TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATIONS

ABL	Atmospheric Boundary Layer
AEP	Annual Energy Production
CDS	Central Difference Scheme
CF	Capacity Factor
CFD	Computational Fluid Dynamics
FE	Finite Element Method
FVM	Finite Volume Method
GIS	Geographic Information Systems
LAI	Leaf Area Index
LES	Large Eddy Simulation
OpenFOAM	Open Source Field Operation and Manipulation Software
QUICK	Quadratic Upwind Interpolation
RANS	Reynolds-Averaged Navier-Stokes Simulation
SIMPLER	Semi-Implicit Method for Pressure-Linked Equations Revised
TI	Turbulence Intensity
TKE	Turbulent Kinetic Energy
UDS	Upwind Interpolation

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

Co-ordinates

x, y, z	Cartesian

Upper-case Roman symbols

A	Cross-sectional area of the face of the control volume, m^2
A_{eff}	Effective area of the disk, m^2
C_D	Drag coefficient of the forest
$C_{\varepsilon 1}$	Constant of the $k-\varepsilon$ canopy model
$C_{\varepsilon 2}$	Constant of the $k-\varepsilon$ canopy model
$C_{\varepsilon 4}$	Constant of the $k-\varepsilon$ canopy model
$C_{\varepsilon 5}$	Constant of the $k-\varepsilon$ canopy model
C_μ	Constant of the $k-\varepsilon$ forest model
C_P	Wind turbine power coefficient
E_{farm}	Total energy yield of a wind farm, Wh
E_i	Total energy yield of a wind turbine, Wh
G_k	Production of turbulent kinetic energy, m^2/s^3
$G_{k,P}$	Production of turbulent kinetic energy in first cell attached to the ground, m^2/s^3
L	Domain diameter, m
$\overline{P}_{\text{rated}}$	Rated turbine power, W
P_w	Turbine power curve, W

S	Speed-up factor
S_k	Source term of the $k-\varepsilon$ canopy model
S_ε	Source term of the $k-\varepsilon$ canopy model
S_ϕ	Source term of general transport equation
S_u	Source term of forest model
U	Averaged velocity field, m/s
$\overline{U}$	Mean velocity, m/s
U_f	Velocity at the top boundary, m/s
U_n	Velocity of cell-centre-top, m/s
U_{rated}	Wind speed at rated turbine power, m/s
U_{ref}	Wind speed at reference height, m/s
U_{out}	Wind speed at wind turbine shut down, m/s

Lower-case Roman symbols

d	Displacement height based on the mean height of forest, m
f	Representation of convection or diffusion in the general transport equation
h	Tree's height, m
k	Turbulent kinetic energy, m^2/s^2
l	Characteristic length, m
$\mathbf{n}$	Unit outward normal to the surface of the control volume
p	Pressure, Pa

$\overline{s}$	Mean speed-up factor
t	Time, s
u	Instantaneous velocity field, m/s
u'	Fluctuating part of given velocity, m/s
$\overline{u'}$	Temporal averaged velocity, m/s
$\overline{u'_i u'_j}$	Reynolds stresses, m^2/s^2
u_*	Friction velocity, m/s
u_{*g}	Friction velocity at the ground, m/s
y^+	Non-dimensional wall-normal distance, $= u * z/\nu$
z	Altitude, m
z_0	Roughness length, m
$z_{0\text{inlet}}$	Roughness length used at the inlet boundary condition, m
z_f	Location of top boundary, m
z_g	Computational ground height, m
z_n	Cell-centre-height of top cell, m
z_{n-1}	Cell-centre-height of penultimate cell, m
z_P	Height of near-wall cell-centre, m

Greek symbols

α	Leaf area density, m^2/m^3
β_d	Constant of the $k-\varepsilon$ canopy model

β_P	Constant of the $k-\varepsilon$ canopy model
γ	Ratio of rated to mean wind speeds
Γ	Diffusion coefficient of the general transport equation
Γ_d	Dry adiabatic temperature gradient
$\delta_{i3}g$	Gravitational force
δ_{ij}	Kronecker delta
δ	Absolute error
η	Drive train efficiency
ε	Turbulence dissipation rate, m^2/s^3
κ	Von Karman constant
μ	Molecular viscosity, $kg/(m \cdot s)$
ν	Kinematic viscosity, m^2/s
ν_t	Turbulent viscosity, m^2/s
ϕ	General variable
ρ	Air density, kg/m^3
σ_k	Constant of the $k-\varepsilon$ canopy model
σ_ε	Constant of the $k-\varepsilon$ canopy model
τ_{ij}	Viscous stresses, Pa
$\tau_{t,ij}$	Reynolds stress tensor, Pa
τ'_{ij}	Kinematic shear stress, m^2/s^2

$\tau'_{t,ij}$	Kinematic Reynolds stress tensor, m^2/s^2
τ'_w	Kinematic shear stress at the wall, m^2/s^2
Θ	Potential temperature, K
Ω_j	Rotational speed of the earth, rad/s

Subscripts

CV	Control volume
P	Refers to current cell
N	Refers to 'north' neighbour
n	Refers to cell face between P and N
S	Refers to 'south' neighbour
s	Refers to cell face between P and S
E	Refers to 'east' neighbour
e	Refers to cell face between P and E
W	Refers to 'west' neighbour
w	Refers to cell face between P and W
T	Refers to 'top' neighbour
t	Refers to cell face between P and T
B	Refers to 'bottom' neighbour
b	Refers to cell face between P and B

INTRODUCTION

Context

The advancement of societies is based on economic growth which relies heavily on energy production. Nowadays, this demand is still mainly supplied by fossil fuels (as much as 63% of the world energy supply (IEA, 2016a)). This representative amount is leaving a considerable footprint on the environment. Research and development in energy technologies aim to decrease this trace; nevertheless, it is not sufficient since the growth rate of mature less polluting forms of energy does not parallel the growth in the energy demand rate (Nakićenović, 1996). This concern is clearly reflected in national efforts to improve energy systems with attention to climate change. As proof, the Paris Agreement of 2015 was mainly orientated to increase the use of renewable energies with support to lessen the involved costs (IEA, 2016a). As a result of these efforts, it has been (ambitiously) predicted that nearly 60% of power generation may come from renewables by 2040 (IEA, 2016b).

One of the contributors to these targets, onshore wind energy, currently represents the second largest renewable generation source. It provided 2.5% of global electricity demand in 2015 with an installed capacity of approximately 500 GW (IEA, 2016c) and represented more than one third of the worldwide total investment in renewable capacity (GWEC, 2016). In Canada, the current installed capacity of 11.2 GW resulted from an average annual growth of 23% and represents $\sim$5% of the total Canadian electricity demand (CanWEA, 2016). These numbers may not seem significant at macro-scale. But the local contribution of wind power generation favors its ambitious growth, creates diversified jobs, and increases economic competitiveness.

Project scope

Motivated by this framework of support, certain technological challenges have to be faced. According to the European Technology and Innovation Platform on Wind Energy (ETIPWind) one of the main research priorities is to minimize the uncertainty associated with wind energy

yield calculations (ETIPWind, 2016). Energy production forecasts strongly depend on the available wind resource. The more accurately it is predicted, the better decisions related to wind energy development can be made.

There is a wide variety of places to site wind farms, and the emergence of taller and more powerful wind turbines is making the development of heavily forested complex sites increasingly attractive. And while the resource is often abundant, the viability of such sites must nonetheless be demonstrated through the calculation of the annual energy production (AEP) or, equivalently, the wind farm capacity factor (CF). But accurate estimation of these quantities represents a significant modeling challenge for forested, complex sites.

Typically, the energy yield is estimated by integrating the power curve over the wind speed probability density function (pdf) for each turbine. The pdf can be obtained via statistical methods or a discrete version can be derived directly from wind speed observations (Manwell *et al.*, 2009). For the latter approach, a lack of data is often the primary setback (Manwell *et al.*, 2009). In either case, it is rare to have wind speed data at more than a few locations and some form of spatial extrapolation is needed to carry out the energy calculation for each prospective turbine. In order to make the most of the limited data available, a combination of computational fluid dynamics (CFD) models and statistical tools is often the most reliable method to obtain satisfactory predictions (Sumner *et al.*, 2010). Many CFD wind flow models have been proposed with their primary difference being the degree of empiricism in their development. Models based on the Reynolds-Averaged Navier-Stokes (RANS) equations sit somewhere in the middle of this spectrum: their solutions respect mass, momentum, and energy conservation in the mean variables, but model turbulent effects in a somewhat *ad hoc* way.

The primary CFD challenges in wind resource assessment lie in adapting turbulence models to accurately consider topographic and forest effects with economical computational resources.

Objective

The main objective of this master's thesis is *to evaluate the uncertainty in wind flow predictions over moderately complex forested terrain and its impact on capacity factor using RANS models coupled with a modified k-ε turbulence closure in the open-source software OpenFOAM.* The specific objectives of this study are:

- To implement computational source terms in the RANS and k-ε turbulence models to account for the effects of both terrain and forest in the neutral atmospheric boundary layer;
- To quantify its accuracy by comparing the simulated results with experimental data in terms of speed-up factors S;
- To evaluate the impact of RANS wind flow modeling uncertainty on capacity factor uncertainty.

Thesis organization

The present master's thesis is structured in one introduction, four chapters: literature review, methodology, validation of forest modelling, and wind flow modelling over a real site.

The literature review describes the common practices for forecasting the wind energy yield, how wind flow is assessed through CFD modelling, and how the uncertainty in the speed-up factor plays a role in the energy yield calculation. It also defines the speed up factor in the RANS context.

Chapter two describes the methodology, details the physical understanding of the atmospheric boundary layer (ABL), its representation as a mathematical model, and the employed numerical method.

Chapter three presents the reproduction of a fully-developed wind flow within and above a horizontally homogeneous forest from a published case. In addition, further studies were car-

ried out to address certain limitations in obtaining foliage characteristics and in implementing inlet-outlet boundary conditions.

Chapter four considers wind flow modelling over a real site, and describes the CFD simulations of four model cases that were performed for a moderately complex forested terrain. The CFD results are compared with two-years of experimental data from the site to finally estimate the uncertainty in capacity factor due to RANS modelling.

CHAPTER 1

LITERATURE REVIEW

1.1 Wind energy yield forecasting

As wind resource potential is a primary factor in the approval of wind farm projects, it is very important to accurately assess its value. But in contrast to conventional power generation plants, the power generation from wind farms cannot be exactly predicted due to the intermittent nature of the wind resource. Nevertheless, an estimation is needed whether for the purposes of wind farm approvals, long/short term forecasting, increasing the energy yield, or for the maintenance of the equipment itself (Lange & Focken, 2006). The forecasting of energy yield is generally obtained by integrating two terms: the turbine power curve and the wind speed probability density function. The former belongs to the engineering design domain, and the latter has to be accurately assessed but is a challenge in meteorological terms (Ayotte, 2008).

1.1.1 Wind assessment

Atmospheric motions fluctuate in time and space and in order to be assessed a correspondent meteorological scale has to be used. This will allow to situate the scope of study by focusing on the driving physical phenomena that are involved (Stull, 1988). Figure 1.1 shows this relation between scale and physical phenomena. According to the scale, different methods to assess the wind can be utilized. Some of these methods are: direct measurements, measure-correlate-predict, global databases, wind atlases, mesoscale and microscale modelling (Landberg *et al.*, 2003). Since the scope of this master's thesis is centered in microscale modelling, a description of its main characteristics will be detailed.

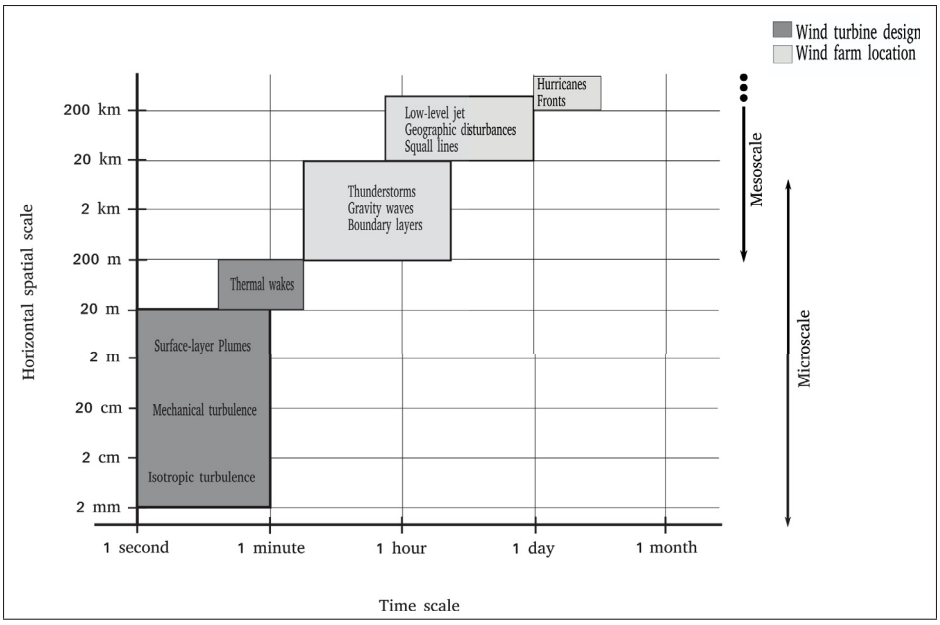


Figure 1.1 Meteorological scale of time and space that shows the physical phenomena that are involved. The figure is in analogy with Stull (1988)

1.1.2 Microscale modelling of atmospheric flows

Nowadays with easy access to meteorological data, in the form of *e.g.* a wind atlas, it may be considered straightforward to plan a wind energy project. But in reality this information alone is not sufficient to plan entire wind farms. Detailed local information as a result of wind measurement campaigns of minimum $\sim$ 1 year (Burton *et al.*, 2001; IEC 61400-1, 2005) are required. These measurements are extracted from anemometers set on masts commonly placed in a few locations. The measurement results from this limited sample then must be spatially extrapolated to produce a wind map of the whole site. For this approach, microscale modelling gives reliable predictions (Sumner *et al.*, 2010).

The microscale modelling of the atmosphere is found within a spatial scale of several hundred meters down to 0.01 m and within a time scale of seconds to minutes. Here the main force involved is the friction generated by the surface. Microscale modelling consists in using computational fluid dynamics (CFD) to solve the partial differential equations that describe

atmospheric motions. CFD discretizes these partial differential equations in order to solve them algebraically, leading to acceptable results. These equations are easily solved when flat terrain cases are analyzed; nevertheless, these ideal cases are rarely present. In reality, the attractiveness of sites located in complex terrains encourages the use of CFD for wind farm purposes. Particularly because in these complex terrain the wind profile may not follow a predicted logarithmic behavior and wind speed accelerations may be present (Panofsky & Ming, 1983). Eventually, these accelerations bring two possibilities: higher power generation and/or possible loads in the turbines with an implicit increase of the maintenance costs (Ayotte, 2008).

Many wind flow models have been proposed for a wide variety of exigent accuracy in turbulence, and they are mainly differentiated in their development and computational costs. Models based on a linearized assumption have been popular in the last few decades in the wind industry; to mention a few: WAsP, MS-Micro, Raptor, Raptor-NL. These models are known to provide reliable results at economical computational cost for flat terrain cases (Petersen *et al.*, 1998); nevertheless, for cases of complex terrain they will weaken the treatment of frictional and thermal effects (Lange & Focken, 2006) by under- and over-estimating the resource (Landberg *et al.*, 2003). At the other extreme, more sophisticated models that focus on the nonlinear behavior of the flow, like Large-Eddy Simulations (LES), provide greater detail of the turbulence and intermittently separated flow which are the main characteristics of complex topographies (Ayotte, 2008). Their improved ability to capture turbulent behavior opens the question if their expensive computational resources are justified and necessary for the wind energy community. Models based on the Reynolds-Averaged Navier-Stokes (RANS) equations sit somewhere in the middle of this spectrum: their solutions respect mass, momentum, and energy conservation in the mean variables. They model turbulent effects in a somewhat *ad hoc* manner that has been well accepted for the purposes of wind energy forecasting (Richards & Hoxey, 1993; Kim *et al.*, 2000; Hargreaves & Wright, 2007; Sumner *et al.*, 2010).

1.1.2.1 RANS modelling of atmospheric flows

The RANS equations are derived from the Navier-Stokes equations for a Newtonian fluid and an atmospheric flow. These Navier-Stokes equations in the rotating frame of the Earth are:

$$\underbrace{\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_j)}{\partial x_j} = 0,}_{\text{conservation of mass}} \tag{1.1}$$

$$\underbrace{\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j}}_{\text{change of momentum}} = - \underbrace{\delta_{i3} g}_{\text{gravity}} - \underbrace{2\varepsilon_{ijk}\Omega_j u_k}_{\text{Coriolis force}} \underbrace{- \frac{1}{\rho}\frac{\partial p}{\partial x_i}}_{\text{pressure gradient}} + \underbrace{\frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j}}_{\text{viscous stresses}} . \tag{1.2}$$

Where x_j designates the three cartesian directions, u_i are the velocities in these directions, Ω_j is the rotational speed of the Earth, p the pressure, ρ the density and τ_{ij} the viscous stresses. By not considering the effect of Coriolis forces and by calculating implicitly the gravity force (Lange & Focken, 2006), the variables that remain are the velocity vector, pressure and τ_{ij}. The non-linear behavior of $\frac{\partial (u_i u_j)}{\partial x_j}$ at large Reynolds number makes it impossible to calculate an exact solution and methods that focus on the important effects of turbulence are utilized (Versteeg & Malalasekera, 2007). One of these methods, the RANS approach, gives attention to mean flow statistics (Versteeg & Malalasekera, 2007) and is well accepted by the wind energy field for its practicality, maturity, and modest demand of computing resources (Lange & Focken, 2006; Versteeg & Malalasekera, 2007; Sumner *et al.*, 2010). Additionally, it requires few *in situ* measurements (Prospathopoulos & Voutsinas, 2006). Although the focus is on the mean behavior of the flow, attention has to be given to the effects of turbulence (Versteeg & Malalasekera, 2007) for which an additional term is included in the governing equations, the symmetric Reynolds stress tensor τ_t which arises from the averaging process and gives six additional terms to solve. A turbulence model is required to mathematically close the system of equations. The $k-\varepsilon$ turbulence model will be used here, it was originally developed by Launder and Spalding (Launder & Spalding, 1972, 1974). Subsequently, in an effort to improve predictions, several authors have proposed diverse values for the model coefficients in order to obtain better agreement with flow measurements (Sumner *et al.*, 2010). This model

has been chosen for its good performance in cases with small pressure gradients (Bardina *et al.*, 1997) as well as being available in a wide range of CFD codes (Pope, 2000) used by the wind engineering community (Hargreaves & Wright, 2007).

1.1.2.2 RANS validation over complex terrain

Two difficulties exist on assessing resources in complex topographies: no available experimental data and no systematic comparison of different wind flow models (Bechmann *et al.*, 2011). To face these limitations, several measurement campaigns over complex terrain have been carried out to deliver experimental data in order to pursue comparisons. Some of these projects are: Black Mountain (Bradley, 1980), Blashaval Hill (Mason & King, 1985), Askervein Hill (Taylor & Teunissen, 1987), Kettles Hill (Salmon *et al.*, 1988), Hjardemål (Emeis *et al.*, 1993) and Bolund hill (Bechmann *et al.*, 2009; Berg *et al.*, 2011).

Among these projects, Askervein Hill is the most commonly used field campaign as it is well documented and represents a proper benchmark for microscale modelling (Bechmann *et al.*, 2011). The field measurement campaign was performed in the early 1980s over the 116 m-high Askervein Hill located in Scotland. Several computational models have been tested to replicate the experimental data of this project. Both linear models (*e.g.* WAsP) and non-linear models (based on RANS equations) have been shown to provide good agreement with the hilltop speed-up factor; however, non-linear models are much more accurate on the lee side. A brief description of some of the studies that used RANS modelling are chronologically presented: First, Raithby *et al.* (1987) used a RANS approach and obtained accurate results albeit with over-predicted turbulence levels. Second, Kim & Patel (2000) and Kim *et al.* (2000) focused on the wall treatment and used the renormalization group (RNG) model[1] to deliver improved predictions in the recirculation zones. Third, in an effort to improve lee-side predictions, Castro *et al.* (2003) used an unsteady Reynolds-Averaged Navier-Stokes (URANS) model to analyze the low frequency unsteadiness of the flow in the hill's wake; also, they focused on the roughness characterization and justified the used of a coarse grid. As a result, Castro *et al.* (2003)

[1] k and ε are modeled by double expansion. For more detail see (Kim & Patel, 2000)

predicted the wind speed at 10 m AGL in good agreement with the experimental data but with an overestimated k in the upstream area. Finally[2], Prospathopoulos & Voutsinas (2006) concluded that by using a RANS solver with the proper refinement in the vertical and main flow directions, and with the appropriate roughness distribution and boundary conditions, satisfactory results are obtained.

In this context, the present work continues the tradition of RANS model validation against experimental data, but a) for a more difficult flow situation involving complex terrain and forest cover; and b) explicitly evaluating the uncertainty and its impact in the context of wind energy development.

1.1.2.3 Forest modelling in atmospheric flows

As part of the scope of this master's thesis, the effect of a forest will be considered. This effect is tackled via two models: a displacement height model and a canopy model.

