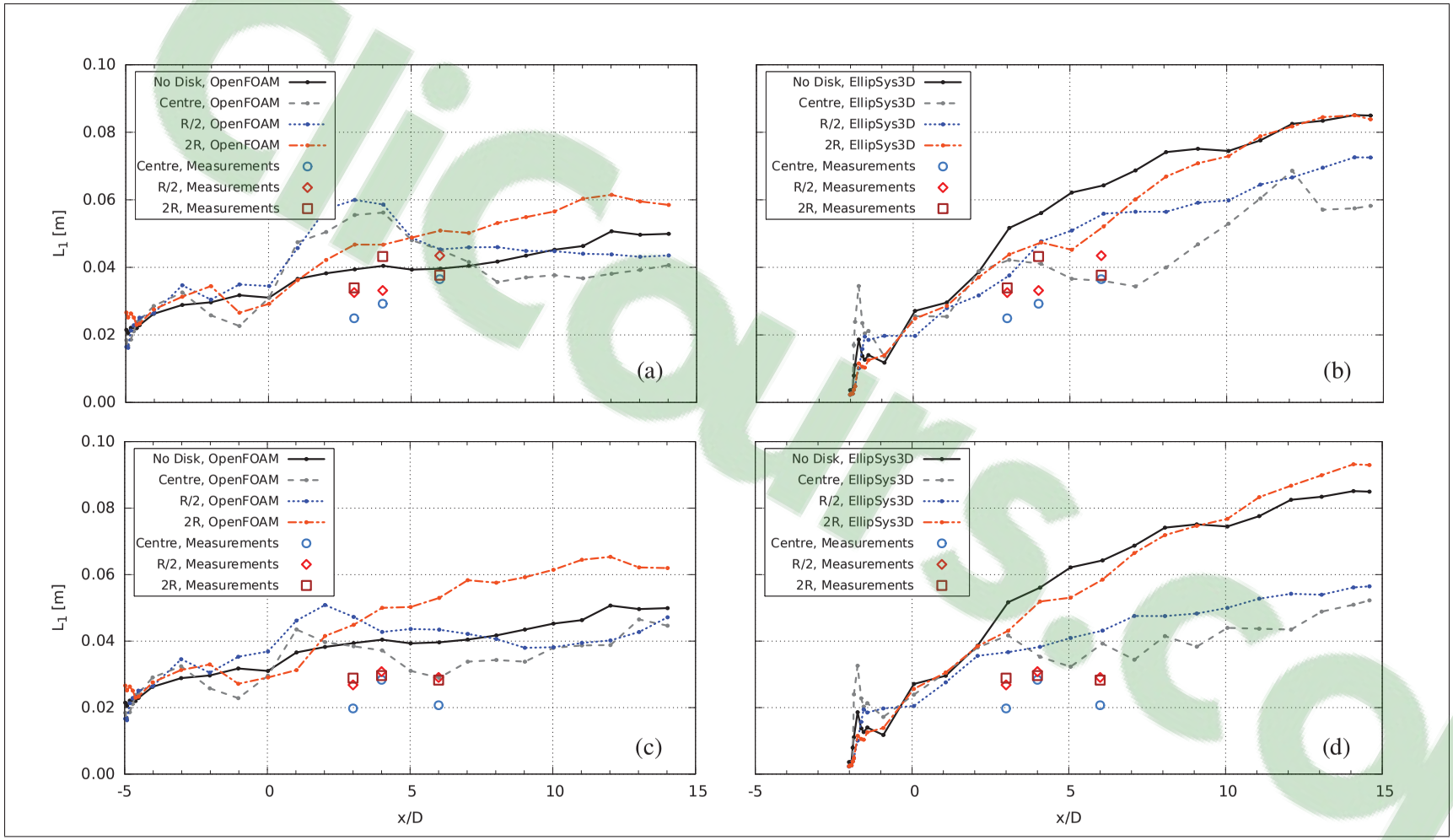

Figure 4.24 Longitudinal evolution of L_1 for the inflow Ti12. Results with disk $C_T = 0.45$ in the top row with: (a) OpenFOAM and (b) EllipSys3D. In bottom row, results with disk $C_T = 0.71$: (c) OpenFOAM and (d) EllipSys3D.

Results for the Ti12 inflow are shown in Figure 4.26. Notably, the computed values from the experimental time-series do not reveal a variation of the lengthscale values at the shear layer. In fact, there is no evident change in L_1 within the wake. This trait is similarly observed in the results of OpenFOAM. With it, the only variations are observed at the upper part of the curves or, in the case of the disk $C_T = 0.45$, towards the bottom part where L_1 is larger (but this effect is reduced further downstream). Meanwhile, EllipSys3D computations yield large fluctuations in the lengthscale values along every profile. Although the local level of turbulence is lower than in OpenFOAM, the cause of this variations has yet to be found.

Previous experimental work by Thacker *et al.* (2010) showed that in the wake of a porous disk with a solidity of 45%, L_1 is approximately 1.5 times larger within the shear layer with respect to the values within the wake or outside the envelope. However, these measurements were obtained using an inflow with very low turbulence ($\mathrm{TI} < 0.4\%$), which clearly sets a different scenario in comparison to our study. Precisely, the absence of a variation of L_1 in the shear layer can be explained considering our previous results, which point at a dominance of the ambience turbulence characteristics over the wake in the case of the inflow Ti12. Although the turbulence production is visibly higher when the disk thrust is larger (e.g. Figures 4.8 and 4.9), its effect does not appear to have an impact in the turbulence lengthscales. Similarly, the use of a lower turbulence inflow (Ti3) does not seem to decidedly increase the magnitude of the lengthscales in the area of turbulence production, or at least not in our computations. In this regard, the fact that the characteristic lengthscales of the Ti12 inflow are better resolved by the mesh and the LES compared to the Ti3 cases can be a factor to consider. This is, if resolution is not adequate within the shear layer, it is to be expected that a sizeable part of the turbulence being produced would fall into the modelled part instead of being resolved, therefore affecting the magnitude of the computed scales. This has been studied in Sec. 4.2.4, where it is shown that the LES modelling does not vary within the wake with respect to the external flow aside from very close to the disk ($x = 1D$), in both codes. Moreover, we have seen that despite the limited resolution, our LES computations have been able to reproduce other principal features along the wake, such as the turbulence levels.

Although not equal, the computed profiles of L_1 are consistent with the previous results for the longitudinal distribution of Figures 4.23 and 4.24. Small differences are due to the fact that each of the points along the vertical profiles corresponds to the value computed at one location, whereas in the longitudinal instance each point represents a mean taken from different locations, as previously described. The curves could potentially be improved if instead of computing a lengthscale from the autocorrelation of one-time series, its value could be calculated from an ensemble average, as it is the case of the experimental data. But such scenario was not contemplated for this work. Nevertheless, the lengthscale computation seems adequate to provide a picture of its evolution and its development in the wake.

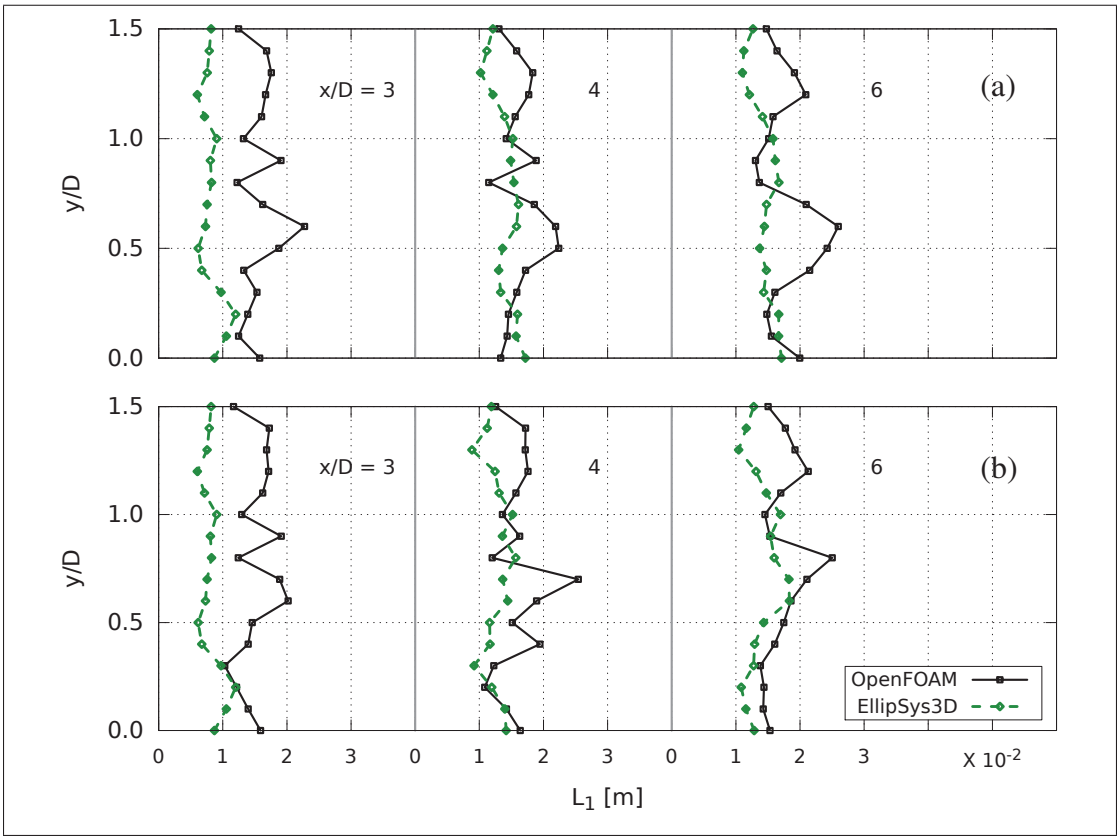

Figure 4.25 Vertical profiles of L_1 behind the AD with inflow Ti3, disks: (a) $C_T = 0.42$ and (b) $C_T = 0.62$.

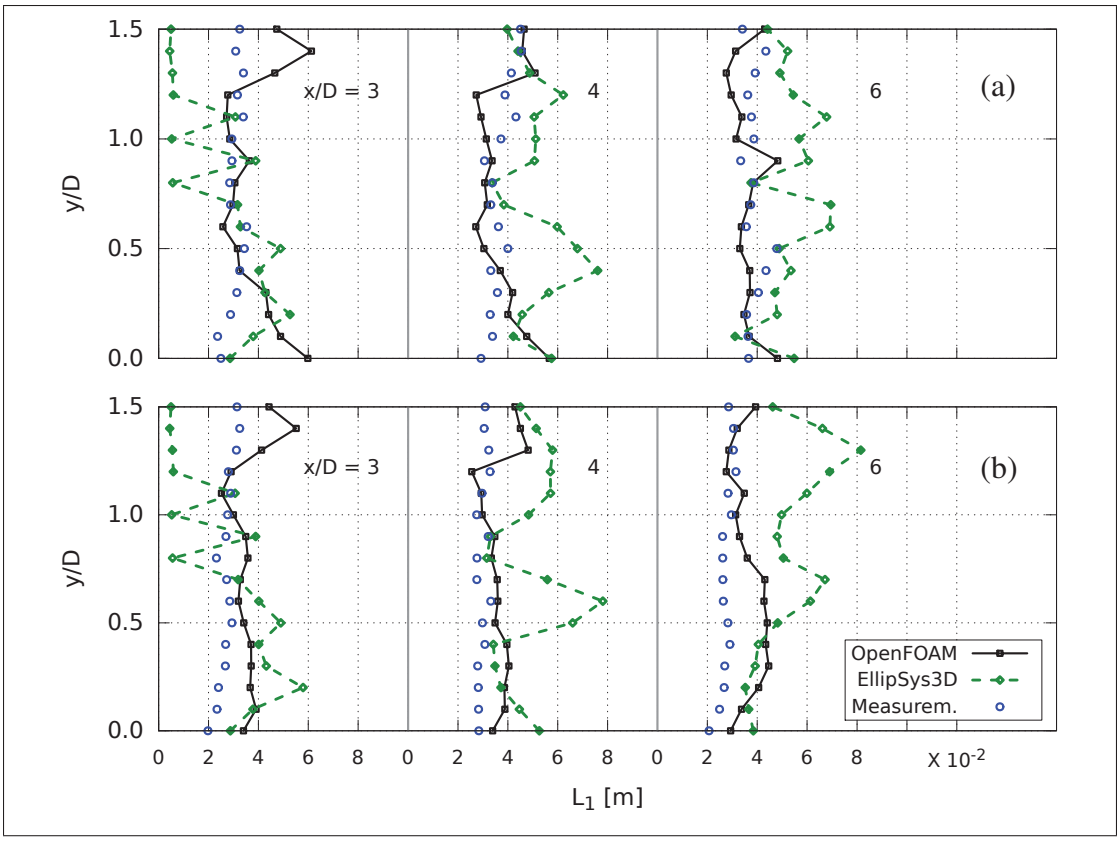

Figure 4.26 Vertical profiles of L_1 behind the AD with inflow Ti12, disk (a) $C_T = 0.45$ and (b) $C_T = 0.71$.

4.2.7 Spectra behind disks

To study the redistribution of turbulence energy along the wake, we compare the spectra obtained at different longitudinal positions for every disk with the spectra from the free decaying turbulence. Power spectral density curves are calculated from only one measuring position at centreline, so unlike the spectra in the decaying-HIT, no spatial averaging is performed. To reduce the noise in the curves that would otherwise make the comparison very difficult, we need to perform a smoothing (in addition to having averaged the spectra from eight blocks, as explained in Sec. 3.3.12). To this aim, an exponential moving average is used to filter[4] the spectra computed at each longitudinal position. Hence, the spectra shown in the following

[4] A rational transfer function is employed for this, see Oppenheim *et al.* (1999).

figures have been processed with this technique, with the sole exception of that obtained from measurements without a disk, which was spatially averaged.

The results for the inflow Ti3 are shown in Figure 4.27. In the results without the AD, we observe a constant decay of energy as the flow moves downstream. The spectra from the synthetic box serves to mark the extension of the resolved wavenumbers ($\kappa_{max} = \pi/\Delta = 1571$ m^{-1}) since the spatial resolution in the box is the same as in the LES. Although the decay at the measured locations is similar both codes, some differences arise in the energy distribution. We notice that in EllipSys3D the highest energies reach a bit deeper into the high wavenumbers than in OpenFOAM, which has been commented before in Sec. 3.3.12. Likewise, it was mentioned there that the abrupt drop in the spectra has been attributed to a combination of numerical diffusion and the limited spatial resolution (Troldborg, 2008). Differences between codes over this region become more evident here than in the previously studied spectra in the vicinity of the inlet/TP or at the target position. Therefore, the differences in the handling of numerical diffusion seem to be enhanced in the limited grid resolution as the flow moves further downstream. Precisely, these disparities are largely reduced for the cases with the Ti12 inflow, where the spatial resolution of turbulence fluctuations is improved.

In case of the disk $C_T = 0.42$, the results are analogous for both codes. First, we observe a gain in fluctuating energy immediately behind the disk, as the curves at $-1D$ and $1D$ are almost identical. Secondly, we see a small decay for the energy at $4D$ and from there, an increase in turbulence energy around the highest levels (lowest κ). This rise is clearer in EllipSys3D, where the increment can also be noticed near the highest resolved wavenumbers, before the energy drop ($\kappa \sim 10^5$). This is consistent with previous observations which suggest that disks in EllipSys3D add more shear and the wake becomes fully turbulent within a shorter length than in OpenFOAM (e.g. Figures 4.6 and 4.7), under the inflow of Ti3. For the disk $C_T = 0.62$, the effects are accentuated, the curves at $4D$ are the only ones displaying a decay and yet only around the inertial range. The energy of the next two longitudinal positions, $10D$ and $14D$ increases for all wavenumbers, which represents an increment of about one order of magnitude at the lowest wavenumber, with respect to the levels displayed by the decaying-HIT. Notably,

the spectra of the last two positions seemingly exhibit an inertial range, characterized by the slope of $-5/3$ in the decay rate.

The results for the Ti12 inflow are shown in Figure 4.28. In this case, the spectra computed from experimental results are also included. The spectra obtained from measurements with disks extends to larger wavenumbers than in the cases without disk, which seem to arise from the use of a different frequency in the low-pass filter. For the decaying-HIT, the energy at the lowest wavenumbers proves to decay less in OpenFOAM, as it has been shown before. Conversely, it is observed that energy levels are more or less conserved in EllipSys3D until the drop, as opposed to OpenFOAM where they display a steady decay which adjusts better to the slope of the intertial range. This feature occurs also upstream of the disks and in the near wake ($x = 1D$). Moreover, when comparing with the experimental results, we see that the curves from OpenFOAM approach better to the slope of such spectra. From these observations, it can be inferred that EllipSys3D overestimates the energy distribution in the inertial range except only for the last two positions ($x = 10D$ and $14D$). Considering this differences, the effects of the disk $C_T = 0.45$ are analogous between both codes. In contrast with the Ti3 inflow where energy is seen to increase beyond $x = 4D$ for the disk with the same porosity, we see here a reduction in the contribution of shear towards the increase of energy along the wake. Although the overall levels of turbulence energy in the wake are higher than in the decaying-HIT, they maintain more or less the same relative decay from one to another (still, a slightly larger decay is discernible in OpenFOAM). This behaviour is similar in the case of the disk $C_T = 0.71$. In OpenFOAM, only the curve at $4D$ shows an increase in energy compared to the previous disk (also matching fairly well the experimental results in the inertial range). Meanwhile, EllipSys3D shows a small increase of energy in the wake at the lowest wavenumbers, which can occur due to the increasing influence on of the wake turbulence caused by the lower level of ambience turbulence compared with OpenFOAM.

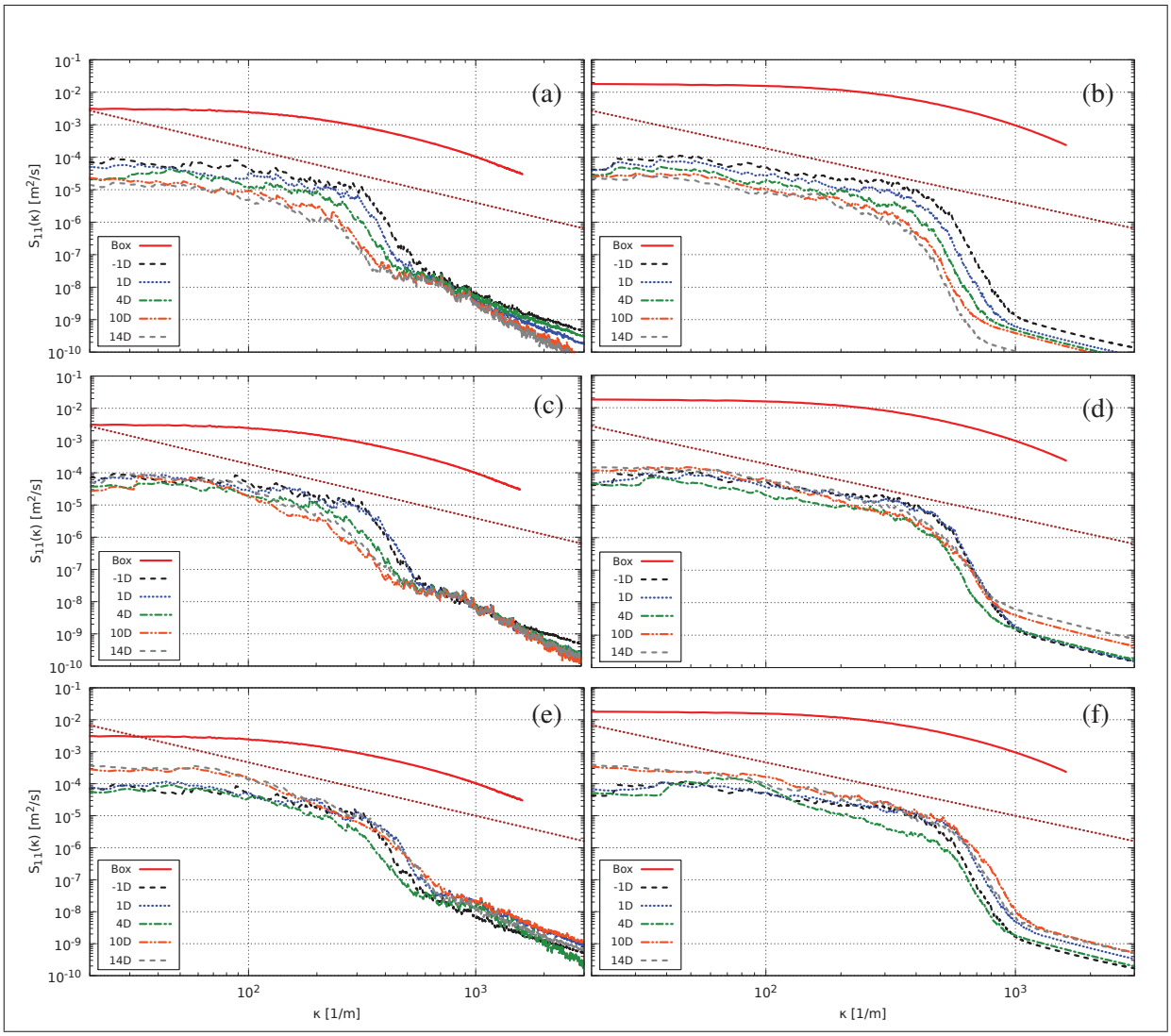

Figure 4.27 Longitudinal evolution of spectra at centreline using the Ti3 inflow. The results for the decaying-HIT (without AD) are shown in the top row: (a) OpenFOAM and (b) EllipSys3D, results with disk $C_T = 0.42$ are shown in the middle row for (c) OpenFOAM and (d) EllipSys3D, results with disk $C_T = 0.62$ are shown in the bottom row for (e) OpenFOAM and (f) EllipSys3D. The straight dotted line marks the -5/3 slope that characterizes the inertial range.

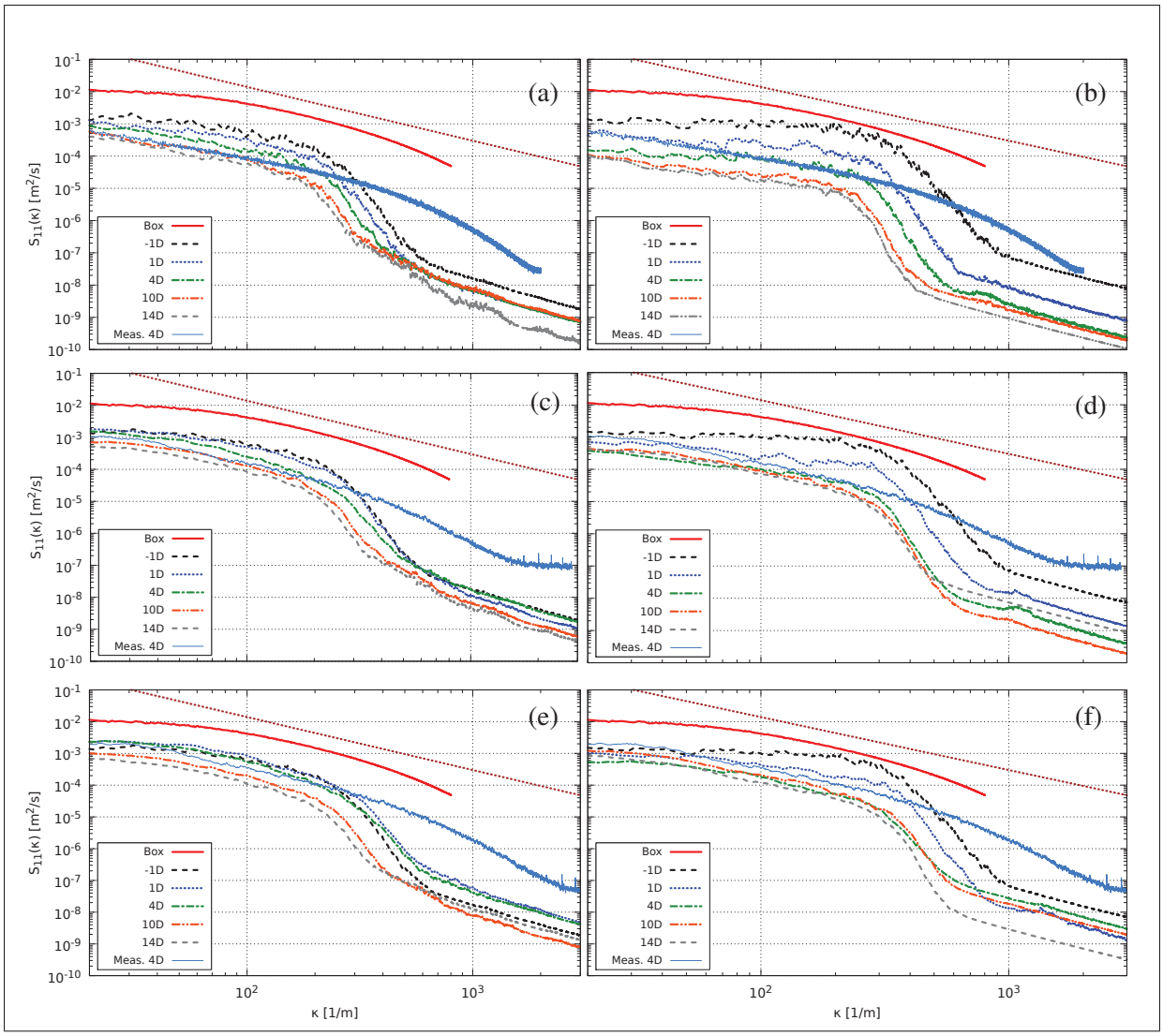

Figure 4.28 Longitudinal evolution of spectra at centreline using the Ti12 inflow. The results for the decaying-HIT (without AD) are shown in the top row: (a) OpenFOAM and (b) EllipSys3D, results with disk $C_T = 0.45$ are shown in the middle row for (c) OpenFOAM and (d) EllipSys3D, results with disk $C_T = 0.71$ are shown in the bottom row for (e) OpenFOAM and (f) EllipSys3D. Spectra computed from measurements is included only for the position $x = 4D$. The straight dotted line marks the -5/3 slope that characterizes the inertial range.