The displacement height model assumes an average tree height with a logarithmic wind profile starting from this edge (Stull, 1988). Several studies have been developed with this method like the work of Raupach (1994) and Verhoef *et al.* (1997), which have given promising results. Additionally, the practicality and required modest computational resources make this model an attractive tool for wind flow modelling over forested sites. Nevertheless, this method has two main limitations: 1) it is difficult to implement for a non-homogeneous forest, especially when the variation in the average tree height becomes large; and 2) it does not consider the aerodynamic drag due to the particular foliage, which can considerably impact the accuracy of the wind speed predictions, especially with season changes.

In order to take into account the mentioned limitations, canopy models have been developed. The canopy model aims to more accurately represent the physical action of the forest by calculating the pressure loss in a porous media. This model was originally developed in the work of Svensson and Haggkvist (1990) which gave good results on a qualitative basis. Subse-

[2] Final case detailed in this work, more recent studies can be found.

quently, their work motivated further research to compare the model with field measurements as done in the work of Liu *et al.* (1996). Later, improvements in the canopy model with attention to the physical characteristics of the forest were achieved in the work of Katul *et al.* (2004) and Lopes da Costa *et al.* (2006). This state of the art has been implemented in several computational codes with attention to the atmospheric boundary conditions (*i.e.* Dalpé and Masson (2008)) and complex topographies (*i.e.* Jeannotte (2013), Arroyo *et al.* (2014), Boudreault *et al.* (2014), and Grant *et al.* (2016)). The canopy model is characterized by requiring detailed information of the physical characteristics of the forest. But this information represents a constraint, mainly because it is obtained by special measurement methods that are not always available for a given forest. Some of these methods are destructive testing, satellite remote sensing, LiDAR sensing, and stereoscopic particle image velocimetry measurements; detail of theses methods are found in the cited literature (Chen & Cihlar, 1996; Omasa *et al.*, 2007; Desmond *et al.*, 2014; Boudreault *et al.*, 2015). The specific information derived from these methods can lead to accurate results for the wind flow modelling (without adding computational expenditures), as presented in the work of Desmond (2014). But what happens with limited information? Can limited physical characteristics of the forest give reasonable results for the prediction of the wind flow? These questions will be addressed when the validation of the canopy model is presented in chapter 3

1.1.2.4 Speed-up factor and its error

The RANS equations model the mean velocity and the average covariances of the velocity components. By definition then the speed-up factor predicted via a RANS solution is the ratio of the time-averaged speeds at two points, viz.

$$S = \frac{U}{U_{\text{ref}}}. \tag{1.3}$$

Experimentally, the speed-up factor is often calculated as the mean of the ratio of instantaneous speeds,

$$\overline{s} = \overline{\left(\frac{U}{U_{\text{ref}}}\right)}. \tag{1.4}$$

The former S is the correct definition for the calculation of an important parameter for the wind energy estimation: the capacity factor (CF). The capacity factor is defined as the ratio of the energy generated, for a period of time (commonly one year), to the energy that could have been generated if the machine ran continuously at the designed rated power for the same period of time (Manwell *et al.*, 2009). Additionally, S is more appropriate for comparison with RANS simulations. Nonetheless, the instantaneous speed-up factor s can be used to provide a useful measure of variance.

The predicted S cannot exactly predict the real wind flow, there is a certain degree of error when comparing with experimental data. This is mainly because models are an approximation of the real physics and because the measurement equipment itself introduces a source of uncertainty (Pinson, 2006). Additionally, factors such as the terrain complexity, roughness and obstacles expose weaknesses in turbulence closures (Lira *et al.*, 2016).

In the next section the importance of S in the calculation of CF will be detailed; therefore, one of the primary goals is to quantify the typical error in S for moderately complex sites using a RANS approach. This can be seen for example in the attempts to model the wind flow over Askervein Hill with the use of higher-order computational models. In the first approaches of Raithby (Raithby *et al.*, 1987), the predicted S from a 3D nonlinear model was shown to be more accurate than those predicted with a linear model. Following this path, Castro *et al.* (Castro *et al.*, 2003) used unsteady RANS (URANS) to capture some time-dependent effects, which resulted in and underprediction of S by less than 10%. Also for this case, Prospathopoulos has documented the quality of results that can be obtained by using RANS and with reasonable computational costs (Prospathopoulos & Voutsinas, 2006). Additionally, Kim and Patel (Kim & Patel, 2000; Kim *et al.*, 2000) presented speed-up factors in good agreement with their measurement data. Finally, when evaluating the accuracy of RANS models,

Milashuk (Milashuk & Crane, 2011) recommends using the error in the calculated speed-up factors instead of using the direct error in the wind speed prediction.

1.2 Uncertainty propagation in wind farm energy yield calculations

While the body of literature on so-called microscale wind speed prediction is vast, there is relatively little analysis on how the uncertainty in these models affects the financial risk associated with wind energy yield forecasting. Considering the wind resource uncertainty (previously discussed) and the energy production uncertainty, the latter is primarily caused by uncertainties in the power curve and related factors like electrical losses and availability (Lira *et al.*, 2016). As these are not considered in this work, the uncertainty in the energy yield calculation becomes a pure function of the predicted wind resource and its uncertainty. In this context, the effect of speed-up factor errors on wind farm energy yield calculations can be easily estimated: The cubic relationship between instantaneous wind speed and energy content (see equation (1.5)) provides an estimate for the relative uncertainty in energy produced by a given turbine of three times the relative uncertainty in the local wind speed estimate.

$$P_w(U) = \frac{1}{2}\rho A_{eff} C_P \eta u^3. \tag{1.5}$$

Equation (1.5) represents the turbine power curve P_w based on the power available in: the air density ρ, the effective area of the disk A_{eff}, wind turbine power coefficient C_P, the drive train efficiency η, and the instantaneous wind speed u. (Manwell *et al.*, 2009).

This kind of argument has led to targets of 1% uncertainty in wind speed to limit the uncertainty in energy calculations to, at most, 3% (ETIPWind, 2016). The goal of this section is to more clearly establish the link between wind speed uncertainty and energy uncertainty and in subsequent sections to evaluate, using common CFD techniques, the current level of uncertainty in wind speed estimation for a complex forested site.

1.2.1 Annual energy production

The total energy yield of a wind farm consisting of N turbines over period N_h is

$$E_{\text{farm}} = \sum_{i}^{N} E_i \tag{1.6}$$

where

$$E_i = \int_0^{N_h} P_i(t)dt. \tag{1.7}$$

This integral is often modeled based on the wind speed probability density function (pdf) $p(U)$ at the location of the turbine along with the turbine power curve. The integral is transformed via $dt = N_h p(U)dU$ to

$$E_i = N_h \int_0^{\infty} \overline{P}(U)p(U)dU \tag{1.8}$$

where $\overline{P}$ is the bin average power produced at wind speed U.

The wind speed frequency distribution at the location of a future turbine is not generally known. Rather the measured wind statistics at a given location are spatially extrapolated through the use of speed-up factors, S_i. The wind speed at turbine i is given by $U_i = S_i U_j$ while its frequency distribution is assumed to follow $p_j(U)$ where j indicates the reference position. Integrating the pdf at the reference location gives

$$E_i = N_h \int_0^{\infty} \overline{P}(S_i U_j)p_j(U_j)dU_j. \tag{1.9}$$

This integral can be split into two parts

$$E_i = N_h \Big[\int_0^{U_{\text{rated}}/S_i} \overline{P}(S_i U_j)p_j(U_j)dU_j + \overline{P}_{\text{rated}} \left[F(U_{\text{out}}/S_i) - F(U_{\text{rated}}/S_i)\right] \Big] \tag{1.10}$$

where $F(U)$ is the cumulative probability distribution function and U_{out} and U_{rated} are the cut-out and rated wind speeds of the turbine. Also, it is assumed that $P(U) = \overline{P}_{rated}$ if $U > U_{rated}$, and $P(U) = 0$ if $U > U_{out}$.

1.2.2 Uncertainty in capacity factor due to modelling

The interest here is in determining the effect of the uncertainty in S on the uncertainty in the energy calculation. For this purpose, it makes sense to assume that both $p(U)$ and $\overline{P}(U)$ are statistically representative and without associated uncertainty. Furthermore, S is assumed to be spatially variant but independent of U_j. Finally, to simplify the analysis, it will be assumed that turbines operate in two discrete states: at constant $\overline{C_P}$ below U_{rated} and at constant $\overline{P}$ after U_{rated}. With these refinements,

$$E_i = N_h\left[\frac{1}{2}\rho A\overline{C_P}S_i^3\int_0^{U_{\text{rated}}/S_i} U_j^3 p(U_j)dU_j + \overline{P}_{\text{rated}}\left[F(U_{\text{out}}/S_i) - F(U_{\text{rated}}/S_i)\right]\right]. \tag{1.11}$$

It is convenient to adimensionalize the analysis by working with capacity factor:

$$CF_i = \frac{E_i}{\overline{P}_{\text{rated}}N_h} \tag{1.12}$$

where

$$\overline{P}_{\text{rated}} = \frac{1}{2}\rho A_{eff}C_P\eta U_{\text{rated}}^3. \tag{1.13}$$

Then, CF is expressed as

$$CF_i = \frac{S_i^3}{U_{\text{rated}}^3}\int_0^{U_{\text{rated}}/S_i} U_j^3 p(U_j)dU_j + F(U_{\text{out}}/S_i) - F(U_{\text{rated}}/S_i). \tag{1.14}$$

Assuming the pdf is Rayleigh distributed with mean wind speed $\overline{U}$,

$$p(U) = \frac{\pi}{2}\frac{U}{\overline{U}^2}\exp\left(-\frac{\pi}{4}\left(\frac{U}{\overline{U}}\right)^2\right) \tag{1.15}$$

$$F(U) = 1 - \exp\left(-\frac{\pi}{4}\left(\frac{U}{\overline{U}}\right)^2\right) \tag{1.16}$$

the integration can be carried out:

$$CF_i = \frac{\pi}{2\overline{U}^2}\frac{S_i^3}{U_{\text{rated}}^3}\int_0^{U_{\text{rated}}/S_i} U_j^4 \exp\left(-\frac{\pi}{4}\left(\frac{U_j}{\overline{U}}\right)^2\right) dU_j + F(U_{\text{out}}/S_i) - F(U_{\text{rated}}/S_i). \quad (1.17)$$

After integration, simplification and the substitution

$$\beta = \frac{\sqrt{\pi}}{2S_i}\frac{U_{\text{rated}}}{\overline{U}} \quad (1.18)$$

the general expression for capacity factor is

$$CF = \frac{3\sqrt{\pi}}{4\beta^3}\text{erf}(\beta) - \frac{3}{2\beta^2}\exp\left(-\beta^2\right) - \exp\left(-\beta^2\frac{U_{\text{out}}^2}{U_{\text{rated}}^2}\right). \quad (1.19)$$

The last term can be safely neglected as U_{out} is generally much larger than $\overline{U}$. This leaves

$$CF = \frac{3\sqrt{\pi}}{4\beta^3}\text{erf}(\beta) - \frac{3}{2\beta^2}\exp\left(-\beta^2\right). \quad (1.20)$$

Now we consider the absolute error in capacity factor δCF for a given absolute error in speed-up factor δS:

$$\delta CF = \left(\frac{\partial CF}{\partial \beta}\frac{\partial \beta}{\partial S}\right)\delta S \quad (1.21)$$

$$\frac{\partial CF}{\partial S} = \frac{9\sqrt{\pi}\text{erf}(\beta)}{4\beta^3 S} - \left(\frac{9}{2\beta^2 S} + \frac{3}{S}\right)\exp(-\beta^2). \quad (1.22)$$

It makes sense here to reorganize the error function in terms of two separate non-dimensional quantities: the speed-up factor S, and the ratio of rated to mean wind speeds:

$$\gamma = U_{\text{rated}}/\overline{U}. \quad (1.23)$$

Thus, the error equation is written as:

$$\frac{\partial CF}{\partial S} = \frac{9\text{erf}\left(\frac{\sqrt{\pi}}{2}\frac{\gamma}{S}\right)}{\left(\frac{\pi}{2}\frac{\gamma^3}{S^2}\right)} - \left(\frac{9}{\left(\frac{\pi}{2}\frac{\gamma^2}{S}\right)} + \frac{3}{S}\right)\exp\left(-\frac{\pi}{4}\frac{\gamma^2}{S^2}\right). \tag{1.24}$$

This is illustrated in figure 1.2. As can be seen, the **absolute** error in capacity factor is a slight function of speed-up factor for γ close to 2, with δCF roughly half of δS, but varies greatly for small γ. This analysis shows that low wind speed sites ($S < 1$) will lead higher uncertainty than low wind speed sites ($S > 1$) and thus will require more accurate modelling.

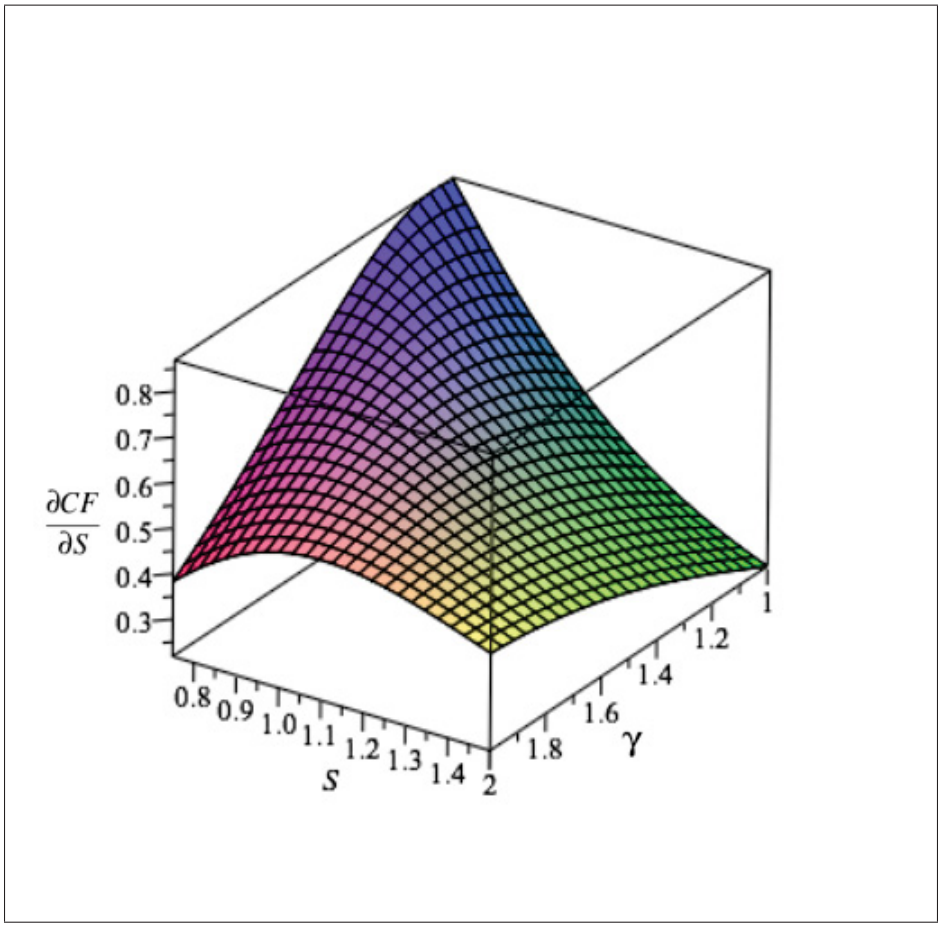
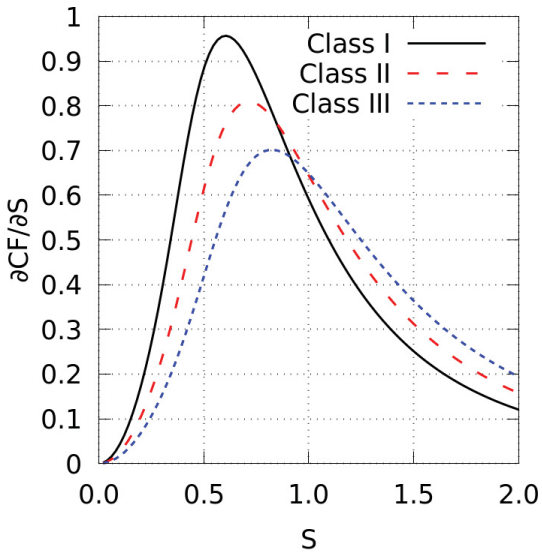


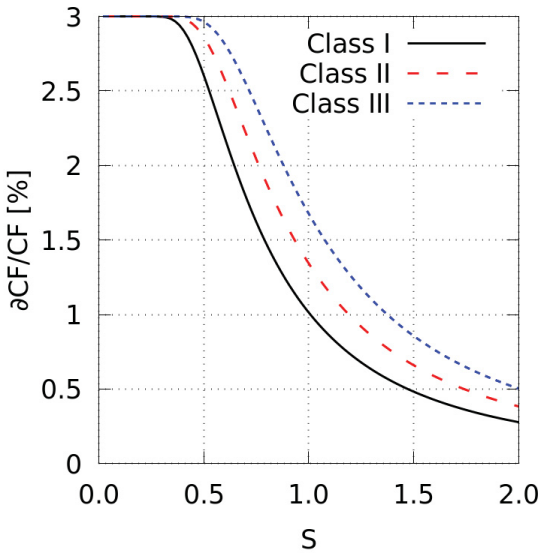
Figure 1.2 Absolute error in capacity factor for a unit absolute error in speed-up factor

This same analysis is reframed in the context of wind classes from the IEC 61400-1 standard (IEC 61400-1, 2005). Three wind classes I, II and III are defined with a characteristic annual average wind speed of 10, 8.5 and 7.5 m/s, respectively (IEC 61400-1, 2005). Considering these values as mean wind speed $\overline{U}$ and an assumed rated wind speed (U_{rated}) of 11 m/s, the absolute error in capacity factor for a unit absolute error in speed-up factor ($\frac{\partial CF}{\partial S}$) is calculated for each wind class with equation (1.24). Slices taken from these calculated errors are shown in figure 1.3a. In addition, figure 1.3b gives the same information in relative terms, *i.e.* the percent error in capacity factor ($\frac{\partial CF}{CF}$) for a 1% error in speed-up factor ($\frac{\partial S}{S}$).

A noteworthy result is that for unity speed-up factor on a class I site, the relative error in S and CF are **identical**. Furthermore, for $S<1$ the relative error in CF is *bigger* than that of S, which scenario is assumed as a representation of low wind speed sites. But what provokes low wind speeds? Are the effects of complex forested terrains some of the causes? On the other hand, for $S>1$ the relative error in CF is *less* than that of S, which is assumed as high wind speed sites. In general, regardless of the wind class, it is only in the limit of small S where the relative error in CF will match the 3-to-1 ratio predicted by the energy content. These results imply that the required accuracy of flow solvers is actually somewhat less than previously estimated for a given target uncertainty in CF. Now the question arises: What is S and its error for a complex forested site in order to evaluate its CF uncertainty? This will be traced in the next chapters, starting with the methodology in chapter 2.



a) Absolute error in capacity factor for a unit error in speed-up factor



b) Relative error in capacity factor, *i.e.* percent error in capacity factor for a unit percent error in speed-up factor

Figure 1.3 Uncertainty in capacity factor as a function of speed-up factor for specific wind classes (assuming U_{rated}=11 m/s)

CHAPTER 2

METHODOLOGY

One of the main focus of this master's thesis is to properly model the atmospheric wind flow through CFD. This is achieved by following a methodology that aims to respect the dominant physical processes of the lower atmosphere at a microscale scale with computational methods. This chapter describes this methodology and addresses the first specific objective of this work: to implement computational source terms in the RANS and k-ε turbulence models to account for the effects of both terrain and forest in the neutral atmospheric boundary layer

2.1 Physics of the atmospheric boundary layer

The physics of the atmosphere are described by the system of equations derived from the principles of conservation of mass, momentum, and heat. But their inherent non-linear behavior makes it impossible to solve them analytically; therefore, numerically methods are relied on (Lange & Focken, 2006). To facilitate the application of these equations, the atmosphere is divided into several layers according to the dominant effects that influence their dynamics (Lange & Focken, 2006). Considering only the layer involved in microscale modelling, the troposphere, it comprises the region from the ground to the first $\sim$20km in the tropics or $\sim$6km in the polar regions. In its broadest sense, it is composed of the free atmosphere and the atmospheric boundary layer (ABL). The free atmosphere is the upper part that is influenced by the rotation of the Earth, it has as its main driving force the geostrophic wind and it does not depend on the topography. On the other hand, the ABL is the part of the troposphere that is influenced by the surface of the Earth (Stull, 1988). Since the ABL is of greatest interest for wind energy purposes it is considered in more detail. The ABL is divided in three sublayers:

Roughness sublayer. This is the smallest region dominated by molecular viscosity. It starts from the roughness length z_0 until approximately ten times this length ($\sim 10 z_0$) for flat terrain (Emeis, 2011), or higher values for forested terrain as this layer includes the canopy air space (Kaimal & Finnigan, 1994). The roughness length z_0 is proportional to the roughness

of the ground terrain; theoretically, this is where the value of velocity should be zero. This roughness length is seldom taken as zero since real topographies are never found as even and smooth (Manwell *et al.*, 2009). A summary of roughness length values for common surfaces are found in table 2.1.

Table 2.1 Proximated values of roughness lenght for diverse types of terrain. Values taken from (Manwell *et al.*, 2009)

Type of terrain	z_0 **(m)**
Very smooth, ice or mud	0.00001
Calm open sea	0.0002
Lawn grass	0.008
Crops	0.05
Few trees	0.1
Forest and woodlands	0.5
Suburbs	1.5
Centers of cities with tall buildings	3.0

Surface or Prandtl layer. Limited to a height of $\sim$10% of the whole ABL, Coriolis and pressure gradients forces are negligible in this layer. In this layer, under neutral and horizontally homogeneous conditions, the constant shear stress leads to a wind profile (with respect to height) defined with the logarithmic law of the wall:

$$U(z) = \frac{u_*}{\kappa} \ln \frac{(z+z_0)}{z_0}. \tag{2.1}$$

Ekman layer. This is a major component of the ABL, where the horizontal wind speed increases with respect to height until attaining its maximum at the free atmosphere. At this limit, it is expected that pressure gradients are in equilibrium with the Coriolis forces (Emeis, 2011).

Now with an idea of the structure of the ABL, the dominant physical phenomena, and their causes and effects will be identified. The main interest here is atmospheric turbulence. In general, it is composed of thermal and mechanical contributions that both represent some of the

most difficult aspects to computationally model in the ABL (Apsley & Castro, 1997). While the thermal turbulence is originated in the vertical heat flux (thermal stratification), the mechanical turbulence is due to surface shear stress and terrain roughness (effects of complex topographies) (Apsley & Castro, 1997).

2.1.1 Thermal stratification

This master's thesis assumes neutral atmospheric stability which does not consider thermal effects; thus, the mathematical model does not include the energy equation. Notwithstanding, considering the importance of the thermal turbulence on the wind profile, it will be briefly described. The information here is mainly from Lange and Focken (Lange & Focken, 2006) and the reader is encouraged to refer to these authors for more detail.

Thermal stratification is the vertical temperature distribution of the atmosphere (Lange & Focken, 2006). It is caused by the daily radiation cycle (Manwell *et al.*, 2009). During the day, the ground is heated by solar radiation, consequently the air is heated from below creating a parcel of air with lower density that tends to rise; this is known as the buoyancy effect (Lange & Focken, 2006). When this parcel of air is rising, it experiences the decreasing atmospheric pressure. As result, the parcel of air expands; thus, its volume increases and its internal energy decreases. In other words, the parcel of air is decreasing its temperature and doing work on its surroundings. All this process is considered as adiabatic since it is not originated neither by adding nor subtracting heat from the system (Lange & Focken, 2006). This adiabatic process of the thermal stratification involves the potential temperature Θ which (with respect to height z) is defined as:

$$\Theta(z) = T(z) + \Gamma_d z, \tag{2.2}$$

where T is the absolute temperature and Γ_d is the dry adiabatic lapse rate, measured to be

$$-\partial_z T = \Gamma_d \simeq 0.01 \frac{K}{m}. \tag{2.3}$$

The vertical gradient of the potential temperature can be used to define three types of thermal stratification:

- $\partial_z \Theta < 0$: **unstable**. The vertical temperature stratification enhances the vertical momentum flux due to buoyancy, *e.g.* during a sunny day with high solar radiation (Lange & Focken, 2006);
- $\partial_z \Theta = 0$: **neutral**. The vertical temperature stratification does not influence the vertical momentum flux (Lange & Focken, 2006);
- $\partial_z \Theta > 0$: **stable**. The vertical temperature stratification dampens the vertical momentum flux; mainly, as a result from the ground being cooled at night (Lange & Focken, 2006).

The main difference between the gradients is the sign which indicates if the vertical movement of air is enhanced or dampened (Lange & Focken, 2006).

As can be deducted, this temperature gradient has a considerable effect on the wind profile as it is the force of the buoyancy effect. In order to be considered in the turbulence model it has to be added as source since it is not provided by the state of the flow (Lange & Focken, 2006). Broadly, this gradient is added to the momentum equations with a turbulence closure that includes sources of atmospheric turbulence (Lange & Focken, 2006). This is done with the use of a scale parameter that describes the effects of buoyancy on turbulent flows, the Monin–Obukhov length (Obukhov, 1971). This procedure is not detailed here, but for completeness, the reader is recommended to revise the cited literature (Monin & Obukhov, 1954; Stull, 1988; Lange & Focken, 2006).

2.1.2 Effects of complex topographies

For flat terrain, the prediction of wind speed is based on the well-known logarithmic law of the wall (equation (2.1)). This is accepted because it is assumed that an equilibrium flow will be maintained. In contrast, for a complex topography this implementation is no longer

valid: spatial inhomogeneities in elevation and ground roughness play an important role in creating mechanical turbulence (Panofsky & Ming, 1983; Apsley & Castro, 1997). Specifically, since equation (2.1) is a function of height, shear stress and roughness it makes sense to assume that when air experiences drastic changes in these parameters the equilibrium will be lost (Kaimal & Finnigan, 1994). The acceleration or deceleration caused in the wall region will affect the velocity gradient as it changes with altitude. Likewise, when atmospheric flow faces an obstacle (such as trees, building, slopes, hills, etc.) it is accelerated due to the change in the pressure gradients. But, downstream of such obstacles, other phenomena could be present like wake vortices or separation zones (Ayotte, 2008; Bautista, 2015). Here is where, if accuracy is required, more sophisticated models not based on the linearization assumption must be used. It has already been mentioned that RANS models will be used here since their results are well-accepted for moderately complex terrain (Lange & Focken, 2006; Versteeg & Malalasekera, 2007; Sumner *et al.*, 2010). Non-linear effects common to complex topographies will be implicitly captured by the RANS equations.

To precisely account for these characteristics, this master's thesis will use two distribution maps: topography and roughness. A description and pre-treatment of these maps will be described later when a real case is examined and likewise it will be detailed in Appendix I.

2.2 Mathematical model

2.2.1 Fundamental equations

Previously, the Navier-Stokes equations for a Newtonian fluid, incompressible atmospheric flow that does not consider thermal effects were described in the chapter of literature review (chapter 1) as equations (1.1) and (1.2). Now they are simplified as equations (2.4) and (2.5) by adding the following assumptions: steady flow, no Coriolis forces, and gravity forces calculated

implicitly:

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{2.4}$$

$$\frac{\partial (u_i u_j)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}. \tag{2.5}$$

Where x_j designates the three cartesian directions, u_i are the velocities in these directions, p the pressure, ρ the density and ν the kinematic viscosity.

2.2.2 RANS modelling of the atmospheric boundary layer

The preceding system of equations describes the exact details of the variables that fluctuate in the turbulent atmospheric flow (Lange & Focken, 2006). As previously explained, the present work is only interested in the mean behavior of these fluctuations. Therefore, equations (2.4) and (2.5) will be written in their time-averaged form by using the Reynolds Averaged Navier-Stokes model. Specifically, RANS model consists in decomposing the velocity and pressure in equations (2.4) and (2.5) to a mean and a fluctuating part over a certain interval of time (Reynolds, 1895). Thus *e.g.*, velocity will be represented as (Lange & Focken, 2006)

$$u_i(t) = U_i(t) + u_i'(t), \tag{2.6}$$

where

$$U_i(t) = \overline{u_i'}(t). \tag{2.7}$$

The overbar represents the temporal average over an interval of time Δt

$$\overline{u_i'}(t) = \frac{1}{\Delta t}\int_0^{\Delta t} u_i'(t)dt'. \tag{2.8}$$

Where Δt should be bigger than the longest time scale of the smallest variation.

By applying equation (2.6) and certain divergence and gradient differentiations rule (that are explained in detail in (Versteeg & Malalasekera, 2007)), the governing equations (2.4) and (2.5)

in their time-averaged form are represented as follows:

$$\frac{\partial U_j}{\partial x_j} = 0, \tag{2.9}$$

$$\frac{\partial (U_i U_j)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j} + S_{Ui}. \tag{2.10}$$

It is seen that the averaging process (equations (2.6) – (2.8)) introduced the products of fluctuating velocities $\overline{u_i' u_j'}$ in the momentum equation (2.10). These products are commonly placed on the right hand side to emphasize their effects as turbulent stresses (Versteeg & Malalasekera, 2007). These products of fluctuating velocities represent six additional terms to solve that are known as the symmetric Reynolds stress tensor τ_t

$$\tau_{t,ij} = \rho \overline{u_i' u_j'}. \tag{2.11}$$

In order to close the system of equations, a turbulence model is required. Additionally, a source term is included which in the present work will account for the forest effect in the canopy model and will be discussed in the canopy model section.

2.2.3 Turbulence closure

The $k-\varepsilon$ turbulence model will be used with the Boussinesq approximation to deal with the Reynolds stresses

$$\tau_{t,ij}' = -\overline{u_i' u_j'} = \nu_t \left[\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right] - \frac{2}{3} k \delta_{ij} \tag{2.12}$$

where ν_t is the turbulent or "eddy" viscosity and k is the turbulent kinetic energy, which are respectively expressed as

$$\nu_t = C_\mu \frac{k^2}{\varepsilon} \tag{2.13}$$

$$k = \frac{1}{2} \overline{u_i' u_i'}. \tag{2.14}$$

Two additional transport equations are necessary to mathematically close the problem, one for the turbulent kinetic energy k and one for its dissipation rate ε

$$\frac{\partial(kU_i)}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k - \varepsilon + S_k \tag{2.15}$$

$$\frac{\partial(\varepsilon U_i)}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + C_{\varepsilon 1}\frac{\varepsilon}{k}G_k - C_{\varepsilon 2}\frac{\varepsilon^2}{k} + S_\varepsilon \tag{2.16}$$

where S_k and S_ε correspond to the source terms of the canopy model. The production of kinetic energy, G_k, is defined as

$$G_k = \nu_t\left(\frac{\partial U_i}{\partial x_j}\right)^2. \tag{2.17}$$

Diverse values for the model coefficients are found in the literature (Sumner *et al.*, 2010), the original values corresponding to Launder and Spalding (Launder & Spalding, 1972, 1974) are found in table 2.2. A different set of coefficients will be used for each forest model. For the displacement height model, the proposed constants of Apsley and Castro (1997) defined in table 2.2 will be used. For the canopy model, the values will be taken as in the work of Dalpé and Masson (2008) to be consistent with the validation; they are referenced in table 2.3. As concerns any difference that may be found in the results due to the choice of coefficients, a sensitivity analysis was performed using the displacement height model with the coefficients of the canopy model C_μ and σ_ε. It was revealed that the difference in predicted speed-up factors is negligible.

Table 2.2 Values for the coefficients used in the $k-\varepsilon$ turbulence model

Author	C_μ	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	σ_K	σ_ε	κ
Launder and Spalding (Launder & Spalding, 1972, 1974)	0.09	1.44	1.92	1.0	1.3	0.42
Apsley and Castro (Apsley & Castro, 1997)	0.09	1.44	1.92	1.0	1.11	0.40

Table 2.3 Turbulence coefficients for the modified $k-\varepsilon$ turbulence model used in the canopy model

C_μ	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	σ_K	σ_ε	$C_{\varepsilon 4}$	$C_{\varepsilon 5}$	β_P	β_d
0.03	1.44	1.92	1.0	2.12	0.78	0.78	1.0	5.03

2.2.4 Forest modelling

2.2.4.1 Displacement height model

The displacement height model (DH) assumes that the trees are very close together and a solid volume of leaves can be considered. As such, a logarithmic wind speed profile is assumed to start at the forest edge (Stull, 1988). This displacement height is considered in the model by elevating the ground height (z_g) in the topography file by this DH value (d) (see figure 2.1). d depends on the mean height of the forest and its density; generally the ratio of d to the mean height of the forest trees is $\sim$2/3, and the roughness value varies between 0.05 to 0.1 m (Mortensen *et al.*, 2011). The elevation of d is created directly in the topography map file with the help of geographic information systems (GIS) software.

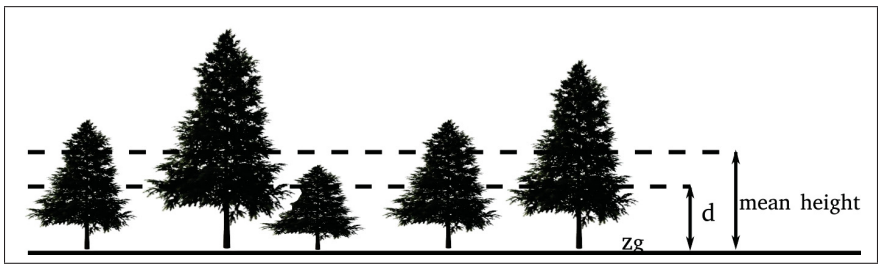


Figure 2.1 Representation of the elevation in the displacement height model. Figure adapted from Stull (1988)

2.2.4.2 Canopy model

The canopy model aims to reproduce the drag effect of the forest (per unit volume) in the governing equations. This effect will result in a wind profile as in figure 2.2. The forest is

considered as a porous media that causes pressure losses and viscous forces in the wind flow. This effect is applied through a new sink term in the momentum equation:

$$S_{Ui} = -C_D \alpha U U_i. \tag{2.18}$$

S_{Ui} stands for the momentum extraction rate of the forest, C_D represents the drag coefficient of the forest, α the leaf area density per unit of volume $(m^2 m^{-3})$ and U the wind speed $(ms^{-}1)$.

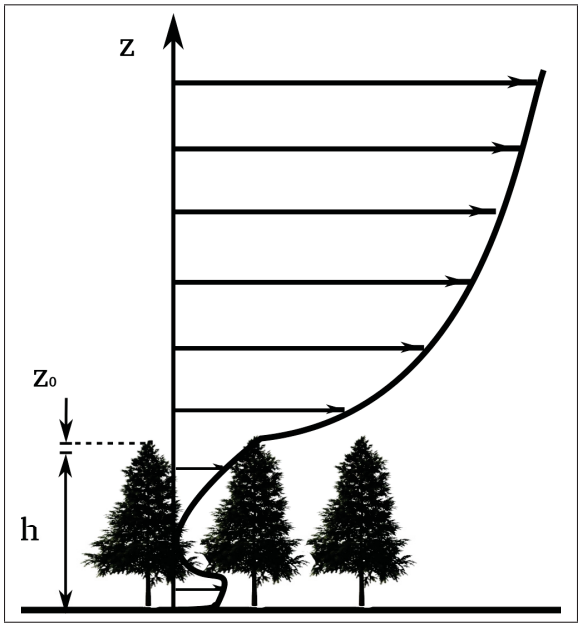


Figure 2.2 Wind profile representation of the canopy model. h represents the height of the trees (m) and z_0 is the roughness length (m). Figure in analogy with Stull (1988)

Each type of forest has its characteristic C_D value and α distribution. The latter, when integrated over the corresponding tree height, defines the leaf area index LAI:

$$LAI = \int_0^h \alpha dz. \tag{2.19}$$

As the α distribution is obtained by destructive methods, satellite remote sensing or LiDAR sensing, the data is not always available for a given forest (Chen & Cihlar, 1996; Omasa *et al.*, 2007; Desmond *et al.*, 2014; Boudreault *et al.*, 2015). For some forest types, the α distribution is known, thus the integration can be precisely made. But, when this information is not available, as in this work, an assumption of a generic leaf area density must be used (Dalpé & Masson, 2008). This generic leaf area density will adopt a triangular shape, in which the *LAI* value will be respected. As an example, the differences between the exact and generic α distributions as a function of height for a typical black spruce forest are shown in figure 2.3. This foliage will be used for model validation and for the real site in the upcoming chapters. This generic distribution assumes a constant and small value α from the ground to $z/h = 0.4$ in an attempt to avoid a near-ground jet in the simulations.

Additionally, following the guidelines on the use of source terms to model forest flows, the transport equations of k and ε are also modified with the new source terms S_k and S_ε to take into account the turbulence generated by the forest (Svensson & Haggkvist, 1990):

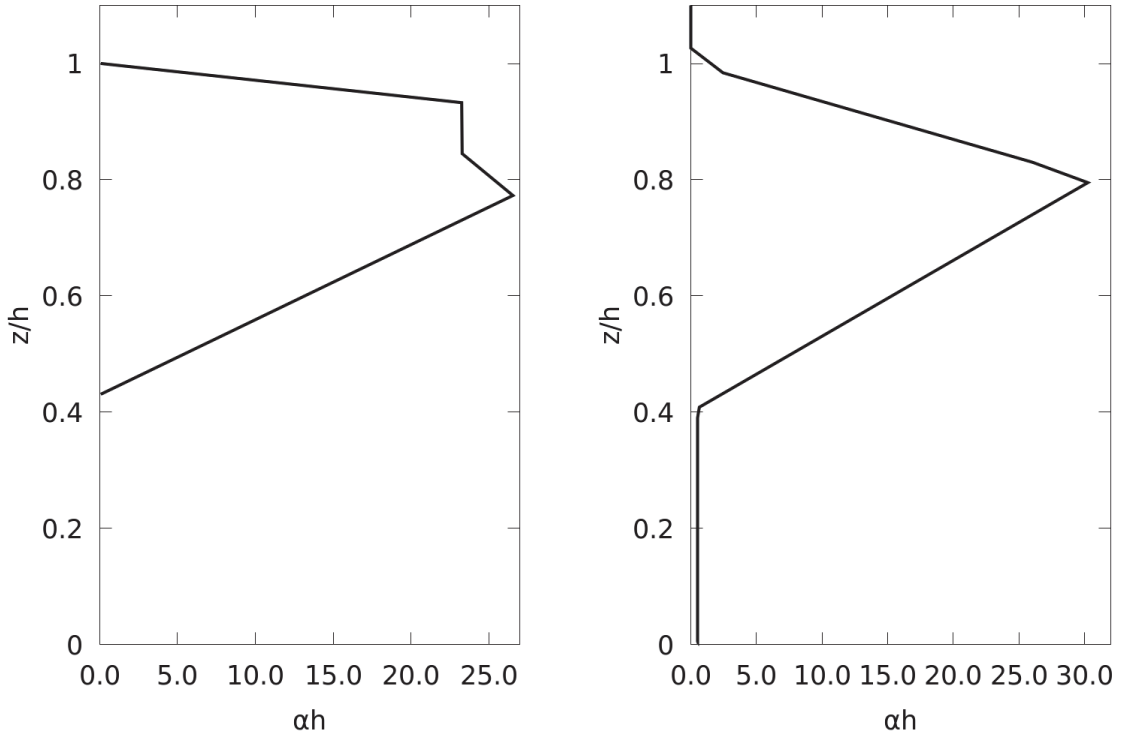
$$S_k = -C_D \alpha (\beta_P U^3 - \beta_d k U) \tag{2.20}$$

$$S_\varepsilon = -C_D \alpha \frac{\varepsilon}{k} (C_{\varepsilon 4} \beta_P U^3 - C_{\varepsilon 5} \beta_d k U). \tag{2.21}$$

The new model coefficients $C_{\varepsilon 4}$, $C_{\varepsilon 5}$, β_P and β_d are also found in table 2.3.

2.3 Numerical method

The previously described system of partial equations that governs the flow is impossible to solve analytically at high Reynolds number. Numerical approaches are thus useful (and well accepted in engineering fields) specifically approximate solutions based on computational fluid dynamics (CFD) (Ferziger & Peric, 2002).



a) Represents the exact α distribution (Amiro, 1990)

b) Assumes a generic α distribution that adopts a triangular shape, in which the LAI value is respected

Figure 2.3 Leaf area density distribution for the black spruce forest

2.3.1 Discretization

The principle of CFD is to use a discretization method that approximates the partial differential equations by a system of algebraic equations that can be numerically solved (Ferziger & Peric, 2002). The most common methods are: finite difference (FD), finite volume (FV) and finite element (FE) (Ferziger & Peric, 2002). All the simulations in this work are performed with the finite volume method; therefore, all the discretization processes will be in terms of this method. For more detail about the other methods the reader is recommended to review the cited literature (Patankar, 1980; Ferziger & Peric, 2002; Versteeg & Malalasekera, 2007).

The discretization process consists of dividing the computational domain into a finite number of contiguous control volumes (CVs) and calculating flow variables at the centroid of each CV. It starts from the assumption that all the partial differential equations that govern the wind flow have a similar structure. Thus, these equations can be represented with the generic form:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla\cdot(\rho\phi\mathbf{u}) = \nabla\cdot(\rho\Gamma\nabla\phi) + S_\phi \tag{2.22}$$

where ϕ is the general variable, Γ is the diffusion coefficient, and $\nabla\cdot$ and ∇ stand for divergence and gradient operations respectively. ϕ is treated as the only unknown whereas the other variables are fixed at their previous iteration values. At this moment the velocity field is assumed to be known and later it will be explained how it is computed. Equation (2.22) is integrated over a control volume (figure 2.4 shows a typical 3D representation). The CV is a polygon of six plane faces which are named with lower-case letters that indicate the direction with respect to the node P. The directions are designated as: T (Top), B (Bottom), N (North), S (South), W (West) and E (East). The points that accompany these directions also represent the centroid of the neighbor cells.