4.2.8 Wake visualization

Lastly, to complement all previous results we present images of the wake representation in each code. This allows us to compare some of the features previously discussed. The images are taken from fields in the $x - y$ plane, at $z = 0$ and correspond to the 1) resolved and instantaneous longitudinal velocity $\overline{u}$, 2) its mean value $\langle \overline{U} \rangle$, marking the wake envelope (defined here through the edge where $\langle \overline{U} \rangle = 0.99 U_\infty$) and accompanied by an image overlapping the envelopes of each code (to compare the wake expansion), 3) the vorticity field and 4) contours of the vorticity field. Each image shown is taken of field values computed at the last time step of the LES runs. Make note that black bars are used to represent the disk position but do not portrait the actual longitudinal region where the forces modelling the AD act.

For the Ti3 inflow, we can confirm that in EllipSys3D the shear layer converges faster towards the centreline than in OpenFOAM. This is particularly noticeable when looking at the vorticity field in Figures 4.31 and 4.35. The vorticity contours, Figures 4.32 and 4.36 are aimed at facilitating this observation. Although these differences were not seen to alter the comparison of the velocity deficit (that differs by a small margin only at the last position in Figure 4.3), we can see in Figure 4.34 that the wake recover is indeed faster for EllipSys3D in the case of the disk $C_T = 0.62$. Note also that the comparison of the envelopes of the wake simulated by each code shown at the bottom of Figures 4.30 and 4.34, respectively, shows that the expansion of the wake computed by each code is almost identical and thus, not affected by the different estimations in k.

In the case of the Ti12 inflow, the roles are reversed and due to the strong TI decay in EllipSys3D, the ambience TI is lower than in OpenFOAM beyond the disk location. Hence, the wake recovery is faster in OpenFOAM due to the dominant ambience TI. This effect can be seen in Figures 4.38 and 4.42 for the average velocity but it can also be discerned from the instantaneous velocity in Figures 4.37 and 4.41. In the vorticity field and its contours (Figures 4.39, 4.40 and 4.43, 4.44), the strong effect of the inflow velocity on the dispersion of the wake boundaries is easily seen: the inflow values are so large that the vorticity contours arising

from the shear layer are either scarce—in the case of EllipSys3D—or hardly identifiable—in OpenFOAM—. Unlike the previous case, the comparison of wake envelopes (at the bottom part of Figures 4.38 and 4.42), shows that OpenFOAM predicts a somewhat larger expansion of the wake. This is explained again by the higher TI content throughout the wake, which induces wider spatial displacements in the shear layer in comparison to the lower ambience TI values computed by EllipSys3D. However, we should make note that for the same experimental setup, Espana (2009) analyses PIV data of the mean wake velocities that indicate a slight reduction of the wake width in the longitudinal direction (measurements at $2D \leq x \leq 6D$) when using the Ti12 inflow. Meanwhile, the wake data obtained with the Ti3 inflow shows that the wake diameter increases along the streamwise direction, in a similar trend to what is observed here. Yet, it should be considered that the criterion used in that work to define the wake boundary employs $\langle \overline{U} \rangle = 0.95 U_\infty$ and this lower value contributes to reduce the diameter of the wake.

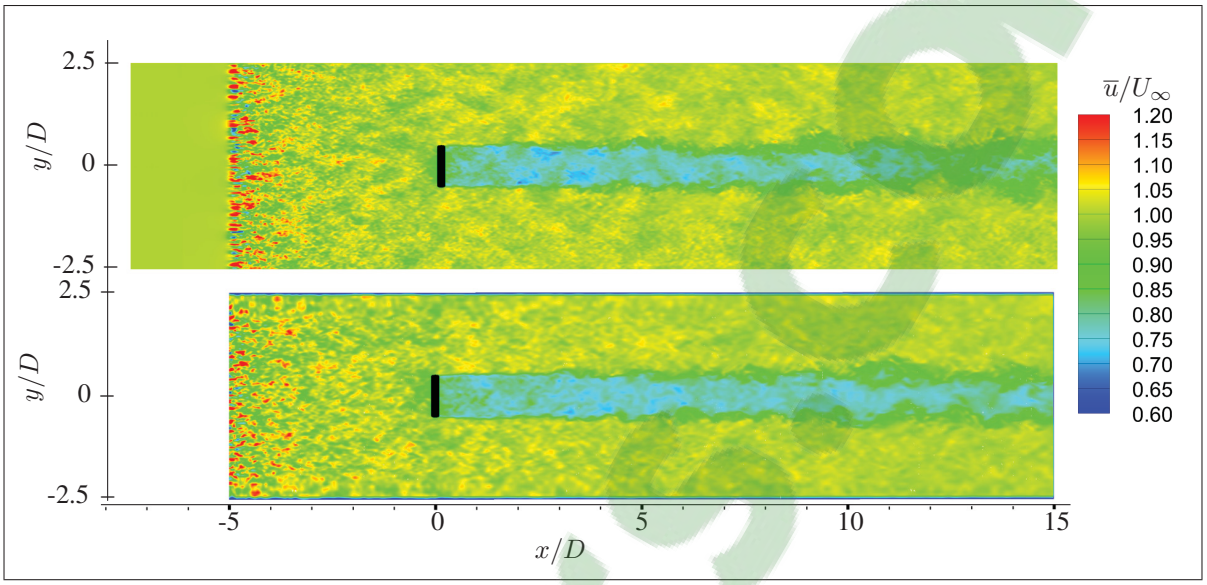

Figure 4.29 Instantaneous streamwise velocity using the Ti3 inflow and disk $C_T = 0.42$. Results of EllipSys3D (top) and OpenFOAM (bottom).

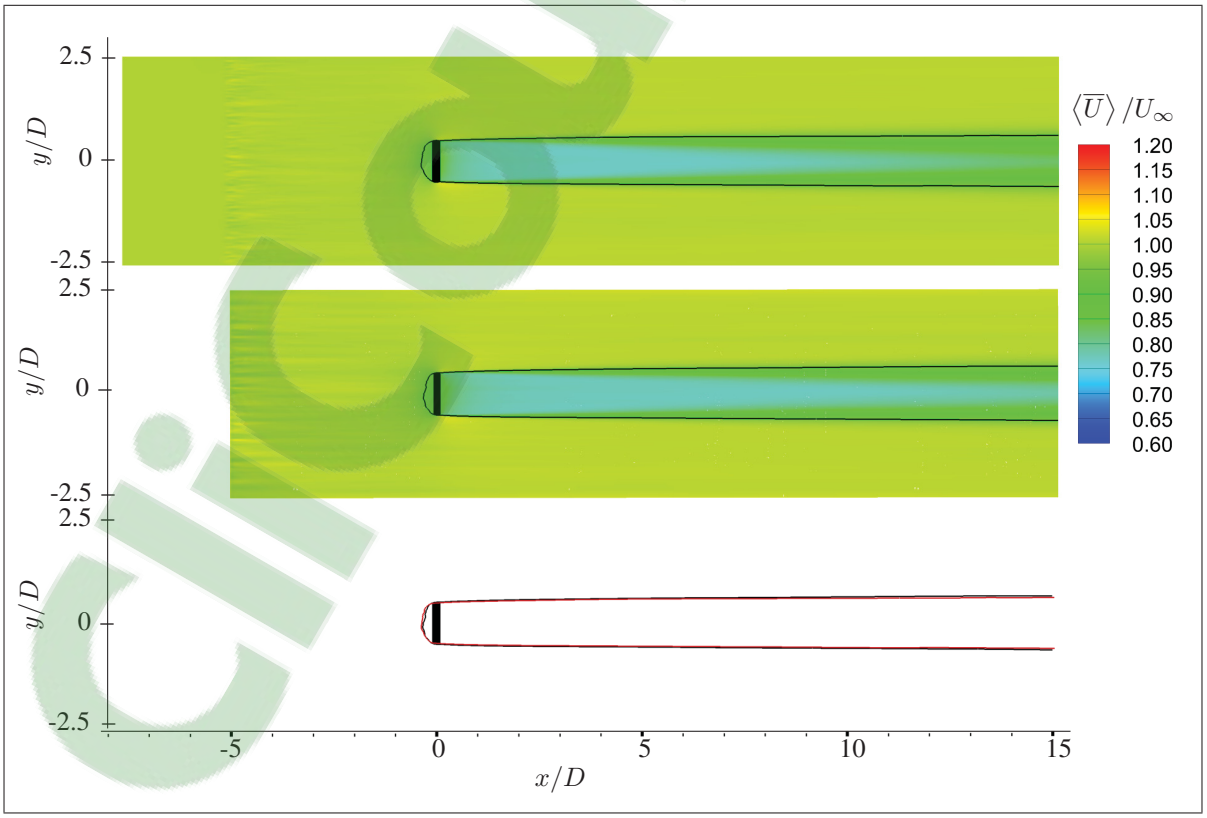

Figure 4.30 Average streamwise velocity using the Ti3 inflow and disk $C_T = 0.42$. A solid line is used to mark the wake envelope (see text for definition). Results of EllipSys3D (top) and OpenFOAM (middle). The bottom figure overlaps both envelopes, OpenFOAM (black) and EllipSys3D (red).

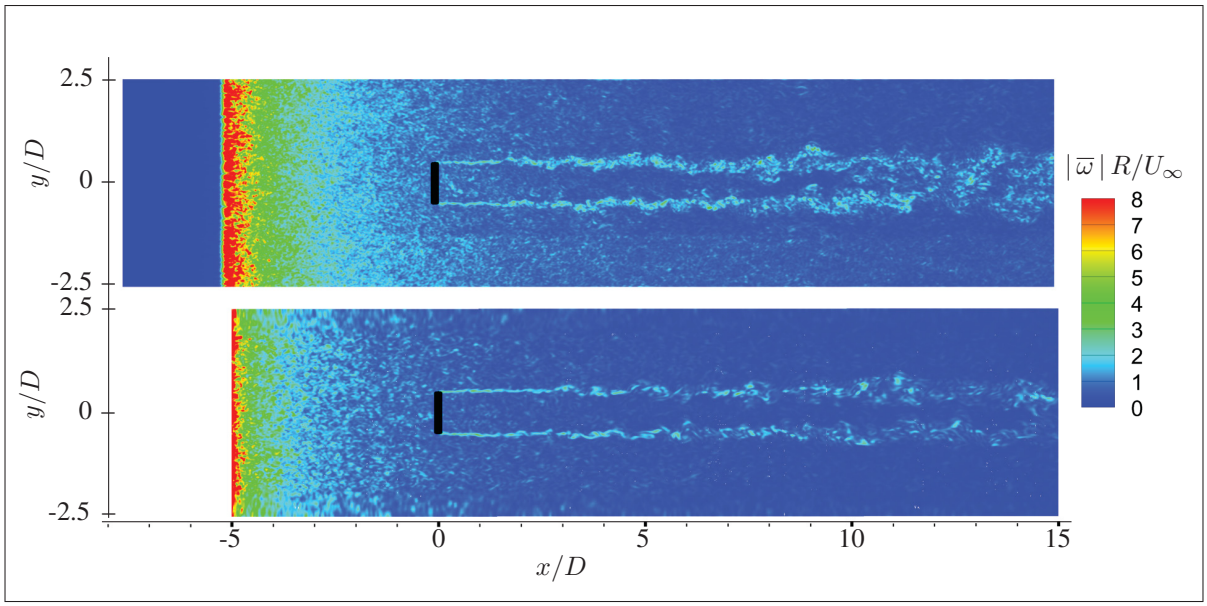

Figure 4.31 Vorticity field obtained with the Ti3 inflow and disk $C_T = 0.42$. Results of EllipSys3D (top) and OpenFOAM (bottom).

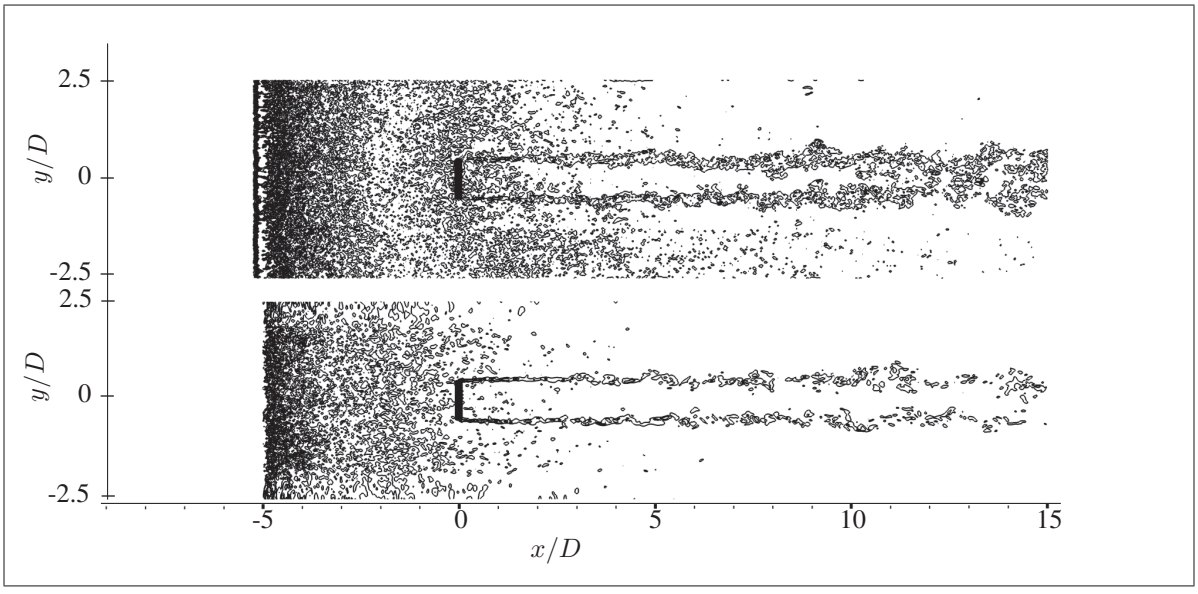

Figure 4.32 Contours of the vorticity field obtained with the Ti3 inflow and disk $C_T = 0.42$. Results of EllipSys3D (top) and OpenFOAM (bottom).

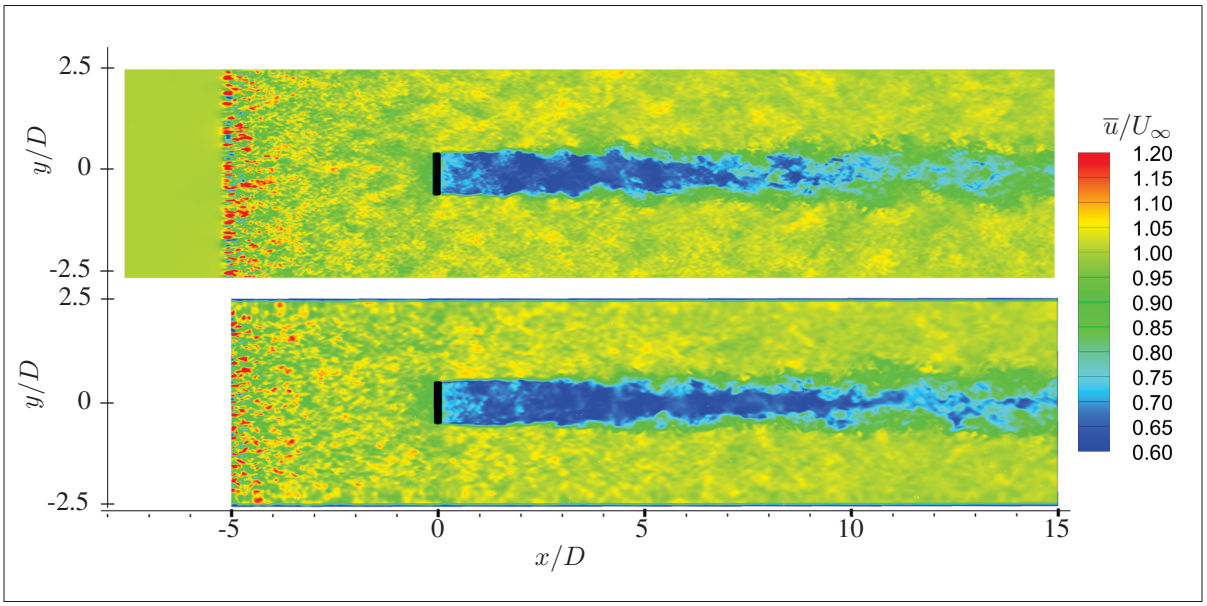

Figure 4.33 Instantaneous streamwise velocity using the Ti3 inflow and disk $C_T = 0.62$. Results of EllipSys3D (top) and OpenFOAM (bottom).

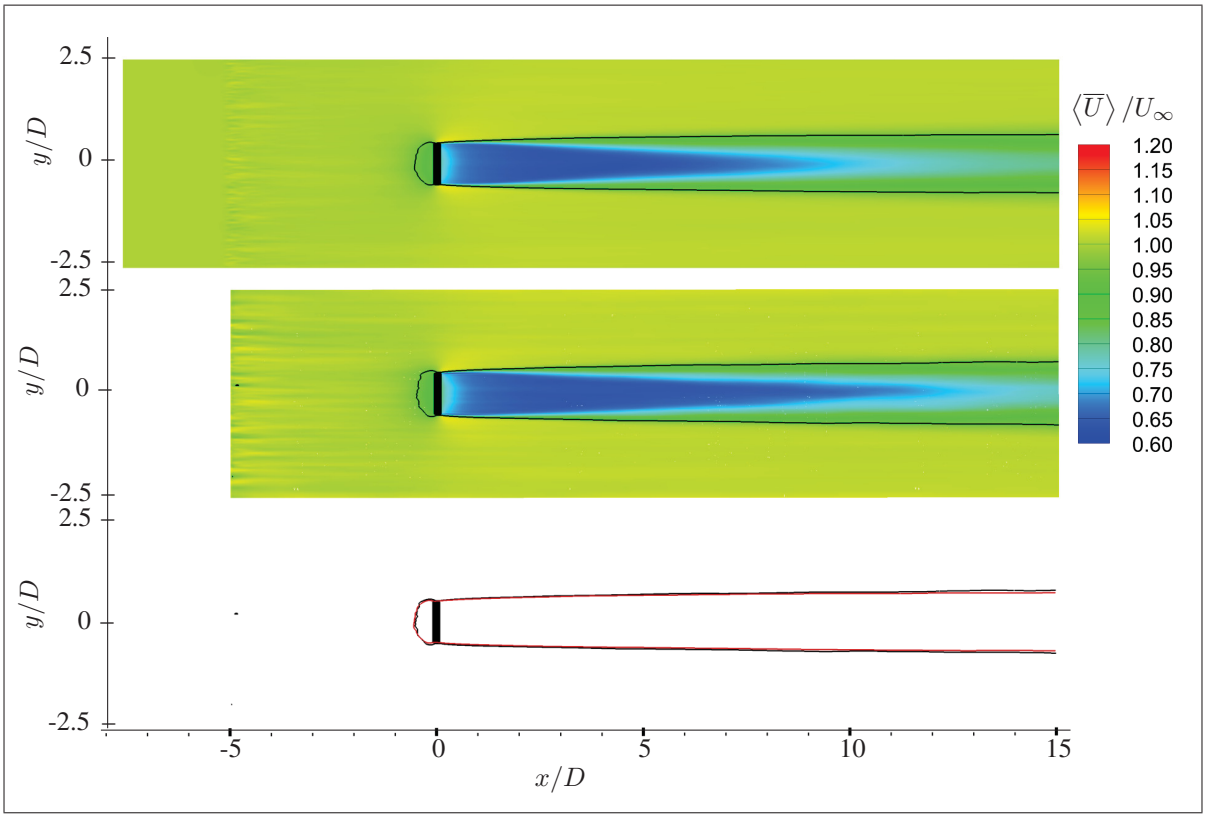

Figure 4.34 Average streamwise velocity using the Ti3 inflow and disk $C_T = 0.62$. Results of EllipSys3D (top) and OpenFOAM (middle). The bottom figure overlaps both envelopes, OpenFOAM (black) and EllipSys3D (red).

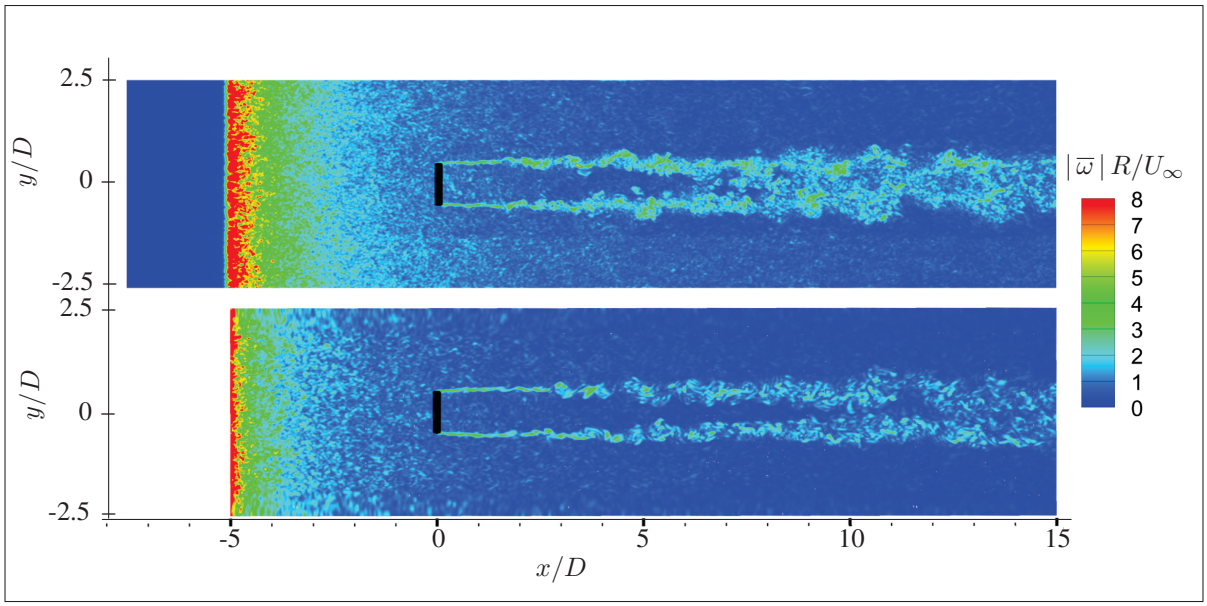

Figure 4.35 Vorticity field obtained with the Ti3 inflow and disk $C_T = 0.62$. Results of EllipSys3D (top) and OpenFOAM (bottom).

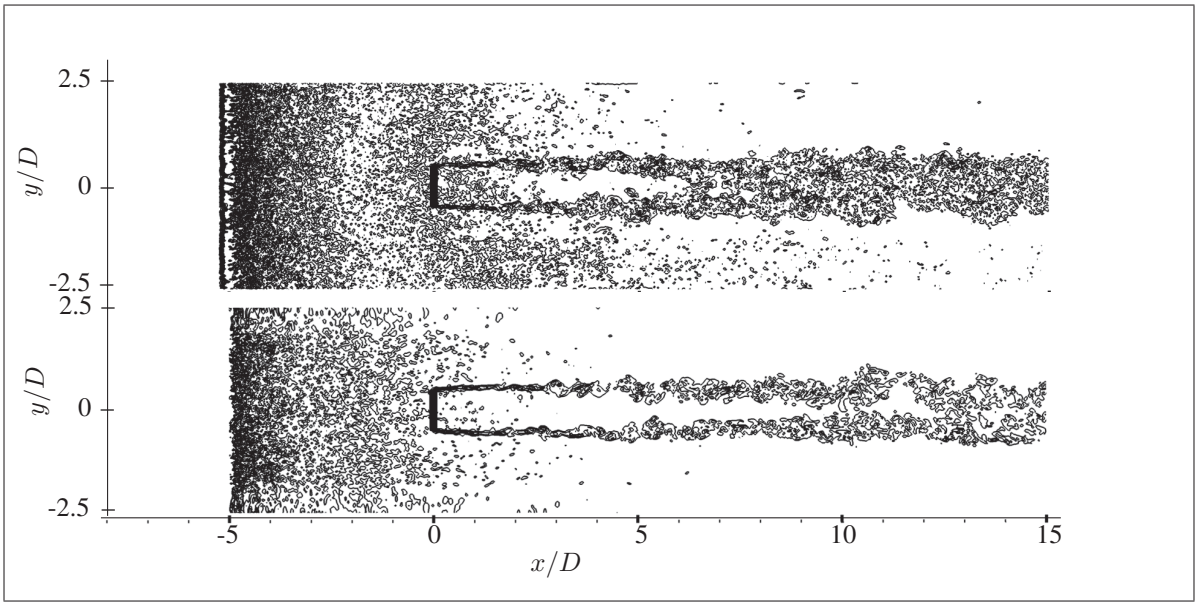

Figure 4.36 Contours of the vorticity field obtained with the Ti3 inflow and disk $C_T = 0.62$. Results of EllipSys3D (top) and OpenFOAM (bottom).

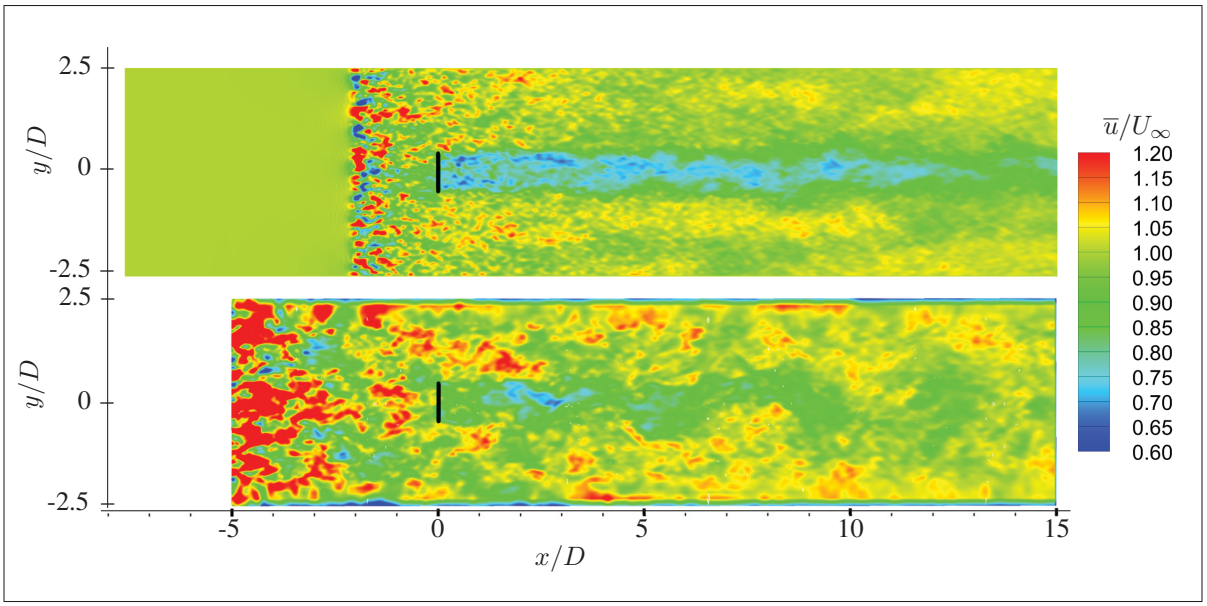

Figure 4.37 Instantaneous streamwise velocity using the Ti12 inflow and disk $C_T = 0.45$. Results of EllipSys3D (top) and OpenFOAM (bottom).

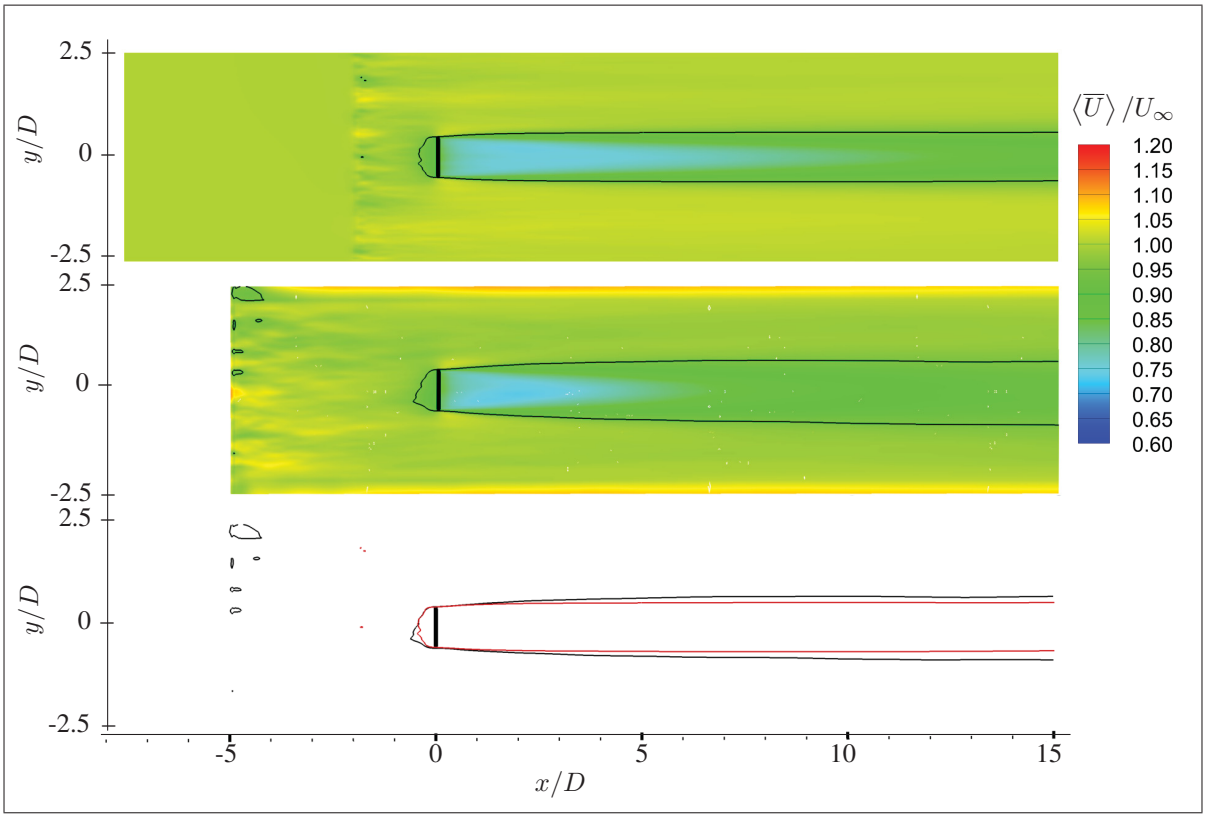

Figure 4.38 Average streamwise velocity using the Ti12 inflow and disk $C_T = 0.45$. Results of EllipSys3D (top) and OpenFOAM (middle). The bottom figure overlaps both envelopes, OpenFOAM (black) and EllipSys3D (red).

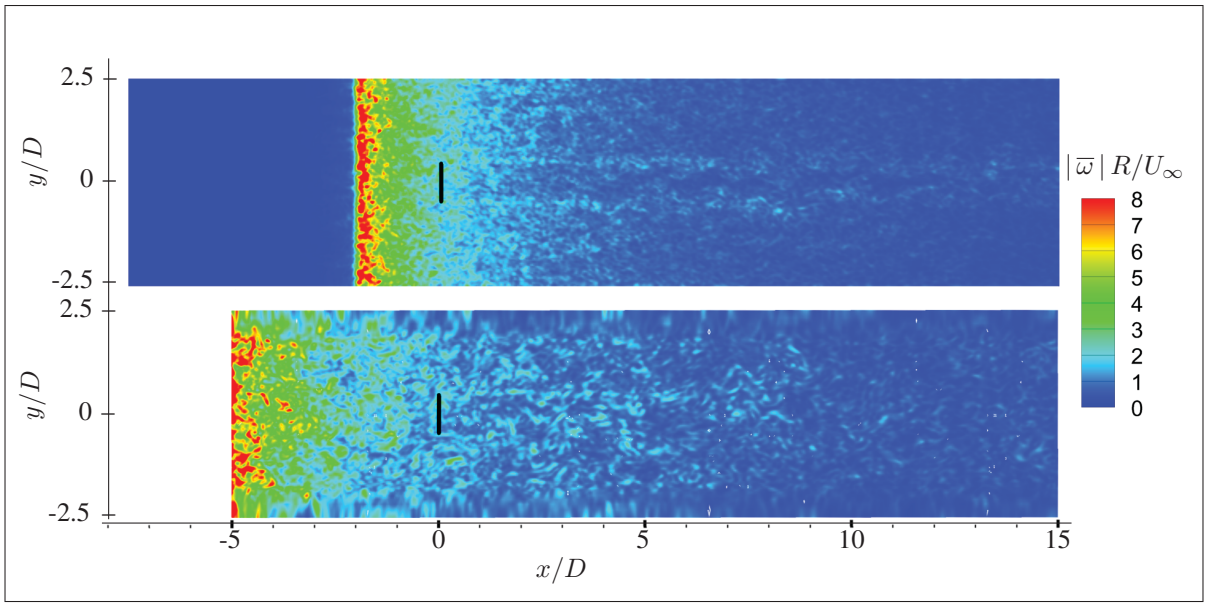

Figure 4.39 Vorticity field obtained with the Ti12 inflow and disk $C_T = 0.45$. Results of EllipSys3D (top) and OpenFOAM (bottom).

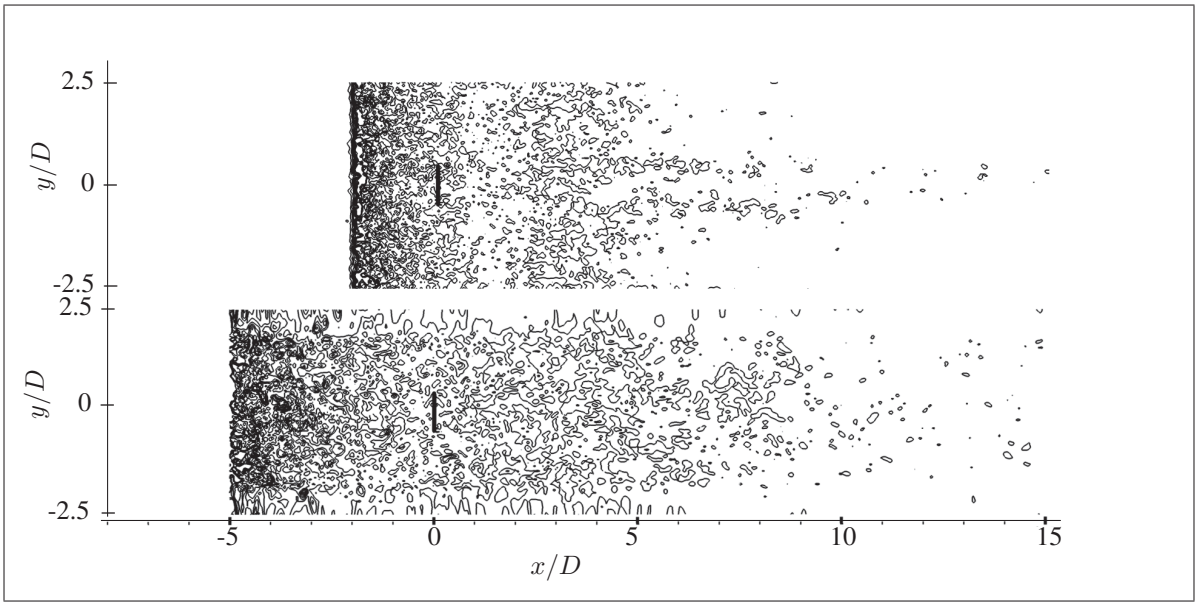

Figure 4.40 Contours of the vorticity field obtained with the Ti12 inflow and disk $C_T = 0.45$. Results of EllipSys3D (top) and OpenFOAM (bottom).

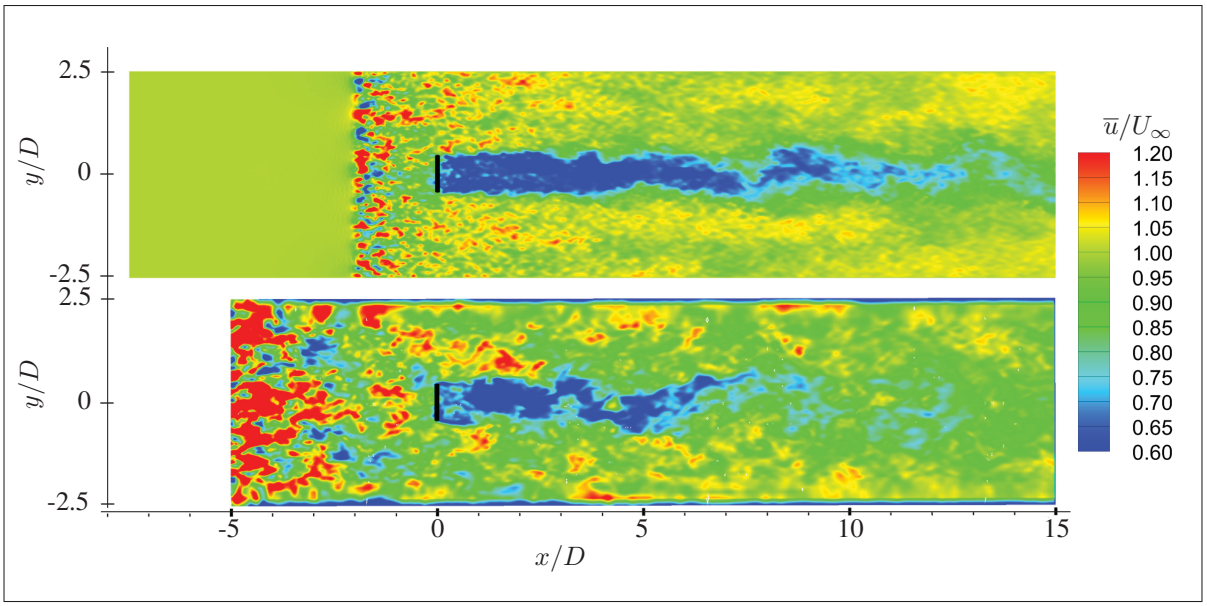

Figure 4.41 Instantaneous streamwise velocity using the Ti12 inflow and disk $C_T = 0.71$. Results of EllipSys3D (top) and OpenFOAM (bottom).

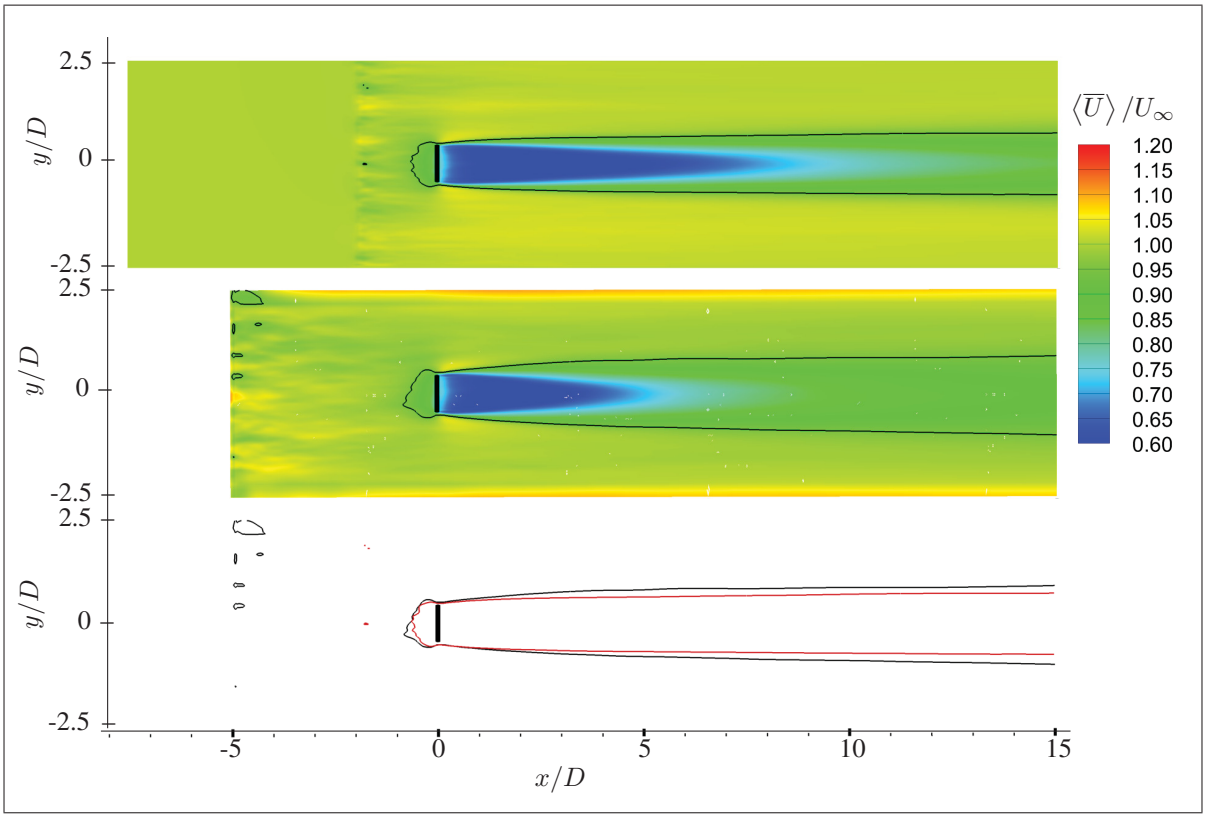

Figure 4.42 Average streamwise velocity using the Ti12 inflow and disk $C_T = 0.71$. Results of EllipSys3D (top) and OpenFOAM (middle). The bottom figure overlaps both envelopes, OpenFOAM (black) and EllipSys3D (red).

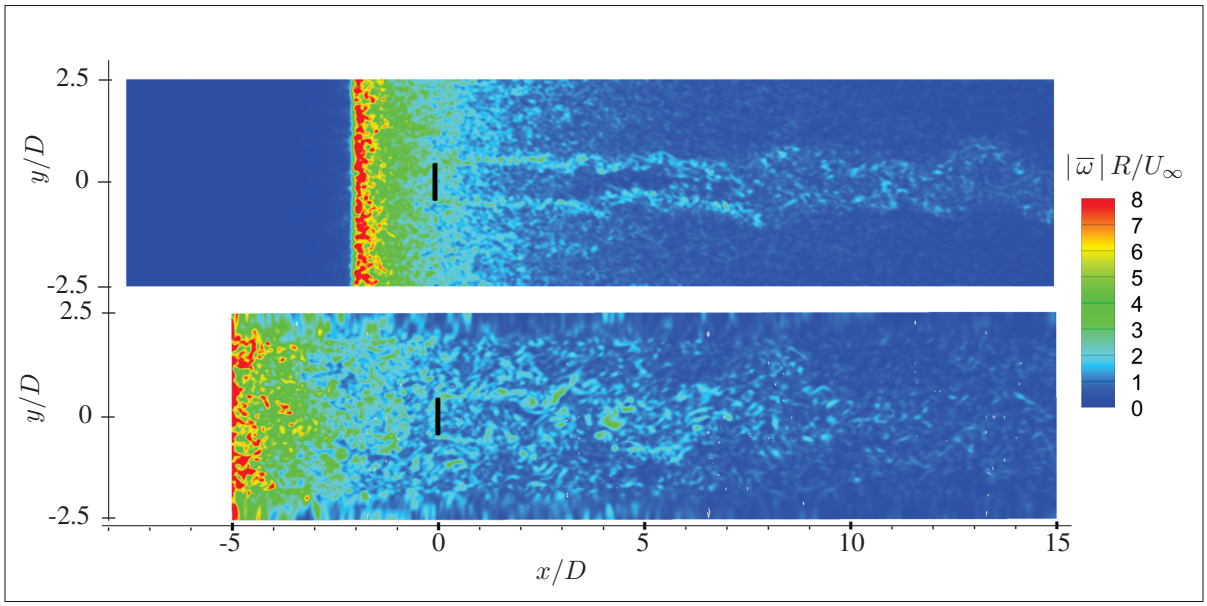

Figure 4.43 Vorticity field obtained with the Ti12 inflow and disk $C_T = 0.71$. Results of EllipSys3D (top) and OpenFOAM (bottom).

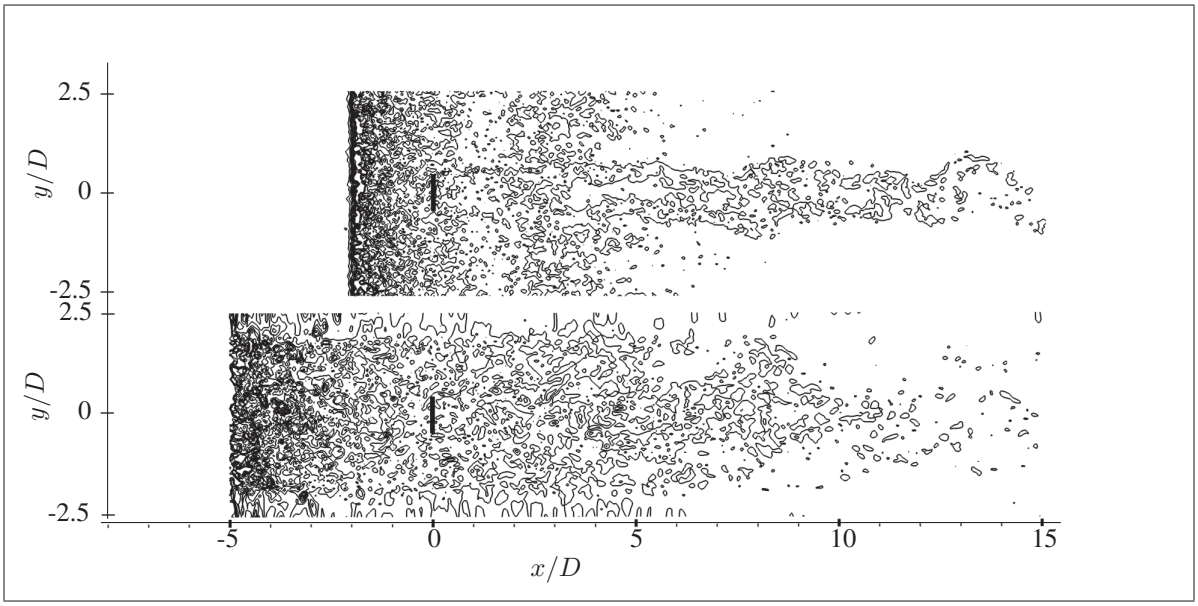

Figure 4.44 Contours of the vorticity field obtained with the Ti12 inflow and disk $C_T = 0.71$. Results of EllipSys3D (top) and OpenFOAM (bottom).

4.3 Summary and conclusions

In this Chapter we have shown a methodology to model and study the wakes produced by porous disks in a homogeneous turbulence inflow. The instances of turbulent inflow correspond to those studied in the previous Chapter. The methodology is employed to reproduce wake measurements made in a wind tunnel experiment, which serves as a validation procedure. Following this procedure, LES computations have been performed in OpenFOAM employing the actuator disk technique. In addition, simulations have been carried out with EllipSys3D. The comparison of the results between these two platforms is complemented by previous work made with RANS, wherever possible. While the numerical setup in OpenFOAM has been chosen for its adequacy to this type of study, the setup in EllipSys3D is taken from previous works of wake simulations on the atmospheric and homogeneous flows. In other words, a *common practice* configuration for wake computations is employed in EllipSys3D to compare with our OpenFOAM implementation.

While the velocity deficit along the wake is well reproduced by both codes, some differences arise in the computation of the turbulence kinetic energy k and its dissipation ε. This can be partly explained due to the choices made to attain each of the desired streamwise turbulence intensity values (TI) of about 3% and 12% at the disk positions. Therefore, for each of these values, we are presented with a different scenario. In the first one, ambience turbulence conditions are similar along the wake in both codes and EllipSys3D predicts a faster convergence of the shear layer towards the centreline than in OpenFOAM. In the second one, TI are approximately the same in both codes only at the disk position, due to the stronger decay observed in EllipSys3D. As a result, the stronger ambience turbulence in OpenFOAM prompts a faster mixing with the shear layer precipitating a fully turbulent wake at a shorter downstream distance than in EllipSys3D. Consequently, in the first scenario we obtain a longer, more turbulent wake in EllipSys3D while in the second one, the situation is reversed. These findings are in general more evident for the disks with higher thrust coefficients, which can also be rapidly identified through different visualizations of the wake structure. For most of the wake, the results obtained with OpenFOAM approach better to quantities acquired from experimental

data than those from EllipSys3D. As argued before for the computations of the decaying-HIT shown in the previous Chapter, a possible explanation for the differences observed between the computations of each code is the disparities in the ratio of the upwind contribution in the advection schemes.

A study of the LES modelling of the wake was also performed, with the additional interest of comparing the results of the distinct SGS methods applied in each code. By studying the ratio of the resolved and subgrid parts with respect to their total value, it is found that the modelling of k in the wake is largely maintained with respect to the outside flow, with a variation only at the shear layer near the disk with the low TI inflow (3%). Likewise, the effect of shear in the modelling of ε is more evident but only under the low TI inflow, with an increase of the subgrid part in this region. While no observable differences can be unequivocally attributed to the use of different SGS models, it can be inferred that modelling in the freestream flow prevails in the wake just as the level of inflow turbulence increases. On the other hand, while the RANS results for the velocity and k behind the wake are fairly good, ε seems to be overestimated in the regions of stronger shear or high TI.

Longitudinal integral lengthscales (L_1) computed at different parts of the wakes evolve, for the most part, as in the decaying homogeneous turbulence. An increase in L_1 can be deduced at the shear layer only from the results of OpenFOAM with the low TI turbulence. Moreover, with the increased TI in the inflow, L_1 computed from measurements do not reveal an appreciable change within the shear layer. While the results obtained with OpenFOAM point in the same direction, fluctuations observed in the results of EllipSys3D difficult the observation of any tendency. Nevertheless, our observations point towards the fact that turbulence scales in the wake appear to be dominated by the inflow characteristics (where $L_1 < D$). This effect increases with the level of TI in the inflow.