In line with the shape of the CV and recalling the steady state assumption, equation (2.22) is integrated over the control volume. The integration is based on Gauss' divergence theorem (Ferziger & Peric, 2002; Versteeg & Malalasekera, 2007):

$$\int_A \mathbf{n}\cdot(\rho\phi\mathbf{u})dA = \int_A \mathbf{n}\cdot(\Gamma\nabla\phi)dA + \int_{CV} S_\phi dV. \tag{2.23}$$

Where $\mathbf{n}$ is the unit outward normal to the surface of the control volume, A is the cross-sectional area of the face (shaded area in figure 2.4) and dV is the volume. Equation (2.23) represents the balance between the net convective flux (left hand side) and diffusive flux and generation or destruction of ϕ (right hand side) (Versteeg & Malalasekera, 2007).

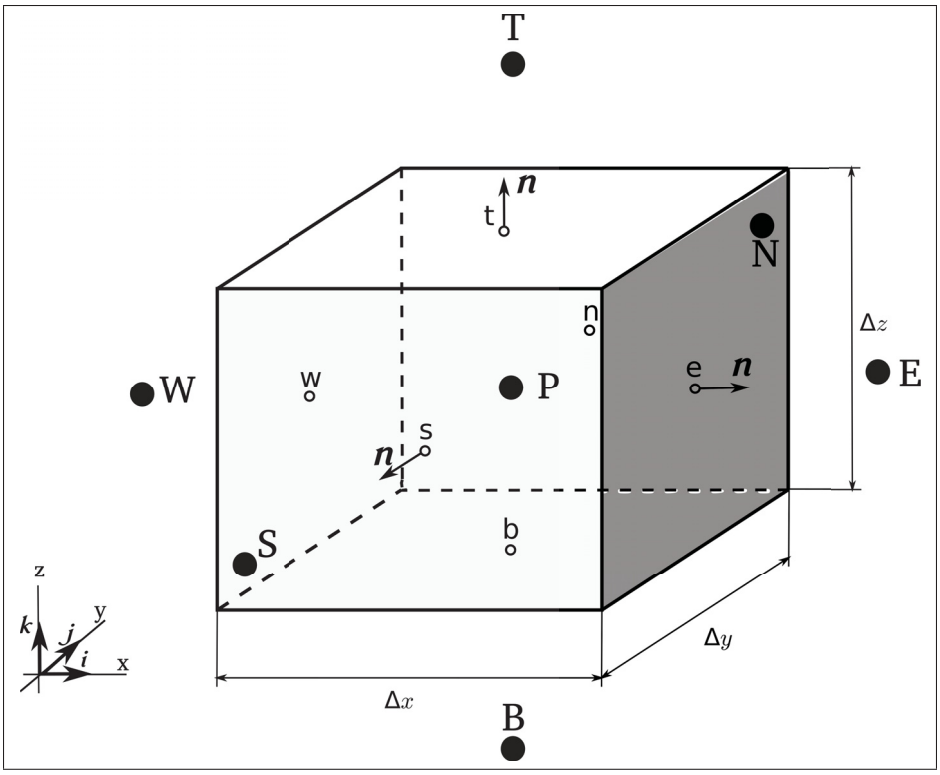


Figure 2.4 3D representation of a CV cell with the respective notations. Figure in analogy with Ferziger and Peric (2002)

Particularly, the total flux through the CV is calculated as the sum of the integrals over the six surfaces:

$$\int_A f dA = \sum_k \int_{A_k} f dA, \tag{2.24}$$

where f represents convection ($\mathbf{n}\cdot\rho\phi\mathbf{u}$) or diffusion ($\mathbf{n}\cdot\Gamma\nabla\phi$) (Ferziger & Peric, 2002).

The source term has to be first linearized as recommended in the literature (Patankar, 1980; Jasak, 1996; Versteeg & Malalasekera, 2007):

$$S_\phi(\phi) = Su + S_p\phi, \tag{2.25}$$

to subsequently be treated as a volume integral:

$$\int_{CV} S_\phi(\phi) dV = SuV_P + S_P V_P \phi_P. \tag{2.26}$$

The calculation of the integrals in both equations (2.24) and (2.26) will be explained in a general manner; mainly, because it is not within the scope of this work to alter existing methods. To better describe the procedure, the e node in figure 2.4 will be used [3]. For convection and diffusion terms in equation (2.24), it is necessary to know the integrand f_e on the whole surface A_e, but since its value is unknown (because only f at the node P is calculated), two steps are followed: interpolation and approximation (Ferziger & Peric, 2002).

Interpolation scheme. The value of f_e at the center of the face A_e has to be obtained by interpolation from the centroid P. Several interpolation schemes have been developed, of which the most commonly used are: Upwind Interpolation (UDS), Linear Interpolation (CDS), Quadratic Upwind Interpolation (QUICK), among others. For more detailed information about these methods, see Patankar (1980), Jasak (1996), Versteeg and Malalasekera (2007). These methods were listed according to their order of accuracy, *i.e.* the order of the truncation error. This error is defined as the difference between the discretized and the exact equations (Ferziger & Peric, 2002). Any function can be represented by a Taylor series as the infinite sum of its derivatives. This representation becomes finite by truncating the derivatives at some degree. At higher order truncation error, more accurate results can be obtained, but oscillations in the convergence may be present (Ferziger & Peric, 2002). Commonly, it is considered sufficient first order accuracy for flows aligned with the mesh; but, in the other case (mostly complex geometries), higher order discretization is strongly recommended (Versteeg & Malalasekera, 2007). Therefore, for both accuracy and stability the simplest and the most commonly used second-order scheme, the linear interpolation (CDS), is used in this work. In the FD method, this represents the central-difference scheme.

Approximation method. Now the value of f_e at the center of the face A_e will be used as an approximation to calculate its value over the whole face A_e. The simplest approximation to the integral in equation (2.24) is the midpoint rule. This approximation consists in multiplying the

[3] To obtain the expressions for the other faces, it will suffice to substitute the lower-case letter label.

integrand at the cell-face center f_e and the face area A_e:

$$F_e = \int_{S_e} f dA = \overline{f}_e A_e \approx f_e A_e \tag{2.27}$$

Analogously, for the source term in equation (2.26), the approximation of the integral is made for the CV with the exact value s calculated at the center P

$$S_P = \int_V s dV = \overline{s} \Delta V \approx s_P \Delta V. \tag{2.28}$$

It is noted that no interpolation scheme is required *a priori*. This is because the calculated value of S_P is assumed to be constant or linearly variable in the CV (Ferziger & Peric, 2002). If this is not true then it becomes a second-order approximation (Ferziger & Peric, 2002).

2.3.1.1 Pressure-velocity coupling

Previously, it was mentioned that the velocity field was assumed as known in the discretization equations. This in order to facilitate the explanation of the interpolation and approximation methods. In reality, the velocity is usually not known and is part of the solution variables (Versteeg & Malalasekera, 2007).

The non-linear behavior of the velocity field is already difficult to treat and it appears in all equations of the system. The challenge becomes even greater when dealing with the pressure field (Versteeg & Malalasekera, 2007). For incompressible flows, pressure only appears in the momentum equation and is not linked to transport equations [4] (Versteeg & Malalasekera, 2007). Recalling the governing steady-state assumption, the best option to deal with the pressure-velocity coupling is through the iterative algorithm SIMPLER (Semi-Implicit Method for Pressure-Linked Equations Revised) implemented by Patankar (1980). Broadly speaking, it consists in evaluating the convective fluxes per cell by starting with a guessed velocity field

[4] For compressible flows, the continuity or energy equation can act as transport equations to help to calculate pressure (Versteeg & Malalasekera, 2007).

to first solve the coefficients of the momentum equations. The pressure is then calculated using a discretized equation for pressure. The velocity field is then corrected and subsequently the other ϕ variables are solved. The process loops until finally reaching convergence. For a detailed description, see Patankar (1980) and Versteeg & Malalasekera (2007).

2.3.2 OpenFOAM

Respecting the foregoing description of the discretization processes, all simulations in this master's thesis were performed with *OpenFOAM*[5] v.2.4.0 (Weller & Tabor, 1998). *OpenFOAM* is an open source software written in C++ where the acronym *FOAM* stands for *Field Operation and Manipulation*. It was selected for the present work given its popularity in the wind energy community and likewise for its attractive freedom of manipulation.

Broadly, *OpenFOAM* acts as a library and creates two kinds of executable files named applications: solvers and utilities. While solvers will (as their name implies) solve the cases, utilities provide data handling, pre- and post-processing, etc. A case in *OpenFOAM* is constructed by a group of folders and files that defines the boundary conditions, the mesh, the discretization schemes, and the convergence tolerance. To show the case structure used in this work, an example is included in Appendix II.

2.3.3 Physical domain and mesh

2.3.3.1 Pretreatment of topography and roughness maps

Previously, the importance of considering the effects of topography and roughness when working with real sites was mentioned. In order to work with these parameters their map files are needed. While the topography map defines the physical elevation of the terrain, the roughness map defines the distribution of the roughness length z_0. These maps were available in raster format. This format simply contains three columns: X and Y for the coordinates; and Z for

[5] Copyright ©2004-2015 OpenCFD Ltd (ESI Group).

either the elevation height or z_0. These map files were obtained from the *Énergie et Ressources naturelles Québec* website (Gouvernement du Québec, 2017) and instead of being directly used, a pretreatment is applied to respect mesh constraints and boundary conditions (explained later). The structure of both mesh and boundary conditions is designed in such a manner that $z = 0$ m at the lower boundary of the computational domain. And this boundary is offset from the ground z_g by z_0 (see figure 2.7). Therefore, both maps are first treated to arrange the distribution of z_0 according to each case and to elevate the topography map by this distribution. The specific procedure of this pretreatment is explained in Appendix I and is enumerated in chapter 4 when working with a real site.

2.3.3.2 Mesh generation

The computational domain shown in figure 2.5 is created in *ZephyTOOLS*[6] v.15.06 (Zephy-Science, 2012) whose general approach is to create a cylindrical horizontally unstructured mesh around a refined zone of interest. This refined zone of interest is defined by the size of the topography and roughness maps. Likewise this zone is surrounded by a buffer zone intended to distance the boundary conditions so as to minimally affect flow in the zone of interest. The formula to set the overall diameter (in meters) is

$$L = l \times 2^{0.5} + 1000 + 20000. \tag{2.29}$$

Here l corresponds to the diameter that is set by the area that the measurement masts cover. The mesh is constructed with three parameters: horizontal discretization, vertical discretization, and smoothing.

Horizontal discretization parameter. The refined zone of interest and the characteristic cell size in the horizontal directions are specified.

[6] ZephyTOOLS®, The Open-Source CFD Wind Farm Design Software, ©2012 Zephy-Science®

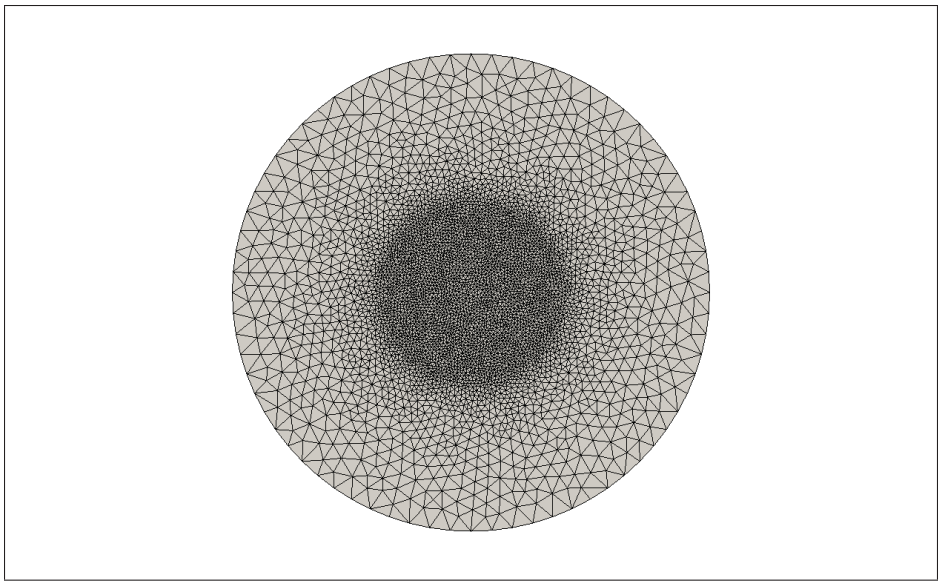


Figure 2.5 Top view of the mesh for the computational domain with a more refined zone in the center

Vertical discretization parameter. Three zones are defined: canopy, turbine, and total. These zones are depicted in figure 2.6. The canopy zone is attached to the ground and covers the forest region and in this zone the height of the first cell Δz is set; for the other zones, expansion coefficients are defined. Nevertheless, maximal vertical resolutions are set for the three vertical zones, which means that cells will increase their size according to the expansion coefficients until reaching the specified maximal value. In this work, the values are 4 m, 8 m and 500 m for canopy, turbine and top zone, respectively. The values for all these parameters are discussed as part of the grid independence study.

Smoothing parameter. A flat terrain smoothing near the lateral boundaries over the ground is utilized to improve mesh robustness.

2.3.4 Boundary conditions

The boundary conditions used in this study are summarized in table 2.4. The indexes are defined in figure 2.7, in which z_n, z_{n-1} and z_f represent the distance to the computational ground z_g.

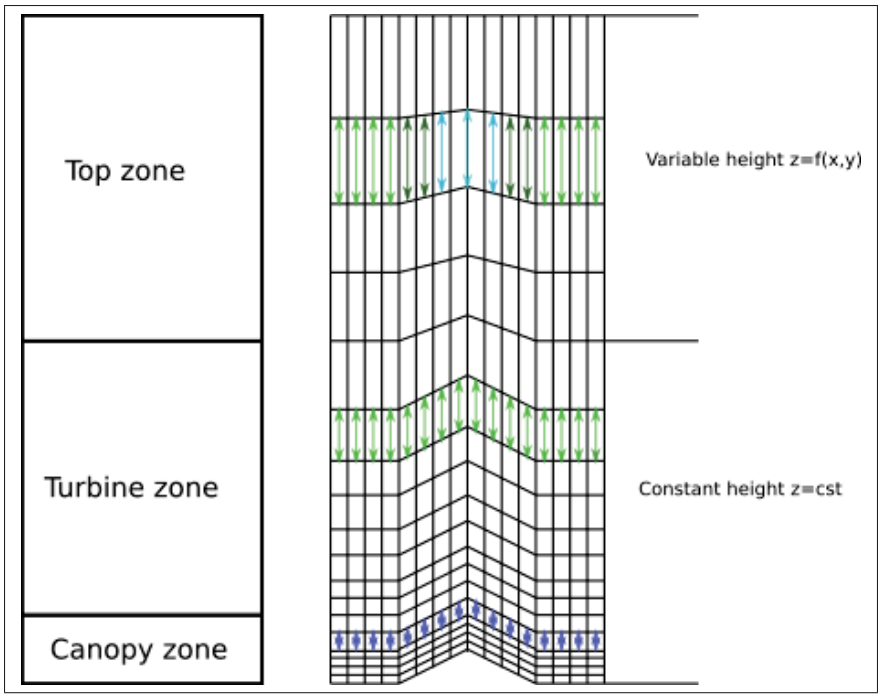


Figure 2.6 Vertical discretization

The boundary conditions attempt to agree with the guidelines of Richards and Hoxey (1993) which rely on the law of the wall for a sustainable atmospheric boundary layer (ABL) when using the $k-\varepsilon$ turbulence model (Richards & Hoxey, 1993). Nevertheless, it has been shown that due to difficulties to completely implement these boundary conditions in commercial codes only some of them are respected; for example, imposing the proper velocity and turbulence profiles at the inlet, and relying that the boundary layer will be maintained as energy is removed by the effect of the ground's shearing (Hargreaves & Wright, 2007). This results in a decay of the inlet velocity profile and an overestimation of the turbulent kinetic energy (TKE) in the cells attached to the wall (Hargreaves & Wright, 2007). In light of this, Hargreaves and Wright (2007) refined the implementation of these boundary conditions to adapt the law of the wall and the ground TKE production rate and to apply a constant shear stress at the top. These implementations improved the reproduction of a sustainable ABL in agreement with Richards and Hoxey (1993) and hence are also implemented in this study.

Table 2.4 Summary of boundary conditions

variable	boundary	condition
U	inlet	$U = \frac{u_*}{\kappa} \ln\left(\frac{z+z_0}{z_{0\text{inlet}}}\right)$
	outlet	$\frac{dU}{dn} = 0 \vee U = U_{fixed}$
	top	$U_f = U_n + \frac{u_*}{\kappa} \ln \frac{z_f}{z_n}$
	ground	$\lvert U \rvert = 0$
p	all	$\frac{dp}{dn} = 0$
k	inlet	$k = \frac{{u_*}^2}{\sqrt{C_\mu}}$
	outlet	$\frac{dk}{dn} = 0$
	top, ground	$\frac{dk}{dn} = 0$
ε	inlet	$\varepsilon = \frac{{u_*}^3}{\kappa\left(z+z_0\right)}$
	outlet, top	$\frac{d\varepsilon}{dn} = 0$
	ground	$\varepsilon_p = \frac{C_\mu^{3/4} k_p^{3/2}}{\kappa(z_p+z_0)}$
ν_t	inlet, outlet and top	$\nu_t = C_\mu \frac{k^2}{\varepsilon}$
	ground	$\nu_{t,p} = \frac{C_\mu^{1/4} k_p^{1/2} \kappa z_p}{\ln\left(\frac{z_p+z_0}{z_0}\right)}$

2.3.4.1 Inlet boundary

For U, a logarithmic velocity profile is given by imposing a constant homogeneous friction velocity u_* in equation (2.30). The value of u_* is set at 0.6 m/s. Originally, this value was intended to approximate the real value from the wind measurement of the highest anemometer of the real site. However, a linearity test will demonstrate the independence of speed-up factors from the value of the friction velocity. A different $z_{0\text{inlet}}$ from z_0 is defined to control the gradient of velocity to better match the desire velocity at the top boundary. This will be justified in the next chapter. For κ, 0.4 is used in all cases.

$$U = \max\left(0, \left(\frac{u_*}{\kappa} \ln\left(\frac{z+z_0}{z_{0\text{inlet}}}\right)\right)\right). \tag{2.30}$$

For p, zero gradient is set at all boundaries, and a pressure reference value is given in a reference cell for the SIMPLER algorithm. k and ε are specified according to Richards and

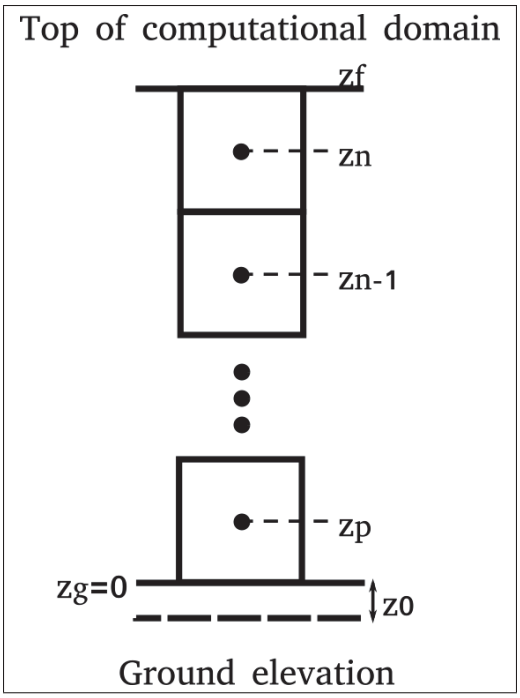


Figure 2.7 Cell notation inside domain. The lower boundary of the computational domain is offset from the ground z_g by z_0 in order to make $z_g = 0$ m

Hoxey (1993) assuming a neutral equilibrium surface layer:

$$k = \frac{{u_*}^2}{\sqrt{C_\mu}} \tag{2.31}$$

$$\varepsilon = \frac{{u_*}^3}{\kappa(z+z_0)}. \tag{2.32}$$

The turbulent viscosity ν_t is calculated at the inlet, outlet, and at the top from its definition:

$$\nu_t = C_\mu \frac{k^2}{\varepsilon}. \tag{2.33}$$

2.3.4.2 Outlet boundary

For U, an outflow condition is used that switches between a fixed value and a zero gradient depending on the direction of U. When flow goes inward, a fixed value $U_{fixed} = 0$ m/s is used; otherwise, a zero gradient is set in the normal direction of the face. For ε and k, a zero gradient condition is also used.

2.3.4.3 Top boundary

For U, a logarithmic profile in the last two cells in the vertical direction is assumed. Hence, the velocity at the top U_f is obtained by extrapolation:

$$U_f = U_n + \frac{u_*}{\kappa} \ln \frac{z_f}{z_n}. \tag{2.34}$$

For k and ε, zero gradient conditions normal to the boundary are set.