Lastly, spectra computed at different axial positions in the wake reveal that shear induces a noticeable boost in the energy content of turbulence, but only in the low TI case. This causes that for the two furthermost positions ($x = 10D$ and $14D$), the energy levels are higher or at

least as energetic as in the upstream region near the disks. Moreover, the turbulence at those positions shows a clear inertial range that was absent in the decaying turbulence at low TI. Conversely, for the high TI inflow, it is seen that despite that turbulence energy levels rise in the wake with respect to the decaying homogeneous flow, the relative decay is maintained from one position to the other. Also, differences in the energy distribution are found between results of each code, as spectra from EllipSys3D show that small-scale, resolved fluctuations are more energetic than in OpenFOAM. This in turn, can be a consequence of the different SGS models employed in the computations.

CHAPTER 5

COMPARISON OF WAKE CHARACTERISTICS USING UNIFORM AND BLADE ELEMENT-BASED ACTUATOR DISKS

In this Chapter we assess the differences in the turbulence characteristics of wakes produced by two rotor models under a non-sheared inflow. To this aim, the Actuator Disk (AD) technique is applied to model a uniformly loaded disk and an AD model based on the blade element theory that employs tabulated airfoil data to calculate the distribution of forces over the disk and other physical parameters from a conceptual 5 MW offshore wind turbine. Moreover, the latter AD model makes use of a control system to adjust the rotational velocity to the conditions of the wind inflow. LES are employed to analyse the main wake properties over non-turbulent and turbulent inflow conditions. In the latter case, the turbulence is pre-generated using the Mann model, to produce a turbulent field with the same characteristics of the atmospheric turbulence. The turbulence is introduced in the computational domain at a position ahead of the rotor instead of at the inlet, to minimize its decay as it is convected downstream in the domain. To achieve this, a method has been implemented in OpenFOAM that resembles the technique previously employed in the computations of EllipSys3D. While the analysis of the wake turbulence features is less detailed than what was showed before, the objective in this part of the work is directed to observe the principal differences in the wake representation by the AD models. Likewise, we assess the accuracy of our implementation of the blade element-based AD with respect to the known performance of the modelled turbine. Lastly, we examine the capabilities of the controller implementation to effectively simulate the rotor response to the inflow conditions.

5.1 Model description

5.1.1 Rotor models

To carry out the comparison of the main turbulence properties in the wake, we employ the models described in Chapter 2; in Sec. 2.3.1 for the uniformly loaded disk and in Sec. 2.3.2 for

the disk with induced tangential velocity, where the lift and drag coefficients are obtained from tabulated airfoil data, simply referred to as the *rotating AD*. For the first model, a validation of our implementation has been provided next to its definition as it has been the disk model used in the wake computations of Chapter 4. Conversely, a validation procedure is incorporated in this Chapter for the rotating disk. Indeed, as this technique is applied to model a particular rotor with a known performance, it is verified that parameters such as rotational velocity and power output agree with the magnitudes provided by the designer. The validation procedure has also the objective of proving the implementation of the rotational control method described in Sec. 2.3.2.1, to represent the actual functioning of wind turbines, where the rotating speed adjusts to the changing wind velocity conditions. Although our simulations are performed with a constant inflow mean velocity, the rotor is expected adjust to the varying inflow velocities of the imposed turbulence field.

5.1.2 Reference turbine

Airfoil parameters are obtained from the concept of a 5 MW offshore wind turbine designed by the National Renewable Energy Laboratory (NREL) (Jonkman *et al.*, 2009). This is a conventional horizontal axis, three bladed (twisted and tapered), pitch-controlled and variable speed turbine created from design information of other turbines, mainly the REpower 5M. The diameter of the rotor is 126 m set at a hub height of 90 m, with a peak power coefficient of $C_P = 0.482$, found when the tip-speed-ratio has a value of $\Lambda = 7.55$ and the blade pitch angle is zero. Information regarding the torque vs. speed response of the turbine is also contained in that report. These data are then used to regulate the angular velocity of the turbine according to the description provided in Sec. 2.3.2.1. The modelling of this wind turbine comprises only the rotor, excluding the tower and nacelle.

5.2 Numerical Setup

5.2.1 Independence of computational domain size, mesh and AD distribution

Before the performing wake computations, we assess the independence of results with respect to the computational domain size, grid density and longitudinal distribution of momentum sources. For these sensitivity studies (as well as for the subsequent wake computations), a uniform inflow of $U_0 = 8$ m/s is set at the inlet. The side boundaries are set to periodic while the top and bottom are symmetry planes. At the outlet, Neumann boundary conditions are imposed. In these simulations, the AD with rotation has a fixed rotational velocity of $\Omega_0 = 9.16$ RPM which corresponds to the peak power coefficient as reported by the designer. These tests are performed using a RANS solver for laminar flow under inviscid conditions and with the SIMPLE algorithm, akin to the AD validation performed in Sec. 2.3.1.1. This examination process is two-folded. First, the uniformly loaded AD with a fixed $C_T = 8/9$ is used to look at the change in the axial induction factor a. Secondly, with the AD with rotation, the variation of the performance of the turbine through its C_P and C_T values is observed. For these coefficients, the values provided by the designer of the reference turbine (see Table 5.1) are used for comparison.

To begin, a set of domain dimensions used in previous, similar studies (Ivanell, 2009; Breton *et al.*, 2012; Olivares-Espinosa *et al.*, 2014) is used as a starting point. This computational domain consists of a rectangular mesh of size $\mathrm{L}_x \times \mathrm{L}_y \times \mathrm{L}_z = 15.2D \times 8.5D \times 8.5D$ in the streamwise, vertical and spanwise directions. A central region where cells are equally spaced in the flow direction x is located at $3.2D$ from the inlet and continues until the outlet. The AD is located within this zone, at $4D$ from the inlet, centred in the crosswise plane. The coordinate system is as in the previous Chapters, i.e. the position $x = y = z = 0$ is located at the disk centre. The uniform cell region is separated from all the lateral boundaries by a distance of $3.45D$. Outside this region, the cells are stretched towards the boundaries. The inlet/outlet boundaries of this domain are thought to be far enough from the AD location to

have a considerable influence in the flow solution around it, so when the domain size is varied only changes in the lateral boundaries are considered.

For the domain size independence, five different lateral sizes are studied, 12, 17, 20, 25 and $32R$ and results are shown in Figure 5.1 (note that radial units are used when describing changes at the disk). There, almost no variation is seen in a for domains larger than $20R$. The case is similar for C_P and C_T and although they exhibit a more obvious asymptotic convergence, their difference is notably small. Therefore, a value of $20R$ is chosen for the domain side.

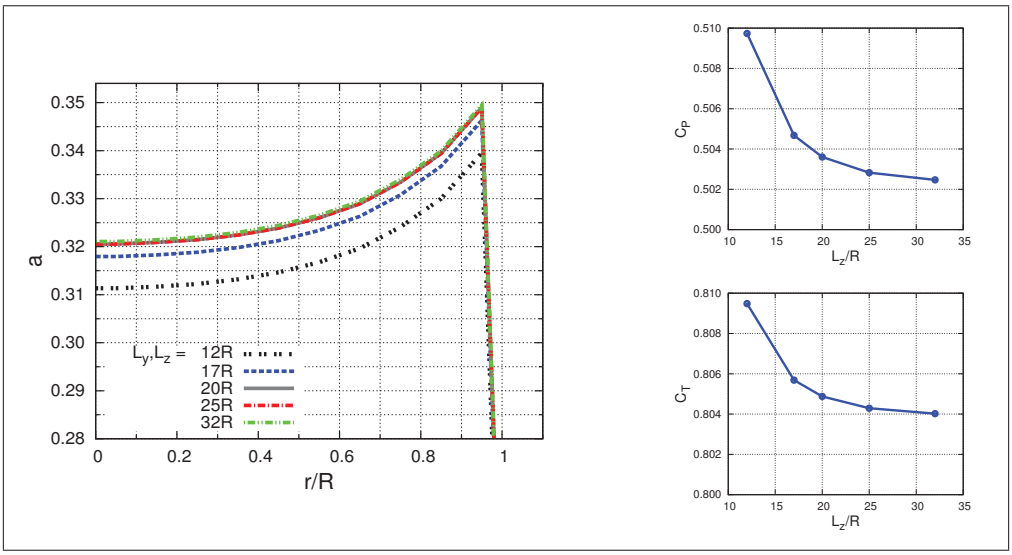

Figure 5.1 Domain independence study. *Left:* the axial induction along the surface of the uniformly loaded disk. *Right:* performance of the wind turbine for the AD with rotation. Values obtained varying the side length of the domain.

For the mesh independence study, the effect of varying the number of cells within an AD radius is analysed. This cell resolution is used throughout the central, uniform mesh region. Outside this region, the aspect ratio of the cells is kept about the same with respect to the reference mesh, maintaining a smooth transition between these and the uniform region cells. With these resolutions, the number of cells $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_z$ is about 1.2×10^6, 9.6×10^6, 16.5×10^6 and 25.5×10^6 cells, for each case. Results are shown in Figure 5.2, where it is observed

that variation in induction factor amongst the different resolution is minimal, especially for a resolution of $20/R$ and larger. Similarly, C_P barely changes after this resolution whereas C_T exhibits a dissimilar increase (although also very small) for the same resolution, perhaps as a result of an oscillatory convergence. We opt to work with a resolution of 20 cells per R, also to maintain the number of cells not too high, considering the computational expense of the turbulent simulations.

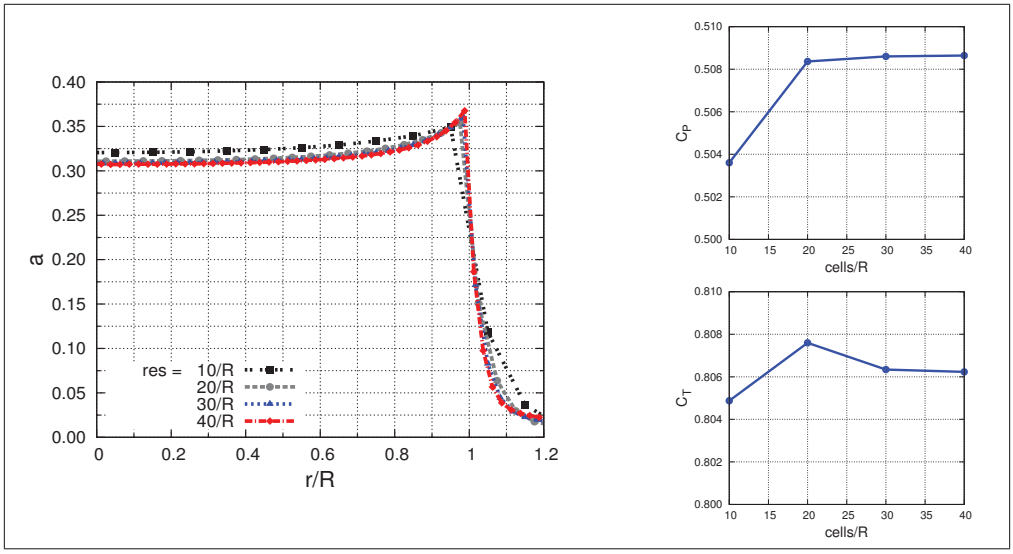

Figure 5.2 Mesh Independence study. *Left:* the axial induction along the surface of the uniformly loaded disk. *Right:* performance of the wind turbine obtained with the AD model with rotation. The cuves show the variation of the results according to the resolution used for the central region of uniform cells, where the AD and wake are located.

Finally, the influence of the extension of the Gaussian force distribution used for the AD is explored. In this case, the value of σ is taken as an integral number of the cell length, varying between Δx and $4\Delta x$. As it can be seen in Figure 5.3, the axial induction is more sensitive to the variation of this parameter. The variation appreciably changes from the case $\sigma = 2\Delta x$. The analysis of C_P and C_T is less evident, as their values move away from the expected values (see Table 5.1). In this case, the election of the distribution width is made considering also the thickness of the AD. Indeed, as σ increases, the AD shape looks less like an actual rotor, so it is preferred to keep its thickness at its minimum. In this regard, it is observed that the first case

when wiggles disappear is when $\sigma = 2\Delta x$ is used. Therefore, this is the value employed in the subsequent computations.

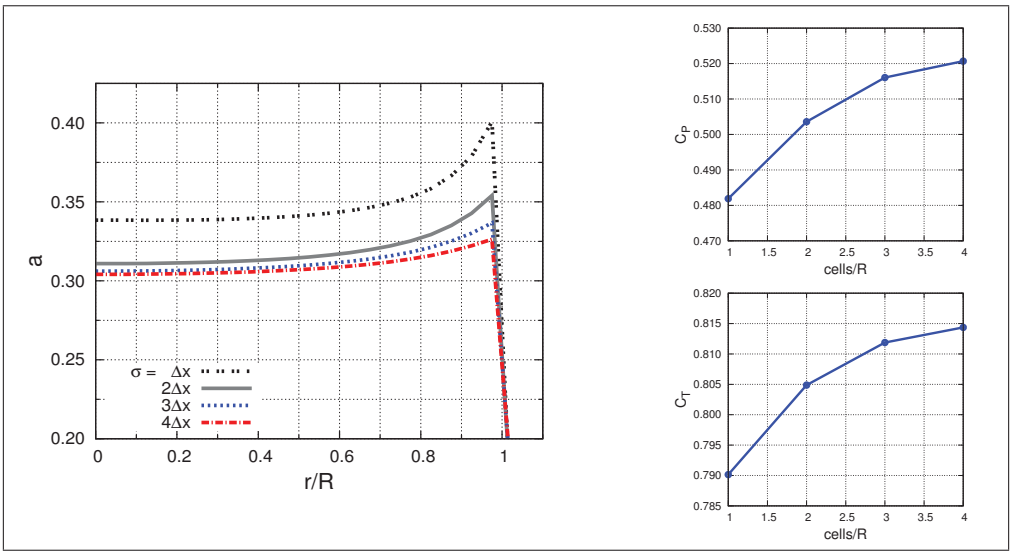

Figure 5.3 Response to the variation of σ of the gaussian distribution of forces. *Left:* the axial induction along the blade for the uniformly loaded disk. *Right:* performance of the wind turbine obtained with the AD model with rotation.

5.2.2 Numerical model

Taking into account the sensitivity studies of the previous section, the computational domain consists of a rectangular mesh of size $\mathrm{L}_x \times \mathrm{L}_y \times \mathrm{L}_z = 15.2D \times 10D \times 10D$, with a number of points equal to $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_z = 240 \times 136 \times 136$. The central region is comprised by uniform cells of side length $\Delta = 0.025D$. AD location, inflow and boundary conditions are the same described in Sec. 5.2.1. Simulations are performed using the LES model coupled with the Smagorinsky technique to model the effect of the subgrid scales. A QUICK interpolation scheme is applied for the solution of convection terms (see Appendix II for the dictionaries containing the numerical parameters).

For the simulations with turbulence, the Mann technique is employed to produce a synthetic velocity field that resembles the characteristics of the atmospheric turbulence. To this aim,

the parameters provided by the standard of the *International Electrotechnical Commission* for wind turbine design (IEC, 2005) are employed within our implementation of the Mann model (described in Sec. 2.5.1). These parameters are in turn based on those obtained from a fit of the model results to the Kaimal spectra by Mann (1998), as shown in Sec. 2.5.2. Then, turbulence is pre-generated in a domain of $\mathrm{L}_{\mathrm{B},x} \times \mathrm{L}_{\mathrm{B},y} \times \mathrm{L}_{\mathrm{B},z} = 102.4D \times 1.6D \times 1.6D$ with $\mathrm{N}_{\mathrm{B},x} \times \mathrm{N}_{\mathrm{B},y} \times \mathrm{N}_{\mathrm{B},z} = 4096 \times 64 \times 64$ uniformly distributed cells, where the fluctuations are imposed over a uniform velocity field equal to U_∞. Make note that ABL turbulence imposed over a non-sheared flow has also been employed in other works to study wake characteristics produced by rotor models, such as Troldborg (2008) and Breton *et al.* (2014).

To introduce the turbulence into the computational domain, we emulate the technique employed in EllipSys3D previously in this work, described in Sec. 3.2.2. This is, the turbulent velocity field is introduced at a plane ahead of the AD instead of the inlet. This technique is applied in order to minimize the turbulence decay, as exposed in previous works (e.g. Troldborg, 2008; Ivanell, 2009; Nilsson *et al.*, 2015) with EllipSys3D. However, unlike what is done in that code, the turbulent velocity field is directly introduced at the turbulence plane (TP) instead of the more sophisticated method of imposing body forces to generate the desired velocity fluctuations. As in the implementation used by Troldborg (2008), the TP consists of a square with a cross-section area smaller than the one of the computational domain (that in our case coincides with that of the uniform region) and located near the rotor, at $3.2D$ from the inlet. In our computations, the TP is set in an analogous manner to a boundary condition, where a convective condition is set at the upstream side, so the uniform flow coming from the inlet exits at the TP while it is been replaced by the turbulent velocity field (the inflow outside the TP area is left intact). Note that the cell resolution of the turbulence box is the same as in the uniform region of the domain. The synthetic velocity field is introduced through the TP with a procedure equivalent to that outlined in Sec. 3.2.2. The scheme employed for the turbulent simulations is presented in Figure 5.4. In this process, crosswise planes are extracted from the synthetic velocity field (turbulence box) and introduced at the turbulence plane. Intermediate velocity values between the available planes (separated by Δx_B) are computed with linear in-

terpolations. Evidently, the introduction of fluctuations at the TP represents a discontinuity in the flow field; however, the continuity and incompressibility are enforced by the LES solver so an adaptation of the turbulence field to the local conditions is to be expected. Therefore, the evolution of the fluctuations next to the TP and along the domain is also studied.

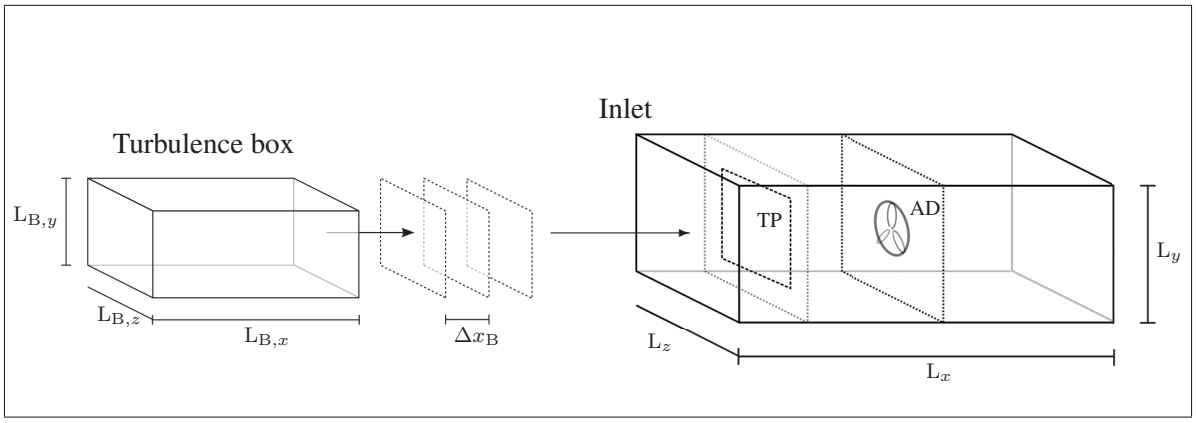

Figure 5.4 Introduction of synthetic turbulence field in the computational field. The turbulence plane (TP) has dimensions $3.45D \times 3.45D$, centred in the $y - z$ plane and located at $3.2D$ from the inlet. The AD is located at $4D$ from the inlet.

The ADs are exposed to two different inflow conditions: a non-turbulent and a turbulent inflows. Each computation is performed first using an adaptive time-step solver where the CFL number is kept below 0.6, during a period equal to 3 longitudinal flow times (LFT), employed to allow the full development of the wake and the stabilization of turbulence in the flow. This initial run provides the time-step that fulfils the CFL condition for the posterior runs, $\Delta t = 0.14$ (the smallest of all computations). In this way, computations are carried out during 10 LFT to record measurements and average values. Simulations are performed with both AD implementations, uniform load and AD with rotation. In the case of the latter, the controller is activated only after 0.5 LFT have passed at the initial run, as it was otherwise observed that a diverging rotational speed is produced by the controller due to the rotational velocity and torque not being well-predicted at the start. The starting value of Ω is 8 RPM. The load of the uniform AD is determined by the average C_T obtained from the AD with rotation under a non-turbulent inflow, which is found to be 0.8.

5.3 Results and discussion

5.3.1 Turbulence decay

As a first step we assess the properties of the turbulence field introduced in the computational domain in the absence of the rotor. To this aim, we track the evolution of the velocity components in the longitudinal direction at 10 positions distributed in the spanwise plane ($x - z$) of the TP, at the mid-height of the domain ($y = 0$) and averaging the results. These are shown in Figure 5.5, where we can see that the variation of the mean velocity components is minimal at the location of the TP and throughout the domain. The evolution of the streamwise turbulence intensity is also in that Figure. A small but noticeable decay occurs next to the TP, from about 6% to almost 4% at $x = 0$. From there, the decay is negligible for the remainder of the domain. Notably, there is also little difference ($< 0.5\%$) in the TI measured in the turbulence box with respect to that measured next to the TP. These results contrast to the large difference observed in Chapter 3 and are most likely the result of the small TI values employed in the current case.

The vertical distribution of the components of the mean velocity along the domain is shown in Figure 5.6. The values there correspond to the averages obtained from 10 vertical lines distributed in planes parallel to the TP, at each $x-$position. Even next to the TP at $x = -0.8D$, we observe that the mean values do not vary much, less than 2% with respect to the mean velocity. The variations are reduced longitudinally, for the rest of the positions. Figure 5.7 shows the evolution of k, for the values extracted and averaged at the same positions. The turbulence decay is appreciable only from next to the TP to $x = 0$, as shown before. Yet, the profiles show an increasing decay close to the edges of the region covered by the turbulent inflow, likely caused by the interaction of the fluctuations with the outer, uniform flow. From these results, we observe that the effects of the discontinuity in the flow field caused by the abrupt introduction of fluctuations are rather minimal. Throughout the domain, we obtain a consistent and sustained turbulence field adequate to be employed in the subsequent wake computations.

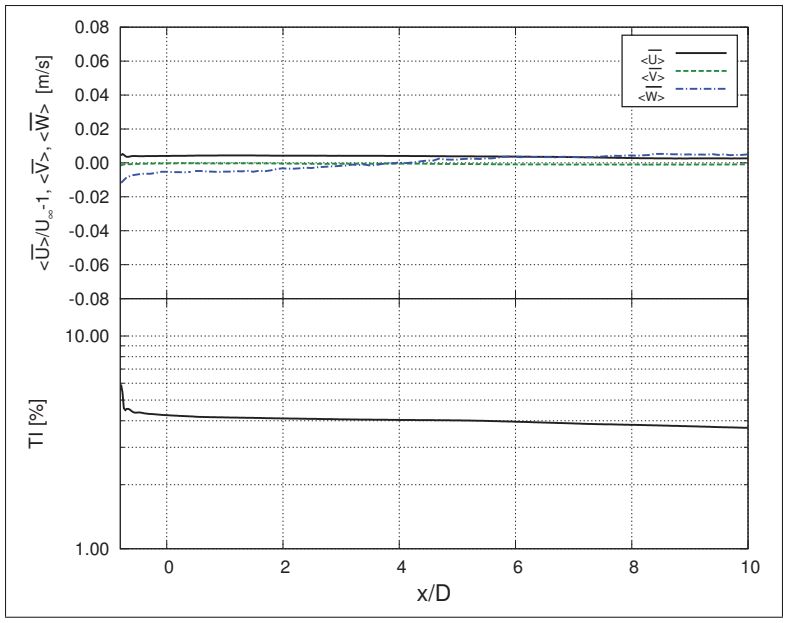

Figure 5.5 Longitudinal evolution of (top) mean velocities and (bottom) streamwise TI. The value of 4% at $x = 0$ (where the ADs are to be set) is used as a reference.

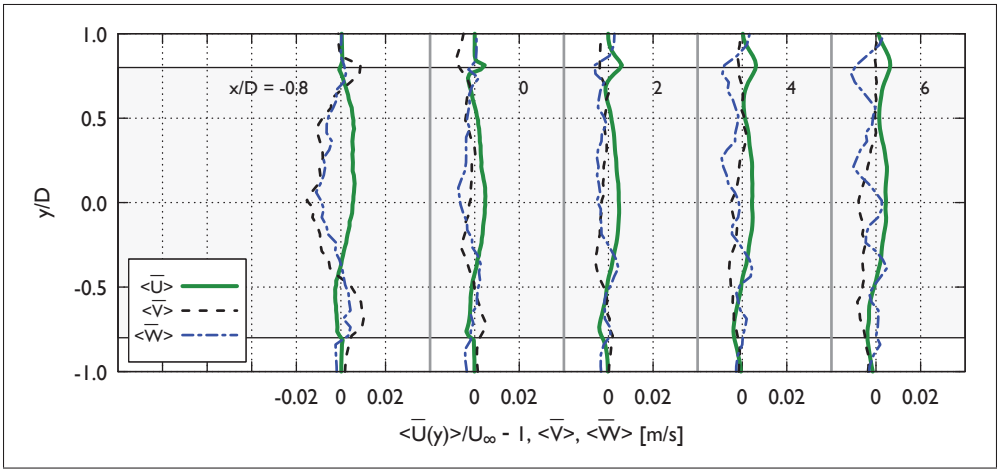

Figure 5.6 Vertical distributions of the velocity components along the domain. The shaded region is used to represent the side length of the turbulence plane.