2.3.4.4 Ground boundary

For U, a non-slip condition $|U|$=0 is used. For k, a zero gradient is specified. For ε, an implementation in agreement with Hargreaves and Wright (2007) is developed, which consists in using equation (2.32) with the definition of u_{*g} at the ground as

$$u_{*g} = C_\mu^{1/4} k_p^{1/2} \tag{2.35}$$

where k represents the turbulent kinetic energy in the first cell p. Hence, the turbulent dissipation rate ε is defined as

$$\varepsilon_p = \frac{C_\mu^{3/4} k_p^{3/2}}{\kappa(z_p + z_0)}. \tag{2.36}$$

The use of this friction velocity (equation (2.35)) at the wall will satisfy the assumption of Hargreaves and Wright (2007) of local equilibrium between the production of kinetic energy G_k and the dissipation rates ($G_k = \varepsilon$) when k satisfies equation (2.31). The definition of G_k in equation (2.17) will be tailored to consider the viscous terms at the wall through the kinematic shear stress at the wall. The kinematic shear stress at the wall is defined either with the effect of the friction velocity or with the viscous terms

$$\tau'_w = (\nu_t + \nu)\frac{\partial U_i}{\partial x_j} = u_{*g}^2. \tag{2.37}$$

Hence, by considering the viscous terms in equation (2.37) and substituting them into equation (2.17), G_k at the wall is now defined as

$$G_{k,p} = \tau'_w \left(\frac{\partial U_i}{\partial x_j}\right). \tag{2.38}$$

For the velocity gradient, an approximation will be used through finite differencing between U_0 (velocity at z_0) and U_p (velocity at the cell's centroid p) (Sumner & Masson, 2012).

$$\frac{\partial U_i}{\partial x_j} \approx \frac{U_p - U_0}{z_p}. \tag{2.39}$$

Since the velocity should be zero at z_0 the velocity gradient is defined as

$$\frac{\partial U_i}{\partial x_j} \approx \frac{U_p}{z_p}. \tag{2.40}$$

To set U_p according to the logarithmic law equation, from its definition in equation (2.30), it is treated in accordance with Richards and Hoxey (1993) to satisfy the assumption that the kinematic shear stress is applied across the ground cell of height $2z_p$ (figure 2.7) so that the mean production rate across the cell will be considered (Hargreaves & Wright, 2007; Richards & Hoxey, 1993):

$$U_p = \frac{u_{*g}}{2\kappa}\ln\left(\frac{2z_p + z_0}{z_0}\right). \tag{2.41}$$

By substituting equation (2.41) into (2.40), the velocity gradient is defined as follows

$$\frac{\partial U_i}{\partial x_j} \approx \frac{u_{*g}}{2\kappa z_p} \ln\left(\frac{2z_p + z_0}{z_0}\right). \tag{2.42}$$

Now, from the definition of $G_{k,p}$ in equation (2.38) and adapting the velocity gradient from equation (2.42), $G_{k,p}$ is readjusted to

$$G_{k,p} = \frac{\tau_w' u_{*g}}{2\kappa z_p} \ln\left(\frac{2z_p + z_0}{z_0}\right). \tag{2.43}$$

Finally, by substituting equation (2.35) into equation (2.37) and by using both definitions of τ_w', the rate of production of k in the near-wall cell appears as follows

$$G_{k,p} = \frac{(\nu_t + \nu)\,\frac{\partial U_i}{\partial x_j} C_\mu^{1/4} k_p^{1/2}}{2\kappa z_p} \ln\left(\frac{2z_p + z_0}{z_0}\right). \tag{2.44}$$

For the turbulent viscosity $\nu_{t,p}$, from equation (2.37), the viscous terms are grouped with the friction velocity

$$\nu_{t,p} = \frac{u_{*g}^2}{(\frac{\partial U_i}{\partial x_j})} - \nu \tag{2.45}$$

where the value of the velocity gradient is taken from equations (2.40) and (2.30) for U to get

$$\nu_{t,p} = \frac{u_{*g}^2 \kappa z_p}{u_{*g} \ln\left(\frac{z_p + z_0}{z_0}\right)} - \nu. \tag{2.46}$$

Finally, by dividing the terms on the right side by ν, the turbulent viscosity is

$$\nu_{t,p} = \nu\left(\frac{y^+ \kappa}{\ln\left(\max(\bar{E}, 1)\right)} - 1\right) \tag{2.47}$$

where

$$y^+ = \frac{u_{*g} z_p}{\nu}$$
$$\bar{E} = \frac{z_p + z_0}{z_0}$$

and u_* is taken from equation (2.35).

2.3.5 Initialization and convergence

The initial value of U is set at 5 m/s, while for k and ε, they are calculated with equations (2.31) and (2.32) respectively. Convergence is satisfied when the normalized residuals are at least of the order $10e^{-7}$ or less.

CHAPTER 3

VALIDATION OF FOREST MODELLING

3.1 Introduction

In order to validate the canopy model three cases were tested and compared with the work of Dalpé and Masson (2008). Their work considered fully developed wind flow within and above a horizontally homogeneous dense forest. Dalpé and Masson (2008) compared three different kinds of trees based on the experimental measurements of Amiro (1990). For the present study, only the black spruce forest is used as validation which represents the dominant type of tree in the boreal forest. The characteristics of this forest are presented in table 3.1.

Table 3.1 Characteristics of the black spruce forest

h (tree height) [m]	LAI [-]	C_D [-]	z_0 [m]
10	9.19	0.15	0.05

The α distributions were previously shown for both specific and generic shapes in figures 2.3a and 2.3b, respectively. They have the same LAI value. In order to evaluate the influence of these shapes on the solutions, simulations performed for both α distributions and their results are compared. The objective is to evaluate the effect of the generic assumption as usually only the forest heights and a general notion of the forest type are known.

The cases consist of an identical 2D replication of Dalpé and Masson (2008) with cyclic inlet-outlet boundary conditions; a 2D replication of Dalpé and Masson (2008) with non-cyclic boundary conditions; and an equivalent 3D flat terrain case that has similar domain dimensions as the real site. For all cases, the U and k results are presented in a normalized manner by taking reference values at a height of $z/h = 1.21$. These results are plotted for the first 30 m and the complete domain height for each case in figures 3.2, 3.8 and 3.11, respectively.

3.2 2D case with cyclic boundary conditions

This domain consists of a 100-m long by 800-m high rectangle composed of 10 columns and 192 rows with a height for the first cell of $\Delta_z = 0.03$ m and an expansion coefficient of 1.036 (figure 3.1). The boundary conditions are the same as previously explained in table 2.4, with the caveat that for the inlet/outlet boundaries a cyclic condition is used. Additionally, by considering that the friction velocity is constant for a neutrally stratified surface layer (Stull, 1988), a constant u_* is utilized at the top. This differs from the treatment of Dalpé and Masson where u_* is calculated at each iteration at the top boundary. At the ground, a slip condition is set for all variables to emphasize the drag effect of the forest.

3.2.1 Results

The results are sampled at a position near the outlet and presented in figure 3.2. It can be seen that for the first 30 meters the U and k profiles with the specific *LAI* density distribution exactly fit the results of Dalpé and Masson (2008), while with the generic *LAI* distribution the profiles are slightly offset. U at the top for the specific *LAI* density distribution deviates slightly from the results of Dalpé and Masson because they use a fixed value for k and ε at the top while a zero-gradient condition is used in this work.

3.3 2D case with inlet-outlet boundary conditions

This case aims to reproduce the results of Dalpé and Masson by using the complete set of boundary conditions previously described in table 2.4. The domain of the preceding case can not be used here because its limited size is not practical for real sites. This impracticality arises mainly because real sites need to approximate fully-developed conditions. Therefore, for the present case an elongated domain in the x-direction is used. The domain consists of a rectangle 40000-m long by 800-m high composed of 1000 columns and 192 rows with the same height for the first cell of $\Delta_z = 0.03$ m and with the same expansion coefficient of 1.036 (see figure 3.3). Furthermore, since the previous case uses cyclic conditions at inlet-outlet

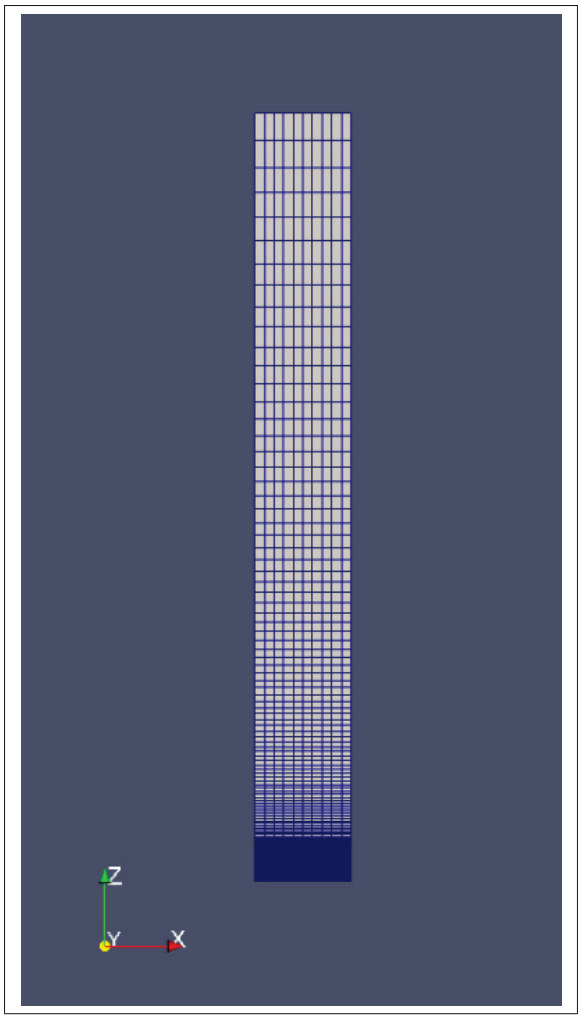


Figure 3.1 Mesh used for the 2D case with cyclic boundary conditions

boundaries, the values of u_* and $z_{0\text{inlet}}$ are irrelevant. But for this 2D case with defined inlet-outlet boundary conditions, two sensitivity analyses on u_* and $z_{0\text{inlet}}$ (see equation (2.30)) were conducted in order to provide the most similar fully-developed flow at the inlet as the one of Dalpé and Masson (2008). Specifically, it is expected to obtain the same average velocity gradient as the fully-developed solution (between the velocity at the ground and the velocity at the top). These following analyses on u_* and $z_{0\text{inlet}}$ aim to identify the parameter that modifies this gradient.

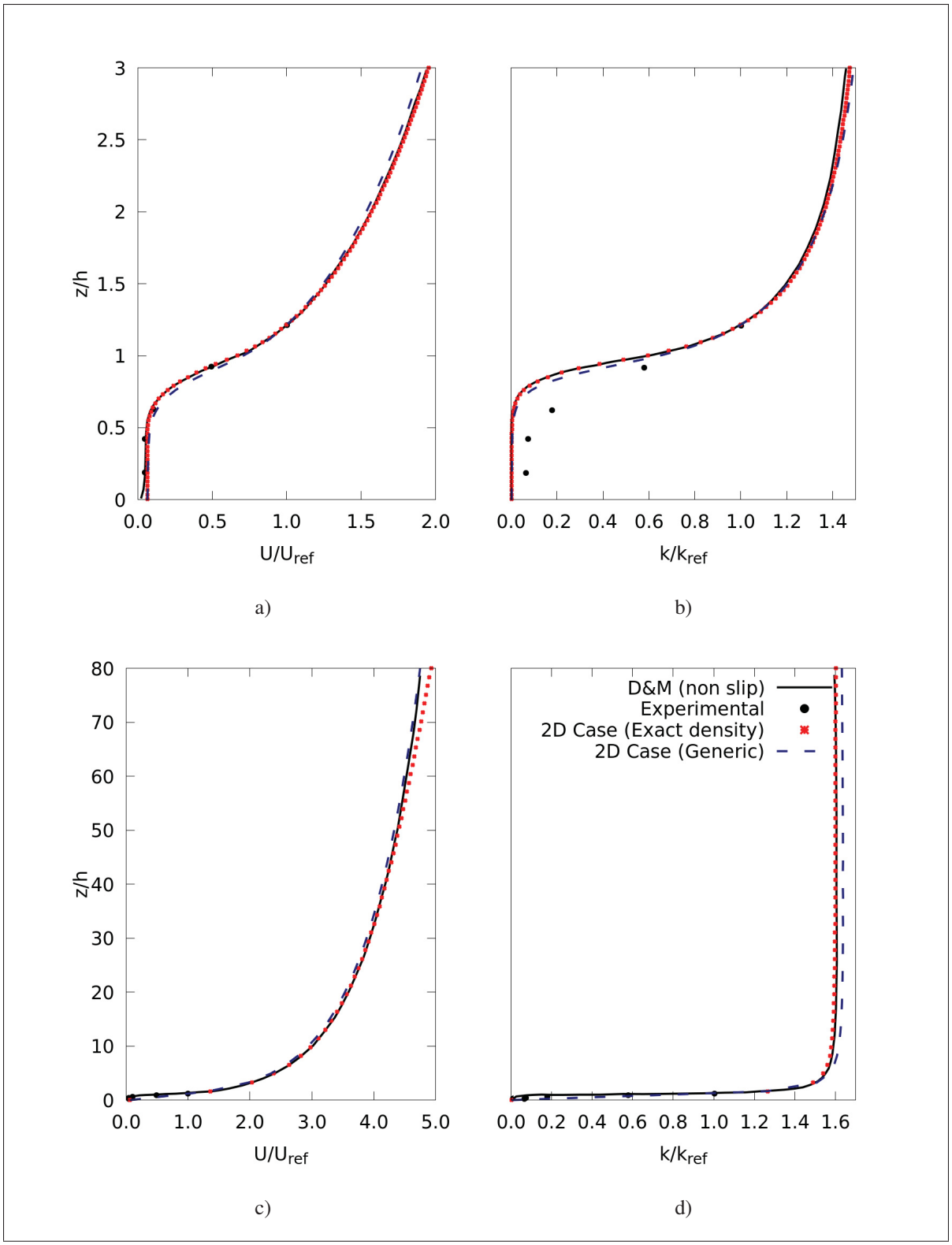
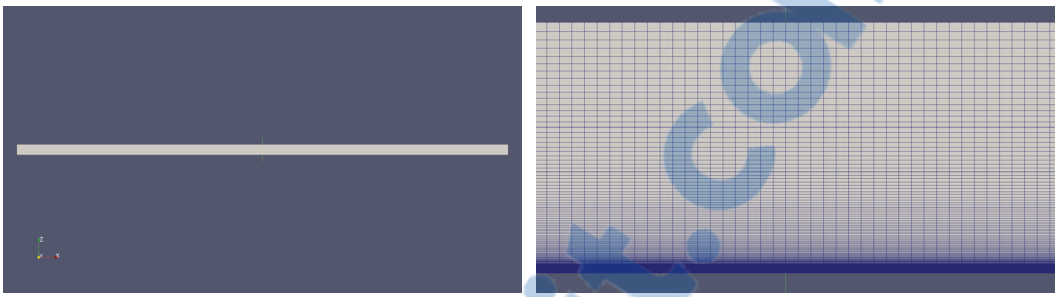


Figure 3.2 U and k distributions for 2D case with cyclic boundary conditions up to 30 and 800 m. Values are normalized at z/h=1.21



a) Elongated domain

b) Same domain with zoom to visualize the structure of the cells

Figure 3.3 Mesh used for the 2D case with inlet-outlet boundary conditions

3.3.1 Sensitivity analysis for friction velocity

It was previously mentioned that all cases will use a value of $u_* = 0.6$ m/s, which roughly agrees with *in situ* measurements. Therefore, for the u_* sensitivity analysis a range from 0.4 to 1.0 m/s is used. With $z_0 = 0.05$ m and $z_{0\text{inlet}} = 0.8$ m, the results of this study are presented in dimensional and non-dimensional forms in figures 3.4 and 3.5 respectively.

The dimensional results present the profiles of U, k, ε and ν_t on a logarithmic scale for the first 100 m (sub-figures 3.4a to 3.4d) and for the complete domain of 800 m height (sub-figures 3.4e to 3.4h). The profiles are presented at two positions per u_* value: at inlet and outlet. Additionally, the results of the preceding 2D case with cyclic boundary conditions is shown. The non-dimensional results present the same profiles normalized by their reference value at a height of $z/h = 1.21$ for the first 30 m (sub-figures 3.5a to 3.5d) and for the complete domain of 800 m height (sub-figures 3.5e to 3.5h). The profiles are presented at the outlet position and they are compared with the results of the 2D case with cyclic boundary conditions and with the results from Dalpé and Masson (2008).

The effect of u_* is clearly seen in the dimensional results (figure 3.4), especially in the tendency of the outlet results to match the inlet profiles. Looking at the normalized results (figure 3.5) to compare them with the published ones of Dalpé and Masson (2008a), this study reveals the

independence to u_*. This makes sense since u_* only changes the magnitude of the velocity profile and not its shape.

3.3.2 Sensitivity analysis for z0 inlet

In an effort to modify the velocity gradient to get better agreement with the fully-developed solution, the effect of $z_{0\text{inlet}}$ is also tested. A range from 0.8 m to (an exaggerated) 8.0 m is used for the $z_{0\text{inlet}}$ sensitivity analysis. With $z_0 = 0.05$ m and $u_* = 0.6$ m/s, the results of this study are presented in dimensional and non-dimensional values in figures 3.6 and 3.7, respectively.

The dimensional results present the profiles of U, k, ε and ν_t on a logarithmic scale for the first 100 m (sub-figures 3.6a to 3.6d) and for the complete domain of 800 m height (sub-figures 3.6e to 3.6h). The profiles are presented at two positions per $z_{0\text{inlet}}$ value: at inlet and outlet. The results of the 2D case with cyclic boundary conditions are also shown. The non-dimensional results present the same profiles normalized by their reference value at a height of $z/h = 1.21$ for the first 30 m (sub-figures 3.7a to 3.7d) and for the complete domain of 800 m height (sub-figures 3.7e to 3.7h). The profiles are presented at the outlet position and they are compared with the results of the 2D case with cyclic boundary conditions and with the results from Dalpé and Masson (2008).

The effect of $z_{0\text{inlet}}$ is clearly distinguishable in the dimensional results (figure 3.6). And this time the normalized results (figure 3.7) are affected by the value of $z_{0\text{inlet}}$. At higher values they agree better with the results of Dalpé and Masson (2008).

3.3.3 Results

The values of either u_* and $z_{0\text{inlet}}$ have no significant effect on the normalized results in the vicinity of the forest, but they do at higher altitudes approaching the top boundary. Specifically, the different results at the top stem from the choice of $z_{0\text{inlet}}$. Therefore, given the results of these analyses, an exaggerated $z_{0\text{inlet}}$ at the inlet of 12.2 m is used for this last case (value taken

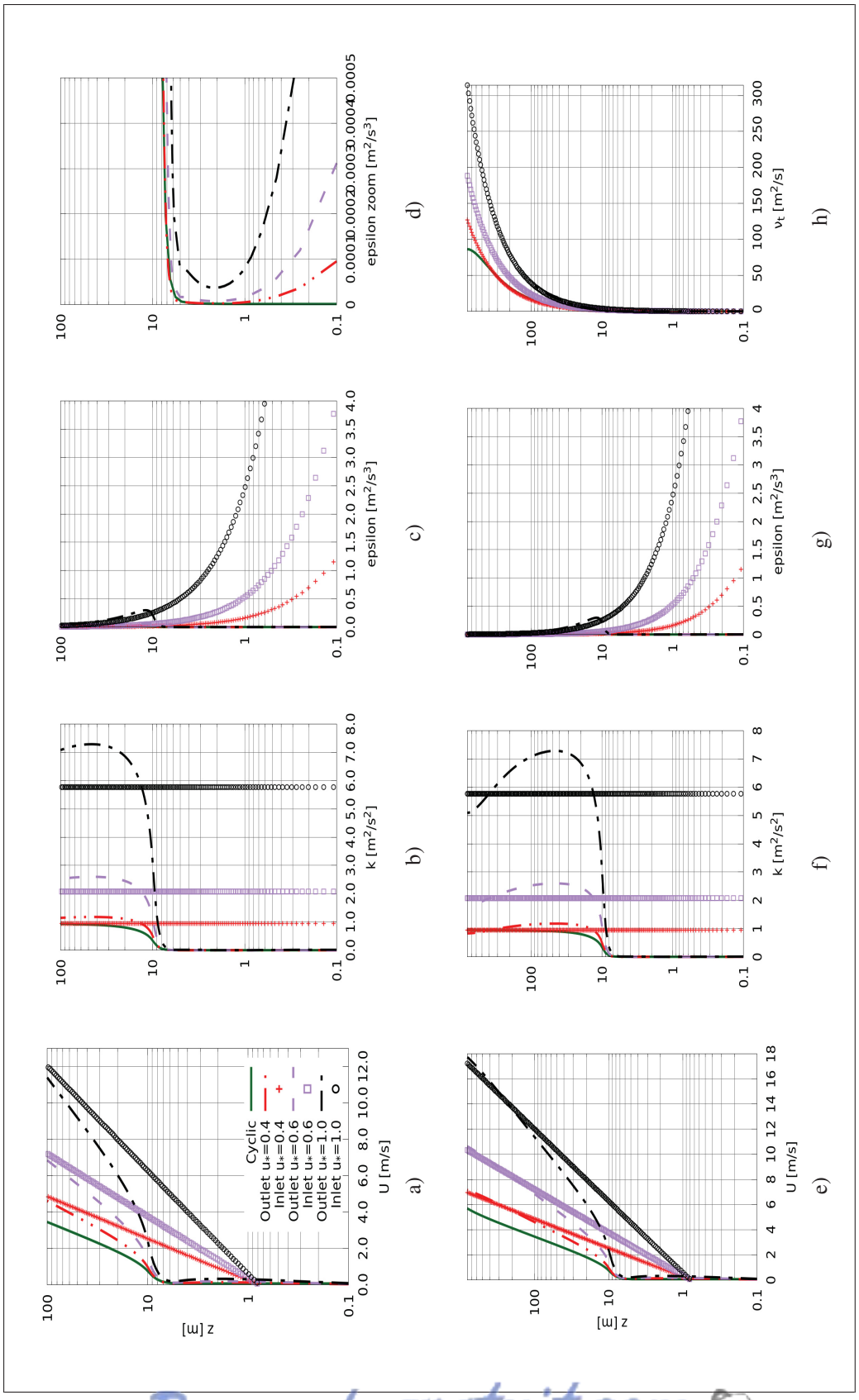


Figure 3.4 Dimensional distributions of U, k and ε for different values of u_* for the 2D case with inlet-outlet boundary conditions up to 100 and 800 m