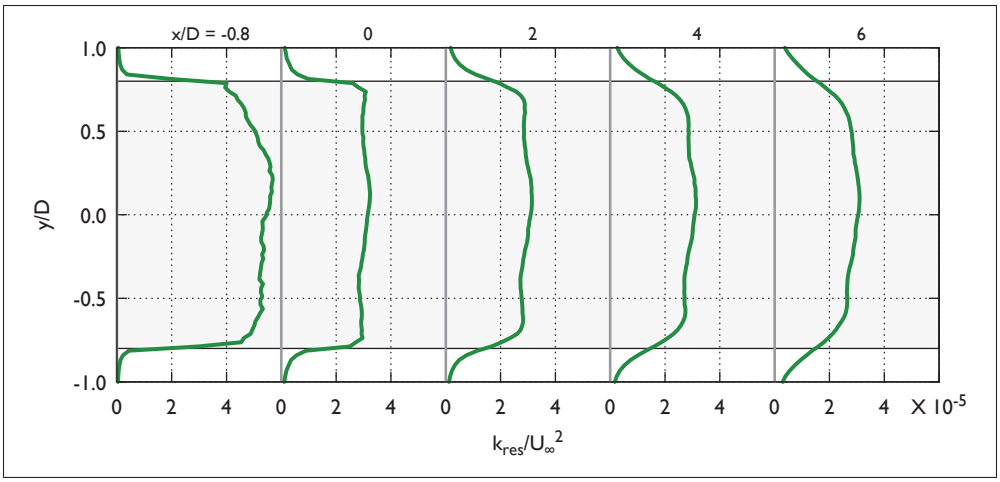

Figure 5.7 Vertical distributions of the turbulent kinetic energy along the domain. The shaded region is used to represent the side length of the turbulence plane.

5.3.2 Wake characteristics

After the assessment of the turbulence decay, we introduce the rotor in our computations to study the main wake characteristics and assess the differences between each AD model. In the following figures of wake results, the curves represent the average between profiles obtained in the vertical and spanwise directions, at each $x-$position. Figure 5.8 shows the velocity deficit obtained with each disk with the non-turbulent inflow. There, it is observed that the largest difference is caused by the absence of thrust force at the centre of the rotating AD. Even at the last position, the differences between the estimated wake velocities are still visible in this region. In Figure 5.9 we observe the results obtained with the turbulent inflow. We immediately recognise the effect of the turbulence in reducing the wake effects, causing the prediction from each model to be closer. It is observed that at $x = 6D$ the difference between the profiles is very small, and inexistant at $x = 10D$. However, it should be remarked that the velocity at the wake envelope estimated with each AD model is very similar in both inflow cases.

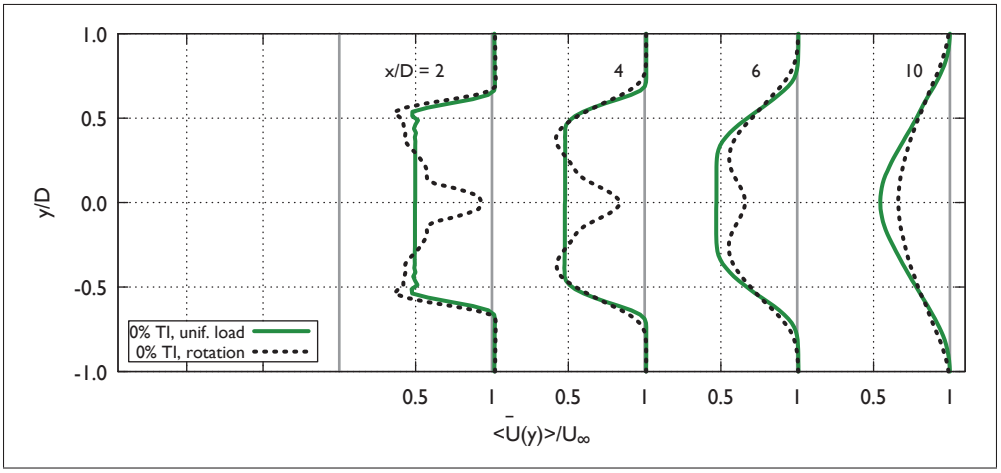

Figure 5.8 Vertical profiles of velocity deficit behind the disks with the non-turbulent inflow.

Differences are larger in the case of the turbulence along the wake. Figure 5.10 shows the results with the non-turbulent inflow (note that the scales are two orders of magnitude larger than in no-disk case). Unlike $\langle \overline{U}(y) \rangle$ at the wake edge, the estimation of k_{res} is appreciably

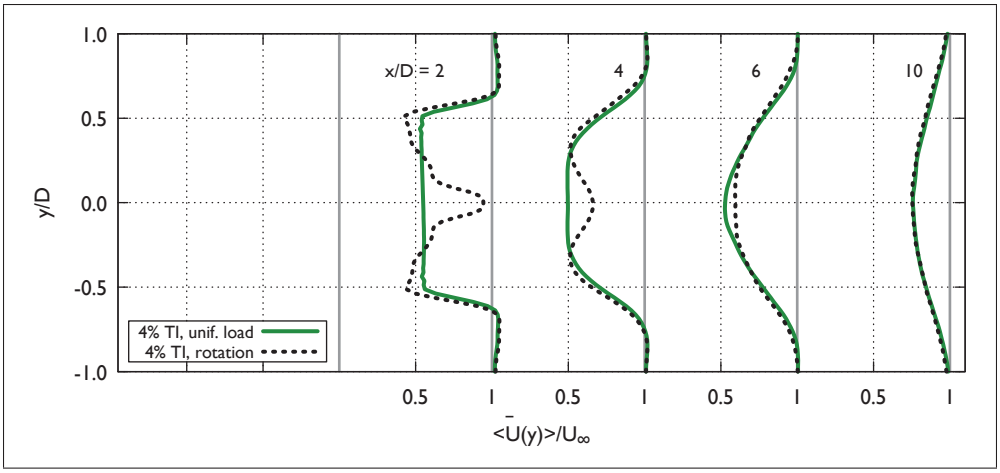

Figure 5.9 Vertical profiles of velocity deficit behind the disks with the turbulent inflow.

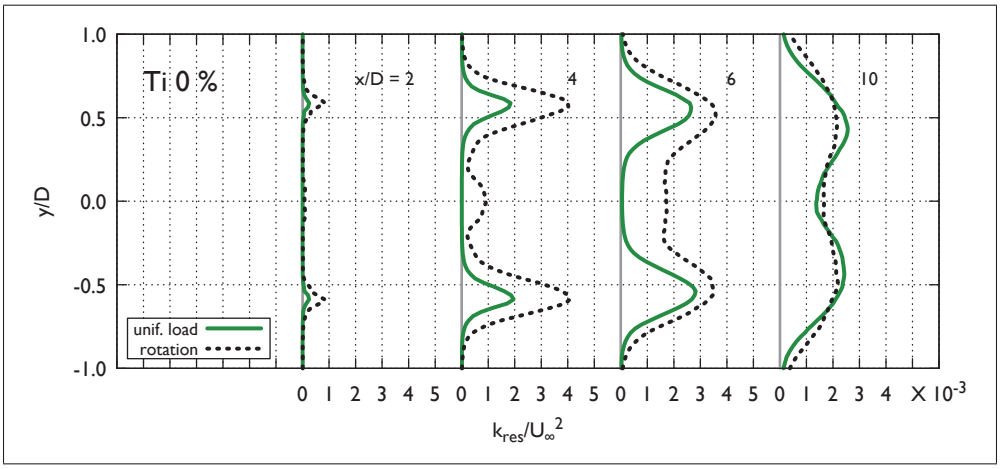

Figure 5.10 Vertical profiles of k behind the disk with the non-turbulent inflow.

different by each disk model. The largest differences occur for the middle longitudinal positions while the values are closer at the opposite ends of the wake. With the turbulence inflow, the differences are reduced and the profiles are basically equal from $x = 6D$, as shown in Figure 5.11. Interestingly, k_{res} is barely increased near the disks ($x = 2D$) with the non-turbulent inflow and largely increased further downstream. Comparatively, less variation is observed in the magnitudes of k along the wake when a turbulent inflow is used, due to the much larger values obtained at the first $x-$position (appreciably larger with the rotating AD). Similar re-

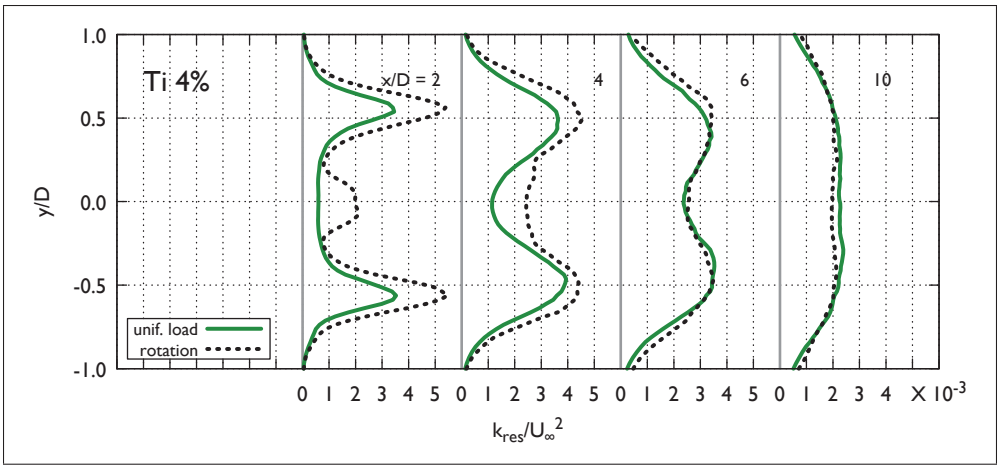

Figure 5.11 Vertical profiles of k behind the disk with the turbulent inflow.

sults were obtained by Troldborg *et al.* (2015) in a comparison of a rotating AD model with AL and a model of a fully resolved rotor geometry (FR) using DES. In that work, it is seen that in the absence of inflow turbulence, the values of k estimated by the rotating AD remain almost unnoticed at $x = 5D$ (the farthest position shown) in comparison with AL and more so with the FR model. Also, the turbulence values estimated with the AL and FR increase continually in the longitudinal direction for all the positions shown. When a turbulent inflow is used (with about the same TI as in this work), the difference in the estimations of each model are dramatically reduced, and the values at each $x-$position are essentially equal. Furthermore, it is mentioned also in that work that more than 70% of k is comprised by the resolved scales in the near wake ($1D$), as opposed to 90% in the far wake ($5D$). To investigate this and also to determine if a larger part of the shear-generated turbulence at $x = 2D$ occurs in the subgrid scales in the non-turbulent inflow as opposed to the turbulent inflow, we plot in Figures 5.12 and 5.13 the subgrid viscosity ν_{SGS} computed in each of these cases, by each rotor model. It is possible to see that in effect, the subgrid viscosity is larger when a non-turbulent inflow is used, specially near the disk. However, the differences are not very large and moreover, the magnitudes do not largely change from $x = 2D$ to the next position in the wakes modelled with the non-turbulent inflow. Conversely, the small values of k in the near disk region could stem from the lack of grid resolution to accurately represent the thin layer of wind shear at the wind

envelope, therefore limiting the production of turbulence. As the shear layer increases in thickness away from the disk, the effect of shear is better represented by the local grid, improving the depiction turbulence.

Figure 5.14 shows the mean velocity magnitude obtained at the middle vertical plane ($x - y$ at $z = 0$) for each rotor model and inflow. We see that in general, the differences in the wake velocities are more evident than when only the streamwise component is considered (Figures 5.8 and 5.9). Precisely, the velocity magnitude predicted in the non-turbulent cases by each disk is shown to be different for all the extension of the wake. The disparities are reduced when the turbulent inflow is used, although the predicted velocities seem still different at around $6D$, where the previous results for the streamwise velocity showed an agreement. In the same figure, we can also see that the extension of the wake is greatly reduced when turbulence is used at the inflow, specially so with the AD with uniform load.

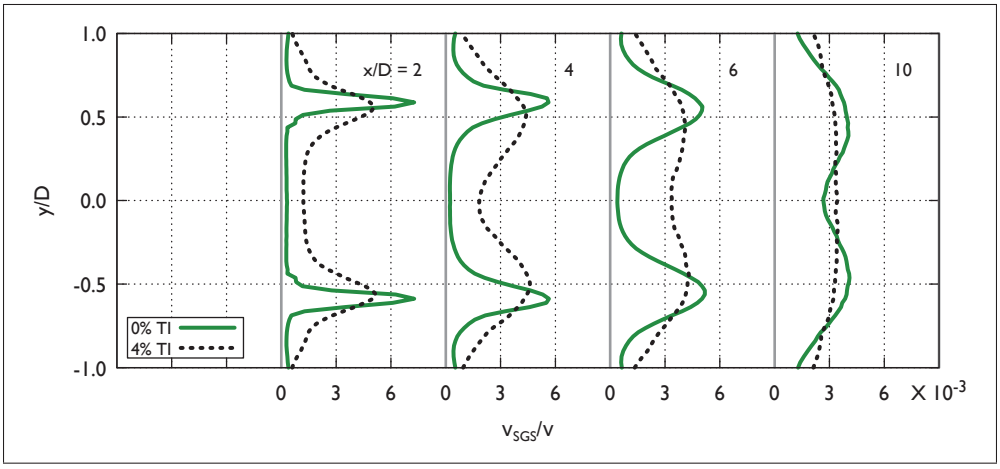

Figure 5.12 Vertical profiles of ν_{SGS}, normalized by the molecular viscosity, behind the uniformly loaded AD with and without inflow turbulence.

In Figure 5.15 the vorticity field magnitude is used to visualize the wake structure in each simulation. In the case of the non-turbulent inflow, disturbances in the shear layer develop earlier when using the AD with rotation (at $x \sim 2D$) than with the uniformly loaded technique (at $\sim 4D$). The turbulence field does not look similar until just before the outlet, from $x \sim 9D$.

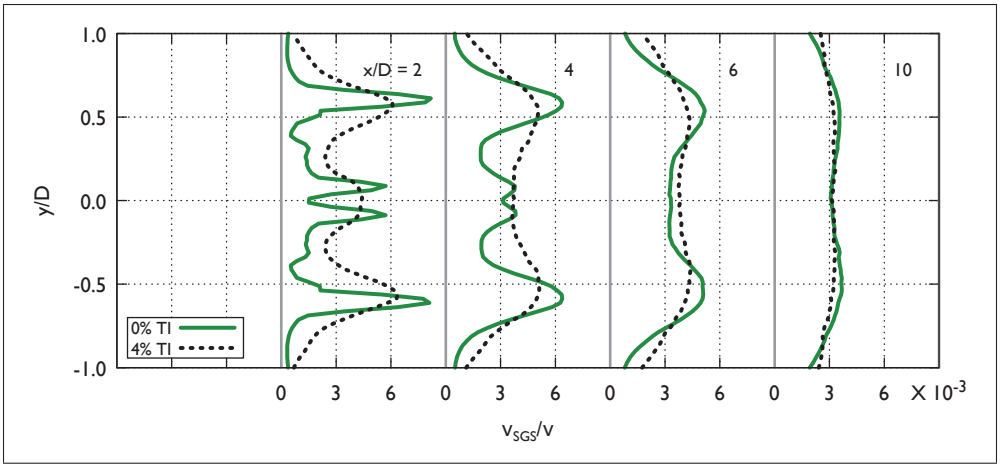

Figure 5.13 Vertical profiles of ν_{SGS}, normalized by the molecular viscosity, behind the rotating AD with and without inflow turbulence.

As expected, the incoming turbulence triggers the apparition of instabilities in the shear layer much sooner than in the non-turbulent inflow cases. We can observe that these structures appear to develop at about the same region behind the rotor when using one or the other AD models (slightly earlier in the rotating AD case, at $x \sim 1.5D$). These observations are complemented by the features observed in Figure 5.16, where the vorticity contours illustrate the turbulence structures appearing in the case of the AD with rotation under the different inflows.

5.3.3 Rotor performance

The values of Ω, C_P and C_T obtained from the simulations of the AD with rotation under the different inflow conditions are shown in Table 5.1. These are compared to the reported values from the turbine design (Jonkman *et al.*, 2009), obtained by means of FAST and AeroDyn simulations at $\Omega_0 = 9.16$. In addition, the values obtained using an in-house BEM code are included next to the results obtained from a steady-state (RANS) computation (performed as in the sensitivity study in Sec. 5.2.1). The agreement between the reported values of the designer and the steady-state simulation are very good, being the largest difference that of the C_T, that is underestimated if the total thrust reported is assumed as 500 kN, which is not exact as it is read from a curve in the publication. In the LES simulations, time-averaged values are presented.

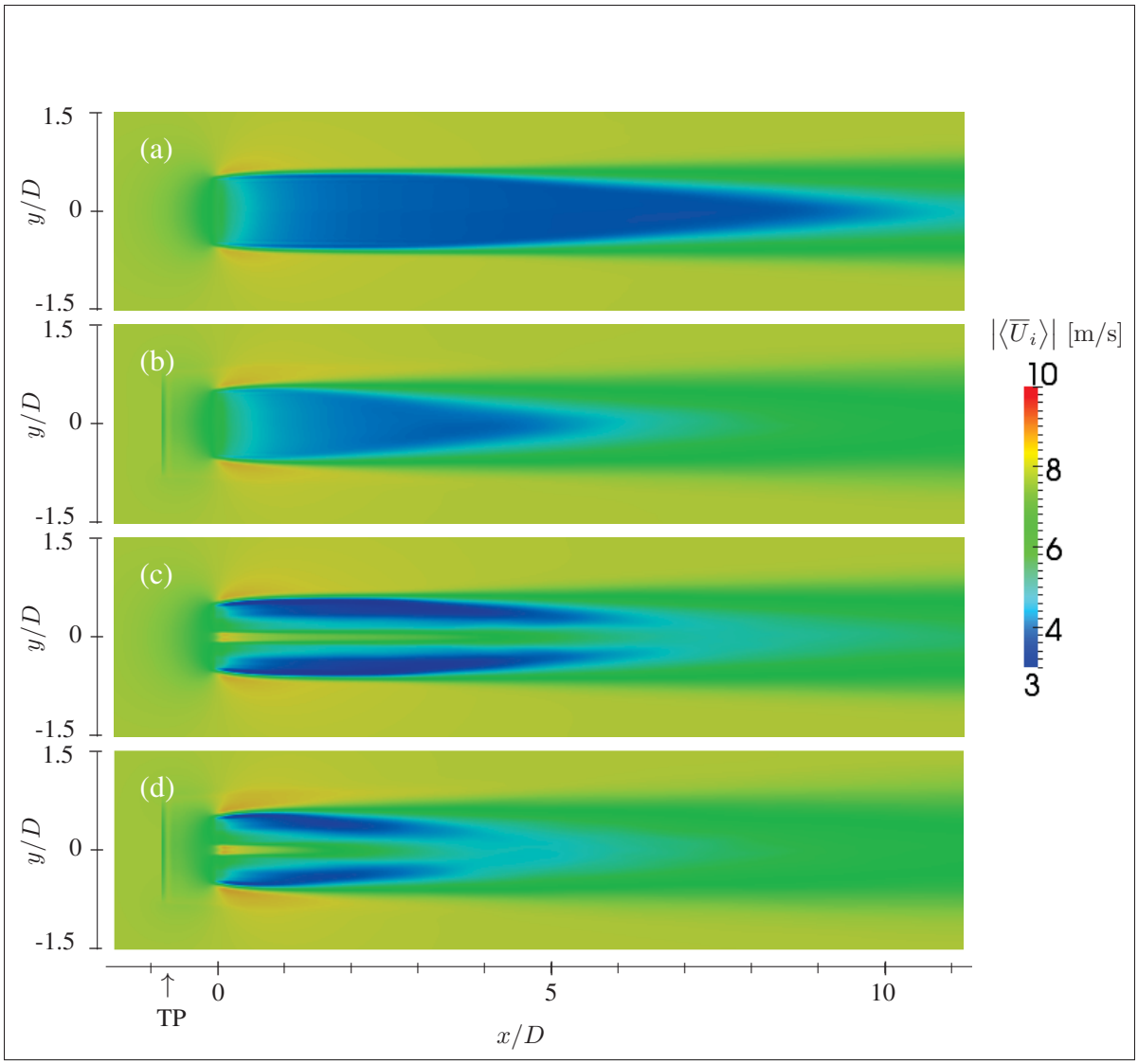

Figure 5.14 Mean velocity magnitude of wakes at the mid-vertical ($x - y$) plane. The images (a) and (b) represent the wake simulation with the uniformly loaded disk, while in (c) and (d) the AD with rotation is used. Turbulence is introduced at the TP in cases (b) and (d) while the non-turbulent inflow is used in cases (a) and (c). The data represent velocity values averaged during 10 LFT.

For these computations, the non-turbulent inflow case produces values very close to those of the designer whereas with the turbulent inflow, the turbine production is found to increase.

In addition to the above observations, we show in Figure 5.17 the variation of the rotational velocity, power coefficient and total power during the simulation (10 LFT). While the values ob-

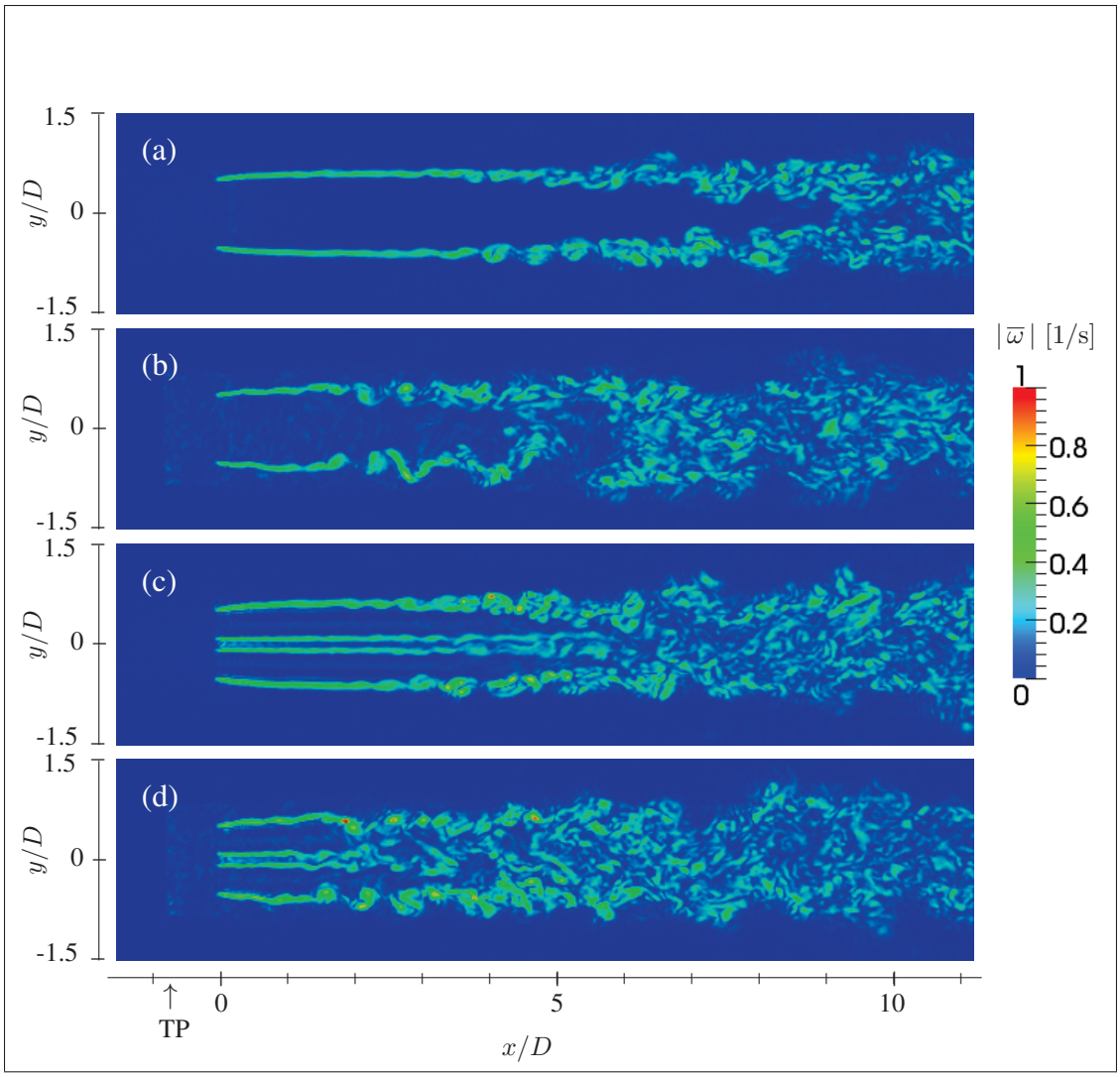

Figure 5.15 Visualization of the turbulence structures in the wakes with vorticity magnitude. The images (a) and (b) represent the wake simulation with the uniformly loaded disk, while in (c) and (d) the AD with rotation is used. Turbulence is introduced at the TP in cases (b) and (d) while the non-turbulent inflow is used in cases (a) and (c). Images are produced at the end of the simulation.

tained with the non-turbulent inflow remain almost unchanged, the quantities oscillate (around the mean shown in Table 5.1) due to the fluctuations in the incoming velocity. Precisely, next to the curve of Ω, the average streamwise velocity taken from two recording positions at the disk location (at the disk centre and at $y = R$, $z = 0$) but without the disk. Even with the velocity extracted from only this two points, it is evident that a correlation exists between the

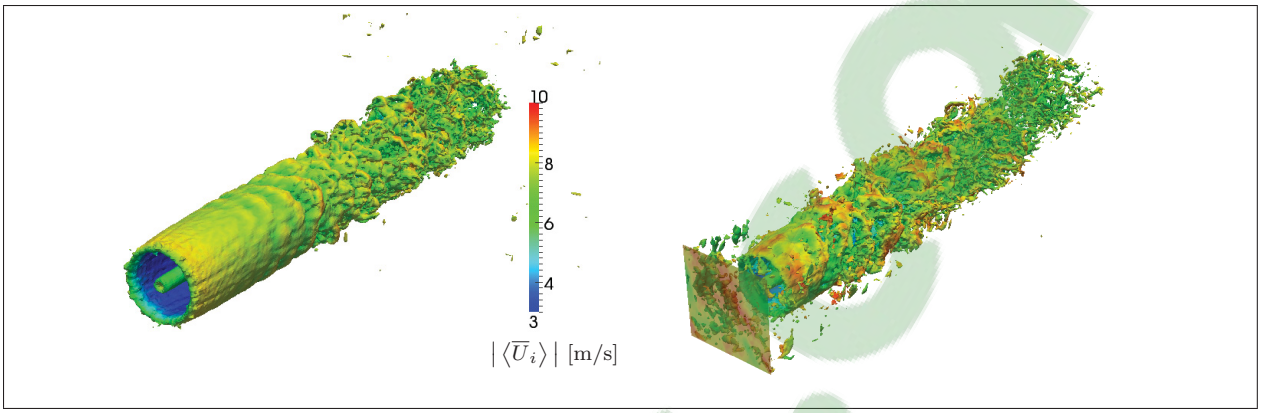

Figure 5.16 3D visualization of wakes in non-turbulent (left) and high turbulence inflow (right) drawn using vorticity contours coloured with the magnitude of the mean velocity. In the latter, the velocity field at the TP is also shown.

Table 5.1 Performance values of the rotor for different simulations.

	Ω [RPM]	C_P	C_T
Reference WT (at peak C_P)	9.16 ^{a}	0.482	$\sim$1
BEM calculation	9.16 ^{a}	0.489	0.863
Steady-state, laminar solver	9.16 ^{a}	0.508	0.808
LES, no inflow turbulence	9.26 ^{b}	0.495	0.800
LES, 4% TI at rotor	9.36 ^{b}	0.512	0.818

$(^a)$ fixed rotational vel.
$(^b)$ determined by controller eq. (2.41)

two curves. Next to the previous performance results, this observation allows us to confirm that the controller regulates the rotational velocity in response to the inflow velocity, as intended. The correlation between the peaks of the incoming velocity and the magnitude of adjustment in rotor velocity is a function of the inertia of the system (drivetrain moment of inertia I_d), considered in the controller design (eq. 2.41). Also, although the variations in C_P are rather large with the incoming turbulence, this can be explained by the observed fluctuation in the total power. In effect, besides agreeing with the expected value, this variation is consistent with the steep change in rotor power with respect to the incoming velocity, as seen in the curves supplied by the designer.

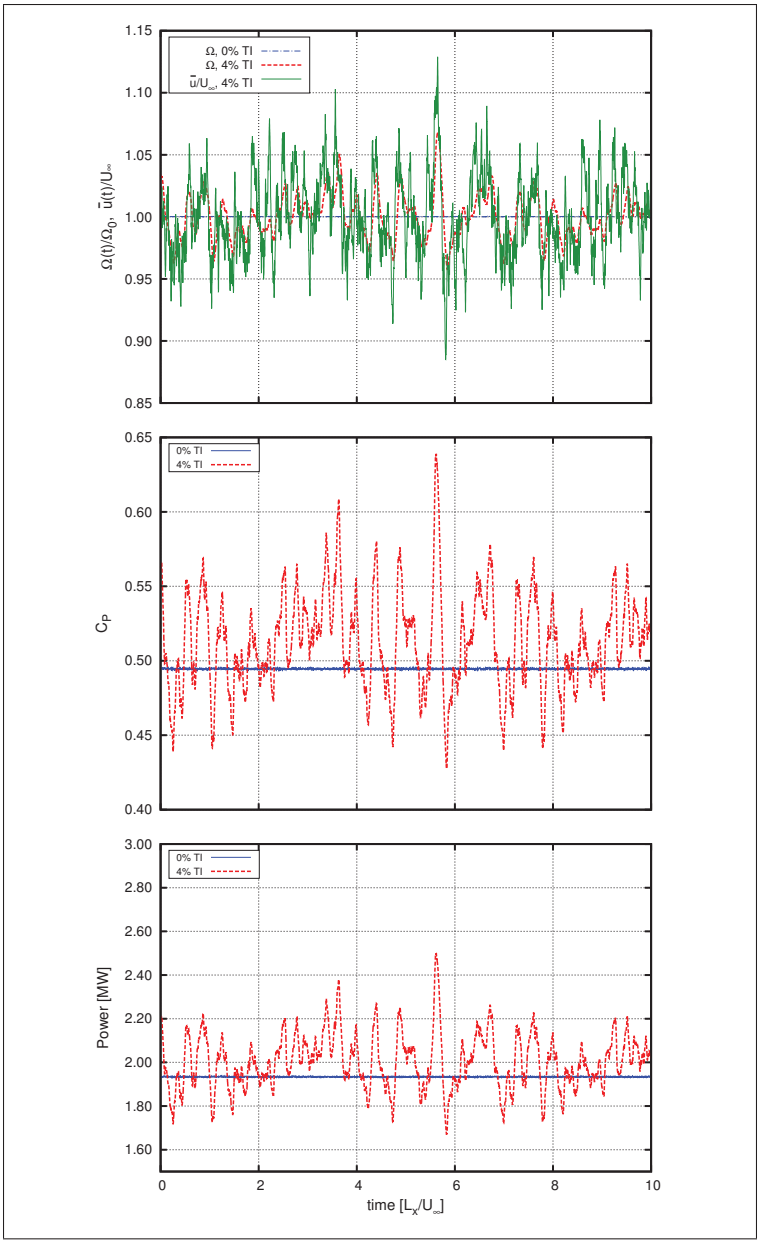

Figure 5.17 Variation of the performance of the rotor for different inflow conditions during the simulation. *Top:* normalized rotational velocity; *middle:* power coefficient and *bottom:* total power. The normalized streamwise velocity recorded at the AD location (without the disk) is added to the top image to highlight the response of the rotor to the inflow conditions.

5.4 Summary and conclusions

The rotor of an horizontal-axis wind turbine is modelled using different techniques with the goal of assessing the differences in the wake characteristics produced by each model. The effect ot the rotor on the incoming wind flow is represented with two Actuator Disk (AD) techniques: 1) the uniformly loaded disk and 2) a disk where the forces are calculated following the blade element theory, where the lift and drag are obtained from tabulated data, called simply *rotating AD*. For the latter model a rotational velocity controller has been also implemented, following the technique presented by Breton *et al.* (2012), with the objective of simulating the "real" conditions of variable speed wind turbines. This device, referred to as the *controller* is designed to work below rated power, in the region where modern wind turbines operate at a constant tip-speed ratio. A first study in laminar, steady-state flow shows a good agreement between the performance obtained from the rotor with respect to the values provided by the designer of the wind turbine that we model.

In our study, we explore the differences between each rotor model under different, non-sheared inflow conditions: a non-turbulent and a turbulent flow. To generate the turbulence, we employ the technique of Mann to create a synthetic turbulent velocity field that possesses the same characteristics of the atmospheric turbulence. The turbulence is introduced in the computational domain just ahead of the AD, inspired by a technique devised by Troldborg (2008). An analysis of the turbulent flow in an empty domain shows that despite the abrupt introduction of fluctuations, turbulence adapts rather quickly to the conditions enforced by the LES. In particular, a very small decay in intensity is observed immediately after the turbulence is imposed and it becomes negligible afterwards. This result, in constrast to the decay observed in Chapter 3, is likely due to the comparatively low TI value of the synthetic turbulence field.

When the turbulent wake is simulated, the computations performed in this work make possible to observe that, in general, differences in the turbulence characteristics are indeed observed near the rotor. This is, the velocity field behind the rotating AD shows the non-uniformity of the thrust distribution (smaller towards the hub, which yields a low velocity deficit behind),

unlike the case of the uniformly loaded disk. As for the turbulence kinetic energy k, both disks cause an increase due to shear behind the disk edges, albeit higher for the rotating model which in addition displays an increase of k behind the hub. These differences are more apparent a few rotor diameters behind the AD, while the values yielded from each model approach to each other when moving further downstream, in the far wake region. When a turbulent inflow is employed, the differences between the predictions of both techniques are largely reduced in the far wake. Unlike the turbulence characteristics, the estimations of the velocity deficit in the wake differ little for the two rotor models. These results comply with what is observed in previous studies. For instance, that the use of a non-uniform load in the AD that also considers the rotational velocity of the rotor leads to different estimations of the turbulence field, particularly in the near wake (Porté-Agel *et al.*, 2011) and that the introduction of a turbulence inflow reduces dramatically the disparities in the turbulence energy predicted by various rotor models (Troldborg *et al.*, 2015).

Lastly, we studied the performance of the rotating AD. By comparing the values obtained for the rotational velocity, C_P as well as for the produced power with the quantities provided by the designer, we show that our implementation represents fairly well the modelled wind turbine. Moreover, the applied velocity control method is shown to respond and adjust to the local inflow conditions by regulating the rotational speed.

CONCLUSION

This work has been dedicated to the modelling and study of turbulence in wakes produced by rotor models using homogeneous inflow conditions. To make this possible, a methodology has been developed and implemented in OpenFOAM that permits to reproduce the main turbulence features of the wake velocity field. In this methodology (inspired by the techniques used by Troldborg, 2008) a synthetic turbulence field generated with an implementation of the Mann algorithm is introduced in a computational domain to simulate different inflow conditions in a flow field computed with Large-Eddy Simulations (LES). The Actuator Disk (AD) technique is used to represent the effect of a wind turbine rotor in the surrounding flow that permits to simulate the ensuing wake and the turbulence field within.

In the first part of this study, the methodology was applied to replicate the turbulence characteristics of wakes arising from the introduction of porous disks in homogeneous flow in a wind tunnel. This part is in turn subdivided in a study of the free decaying turbulence properties (Chapter 3) and the analysis of the turbulence features in the wake (Chapter 4). In this first part, our methodology is validated by comparing our results with quantities computed from the wind tunnel measurements. This comparison is complemented with results obtained from simulations carried out with EllipSys3D, a platform widely used and tested for computations of wind turbine wakes.

It has been shown that the computations of the homogeneous decaying turbulence performed with the presented methodology adequately reproduced the evolution of the streamwise turbulence intensity values (TI) and the longitudinal integral lengthscales (L_1) of the experiments, particularly at the location where the porous disks are later introduced. This fact is further reinforced by comparing with the analytical expressions found in the literature. Unlike most of the wind turbine wake computations performed in wind energy research, the values of L_1 were much smaller than the size of the modelled rotor, which imposed a considerable demand regarding the mesh resolution. Despite this restriction, it has been possible to replicate the evolution of the most significant turbulence structures. The limitations on the representation of these structures at the given mesh resolution have been explored while analyzing the reproduc-

tion of the micro-scales and the dissipative scales. Then, an examination of the LES modelling in the computations permitted to assess if the simulations could be considered sufficiently well-resolved. Also, an investigation of the turbulence development, from the point where the synthetic field enters the domain, shows that the velocity field adapts to the conditions imposed by the LES solver, maintaining the distinctive features of turbulence seen in nature, such as the turbulence kinetic energy (k) distribution in the power spectra. From these observations it is concluded that the methodology and the employed numerical setup were adequate to provide an inflow with the looked for turbulence characteristics for the posterior wake simulations.

The methodology was later applied for the simulations of turbulence in wakes by introducing an AD to replicate the effect of the porous disks used in the experiments. Here, the results obtained with OpenFOAM showed again a good agreement with the wind tunnel measurements for the velocity deficit, turbulence kinetic energy and its dissipation. A satisfactory comparison with the results of EllipSys3D was also obtained, although in one case the setup in this code had to be modified due to differences observed in the turbulence decay. Small differences in the turbulence level in the wake yielded by each code were therefore seen as a direct consequence of the variations in the local value of TI. The OpenFOAM results indicate that L_1 increases in the shear layer created by the AD but only in the case of low TI. With a higher TI, the turbulence lengthscales of the inflow predominate throughout the wake. A study of the LES modelling showed that the ratios of the resolved and Sub-Grid Scale (SGS) parts are largely conserved along the wake with respect to the computations without the disks. Although an increase of the SGS contribution could be observed within the shear layer immediately behind the disk, the increase of the turbulence level of the inflow decreased this effect. This feature exhibits that the modelling of the freestream flow prevails over that of the turbulent flow arising in the wake as the TI level raises. In addition, an investigation of the power spectra showed that shear indeed increases the turbulence energy in the wake, but this was only evident for the low TI inflow. In such case, the level of energy far downstream in the wake was as energetic as the one computed just ahead of the AD, displaying a clear inertial range that was absent in the free flow. Conversely, the TI increase in the inflow turbulence makes the added shear turbulence to

become negligible, so the longitudinal turbulence decay remains largely as in the free flow. It can be inferred that, if the level of turbulence in the inflow is sufficiently high (as in the high TI cases shown here), the characteristics of the inflow turbulence prevail over those arising in the wake.

The second part of this work is presented in Chapter 5. This consists of a comparison of wake modelling results yielded by two different rotor models: a uniformly loaded AD (as described above) and a disk where the thrust and torque are computed following the blade element theory, including a tangential velocity component and where lift and drag are obtained from tabulated data. In the latter model, identified as the *AD with rotation*, a rotational velocity controller has been implemented to reproduce the behaviour of variable speed wind turbines below rated power.

As a first method of validation, a steady-state flow simulation was carried out to observe the performance of the AD with rotation, showing a good comparison with the values provided by the designer of the modelled rotor. Later, a turbulence inflow was produced in a similar way to the procedure shown in the preceding Chapters but unlike the results of Chapter 3, the TI decay observed was very small, likely due to the comparatively low TI values of the synthetic field. The comparison of the wake field generated by each AD model shows differences both in the velocity deficit and in k behind the disks. However, these differences become smaller further downstream, specially for the velocity profiles. When contrasting results obtained with and without a turbulent inflow, the differences in the wake simulations of each rotor model are reduced, confirming the assumption that the *far wake* can be represented with disk models of little sophistication, such as the uniformly loaded AD. In addition, it was seen that the performance simulated by the controller system responded to fluctuations in the incoming velocity. This was observed through variations of the rotational speed and the produced power, which varied around the values predicted by the designer and in accordance to the inflow velocity.

The examination of our work presented above permits to answer, in a general scope, the questions formulated at the introduction of this thesis. More importantly, the main objective set at

the beginning of this work has been reached, after having implemented a i) method of turbulence generation to reproduce an homogeneous turbulence field, ii) an AD model and assess the reproduction of turbulence in the ensuing wake and iii) having evaluated the changes in the turbulence field in the wake of an AD model when rotation and non-uniform load distribution are included. These three elements comprised the specific objectives of this work.

Future work

The validation process and the results obtained show that the presented methodology is adequate to model wind turbine wakes with an emphasis in reproducing the *far wake* turbulence field within. It is important to note that this was accomplished in a context of limited mesh resolution, which is relevant in the wind energy field where the significant wind and wake characteristics should be reproduced while minimizing the computational requirements.

Wind energy research provides the background of this work but its limits are set in a much smaller framework. Wind turbine rotors have been simulated in an isolated setting, which indeed replicates a laboratory setup but is far from the clusters of turbines found in a wind park. Thence, this work can be considered as a first step in the path of performing studies that seek to reproduce conditions of real world operations. However, from the perspective of the study of wake turbulence, it is desirable to simplify the conditions of the problem and investigate first the turbulence arising only from the rotor model in homogeneous turbulence, separately from the turbulence effects that emerge from the interaction with the Atmospheric Boundary Layer (ABL) such as those due to topography variations, vegetation interaction, atmospheric stability, etc. Therefore, in order to study the wakes occurring in wind parks, the methodology exposed in this work should be taken a step further to model the flow of the ABL. A possible path to achieve this is to follow the method presented by Mikkelsen *et al.* (2007), where a synthetic ABL (also produced with the Mann algorithm as shown in Sec. 2.5.2) is introduced in a domain where the wind vertical profile is maintained with the introduction of source terms in the momentum equation. This model is known as the Forced Boundary Layer (FBL) and has the advantage of avoiding the modelling of the flow interaction with the walls (Troldborg,

2008; Nilsson, 2015). However, the complexities of non-uniform ground roughness as well as the stability effects are difficult to include when the FBL method is employed. Therefore, a flow simulation where the ground is included in the model with either wall functions or a forest drag model (Boudreault, 2015; Nebenführ, 2015) that also comprises atmospheric stability seems more adequate, although it is computationally more expensive. In this scenario, the rotating AD model can certainly be used to represent the wind turbine rotor. This technique has been proven capable of representing the far wake turbulence, which is fundamental as this is the region interacting with other turbines in a park, in addition to provide an estimation of the generated power.

As for the modelling assessment, a future investigation could be made to address the impact of numerical dissipation in the simulation of decaying turbulence. In this regard, it should be investigated if the incompressibility of the synthetic turbulence is related to a substantial increase in the numerical dissipation and the consequent loss of turbulence energy. Also, it should be determined if a blend with a bounded interpolation scheme for the advective term (e.g. QUICK) is indeed necessary, or alternatively, to be kept to a minimum. Hence, a simulation where the use of a linear scheme is maximized could, in principle, yield a TI decay that is closer to that caused by viscous dissipation. For the same reason, the impact of different SGS models in the LES simulation should be considered.

APPENDIX I

EFFECTS OF MESH RESOLUTION IN THE REPRODUCTION OF TURBULENCE CHARACTERISTICS

In the Chapter 3 it was shown that despite the limited grid resolution of the longitudinal integral lengthscale (this is, L_1/Δ), the desired values of L_1 were obtained. Furthermore, the development of L_1, TI, k and other values was consistent with the wind tunnel measurements as well as with empirical equations that described observations from previous experiments. In this Appendix, the effect on the turbulence development of different combinations of resolution (based on L_1/Δ) between the synthetic field and the LES computational domain are studied. The aim is simply to compare main characteristics in a few examples, with a focus in the L_1 development, so a detailed analysis is not presented.

The investigation is divided into two parts. First, turbulence lengthscales are highly resolved in the synthetic field and LES simulations are performed with varying resolutions of the computational domain. Later, synthetic fields are produced with different resolutions and employed in LES computations where the resolution of the computational domain is maintained. The homogeneous turbulence fields have been produced with the Mann algorithm described in Chapter 2 and the simulations have been performed with OpenFOAM, following the procedure seen in Chapter 3.

1. Varying the resolution of computational domain

To investigate the effect that different resolutions of the computational domain have in the development of turbulence in the LES, a synthetic field with a high resolution of L_1 is employed. This field is produced with the following parameters:

- Synthetic field:

  $\mathrm{L}_{\mathrm{B},x} \times \mathrm{L}_{\mathrm{B},y} \times \mathrm{L}_{\mathrm{B},z} = 4\,\mathrm{m} \times 0.125\,\mathrm{m} \times 0.125\,\mathrm{m}$

  $\mathrm{N}_{\mathrm{B},x} \times \mathrm{N}_{\mathrm{B},y} \times \mathrm{N}_{\mathrm{B},z} = 4096 \times 128 \times 128$ cells

$\Delta = 9.76 \times 10^{-4}$ m, $L_{1,\mathrm{B}} = 0.01$ m $\Rightarrow$ **10.24 cells per** $L_{1,\mathrm{B}}$

$\mathrm{TI} = 5.5\%$

Three computational domains of size $\mathrm{L}_x \times \mathrm{L}_y \times \mathrm{L}_z = 0.5\,\mathrm{m} \times 0.125\,\mathrm{m} \times 0.125\,\mathrm{m}$ with uniformly distributed cells have been used, this are referred to as a) coarse, b) baseline and c) fine:

a. Coarse

   $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_z = 128 \times 32 \times 32$ cells

   $\Delta = 0.0039$ m, **2.56 cells per** $L_{1,\mathrm{B}}$

   Mesh resolution in computational domain is 4 times coarser than in the Mann box

b. Baseline

   $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_z = 256 \times 64 \times 64$ cells

   $\Delta = 0.00195$ m, **5.12 cells per** $L_{1,\mathrm{B}}$ **m**

   Mesh resolution in computational domain is 2 times coarser than in the Mann box

c. Fine

   $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_z = 512 \times 128 \times 128$ cells

   $\Delta = 9.76 \times 10^{-4}$ m, **10.24 cells per** $L_{1,\mathrm{B}}$ **m**

   Mesh resolution in computational domain is equal to that of the Mann box

In all cases, results are presented for simulations lasting 20 longitudinal flow-times (4 s), after an initial run of 4 flow-times to allow the stabilization of the solution ($U_0 = 2.5\,\mathrm{m/s}$). A comparison of the results of the longitudinal evolution of TI, L_1, k and ε obtained with each of the above mesh resolutions is shown. This is done employing a normalized distance scale equal to the one of Chapters 3 and 4 (x/D with $D = 0.1$ m).

Figure I-1 shows that near the inlet, TI raises with mesh resolution. In the fine case, this value is even higher than that of the synthetic field ($\mathrm{TI} = 5.5\%$). The TI decay is shown to be stronger

for higher mesh resolutions, so the values attained towards the end are essentially identical. Figure I-2 shows an increase in the estimation of L_1 in the coarse case, although the relative increment yielded by each mesh is about the same. Figure I-3 shows k_{res} is appreciably larger for the denser grids, also revealing that the SGS components increase their values (not only their ratio to k_{tot}, but also in absolute terms) for coarser resolutions. The analogous effect is more noticeable for ε in Figure I-4. There, in the coarse case, the SGS component remains larger than the resolved part for most of the domain. As the mesh resolution increases, the ε_{res} becomes larger, with the opposite effect on ε_{SGS}.