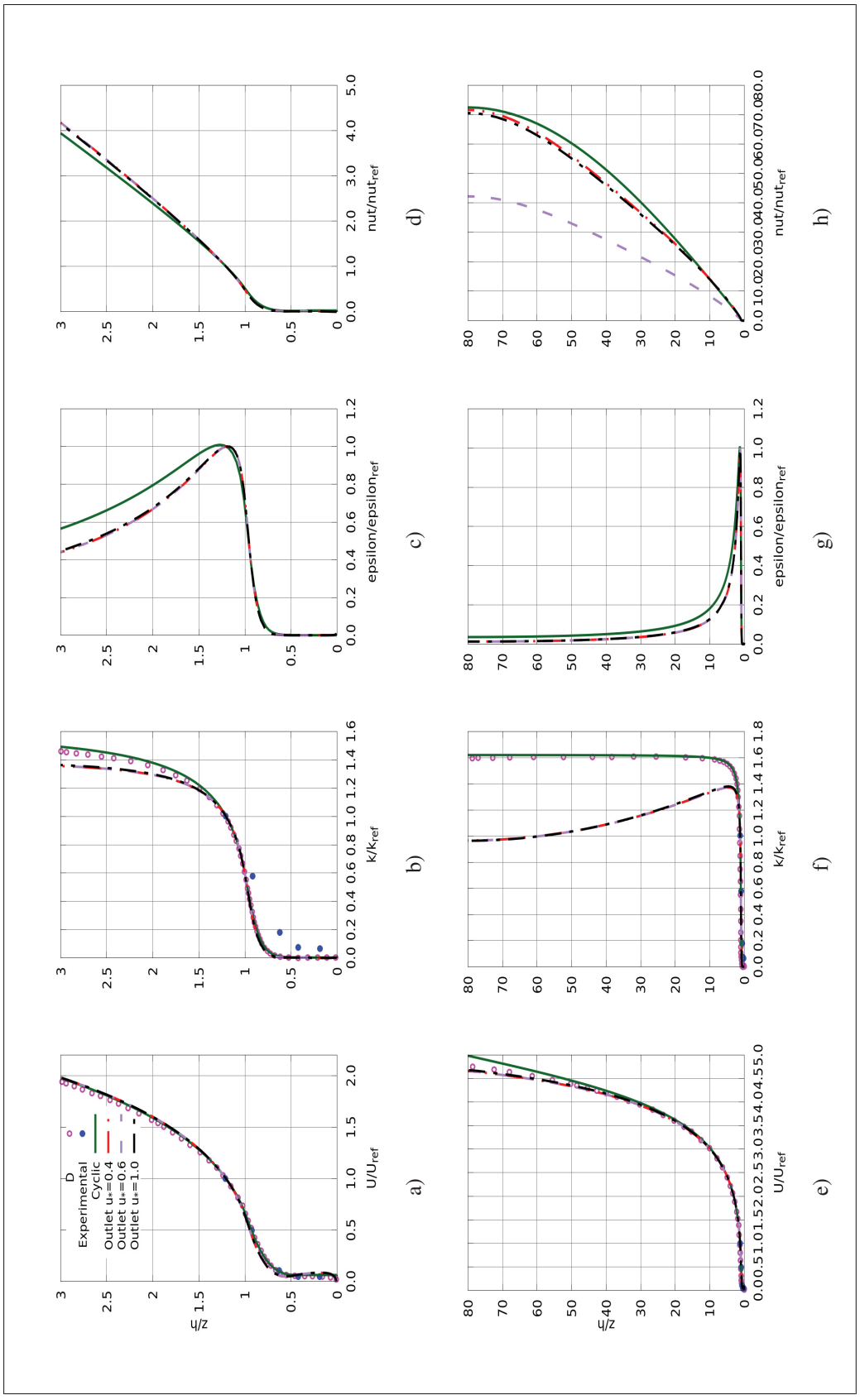


Figure 3.5 Non-dimensional distributions of U, k, ε and ν_t for different values of u_* for the 2D case with inlet-outlet boundary conditions up to 30 and 800 m

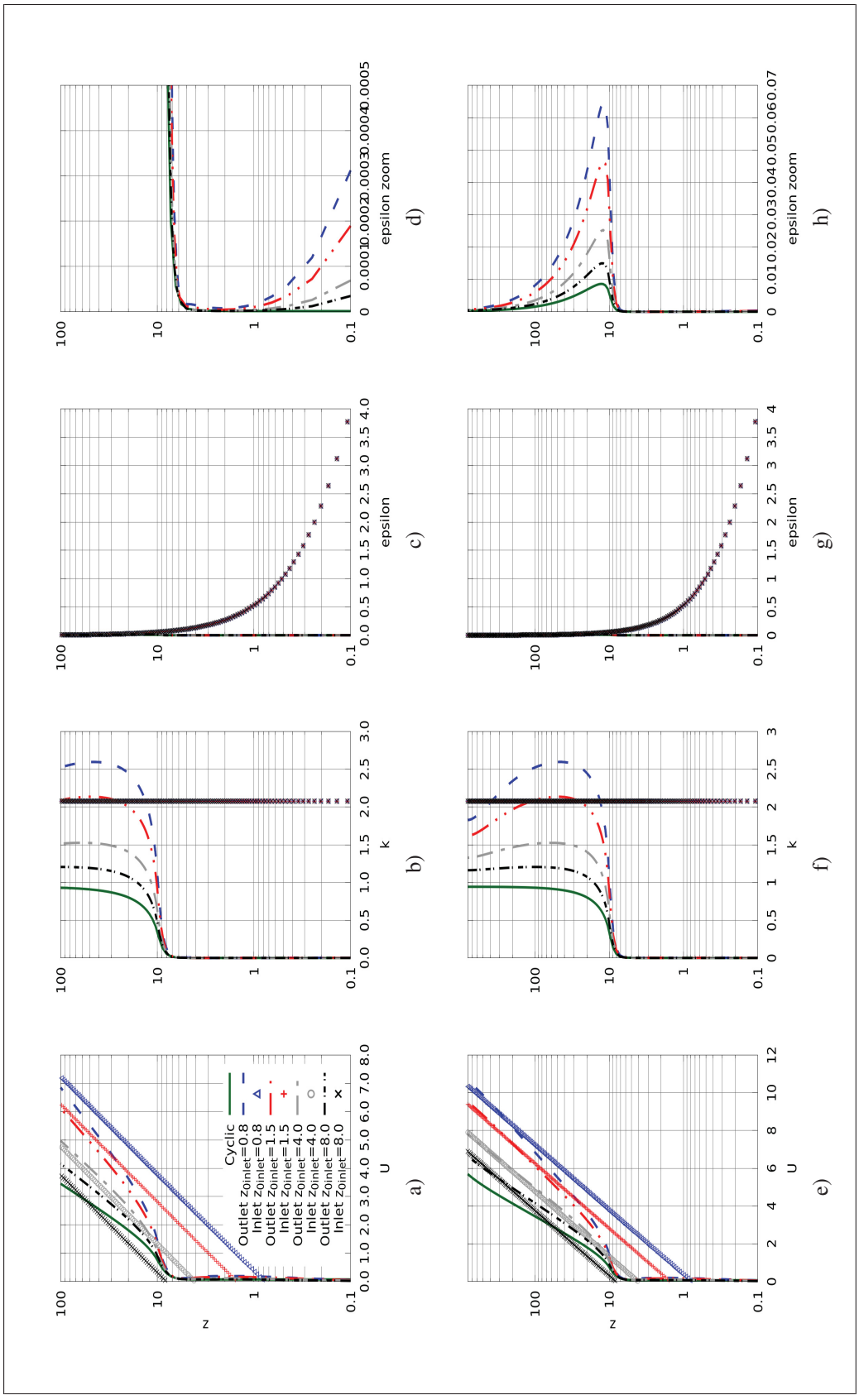


Figure 3.6 Dimensional distributions of U, k and ε for different values of z_0 for the 2D case with inlet-outlet boundary conditions up to 100 and 800 m

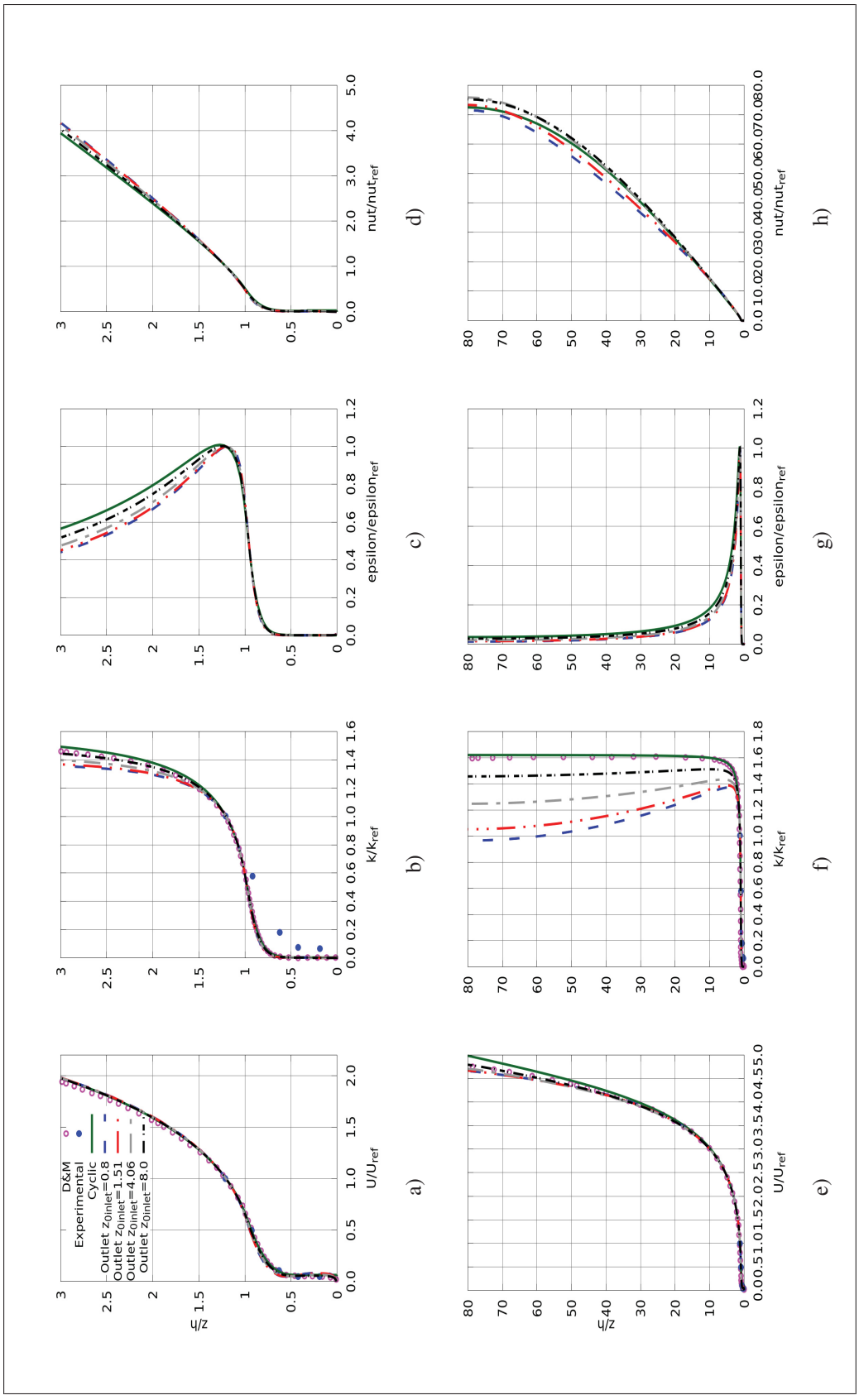


Figure 3.7 Non-dimensional distributions of U, k, ε and ν_t for different values of z_0 for the 2D case with inlet-outlet boundary conditions up to 30 and 800 m

as the same as the reference height) which gives the best agreement with the fully-developed flow of Dalpé and Masson (2008). The results are presented at a position near the outlet in figure 3.8 and they are in almost exact agreement with (Dalpé & Masson, 2008) for both specific and generic *LAI* density distributions.

3.4 Flat terrain 3D case

This case consists of a 32 000-m diameter cylinder that is 2 500-m high with all the boundary conditions previously described in table 2.4. The criteria for choosing the value of the first cell $\Delta_z = 0.03$ m and the expansion coefficient of 1.04 is found in the work of Dalpé and Masson (2008a). Different views for this mesh are shown in figure 3.9. Furthermore, since this domain is bigger than the one of previous case, the value of $z_{0\text{inlet}} = 12.2$ m is no longer used, mainly because it was tested and the results presented a smaller mass flow rate, translated in overpredicted results of k. Therefore, a smaller $z_{0\text{inlet}}$ with a value of 8.0 m is used for this case. Although the results will still overpredict the value of k, it was decided to remain with this value; keeping in mind that for the real site, a value of $z_{0\text{inlet}} < 8.0$ m and $z_{0\text{inlet}} > z_0$ m is recommended to be used.

3.4.1 Grid independent solution

A grid independence study was first conducted; the three meshes analyzed are shown in table 3.2. Since *ZephyTOOLS* requires the coordinates of the met masts to define the refined center diameter, five contrived masts were set, which resulted in a relatively small refined center diameter of 2 000 m. This value was used for mesh *i*, but for the others, this value was manually increased to approximate the diameter used in the real terrain case. The results for the speed-up factor S (wind speed normalized by its value at $z/h = 1.21$) are sampled at the center of the domain at three different heights, these values were compared with the results of Dalpé and Masson (2008) and the relative errors are shown in table 3.3. When the velocity profiles of each mesh were reviewed, the importance of increasing the size of the refined center was confirmed.

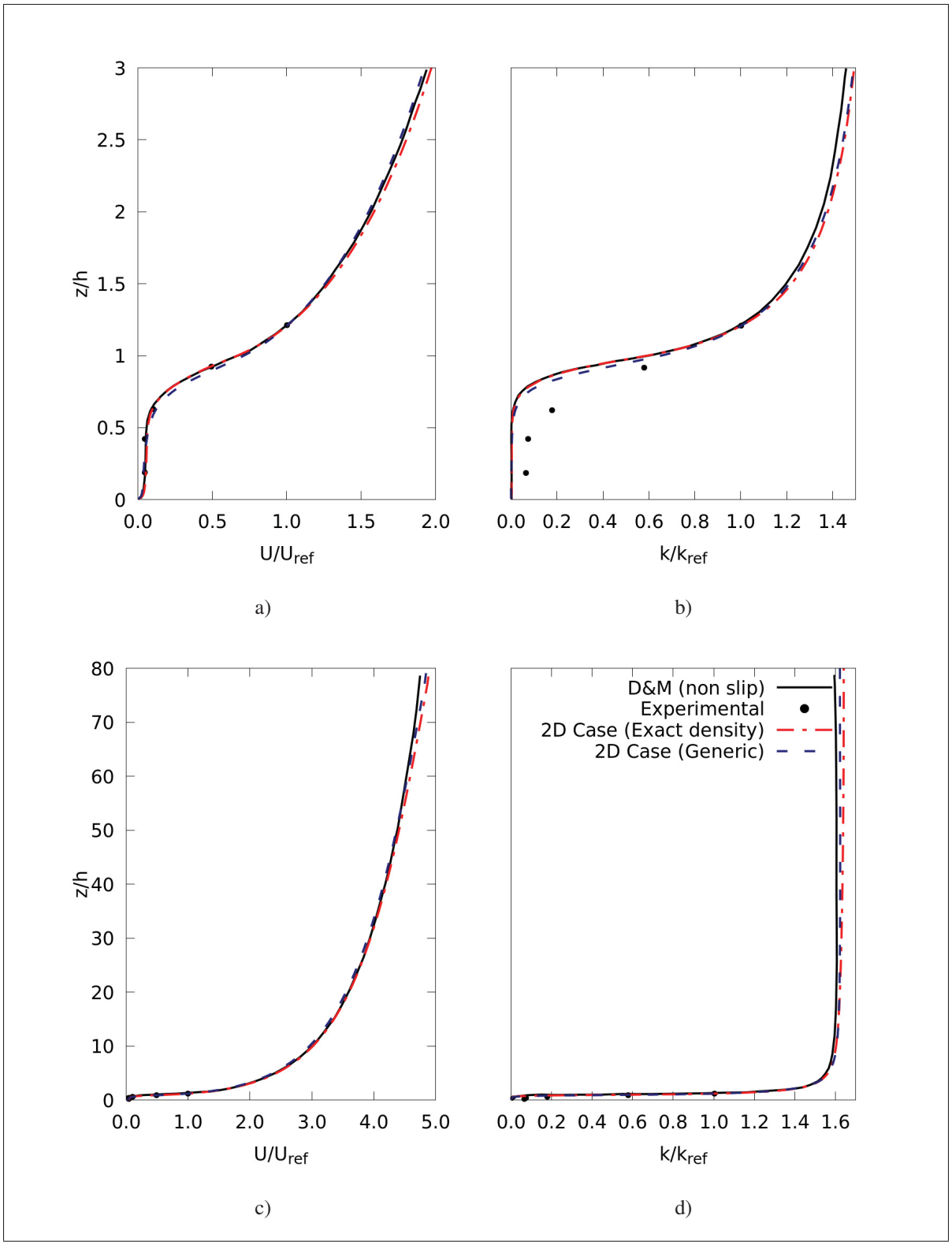
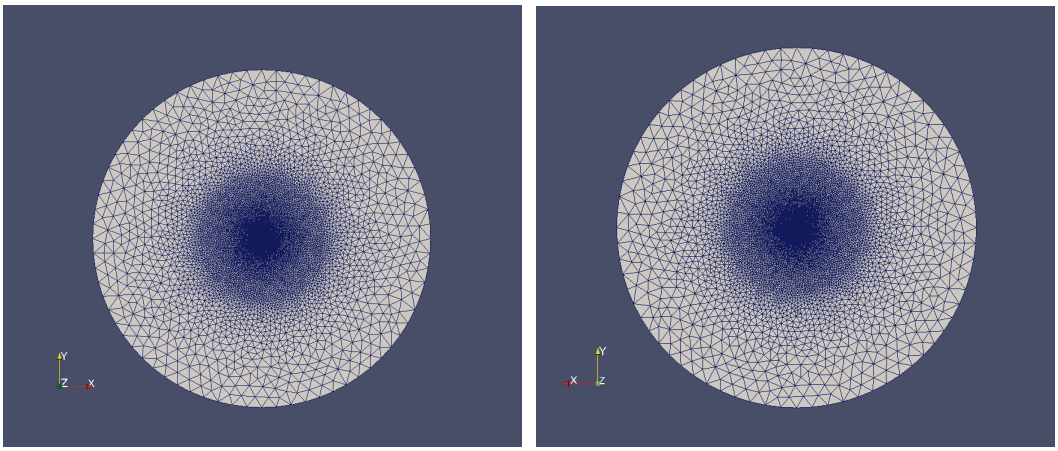
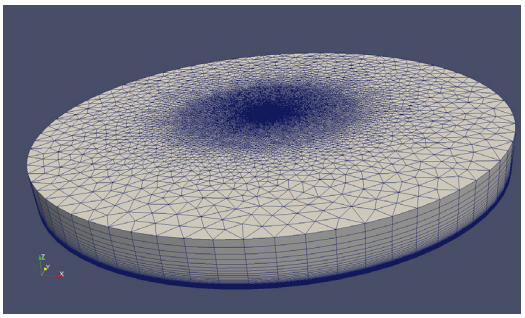


Figure 3.8 U and k distributions for 2D case with inlet-outlet boundary conditions up to 30 and 800 m. Values are normalized at z/h=1.21



a) Top view

b) Ground view



c) Perspective view

Figure 3.9 Mesh used for the flat terrain 3D case

Comparing meshes *ii* and *iii*, grid independence is shown through the minimal difference in errors.