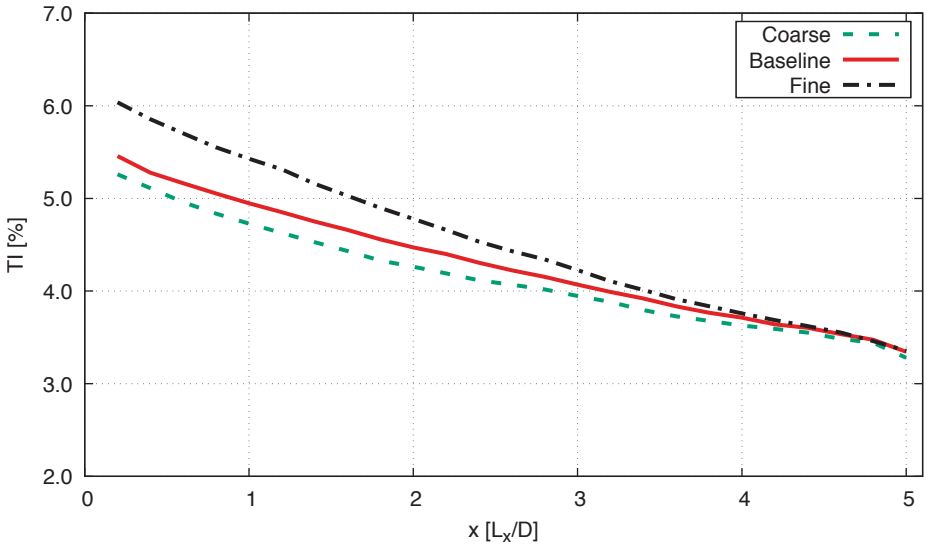

Figure-A I-1 Turbulence intensity decay

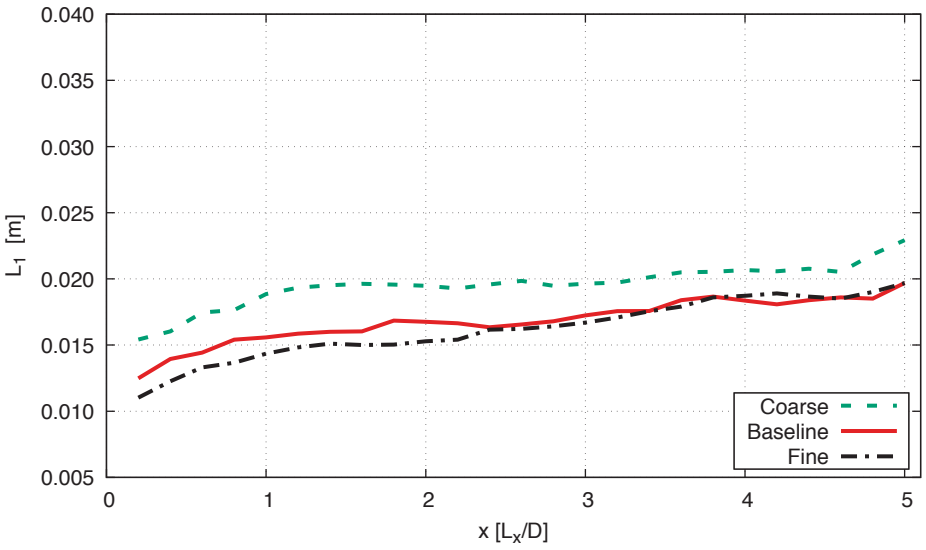

Figure-A I-2 Longitudinal development of the integral lengthscale

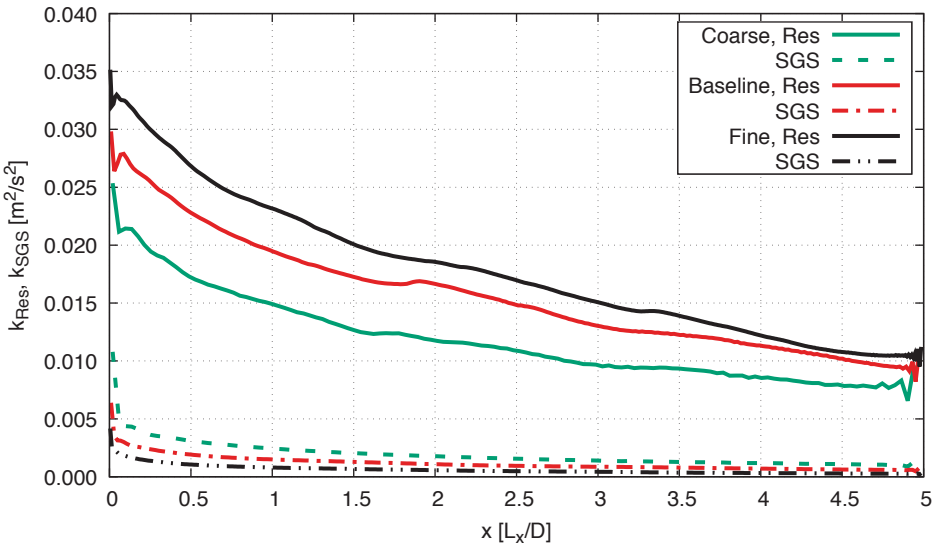

Figure-A I-3 Longitudinal development of the turbulent kinetic energy components

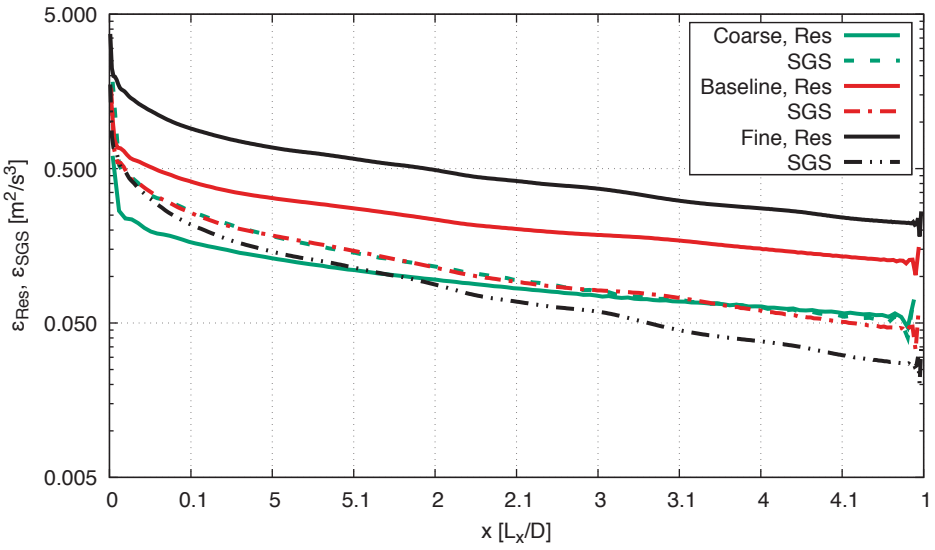

Figure-A I-4 Longitudinal development of the dissipation components

2. Varying the resolution of synthetic field

Now, the effect of the opposite change is compared. This is, synthetic fields of homogeneous turbulence are produced with varying mesh resolutions in order to observe the consequences in the reproduction of turbulence in domains that do not change grid resolution. The setup is slightly different compared to the previous comparison, since the synthetic fields are generated with a small difference in TI but also in domains with different longitudinal sizes, which leads to simulations with a small change in simulation time.

Simulations are performed in a computational domain with dimensions $\mathrm{L}_x \times \mathrm{L}_y \times \mathrm{L}_z = 2\,\mathrm{m} \times 0.5\,\mathrm{m} \times 0.5\,\mathrm{m}$ with a central uniform region of $20\,\mathrm{m} \times 0.36\,\mathrm{m} \times 0.36\,\mathrm{m}$ (computational domain as described in Chapter 3). In all cases, the simulations are let run for 4 longitudinal flow-times before data are registered and averages are calculated. Only 2 configurations are compared:

- Case 1
  - Synthetic field:
    $\mathrm{L}_{\mathrm{B},x} \times \mathrm{L}_{\mathrm{B},y} \times \mathrm{L}_{\mathrm{B},z} = 16 \times 0.5 \times 0.5$ m
    $\mathrm{N}_{\mathrm{B},x} \times \mathrm{N}_{\mathrm{B},y} \times \mathrm{N}_{\mathrm{B},z} = 4098 \times 128 \times 128$ cells
    $\Delta = 0.00391$ m, $L_{1,\mathrm{B}} = 0.01$ m $\Rightarrow$ **2.56 cells per** $L_{1,\mathrm{B}}$
    $\mathrm{TI} = 5.5\%$
  - Computational domain:
    $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_z = 500 \times 104 \times 104$ cells
    in uniform region $\Delta = 0.004$ m, **2.5 cells per** $L_{1,\mathrm{B}}$ m
    Total simulation time: 24 longitudinal flow-times (16 s), synthetic field is recycled 3 times
    Mesh resolution of synthetic field almost the same as in computational domain
- Case 2
  - Synthetic field:
    $\mathrm{L}_{\mathrm{B},x} \times \mathrm{L}_{\mathrm{B},y} \times \mathrm{L}_{\mathrm{B},z} = 4 \times 0.5 \times 0.5$ m
    $\mathrm{N}_{\mathrm{B},x} \times \mathrm{N}_{\mathrm{B},y} \times \mathrm{N}_{\mathrm{B},z} = 2048 \times 256 \times 256$ cells
    $\Delta = 0.00195$ m, $L_{1,\mathrm{B}} = 0.01$ m $\Rightarrow$ **5.12 cells per** $L_{1,\mathrm{B}}$
    $\mathrm{TI} = 4.8\%$
  - Computational domain:
    $\mathrm{N}_x \times \mathrm{N}_y \times \mathrm{N}_z = 500 \times 104 \times 104$ cells
    in uniform region $\Delta = 0.004$ m, **2.5 cells per** $L_{1,\mathrm{B}}$ m
    Total simulation time: 20 longitudinal flow-times (13.334 s), synthetic field is recycled 6 times
    Mesh resolution of synthetic field twice as fine as in computational domain

In Figure I-5 we observe a slightly larger TI next to the inlet for the higher synthetic field resolution. But it is unclear if this is due to the resolution effects of the higher TI of the

synthetic field, and after 5D the values are practically the same. The effect in the estimation of L_1 in Figure I-6 is less apparent and the values yielded using both synthetic field resolutions are very close throughout the domain.

In the case of k in Figure I-7, the higher values obtained with the coarser synthetic field could be also due to the larger TI of the generated turbulence. However, this could also be caused by the *filtering* of small fluctuations created in the synthetic field with a finer mesh that cannot be resolved by the grid in the LES, which results in a loss of k. This argument would have to be investigated in future research. Notably, the k_{SGS} also decreases for higher synthetic turbulence resolutions, unlike the previous comparison when the computational domain resolution is varied. As with TI, values from both Case 1 and 2 seem to match after 5D. The SGS component of dissipation seem to be larger next to the inlet, for both cases, to later decrease to values below the resolved component, as seen in Figure I-8. However, it should be noted that ε_{tot} is larger for the case with coarser synthetic field for most of the domain. Here, we argue an analogous reasoning to that employed to explain the observations made for the development of k.

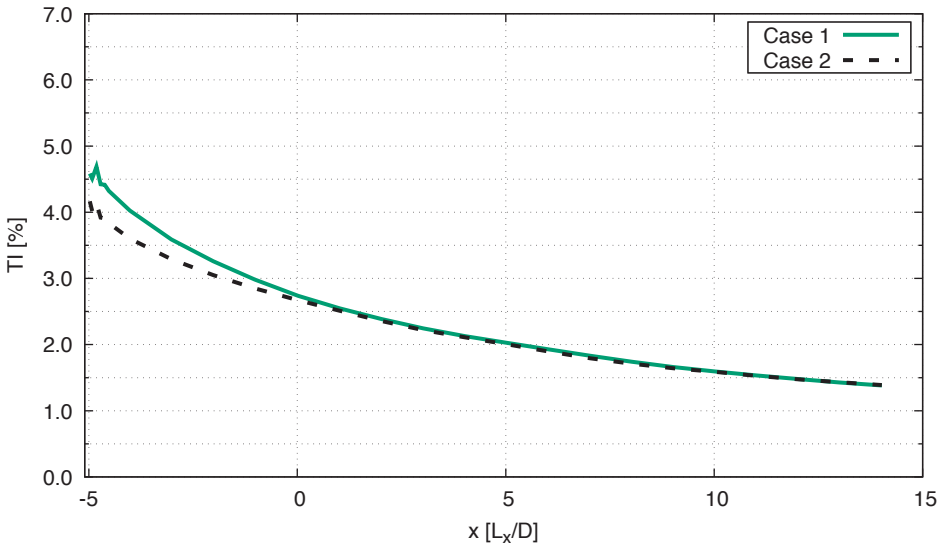

Figure-A I-5 Turbulence intensity decay

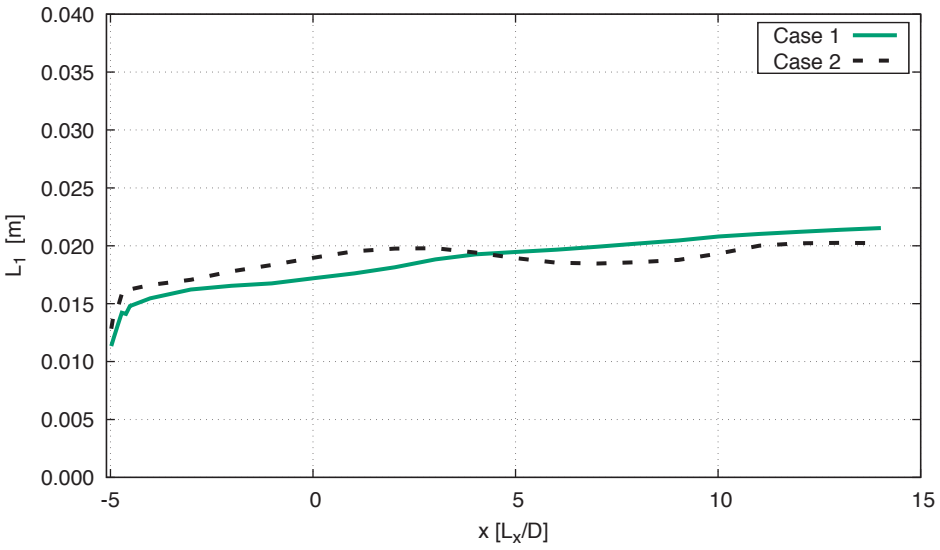

Figure-A I-6 Longitudinal development of the integral lengthscale.

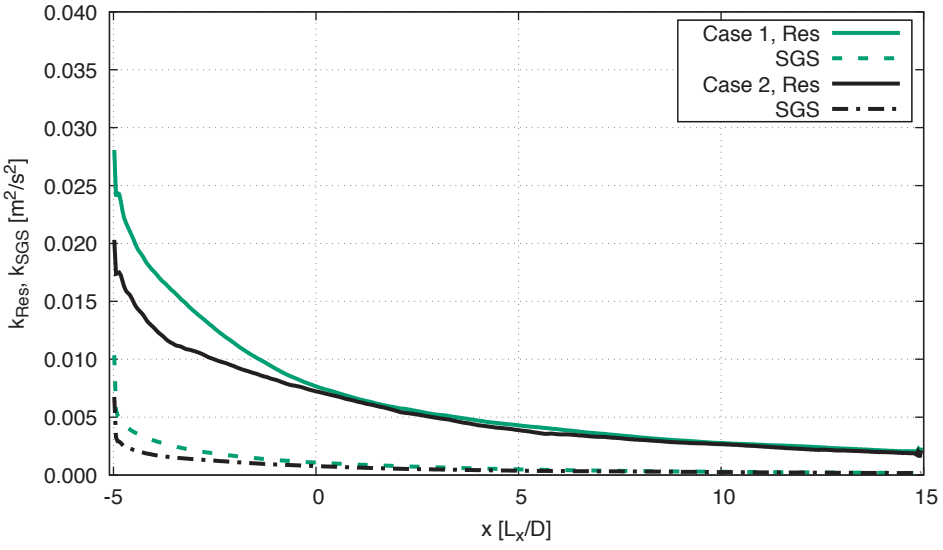

Figure-A I-7 Longitudinal development of the turbulent kinetic energy components

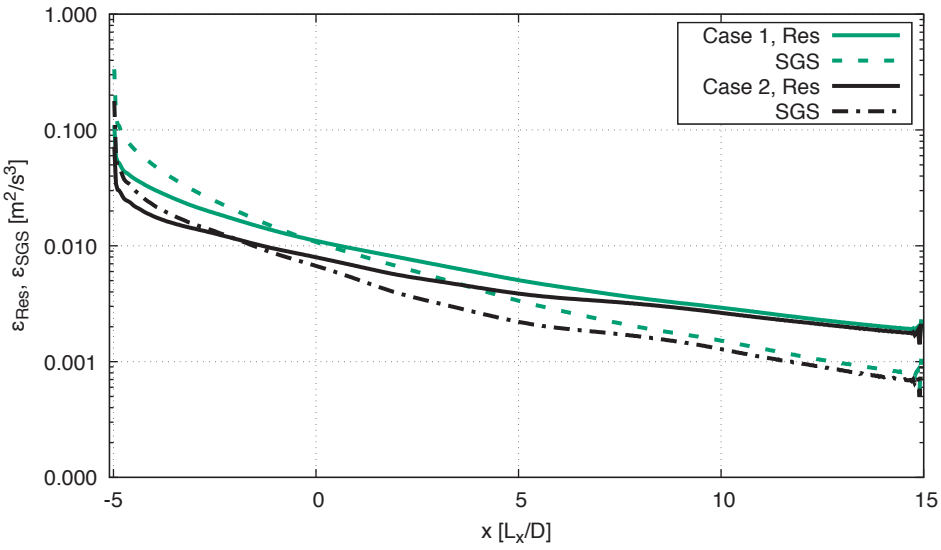

Figure-A I-8 Longitudinal development of the dissipation components

APPENDIX II

OPENFOAM DICTIONARIES

We present a copy of the two dictionaries containing the numerical methods applied in the computation of the flow solution in OpenFOAM. In general, *fvSchemes* controls the methods for interpolation of quantities between cell centres and faces while *fvSolution* defines the algorithm employed for the solution of the discretized flow equations as well as the techniques and parameters used for the solution of the matrices and equations involved in that process.

The methods and parameters used in Chapters 4 and 3 vary slightly from those used in Chapter 5. For that reason, we present the dictionaries used in each of these two occurrences.

1. Dictionaries used in Chapters 4 and 3

1.1 fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.5                                   |
|   \\  /    A nd           | Web:      http://www.OpenFOAM.org               |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
 default           backward;
}

gradSchemes
```

```
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         fourth;
}

divSchemes
{
    default          none;
    div(phi,U)       Gauss filteredLinear;
    div(phi,k)       Gauss linear;
    div(phi,epsilon) Gauss linear;
    div(phi,R)       Gauss linear;
    div(R)           Gauss linear;
    div(phi,nuTilda) Gauss linear;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default                         none;
    laplacian(nuEff,U)              Gauss linear corrected;
    laplacian((1|A(U)),p)           Gauss linear corrected;
    laplacian(DkEff,k)              Gauss linear corrected;
    laplacian(DepsilonEff,epsilon) Gauss linear corrected;
    laplacian(DREff,R)              Gauss linear corrected;
    laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
    interpolate(U)  linear;
}

snGradSchemes
{
    default         corrected;
}

fluxRequired
{
    default         no;
    p;
```

```
}

// ************************************************************************* //
```

1.2 fvSolution

```
/*---------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.5                                   |
|   \\  /    A nd           | Web:      http://www.OpenFOAM.org               |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver           GAMG;
        tolerance        1e-07;
        relTol           0.01;
        smoother         DICGaussSeidel;
        nPreSweeps       0;
        nPostSweeps      2;
        cacheAgglomeration true;
        agglomerator     faceAreaPair;
        nCellsInCoarsestLevel 20;
        mergeLevels      1;
    };
    pFinal
    {
        solver           GAMG;
        tolerance        1e-06;
        relTol           0.01;
        smoother         DICGaussSeidel;
        nPreSweeps       0;
```

```
        nPostSweeps      2;
        cacheAgglomeration true;
        agglomerator     faceAreaPair;
        nCellsInCoarsestLevel 20;
        mergeLevels      1;
    };
    U
    {
        solver           smoothSolver;
        smoother         DILUGaussSeidel;
        tolerance        1e-06;
        relTol           0;
        nSweeps          1;
    };
    k PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0.1;
    };
    epsilon PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0.1;
    };
    R PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0.1;
    };
    nuTilda PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0.1;
    };
}

PISO
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
```

```
    pRefCell        1001;
    pRefValue       0;
    convergence     1e-05;
}

relaxationFactors
{
}


// ************************************************************************* //
```

2. Dictionaries used in Chapter 5

2.1 fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.5                                   |
|   \\  /    A nd           | Web:      http://www.OpenFOAM.org               |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
 default           backward;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}
```

```
divSchemes
{
    default          none;
    div(phi,U)       Gauss QUICK;
    div(phi,k)       Gauss QUICK;
    div(phi,epsilon) Gauss QUICK;
    div(phi,R)       Gauss QUICK;
    div(R)           Gauss QUICK;
    div(phi,nuTilda) Gauss QUICK;
    div((nuEff*dev(T(grad(U))))) Gauss QUICK phi;
}

laplacianSchemes
{
    default                      none;
    laplacian(nuEff,U)            Gauss linear corrected;
    laplacian((1|A(U)),p)         Gauss linear corrected;
    laplacian(DkEff,k)            Gauss linear corrected;
    laplacian(DepsilonEff,epsilon) Gauss linear corrected;
    laplacian(DREff,R)            Gauss linear corrected;
    laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
    interpolate(U)  linear;
}

snGradSchemes
{
    default         corrected;
}

fluxRequired
{
    default         no;
    p;
}

// ************************************************************************* //
```

2.2 fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.5                                   |
|   \\  /    A nd           | Web:      http://www.OpenFOAM.org               |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver           GAMG;
        tolerance        1e-07;
        relTol           0.01;
        smoother         DICGaussSeidel;
        nPreSweeps       0;
        nPostSweeps      2;
        cacheAgglomeration true;
        agglomerator     faceAreaPair;
        nCellsInCoarsestLevel 20;
        mergeLevels      1;
    };
    pFinal
    {
        solver           GAMG;
        tolerance        1e-06;
        relTol           0.01;
        smoother         DICGaussSeidel;
        nPreSweeps       0;
        nPostSweeps      2;
        cacheAgglomeration true;
        agglomerator     faceAreaPair;
        nCellsInCoarsestLevel 20;
```

```
        mergeLevels      1;
    };
    U
    {
        solver           smoothSolver;
        smoother         DILUGaussSeidel;
        tolerance        1e-06;
        relTol           0;
        nSweeps          1;
    };
    k PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0.1;
    };
    epsilon PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0.1;
    };
    R PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0.1;
    };
    nuTilda PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0.1;
    };
}

PISO
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
    pRefCell        1001;
    pRefValue       0;
    convergence     1e-05;
}
```