Table 3.2 Three different numerical meshes for the 3D flat terrain case

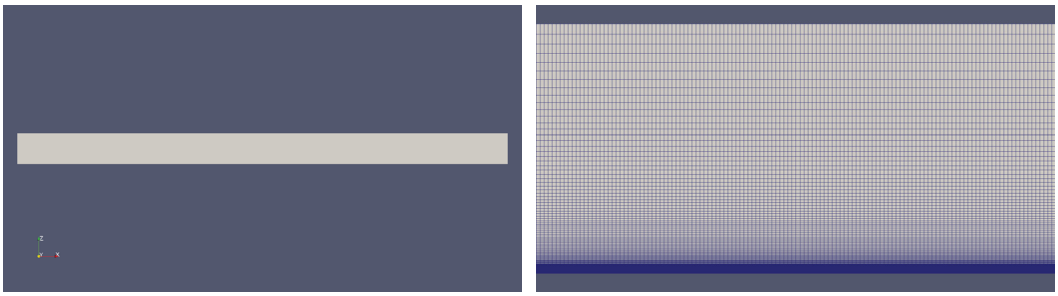
Mesh	Grid cells	Refined diameter [m]	Δ_x [m]	Δ_{zmin} [m]	Expansion coefficient		
					Canopy	Turbine	Top
i	600 930	2 000	80	0.03	1.04	1.04	1.2
ii	2 709 300	12 000	80	0.03	1.04	1.04	1.2
iii	4 708 692	12 000	60	0.03	1.04	1.04	1.24

Table 3.3 Grid independent solution for the 3D flat terrain case used for model validation. The errors in speed-up factor are shown for three different heights

Position	Dalpé and Masson S (Dalpé & Masson, 2008)	Error		
		mesh i	mesh ii	mesh iii
z/h=1	0.66	3.96%	2.76%	2.76%
z/h=2	1.57	1.05%	0.77%	0.76%
z/h=3	1.94	1.54%	1.21%	1.19%

3.4.2 Analogous 2D case

Since the results of Dalpé and Masson (2008) are for a smaller domain, this case was reproduced with a much higher domain for comparison purposes. The case consists of a 2D rectangle with all the boundary conditions in table 2.4 and with $z_{0\text{inlet}} = 8.0$ m. Its dimensions are 40 000-m long by 2 500-m high with 1000 columns and 192 rows with the height for the first cell of $\Delta_z = 0.03$ m and with an expansion coefficient of 1.036 (see figure 3.10).



a) Completed domain

b) Same domain with zoom to visualize the structure of the cells

Figure 3.10 Mesh used for the analogous 2D case

3.4.3 Results

For the 3D case, four results are plotted: at the center and outside the refined zone (16 000 m from the center of the domain) for both the specific and generic *LAI* density distributions. For

the 2D case, two results are plotted: at the center and near the end (30 000 m) of the domain for the specific *LAI* density distribution. The results are shown in figure 3.11. From the previous analysis, it is clear that a large value of $z_{0\text{inlet}}$ helps to approach the fully-developed flow inlet condition and thus yields better overall results. Since no comparison is available for a higher domain, the validation of the 3D case aims to better agree with the results of the analogous 2D case than with the original results of Dalpé and Masson (2008).

Velocity. In the first 30 meters (figure 3.11a) all results show good agreement with the results of the Dalpé and Masson model. And, as expected, the 3D results with the generic *LAI* distribution present the largest deviation. For larger heights (figure 3.11c), good results are found for all cases, although the generic distribution gives a slight underestimation for most of the domain.

Turbulent kinetic energy. All results in the first 30 m (figure 3.11b) show good agreement with an overestimation towards the top compared to the results of the Dalpé and Masson model. The 3D results with the specific *LAI* density distribution agree well with their 2D case counterpart. For the whole profile (figure 3.11d) the overestimation continues compared to the results of Dalpé and Masson model. Moreover, the results agree well with their counterpart in the 2D case.

In general, the results show the importance of using the more accurate *LAI* integration to better fit the wind shear. However, the generic α distribution allows preliminary studies to focus on the C_D and *LAI* parameters in an effort to improve accuracy of wind flow predictions over forested terrain. Additionally, in order to obtain more accurate results it is crucial to set as inlet condition a fully-developed flow that can be adjusted with different values of $z_{0\text{inlet}}$.

With the results of this validation study, the canopy model is ready to be used for a real site which is presented in the next chapter.

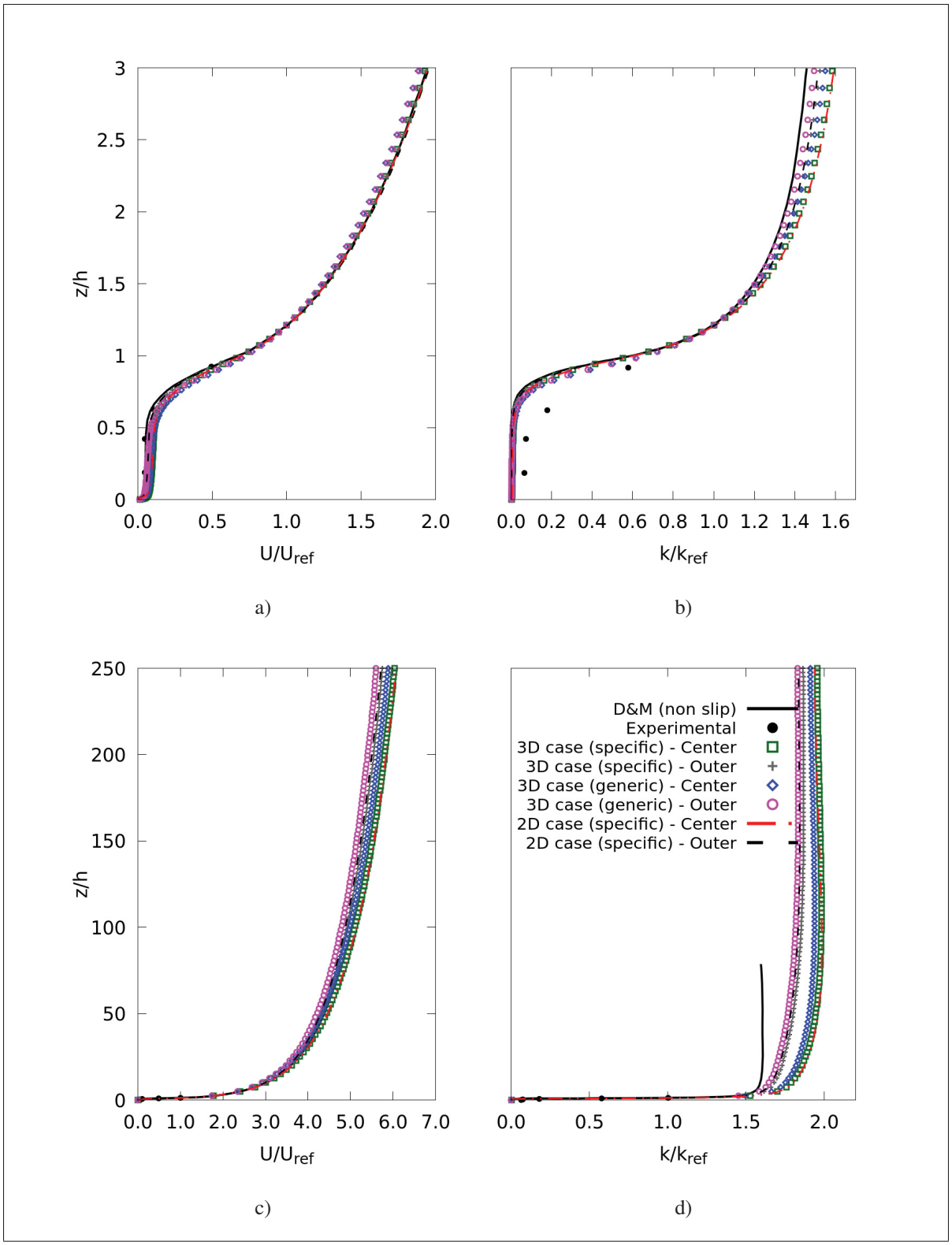


Figure 3.11 U and k distributions for 3D flat terrain case up to 30 and 2 500 m. Values are normalized at z/h=1.21

CHAPTER 4

WIND FLOW MODELLING OVER A REAL SITE

With the canopy model validated, the present and final chapter addresses two of the specific objectives of this master's thesis:

1. To quantify the accuracy of the canopy model by comparing simulated results with experimental data in terms of speed-up factors S. This is done by using the canopy model to simulate the wind flow over a real site and comparing the results with two years of measurement data. Furthermore, models that do not take into account the turbulence within the forest will be simulated and compared.

2. To evaluate the impact of RANS wind flow modeling uncertainty on capacity factor uncertainty. This is carried out by calculating the uncertainties described in the literature review using the simulation results and the field measurements.

4.1 Case study

4.1.1 Site description and instrumentation

The site is located in Quebec, Canada, and because it belongs to a private wind energy developer, its specific location is withheld for confidentiality reasons. Still, the characteristics that are important for this study can be mentioned. It is positioned at high elevation, expecting with this good wind speeds and to be attractive for wind energy purposes. The site is moderately complex as can be seen in the discretized form (elevation normalized by the smallest altitude) in figure 4.1. The site is mainly covered with forest with a slow descent to a plain towards the west, which is the dominant wind direction as shown in the windrose in figure 4.2. The wind direction frequency of all masts show very similar distributions; thus, the distribution of only mast M3 is shown in figure 4.2. The three anemometers of each of the three masts (M1, M2,

M3 in figure 4.1) were taken into consideration; the instruments have a measurement precision of ± 0.1 m/s. As the work considers different wind directions, the top wind vanes were also used which have a measurement precision of $\pm 3°$.

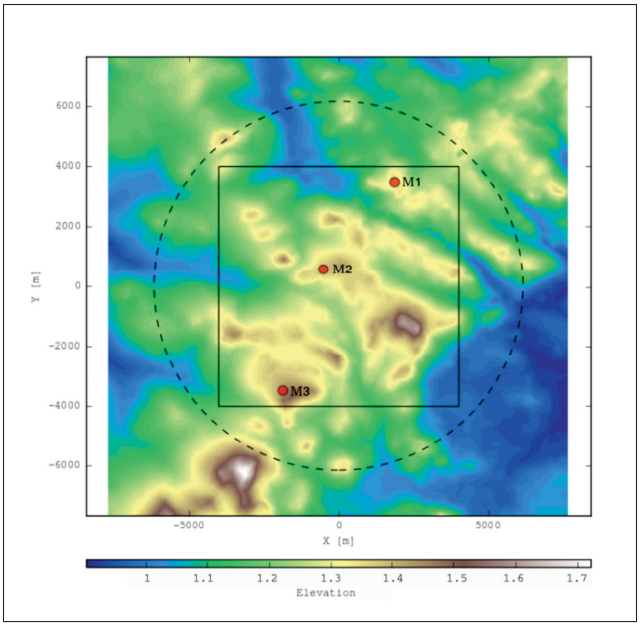


Figure 4.1 Discretized elevation for the zone of interest. The elevation is normalized by the smallest altitude

4.1.2 Data treatment

The site is equipped with three masts that have a common operating time of two years from which the wind speed time series are taken. The mean velocities are extracted from the primary anemometers at each height (30 m, 45 m, 58 m) by appending the reconstituted wind speeds for each height to a single file. Then a subset of wind speed data is exported with the wind coming from $270° \pm 5°$. In order to compare the experimental results to the simulations, the speed-up factors at M1 and M2 are calculated with equation (1.3) by using M3 as a reference mast.

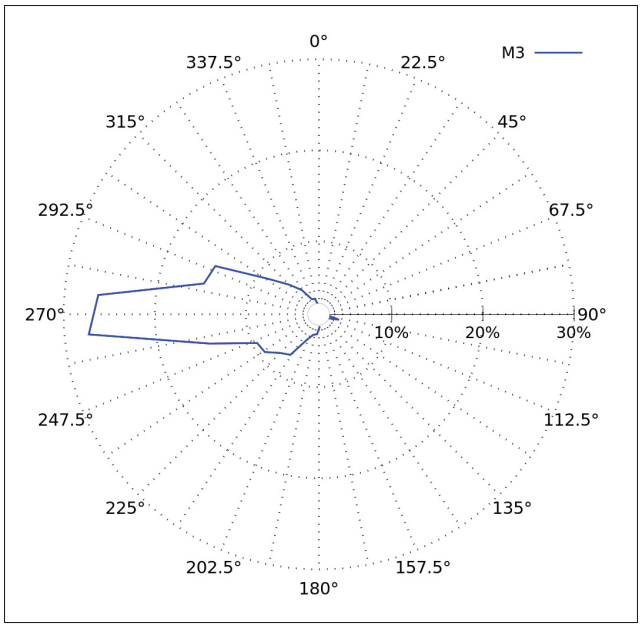


Figure 4.2 Windrose showing clearly a predominant wind from western direction

4.2 Simulations setup

4.2.1 Model cases

Four model cases with different forest treatments are simulated and their results compared with measurements in terms of speed-up factor. These cases are designated as follows:

(A) Terrain only case. It will represent the wind flow over the terrain neglecting the presence of forest. This case will be used mainly to demonstrate the importance of considering the forest effect;

(B) Displacement height case. It will elevate the terrain by an amount proportional to the average tree height implicitly assuming the wind flow starts at this location;

(C) Uniform canopy case. It will resolve the turbulence within the forest assuming a uniform forest distribution;

(D) Non-uniform canopy case. It will resolve the turbulence within the forest for the actual forest distribution of the site.

These cases are summarized in table 4.1. The simulations are carried out for the 270° dominant wind direction as previously justified by the windrose in figure 4.2. The kind of wind speed predictions expected by each model are illustrated in figure 4.3.

Table 4.1 Summary of main characteristics of the four model cases under study

Case	Model	Turbulence closure	Logarithmic wind speed profile trough
A	Terrain only	Standard (Apsley & Castro, 1997)	No obstacles
B	Displacement height (DH)		No obstacles and terrain elevated
C	Canopy	Modified (Dalpé & Masson, 2008)	Uniform forest distribution
D			Real forest map distribution

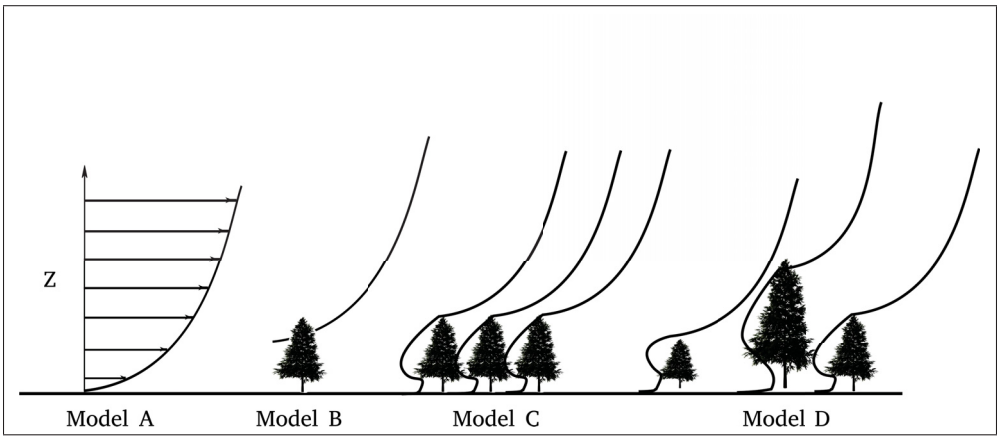


Figure 4.3 Wind speed profile expected for each model. Figure in analogy with Stull (1988)

4.2.2 Pretreatment of roughness and topography

The pretreatment of roughness and topography will be summarized for the current model cases (the detailed procedure is found in Appendix I). The pretreatment starts by obtaining the original map files of roughness and topography; for this study they were obtained from the website

Géoboutique Québec (Gouvernement du Québec, 2017). Specific treatments are then performed for each map and model case.

Roughness. Since cases *A* and *B* share the characteristic of not resolving the turbulence within the forest, they require a moderately high z_0 value (as shown in table 2.1). Therefore, they keep the original z_0 distribution with a predominant value of $z_0 = 0.8$ m as seen in figure 4.4a. Cases *C* and *D*, with canopy models that resolve the turbulence within the forest, assume a smaller z_0. Case *C* uses a uniform value of $z_0 = 0.05$ m for the whole domain, *i.e.* the value that agrees with the black spruce forest parameters. Case *D* uses $z_0 = 0.05$ m in the presence of forest with a height between 10-m and 15-m, and the z_0 distribution of cases *A* and *B* for the rest of the domain (figure 4.4b).

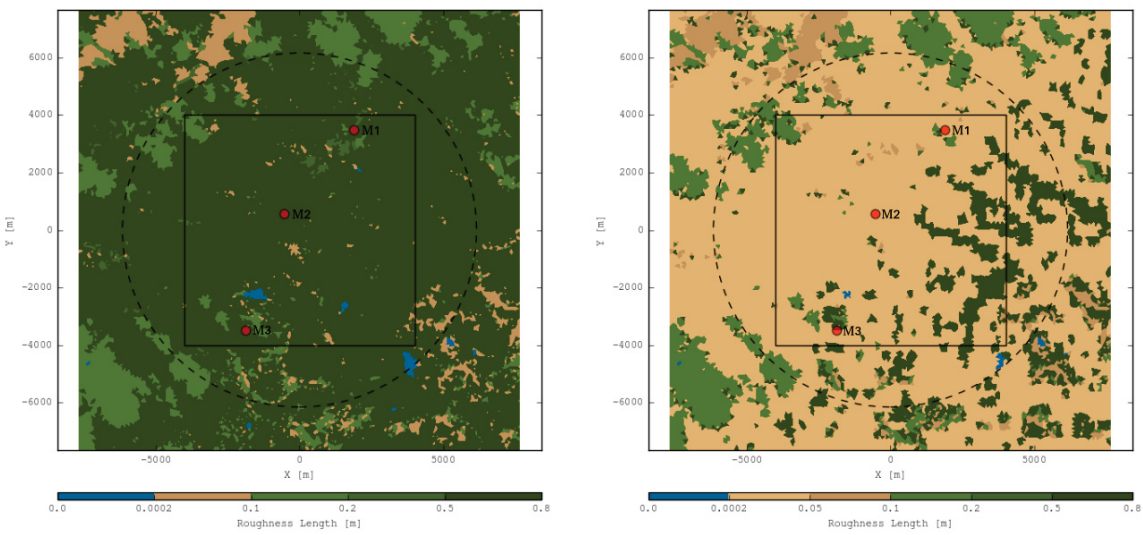
Topography. Cases *A*, *C*, and *D* elevate the ground altitude by the values of their corresponding z_0 distributions. Case *B* first elevates the ground by 5.36 m as a uniform value of d (particularly for this site) and then adds the z_0 distribution.

4.2.3 Boundary conditions and initialization

The boundary conditions used for all cases are those described in the methodology chapter and that are summarized in table 2.4. Likewise, the same initialization parameters are respected: U was set at 5 m/s; k and ε were calculated with equations (2.31) and (2.32) respectively; and ν was set to 1.4e-05 m^2/s.

4.2.4 Mesh

With a domain 32 000 m in diameter and 4 000 m high, each case uses a different mesh which is justified by its grid independent solution. The parameters of theses final meshes are summarized in table 4.2. For cases *C* and *D*, the parameters for the forest correspond to the black spruce used in the canopy validation section *i.e.* $LAI = 9.19$ and $C_D = 0.15$. The difference between these canopy cases lies in the use of the tree heights and the α distribution. For case *C*, a uniform tree height of 10 m distributed over the whole domain is assumed and



a) Roughness distribution for cases A and B. The value of z_0 designed to the forest is of 0.8m

b) Roughness distribution for case D. In the presence of forest (10m$\leq$ tree height $\leq$15m) $z_0 = 0.05$m is set

Figure 4.4 Roughness distribution

the "exact" α distribution (figure 2.3a) from Dalpé and Masson (2008) is used. For case *D*, the actual forest map was taken into account in which the heights ranged from 10 m to 15 m as shown in figure 4.5. In addition, the generic α distribution (figure 2.3b) is used. Different values of z_0 are adopted depending on the case model as previously described: for cases *A* and *B*, z_0 is based on figure 4.4a, for case *C* a uniform z_0 of 0.05 m, and for case *D* z_0 is from figure 4.4b. Notably, the dominant value for cases *A* and *B* is $z_0 = 0.8$ m and for cases *C* and *D* it is $z_0 = 0.05$ m.

Furthermore, the turbulence intensity (TI) at the inlet is shown. TI represents a characteristic of the flow rather than a boundary condition. TI is defined as the ratio of the standard deviation of the wind speed to its mean (Manwell *et al.*, 2009) and is based on the turbulent kinetic energy *k*:

$$TI = \frac{\sigma}{\overline{U}} \cong \frac{\sqrt{\frac{2}{3}k}}{\overline{U}}. \tag{4.1}$$

Therefore, by combining equations (2.30) and (2.31), and substituting them into equation (4.1), the TI at the inlet is deduced as:

$$TI = \frac{\sqrt{\frac{2}{3}}}{C_\mu^{1/4}} \frac{\kappa}{\ln\left(\frac{z+z_0}{z_0}\right)}, \tag{4.2}$$

where z is taken at a reference height $z_{\text{ref}} = 500$ m. Thus, by calculating TI with equation (4.2), the difference between the TI for cases A and B vs cases C and D lies in the values of C_μ which differ depending on the turbulence closure used for each case. To illustrate the meshes, the mesh used for case D is shown with different views in figure 4.6.