```
relaxationFactors
{
}


// ************************************************************************* //
```

BIBLIOGRAPHY

Ammara, Idriss. 1998. "Modélisation aérodynamique tridimensionnelle d'un parc d'éoliennes à axe horizontal". Master's thesis, École polytechnique de Montréal, Montreal, Canada. (In french).

Ammara, Idriss, Christophe Leclerc, and Christian Masson. 2002. "A viscous three-dimensional differential/actuator-disk method for the aerodynamic analysis of wind farms". *Journal of Solar Energy Engineering*, vol. 124, n° 4, p. 345–356.

Aubrun, Sandrine, Philippe Devinant, and Guillaume Espana. 2007. "Physical modelling of the far wake from wind turbines. Application to wind turbine interactions". In *Proceedings of the European Wind Energy Conference, Milan, Italy*. p. 7–10.

Aubrun, Sandrine, Stéphane Loyer, PE Hancock, and Paul Hayden. 2013. "Wind turbine wake properties: Comparison between a non-rotating simplified wind turbine model and a rotating model". *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 120, p. 1–8.

Bailly, Christophe and Geneviève Comte-Bellot, 2003. *Turbulence*. Paris, France : CNRS éditions. (In french).

Bardina, Jorge, Joel H Ferziger, and WC Reynolds. 1980. "Improved subgrid-scale models for large-eddy simulation". In *American Institute of Aeronautics and Astronautics, Fluid and Plasma Dynamics Conference, 13th, Snowmass, Colo., July 14-16, 1980, 10 p.*

Bautista, Mary. 2015. "Turbulence modelling of the atmospheric boundary layer over complex topography". PhD thesis, École de technologie supérieure, Université du Québec, Montreal, Canada.

Bechmann, Andreas. November 2006. "Large-Eddy Simulation of Atmospheric Flow over Complex". PhD thesis, Risø Technical University of Denmark, Roskilde, Denmark.

Betz, Albert, 1926. *Wind-energie und ihre ausnutzung durch windmühlen*. Göttingen, Germany : Vandenhoeck. (In german).

Boudreault, Louis-Étienne. 2015. "Reynolds-averaged Navier-Stokes and Large-Eddy Simulation Over and Inside Inhomogeneous Forests". PhD thesis, Technical University of Denmark, Department of Mechanical Engineering.

Boussinesq, Joseph, 1897. *Théorie de l'écoulement tourbillonnant et tumultueux des liquides dans les lits rectilignes a grande section*. Paris, France : Gauthier-Villars et fils. (In french).

Breton, S-P, K Nilsson, H Olivares-Espinosa, C Masson, L Dufresne, and Stefan Ivanell. 2014. "Study of the influence of imposed turbulence on the asymptotic wake deficit in a very long line of wind turbines". *Renewable energy*, vol. 70, p. 153–163.

Breton, Simon-Philippe, Karl Nilsson, Stefan Ivanell, Hugo Olivares-Espinosa, Christian Masson, and Louis Dufresne. 2012. "Study of the effect of the presence of downstream turbines on upstream ones and use of a controller in CFD wind turbine simulation models". In *The Science of Making Torque from Wind*.

Calaf, Marc, Charles Meneveau, and Johan Meyers. 2010. "Large eddy simulation study of fully developed wind-turbine array boundary layers". *Physics of Fluids*, vol. 22, n° 1, p. 015110.

Celik, IB, ZN Cehreli, and I Yavuz. 2005. "Index of resolution quality for large eddy simulations". *Journal of Fluids Engineering*, vol. 127, n° 5, p. 949–958.

Chamorro, Leonardo P and Fernando Porté-Agel. 2009. "A wind-tunnel investigation of wind-turbine wakes: boundary-layer turbulence effects". *Boundary-layer meteorology*, vol. 132, n° 1, p. 129–149.

Chougule, A, Jakob Mann, Antonio Segalini, and Ebba Dellwik. 2015. "Spectral tensor parameters for wind turbine load modeling from forested and agricultural landscapes". *Wind Energy*, vol. 18, n° 3, p. 469–481.

Chow, Fotini Katopodes, Robert L Street, Ming Xue, and Joel H Ferziger. 2005. "Explicit filtering and reconstruction turbulence modeling for large-eddy simulation of neutral boundary layer flow". *Journal of the Atmospheric Sciences*, vol. 62, n° 7, p. 2058–2077.

Churchfield, Matthew J, Sang Lee, Patrick J Moriarty, Luis A Martinez, Stefano Leonardi, Ganesh Vijayakumar, and James G Brasseur. 2012. "A large-eddy simulation of wind-plant aerodynamics". *AIAA paper*, vol. 537.

Churchfield, MJ, PJ Moriarty, G Vijayakumar, and JG Brasseur. 2010. "Wind energy-related atmospheric boundary layer large-eddy simulation using OpenFOAM". In *19th Symposium on Boundary Layers and Turbulence*. (Keystone, Colorado 2010). National Renewable Energy Laboratory (NREL).

Comte-Bellot, Geneviève and Stanley Corrsin. 1966. "The use of a contraction to improve the isotropy of grid-generated turbulence". *Journal of Fluid Mechanics*, vol. 25, n° 04, p. 657–682.

Comte-Bellot, Geneviève and Stanley Corrsin. 1971. "Simple Eulerian time correlation of full-and narrow-band velocity signals in grid-generated,'isotropic'turbulence". *Journal of Fluid Mechanics*, vol. 48, n° 02, p. 273–337.

Conway, John T. 1995. "Analytical solutions for the actuator disk with variable radial distribution of load". *Journal of Fluid Mechanics*, vol. 297, p. 327–355.

Conway, John T. 1998. "Exact actuator disk solutions for non-uniform heavy loading and slipstream contraction". *Journal of Fluid Mechanics*, vol. 365, p. 235–267.

Crespo, Antonio, F Manuel, D Moreno, E Fraga, and J Hernandez. 1985. "Numerical analysis of wind turbine wakes". In *Workshop on Wind Energy Applications, Delphi, Greece*.

Crespo, Antonio, J Hernandez, and Sten Frandsen. 1999. "Survey of modelling methods for wind turbine wakes and wind farms". *Wind energy*, vol. 2, n° 1, p. 1–24.

Davidson, Lars. 2009. "Large eddy simulations: how to evaluate resolution". *International Journal of Heat and Fluid Flow*, vol. 30, n° 5, p. 1016–1025.

Davidson, Lars and Simon Dahlström. 2005. "Hybrid LES–RANS: computation of the flow around a three-dimensional hill". *Engineering Turbulence Modeling and Experiments*, vol. 6, p. 319–328.

El Kasmi, Amina and Christian Masson. 2008. "An extended k–ε model for turbulent flow through horizontal-axis wind turbines". *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, n° 1, p. 103–122.

Eriksson, Ola, Robert Mikkelsen, Kurt Schaldemose Hansen, Karl Nilsson, and Stefan Ivanell. 2014. "Analysis of long distance wakes of Horns Rev I using actuator disc approach". In *Journal of Physics: Conference Series*. p. 012032. IOP Publishing.

Espana, Guillaume. December 2009. "Étude expérimentale du sillage lointain des éoliennes à axe horizontal au moyen d'une modélisation simplifiée en couche limite atmosphérique". PhD thesis, Université d'Orléans. (In french).

Espana, Guillaume, Sandrine Aubrun, Stéphane Loyer, and Philippe Devinant. 2011. "Spatial study of the wake meandering using modelled wind turbines in a wind tunnel". *Wind Energy*, vol. 14, n° 7, p. 923–937.

Espana, Guillaume, Sandrine Aubrun, Stéphane Loyer, and Philippe Devinant. 2012. "Wind tunnel study of the wake meandering downstream of a modelled wind turbine as an effect of large scale turbulent eddies". *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 101, p. 24–33.

Ferziger, Joel H and Milovan Perić, 2002. *Computational methods for fluid dynamics*. ed. 3rd. Berlin : Springer, 423 p.

Fletcher, Clive, 1991. *Computational techniques for fluid dynamics 1*, volume 1. Berlin, Germany : Springer Science & Business Media.

Freitag, Martin and Markus Klein. 2006. "An improved method to assess the quality of large eddy simulations in the context of implicit filtering". *Journal of Turbulence*, vol. 7, p. N40.

Germano, Massimo. 1986. "A proposal for a redefinition of the turbulent stresses in the filtered Navier–Stokes equations". *Physics of Fluids*, vol. 29, n° 7, p. 2323–2324.

Geurts, Bernard J and Jochen Fröhlich. 2002. "A framework for predicting accuracy limitations in large-eddy simulation". *Physics of Fluids*, vol. 14, n° 6, p. L41–L44.

Gilling, Lasse. 2009. *TuGen: Synthetic turbulence generator, manual and user's guide*. Technical Report DCE-76.

Gilling, Lasse and Niels N Sørensen. 2011. "Imposing resolved turbulence in CFD simulations". *Wind Energy*, vol. 14, n° 5, p. 661–676.

Global Wind Energy Council. 2015. "Global Wind Report, Annual Market Update". <http://www.gwec.net>.

Goodfriend, Lauren, Fotini K Chow, Marcos Vanella, and Elias Balaras. 2013. "Large-eddy simulation of decaying isotropic turbulence across a grid refinement interface using explicit filtering and reconstruction". *Journal of Turbulence*, vol. 14, n° 12, p. 58–76.

Hansen, Martin OL, 2003. *Aerodynamics of wind turbines*. London, United Kingdom : James & James.

IEC. 2005. *Wind turbines–part 1: Design requirements*. Technical Report 61400-1. Geneva : International Electrotechnical Commission.

Issa, Raad I. 1986. "Solution of the implicitly discretised fluid flow equations by operator-splitting". *Journal of computational physics*, vol. 62, n° 1, p. 40–65.

Ivanell, Stefan, Robert Mikkelsen, Jens N. Sørensen, and Dan Henningson. 2010. "Stability analysis of the tip vortices of a wind turbine". *Wind Energy*, vol. 13, n° 8, p. 705–715.

Ivanell, Stefan, Thomas Leweke, Sasan Sarmast, Hugo U Quaranta, Robert F Mikkelsen, and Jens N Sørensen. 2015. "Comparison between experiments and Large-Eddy Simulations of tip spiral structure and geometry". In *Journal of Physics: Conference Series*. p. 012018. IOP Publishing.

Ivanell, Stefan A. 2009. "Numerical computations of wind turbine wakes". PhD thesis, Royal Institute of Technology, Stockholm, Sweden.

Jiménez, Ángel, Antonio Crespo, Emilio Migoya, and Javier García. 2007. "Advances in large-eddy simulation of a wind turbine wake". In *Journal of Physics: Conference Series*. p. 012041. IOP Publishing.

Jiménez, Ángel, Antonio Crespo, Emilio Migoya, and Javier García. 2008. "Large-eddy simulation of spectral coherence in a wind turbine wake". *Environmental Research Letters*, vol. 3, n° 1, p. 015004.

Jiménez, Javier (Ed.). 1997. *A selection of test cases for the validation of large-eddy simulations of turbulent flows*. Technical Report AGARD Advisory Report No.345.

Johansson, Peter BV and William K George. 2006. "The far downstream evolution of the high-Reynolds-number axisymmetric wake behind a disk. Part 1. Single-point statistics". *Journal of Fluid Mechanics*, vol. 555, p. 363–385.

Johansson, Peter BV, William K George, and Michael J Gourlay. 2003. "Equilibrium similarity, effects of initial conditions and local Reynolds number on the axisymmetric wake". *Physics of Fluids (1994-present)*, vol. 15, n° 3, p. 603–617.

Jonkman, Jason Mark, Sandy Butterfield, Walter Musial, and George Scott. 2009. *Definition of a 5-mw reference wind turbine for offshore system development*. Technical Report NREL/TP-500-38060.

Kaimal, Jagadish Chandran and John J Finnigan, 1994. *Atmospheric boundary layer flows: their structure and measurement*. Oxford : Oxford University Press.

Kang, Hyung Suk, Stuart Chester, and Charles Meneveau. 2003. "Decaying turbulence in an active-grid-generated flow and comparisons with large-eddy simulation". *Journal of Fluid Mechanics*, vol. 480, p. 129–160.

Kärrholm, Fabian Peng. 2008. "Numerical Modelling of Diesel Spray Injection, Turbulence Interaction and Combustion". PhD thesis, Chalmers University of Technology.

Keck, Rolf-Erik, Robert Mikkelsen, Niels Troldborg, Martin de Maré, and Kurt S Hansen. 2014. "Synthetic atmospheric turbulence and wind shear in large eddy simulations of wind turbine wakes". *Wind Energy*, vol. 17, n° 8, p. 1247–1267.

Kelley, Niel D. 1992. *Full vector (3-D) inflow simulation in natural and wind farm environments using an expanded version of the SNLWIND (Veers) turbulence code*. Technical Report NREL/TP-442-5225.

Kim, Hyun-Goo and VC Patel. 2000. "Test of turbulence models for wind flow over terrain with separation and recirculation". *Boundary-Layer Meteorology*, vol. 94, n° 1, p. 5–21.

Kim, Theodore, Nils Thürey, Doug James, and Markus Gross. 2008. "Wavelet turbulence for fluid simulation". In *ACM Transactions on Graphics (TOG)*. p. 50. ACM.

Klein, Markus. 2005. "An attempt to assess the quality of large eddy simulations in the context of implicit filtering". *Flow, Turbulence and Combustion*, vol. 75, n° 1-4, p. 131–147.

Klein, Markus, Amsini Sadiki, and Johannes Janicka. 2003. "A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations". *Journal of computational Physics*, vol. 186, n° 2, p. 652–665.

Kolmogorov, Andrey Nikolaevich. 1991. "The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers". *Proceedings: Mathematical and Physical Sciences (First published in Russian in Dokl. Akad. Nauk SSSR, 1941)*, vol. 434, n° 1890, p. 9–13.

Koning, C. 1963. Influence of the propeller on other parts of the airplane structure. Durand, W. F., editor, *Aerodynamic Theory: A General Review of Progress Vol. IV*. Dover Publications, New York, USA.

Larsen, Torben Juul. January 2013. *Turbulence for the IEA annex 30 OC4 project*. Technical Report I-3206.

Lee, Sangsan, Sanjiva K Lele, and Parviz Moin. 1992. "Simulation of spatially evolving turbulence and the applicability of Taylor's hypothesis in compressible flow". *Physics of Fluids A: Fluid Dynamics (1989-1993)*, vol. 4, n° 7, p. 1521–1530.

Leonard, A. 1974. "Energy cascade in large-eddy simulation of turbulent fluid flows". *Advances in Geophysics A*, vol. 18, p. 237-248.

Leonard, Brian P. 1979. "A stable and accurate convective modelling procedure based on quadratic upstream interpolation". *Computer methods in applied mechanics and engineering*, vol. 19, n° 1, p. 59–98.

Lund, Thomas S, Xiaohua Wu, and Kyle D Squires. 1998. "Generation of turbulent inflow data for spatially-developing boundary layer simulations". *Journal of Computational Physics*, vol. 140, n° 2, p. 233–258.

Machefaux, Ewan, Gunner C Larsen, Niels Troldborg, and Andreas Rettenmeier. 2013. "Single Wake Meandering, Advection and Expansion-An analysis using an adapted Pulsed Lidar and CFD LES-ACL simulations". In *European Wind Energy Conference & Exhibition 2013*.

Madsen, H Aagaard. 1997. "A CFD analysis of the actuator disc flow compared with momentum theory results". In *IEA Joint Action. Aerodynamics of Wind Turbines*. Technical University of Denmark. Department of Fluid Mechanics.

Madsen, Helge A. 1988. "Application of actuator surface theory on wind turbines". In *Proc. 2nd IEA Symp. on Aerodynamics of Wind Turbines*.

Madsen, Helge Aagaard. 1982. "The actuator cylinder: A flow model for vertical axis wind turbines". PhD thesis, Institute of Industrial Constructions and Energy Technology, Aalborg University Centre, Aalborg, Denmark.

Mann, Jakob. 1994. "The spatial structure of neutral atmospheric surface-layer turbulence". *Journal of Fluid Mechanics*, vol. 273, p. 141–168.

Mann, Jakob. 1998. "Wind field simulation". *Probabilistic engineering mechanics*, vol. 13, n° 4, p. 269–282.

Manwell, James F, Jon G McGowan, and Anthony L Rogers, 2002. *Wind energy explained: theory, design and application*. Chichester, United Kingdom : John Wiley & Sons.

Mason, Paul J. and David J. Thomson. 9 1992. "Stochastic backscatter in large-eddy simulations of boundary layers". *Journal of Fluid Mechanics*, vol. 242, p. 51–78.

Masson, Christian, Arezki Smaïli, and Christophe Leclerc. 2001. "Aerodynamic analysis of HAWTs operating in unsteady conditions". *Wind Energy*, vol. 4, n° 1, p. 1–22.

Medici, Davide. 2005. "Experimental Studies of Wind Turbine Wakes". PhD thesis, Royal Institute of Technology, Stockholm, Sweden.

Michelsen, JA. 1992. *Basis3D - a platform for development of multiblock PDE solvers*. Technical Report AFM 92-05. 92–05 p.

Michelsen, Jess A. 1994. *Block structured multigrid solution of 2D and 3D elliptic PDE's*. Technical Report AFM 94-06.

Mikkelsen, Robert Flemming, Jens Nørkær Sørensen, and Niels Troldborg. 2007. "Prescribed wind shear modelling with the actuator line technique". In *EWEC Conference, Milan*.

Mohamed, Mohsen S and John C LaRue. 1990. "The decay power law in grid-generated turbulence". *Journal of Fluid Mechanics*, vol. 219, p. 195–214.

Muller, Yann-Aël. 2014. "Étude du méandrement du sillage éolien lointain dans différentes conditions de rugosité". PhD thesis, Université d'Orléans. (In french).

Mydlarski, Laurent and Zellman Warhaft. 1996. "On the onset of high-Reynolds-number grid-generated wind tunnel turbulence". *Journal of Fluid Mechanics*, vol. 320, p. 331–368.

Mydlarski, Laurent and Zellman Warhaft. 1998. "Passive scalar statistics in high-Péclet-number grid turbulence". *Journal of Fluid Mechanics*, vol. 358, p. 135–175.

Nebenführ, Bastian. 2015. "Turbulence-resolving simulations for engineering applications". PhD thesis, Chalmers University of Technology, Department of Applied Mechanics.

Nilsen, Katrine M, Bo Kong, Rodney O Fox, James C Hill, and Michael G Olsen. 2014. "Effect of inlet conditions on the accuracy of large eddy simulations of a turbulent rectangular wake". *Chemical Engineering Journal*, vol. 250, p. 175–189.

Nilsson, Karl. 2015. "Numerical computations of wind turbine wakes and wake interaction". PhD thesis, Royal Institute of Technology, Stockholm, Sweden.

Nilsson, Karl, Stefan Ivanell, Kurt S Hansen, Robert Mikkelsen, Jens N Sørensen, Simon-Philippe Breton, and Dan Henningson. 2015. "Large-eddy simulations of the Lillgrund wind farm". *Wind Energy*, vol. 18, n° 3, p. 449–467.

Olivares-Espinosa, Hugo, Simon-Philippe Breton, Christian Masson, and Louis Dufresne. 2014. "Turbulence characteristics in a free wake of an actuator disk: comparisons between a rotating and a non-rotating actuator disk in uniform inflow". In *Journal of Physics: Conference Series*. p. 012081. IOP Publishing.

Oppenheim, Alan V, Ronald W Schafer, John R Buck, et al., 1999. *Discrete-time signal processing*. NJ, USA : Prentice hall.

Palma, José .M.L.M., Fernando A. Castro, Luis F. Ribeiro, Álvaro H. Rodrigues, and Paulo Pinto. 2008. "Linear and nonlinear models in wind resource assessment and wind turbine micro-siting in complex terrain". *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, n° 12.

Peña, Alfredo, Charlotte Bay Hasager, Julia Lange, Jan Anger, Merete Badger, Ferhat Bingöl, Oliver Bischoff, Jean-Pierre Cariou, Fiona Dunne, Stefan Emeis, Michael Harris, Martin Hofsäss, Ioanna Karagali, Jason Laks, Søren Ejling Larsen, Jakob Mann, Torben Mikkelsen, Lucy Y. Pao, Mark Pitter, Andreas Rettenmeier, Ameya Sathe, Fabio Scanzani, David Schlipf, Eric Simley, Chris Slinger, Rozenn Wagner, and Ines Würth. 2013. *Remote sensing for wind energy*. Technical report. Roskilde, Denmark : DTU Wind Energy.

Peña, Alfredo, Sven-Erik Gryning, Jakob Mann, and Charlotte B Hasager. 2010. "Length scales of the neutral wind profile over homogeneous terrain". *Journal of Applied Meteorology and Climatology*, vol. 49, n° 4, p. 792–806.

Perrone, Francesco. 2015. "Turbulent wind field by means of the mann model". MATLAB Central File Exchange. <http://www.mathworks.com/matlabcentral/fileexchange/35722-3d-turbulent-wind-field-by-means-of-the-mann-model>. Last visited Nov 24th, 2015.

Pope, Stephen B, 2000. *Turbulent flows*. Cambridge, United Kingdom : Cambridge Univ press.

Pope, Stephen B. 2004. "Ten questions concerning the large-eddy simulation of turbulent flows". *New journal of Physics*, vol. 6, n° 1, p. 35.

Porté-Agel, Fernando, Charles Meneveau, and Marc B Parlange. 2000. "A scale-dependent dynamic model for large-eddy simulation: application to a neutral atmospheric boundary layer". *Journal of Fluid Mechanics*, vol. 415, p. 261–284.

Porté-Agel, Fernando, Yu-Ting Wu, Hao Lu, and Robert J Conzemius. 2011. "Large-eddy simulation of atmospheric boundary layer flow through wind turbines and wind farms". *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 99, n° 4, p. 154–168.

Rajagopalan, R Ganesh and Jerome B Fanucci. 1985. "Finite difference model for vertical axis wind turbines". *Journal of Propulsion and Power*, vol. 1, n° 6, p. 432–436.

Réthoré, Pierre-Elouan Mikael. 2009. "Wind turbine wake in atmospheric turbulence". PhD thesis, Technical University of Denmark, Risø National Laboratory for Sustainable EnergyRisø Nationallaboratoriet for Bæredygtig Energi.

Réthoré, Pierre-Elouan Mikael and Niels N Sørensen. 2008. "Actuator disc model using a modified Rhie-Chow/SIMPLE pressure correction algorithm. Comparison with analytical solutions". In *European Wind Energy Conference and Exhibition*.

Réthoré, Pierre-Elouan Mikael, Paul Laan, Niels Troldborg, Frederik Zahle, and Niels N Sørensen. 2014. "Verification and validation of an actuator disc model". *Wind Energy*, vol. 17, n° 6, p. 919–937.

Rhie, CM and WL Chow. 1983. "Numerical study of the turbulent flow past an airfoil with trailing edge separation". *AIAA journal*, vol. 21, n° 11, p. 1525–1532.

Sagaut, Pierre, 2006. *Large Eddy Simulations for Incompressible Flows*. ed. 3rd. Berlin, Germany : Springer Berlin.

Sarlak, H, F Pierella, R Mikkelsen, et al. 2014. "Comparison of two LES codes for wind turbine wake studies". In *Journal of Physics: Conference Series*. p. 012145. IOP Publishing.

Sarlak, H, C Meneveau, Jens Nørkær Sørensen, and Robert Mikkelsen. 2015a. Quantifying the impact of subgrid scale models in actuator-line based les of wind turbine wakes in laminar and turbulent inflow. Fröhlich, J., Hans Kuerten, Bernard J Geurts, and Vincenzo Armenio, editors, *Direct and Large-Eddy Simulation IX*, p. 169–175. Springer.

Sarlak, Hamid, C Meneveau, and Jens Nørkær Sørensen. 2015b. "Role of subgrid-scale modeling in large eddy simulation of wind turbine wake interactions". *Renewable Energy*, vol. 77, p. 386–399.

Sarmast, Sasan. 2014. "Numerical study on instability and interaction of wind turbine wakes". PhD thesis, Royal Institute of Technology, Stockholm, Sweden.

Sarmast, Sasan, Reza Dadfar, Robert F. Mikkelsen, Philipp Schlatter, Stefan Ivanell, Jens N. Sørensen, and Dan S. Henningson. 2014. "Mutual inductance instability of the tip vortices behind a wind turbine". *Journal of Fluid Mechanics*, vol. 755, p. 705–731.

Schlichting, Herrmann and Klaus Gersten, 2003. *Boundary-layer theory*. Berlin,Germany : Springer Science & Business Media.

Sibuet Watters, Christophe and Christian Masson. 2010. "Modeling of lifting-device aerodynamics using the actuator surface concept". *International journal for numerical methods in fluids*, vol. 62, n° 11, p. 1264–1298.

Smagorinsky, Joseph. 1963. "General circulation experiments with the primitive equations: I. the basic experiment*". *Monthly weather review*, vol. 91, n° 3, p. 99–164.

Sørensen, Jens N and Carsten Weber Kock. 1995. "A model for unsteady rotor aerodynamics". *Journal of wind engineering and industrial aerodynamics*, vol. 58, n° 3, p. 259–275.

Sørensen, Jens N and Asger Myken. 1992. "Unsteady actuator disc model for horizontal axis wind turbines". *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 39, n° 1, p. 139–149.

Sørensen, Jens N and Wen Zhong Shen. 2002. "Numerical modeling of wind turbine wakes". *Journal of fluids engineering*, vol. 124, n° 2, p. 393–399.

Sørensen, Jens N, Robert F Mikkelsen, Dan S Henningson, Stefan Ivanell, Sasan Sarmast, and Søren J Andersen. 2015. "Simulation of wind turbine wakes using the actuator line technique". *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 373, n° 2035, p. 20140071.

Sørensen, Niels N. 1995. "General purpose flow solver applied to flow over hills". PhD thesis, Risø Technical University of Denmark, Roskilde, Denmark.

Sørensen, Niels N and JA Michelsen. 2000. "Aerodynamic predictions for the unsteady aerodynamics experiment phase-II rotor at the National Renewable Energy Laboratory". *AIAA Paper*, vol. 37, p. 2000.

Spalart, Philippe R. 2001. *Young-person's guide to detached-eddy simulation grids*. Technical Report CR-2001-211032. Hampton, Va. : NASA Langley Reseach Center.

Spalart, PR, WH Jou, and M Strelets. 1997. "Comments on the Feasibility of LES for Wings, and on a Hybrid RANS/LES Approach". In *Advances in DNS/LES, First AFOSR Int. Conf. in DNS/LES*. (Columbus, OH, USA 1997). Greyden Press.

Sumner, Jonathon, Guillaume Espana, Christian Masson, and Sandrine Aubrun. 2013. "Evaluation of RANS/actuator disk modelling of wind turbine wake flow using wind tunnel measurements". *International Journal of Engineering Systems Modelling and Simulation*, vol. 5, n° 1, p. 147–158.

Tabor, Gavin R and Mohammad Baba-Ahmadi. 2010. "Inlet conditions for large eddy simulation: a review". *Computers & Fluids*, vol. 39, n° 4, p. 553–567.

Thacker, Adrien, Stéphane Loyer, and Sandrine Aubrun. 2010. "Comparison of turbulence length scales assessed with three measurement systems in increasingly complex turbulent flows". *Experimental Thermal and Fluid Science*, vol. 34, n° 5, p. 638–645.

The OpenFOAM Foundation. March 2016. "OpenFOAM: The open source CFD toolbox. User guide". <http://www.openfoam.org>.

Troldborg, Niels. June 2008. "Actuator line modeling of wind turbine wakes". PhD thesis, Technical University of Denmark.

Troldborg, Niels, Jens N Sørensen, and Robert Mikkelsen. 2010. "Numerical simulations of wake characteristics of a wind turbine in uniform inflow". *Wind Energy*, vol. 13, n° 1, p. 86–99.

Troldborg, Niels, Jens N. Sørensen, Robert Mikkelsen, and Niels N. Sørensen. 2014. "A simple atmospheric boundary layer model applied to large eddy simulations of wind turbine wakes". *Wind Energy*, vol. 17, n° 4, p. 657–669.

Troldborg, Niels, Frederik Zahle, Pierre-Elouan Réthoré, and Niels N Sørensen. 2015. "Comparison of wind turbine wake properties in non-sheared inflow predicted by different computational fluid dynamics rotor models". *Wind Energy*, vol. 18, n° 7, p. 1239–1250.

Veers, Paul S. 1988. *Three-dimensional wind simulation*. Technical report. Albuquerque, NM, USA : Sandia National Labs.

Veloudis, Ioannis, Zhijia Yang, Jim J McGuirk, Gary J Page, and Adrian Spencer. 2007. "Novel implementation and assessment of a digital filter based approach for the generation of LES inlet conditions". *Flow, turbulence and combustion*, vol. 79, n° 1, p. 1–24.

Vermeer, LJ, Jens N Sørensen, and Antonio Crespo. 2003. "Wind turbine wake aerodynamics". *Progress in aerospace sciences*, vol. 39, n° 6, p. 467–510.

Versteeg, Henk K and Weeratunge Malalasekera , 2007. *An introduction to computational fluid dynamics. The finite volume method*. ed. 2nd. Harlow : Pearson Education Limited, 503 p.

von Kármán, Theodore. 1948. "Progress in the statistical theory of turbulence". *Proceedings of the National Academy of Sciences of the United States of America*, vol. 34, n° 11, p. 530.

Wachtor, Adam J, Fernando F Grinstein, Carl R DeVore, Raymond Ristorcelli, and Len G Margolin. 2013. "Implicit large-eddy simulation of passive scalar mixing in statistically stationary isotropic turbulence". *Physics of Fluids*, vol. 25, n° 2, p. 025101.

Weller, Henry G, Gavin Tabor, Hrvoje Jasak, and Christer Fureby. 1998. "A tensorial approach to computational continuum mechanics using object-oriented techniques". *Computers in physics*, vol. 12, n° 6, p. 620–631.

Wilcox, David C, 1994. *Turbulence modeling for CFD*. La Cañada, CA, USA : DCW industries.

World Wind Energy Association. 2015. "World Wind Energy Report 2014". <http://www.wwindea.org>.

World Wind Energy Association. 2016. "World Wind Energy Website, preliminary figures for 2015". <http://www.wwindea.org>.

Zahle, Frederik and Niels N Sørensen. 2007. "On the Influence of Far-Wake Resolution on Wind Turbine Flow Simulations". *Journal of Physics: Conference Series*, vol. 75, n° 1, p. 012042.

Zahle, Frederik, Jeppe Johansen, and Niels Sørensen. 2007. "Wind turbine aerodynamics using an incompressible overset grid method". In *Proceedings of the European Wind Energy Conference and Exhibition, Ewec 2007*. The European Wind Energy Association.