Table 4.2 Summary of mesh parameters for the four cases

Case	Grid cells	Δ_x [m]	Δ_{zmin} [m]	z_0 [m]	z_{0inlet} [m]	TI (%)
A	2 449 392	60	1.0	0.8	0.8	9
B	2 857 624		1.0			
C	6 123 480		0.03	0.05		12
D	6 197 704					

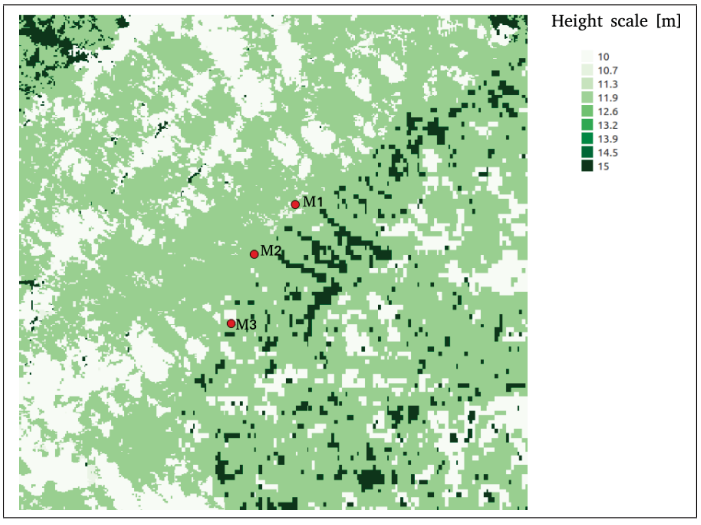
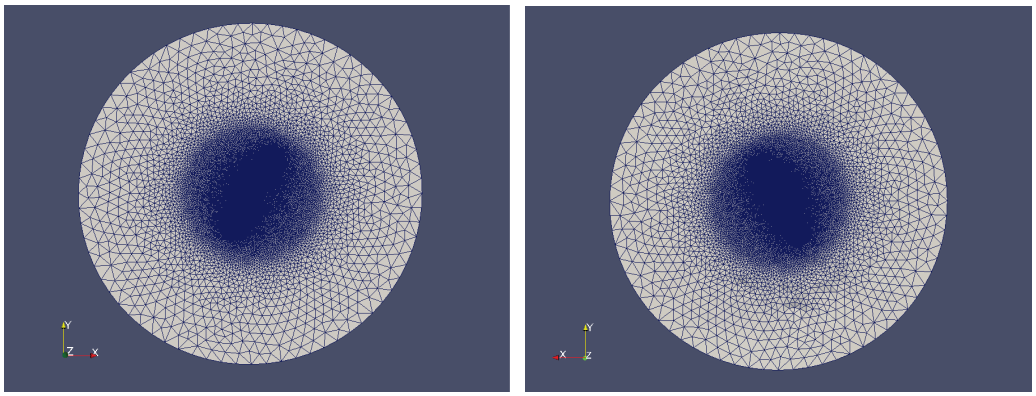
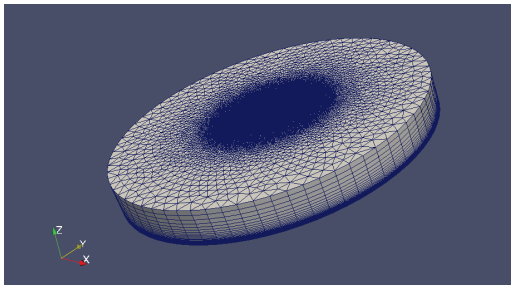


Figure 4.5 Map of forest heights used in case D. The range varies from 10 m to 15 m

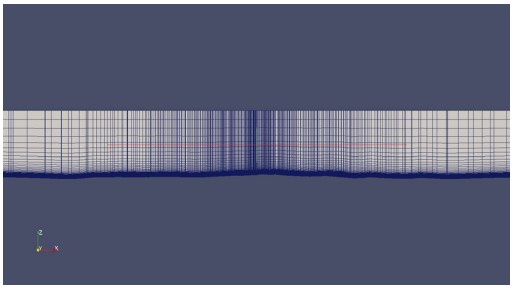


a) Top view

b) Ground view



c) Perspective view



d) Transversal cut in Y direction

Figure 4.6 Mesh used for case *D*

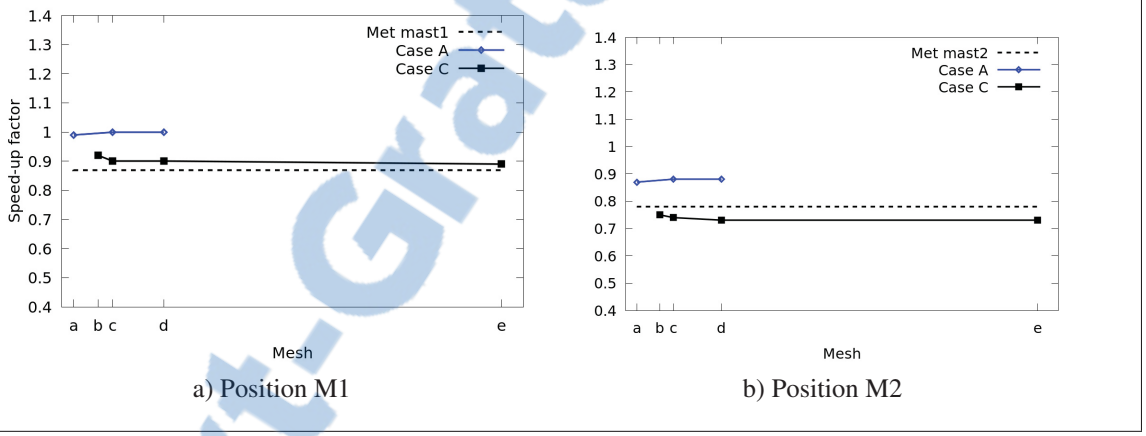
4.2.4.1 Grid independent solution

The five meshes detailed in table 4.3 were used to determine the grid independent solution. The criteria for cell sizes (Δ_x and Δ_{zmin}) and the expansion coefficients were taken from the flat case used in the canopy validation.

The results are presented in figure 4.7 in terms of speed-up factor at positions M1 and M2 at their primary anemometer heights of 45 m and 30 m, respectively. As cases *A* and *B* do not resolve the source terms within the forest, the biggest Δ_z in mesh *a* was used. Modeling case *A* with meshes *a*, *c* and *d* resulted in no significant difference. For case *C*, four meshes were considered. The results of mesh *d* minimally differ from mesh *e* despite the much higher level of refinement.

Table 4.3 Five different numerical meshes used to determine the grid independent solution. See figure 2.6 to recall the vertical zones for the expansion coefficients

Mesh	Grid cells	Refined diameter [m]	Δ_x [m]	Δ_{zmin} [m]	Expansion coefficient		
					Canopy	Turbine	Top
a	2.5×10^6	12 318	60	1.00	1.04	1.04	1.2
b	3.5×10^6		80	0.03			
c	4.2×10^6		60	0.25			
d	6.2×10^6		60	0.03			
e	20.1×10^6		30	0.03			



a) Position M1 b) Position M2

Figure 4.7 Grid independent solution for real site. The sizes of the meshes are found in table 4.3

4.3 Results

4.3.1 Results in terms of mean

The results of the four model cases are shown in figure 4.8 for the three mast positions at the three heights of their primary anemometers (30 m, 45 m and 58 m). These results are compared with the mean measurement results.

Position M1. Model *A* completely ignores the presence of forest and the shape of the velocity profile does not match the measurements. Model *B* offsets the wind profile from the ground by the displacement height, but this is not enough to match the measurements. Model *B* generally

predicts higher wind speeds than the canopy models *C* and *D*. In models *C* and *D*, the effects of the uniform and non-uniform forest are distinguished, although this seems to have little impact on the speed-up factors at measurement heights. These canopy models have a significant improvement over models *A* and *B* by nearly fitting the measurements with the proper shear (see figure 4.8a). Model *D* predicts a slightly larger shear than case *C* and poses the question of whether better tree data would improve predictions further.

Position M2. Model *A* mismatches the experimental results without even adopting the shear. Model *B* brings improved results by approaching the measurements but it does not quite match the shear. Although models *C* and *D* do not pass exactly through the experimental results, they fit smoothly the shear, especially case *D* which is in perfect agreement with the measured velocity gradient (see figure 4.8b). In general, all models underestimate the wind speed. This is explained by the physically complex zone where M2 is located. For a wind coming from the west, M2 is situated downstream of a hill that obstructs the oncoming flow and causes a topographic wake (see figure 4.1). RANS turbulence models often over-predict the size of such wakes resulting in low wind speeds predictions. This explains the improved results of model *B* with respect to measurements: the wake causes a too low wind speed prediction while the displacement height model itself makes the opposite error (over-prediction of wind speed) and thus the effects are canceled.

Position M3. Since all results are normalized by the highest point of M3, the agreement of unity speed-up at this point is present for all models (see figure 4.8c). For the rest, models *A* and *B* mismatch the first point, while models *C* and *D* fit perfectly all the results.

Overall, the canopy model has some advantages over the terrain only and displacement height models. By comparing models *C* and *D*, model *D* gives better agreement with speed-up factors. Nonetheless, model *C* gives good results, which is explained by the fact that the zone where the masts are located is a very dense forest. Thus, the assumption of a uniform dense forest distribution fits the real site and the selection of the forest parameters are adequate for this case.

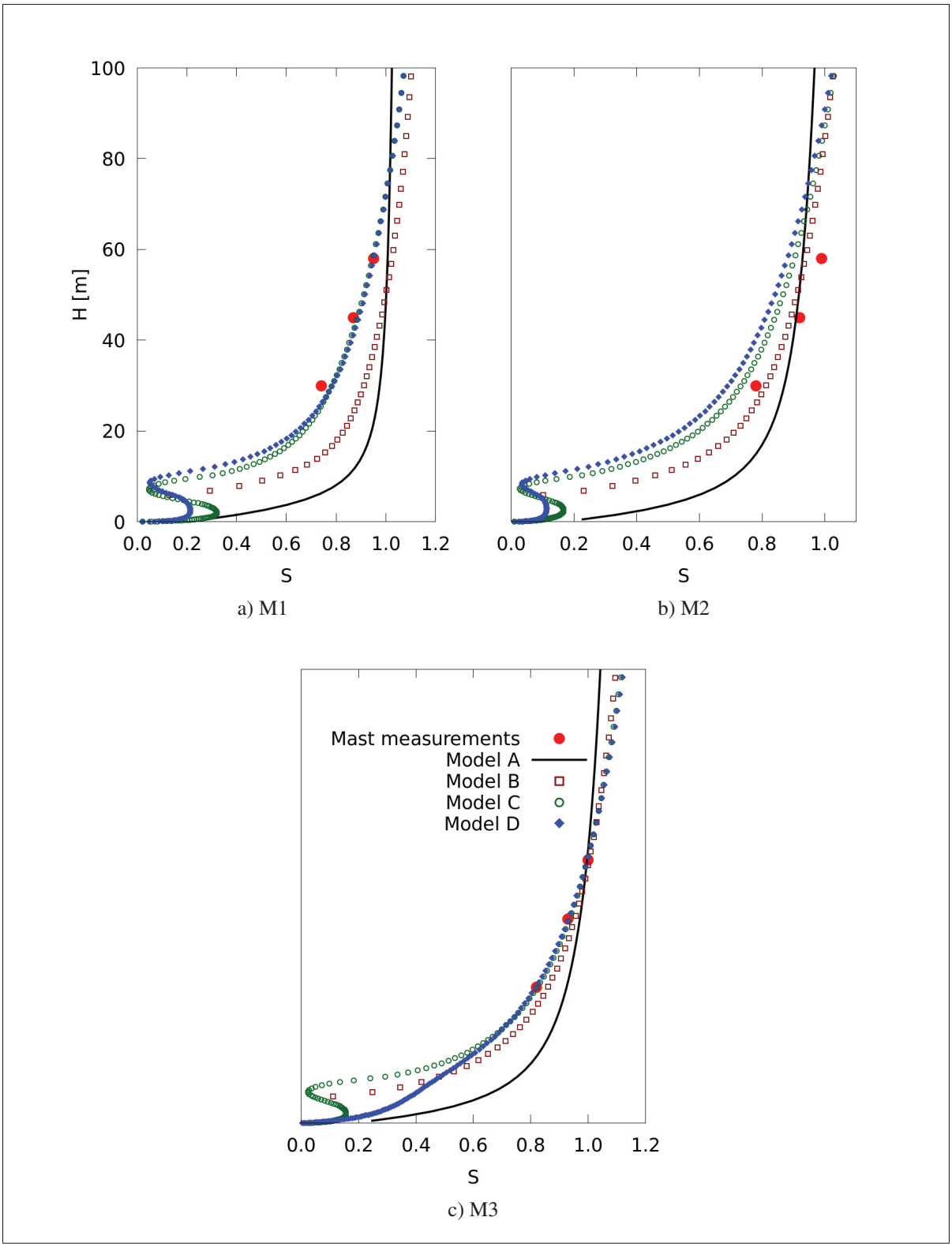


Figure 4.8 Results of the four model cases compared with mast mean results in terms of speed-up factor $S = U/U_{ref,M3-top}$

In other words, for densely forested sites, the assumptions of model *C* should give promising results, whereas for sites with forest patches, the exact forest distribution is likely required.

With these results, it is important to additionally consider the time for case preparation and simulation. With a computer of 12 cores[7], mesh generation with pretreatment of roughness and topography are obtained in a few hours for each model. On the other hand, there is a significant difference in the simulation time for each model that must taken into account. The forest models that do not modify the governing equations (models *A* and *B*), require the least time due to the applicability of a coarse grid. These models require roughly one day of simulation time with a computer of this kind. The forest models that do modify the governing equations (models *C* and *D*) require more time due to the refinement required. A simulation of this type lasts approximately seven days. To overcome these delays, the simulations were performed using the *Guillimin* supercomputer with 84 cores and a duration of two days.

4.3.2 Results in terms of mean, mode and median per bin

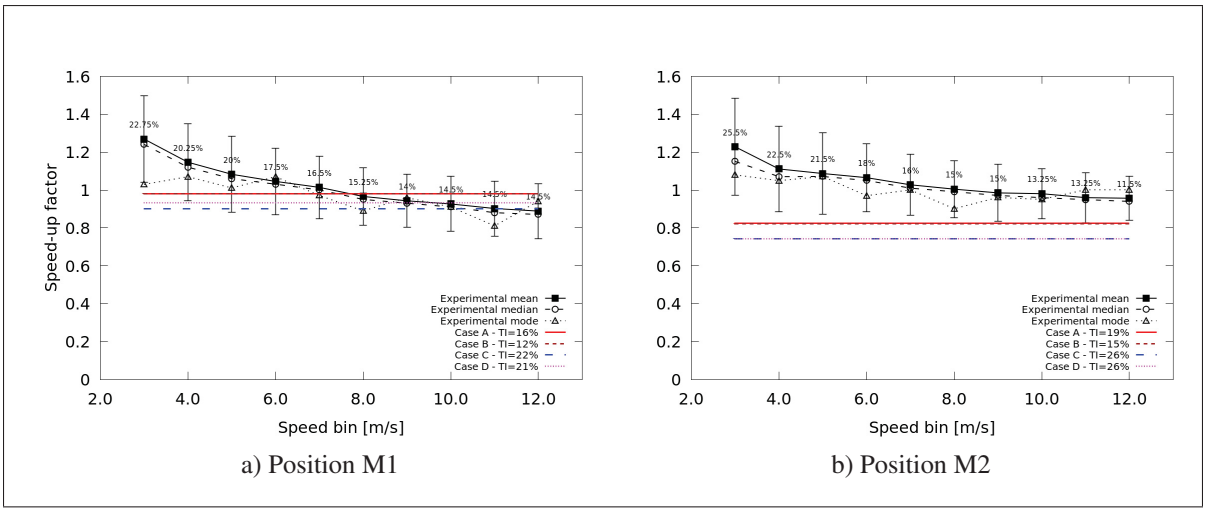
It is important to consider that the previous results rely entirely on the overall mean experimental results. As such, a close-up study was performed to compare the RANS results with the experimental data in terms of mean, mode and median values by wind speed. The experimental data is decomposed with the method of bins (Manwell *et al.*, 2009) in values ranging from 3 m/s to 12 m/s with ± 0.5 m/s. The mean, mode and median values were compared with the four simulations for positions M1 and M2 at the height of their primary anemometers 45 m and 30 m respectively. The results of this study are presented in figure 4.9.

Measurements. For both masts, the measured *S* seems to converge towards higher wind speed bins. The median is quasi-parallel to the mean, while the mode is always fluctuating due to its discrete nature. The figure also presents the values of the TI in terms of percentages and error bars. They were calculated with equation (4.1) previously described as the ratio of the standard

[7] Computer size used in this work that is commonly used in wind energy companies

deviation of the wind speed to its mean. It is seen that at higher wind speed bins, lower TI are present. This behavior is expected because the TI decreases as the mean velocity increases.

RANS results. The comparison with the simulations gives the same results discussed previously, a better agreement at M1 and an underestimation of S at position M2 due to the wake zone. Likewise the TI were calculated for each model with equation (4.1). It is clearly seen that the exaggerated effect of the wake zone impacts the turbulence and S. Overall, agreement is better for higher bins.



a) Position M1 b) Position M2

Figure 4.9 Results of the four model cases compared with mast results in terms of mean, mode and median using method of bins at positions M1 and M2

4.3.3 Linearity

Previously it was mentioned that all simulations will be performed with a friction velocity value of $u_* = 0.6$ m/s mainly because it agrees with the value of the wind measurements of the real site. But since the friction velocity is the only user-defined parameter that can be set in the simulations, the following question may arise: What is the effect of the u_* value on the results? This question was first addressed in the forest model validation with a sensitivity analysis on u_*. This analysis demonstrated u_* independence of the speed-up factor results. To

further strengthen this conclusion, another sensitivity study was performed for the real site. Eight simulations were carried out for case *C* with a u_* ranging from 0.2 to 1.0 m/s. The results are shown in terms of speed-up factor in figure 4.10 for five different locations: M1 and M2 mast coordinates, and three locations positioned within possible topographic wake zones (near M2). It is seen anew that the speed-up factor does not depend on the value of u_*.

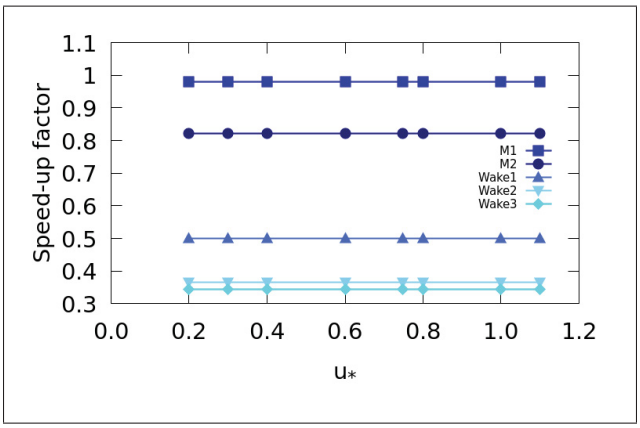


Figure 4.10 Results of the sensitivity analysis to different values of u_*. The results show clearly an independence of this user-defined parameter

4.3.4 Uncertainty in capacity factor due to RANS modelling

The simulation results and the experimental measurements allow to address the third specific objective of this master's thesis: *to evaluate the impact of RANS wind flow modeling uncertainty on capacity factor uncertainty*. This is achieved by calculating the absolute and relative uncertainties in speed-up factor and capacity factor, for the three mast positions at the three heights of their primary anemometers (30 m, 45 m and 58 m). The results of these calculations are shown in figure 4.11. The calculation of these uncertainties is explained in the following paragraph with the results of case *D* at position M1 and height of 45 m.

By assuming a rated wind speed of $U_{\text{rated}} = 11$ m/s (same as in the literature review section 1) and by using the real mean wind speed ($\overline{U}$) of 7.18 m/s, γ takes a value of 1.53 (see equation (1.23)). Then with equation (1.3), the experimental speed-up factor $S_{\text{experimental}}$ is 0.87 while the simulated one ($S_{\text{simulated}}$) is 0.89. This leads to the relative error in speed-up factor $\frac{\partial S}{S} = 2.76\%$. Directly, using $S_{\text{simulated}}$ in equation (1.24), the error in capacity factor per unit error in speed-up factor is predicted as $\frac{\partial CF}{\partial S} = 0.68$. This can be graphically found in sub-figure 1.3a with the analogy that the site itself is approximately class III. This same information can be expressed in relative terms, *i.e.* the percent error in capacity factor ($\frac{\partial CF}{CF}$) for a 1% error in speed-up factor ($\frac{\partial S}{S}$). With equation (1.20), the experimental capacity factor $CF_{\text{experimental}} = 0.28$ and the simulated one $CF_{\text{predicted}} = 0.30$ are calculated, resulting in a relative error in capacity factor $\frac{\partial CF}{CF} = 5.76\%$. Thus, the 2.76% error in S causes a 5.76% error in CF. The ratio of 2.08 can be deducted graphically from sub-figure 1.3b. Moreover, the intersection of these two values is found in sub-figure 4.11b.

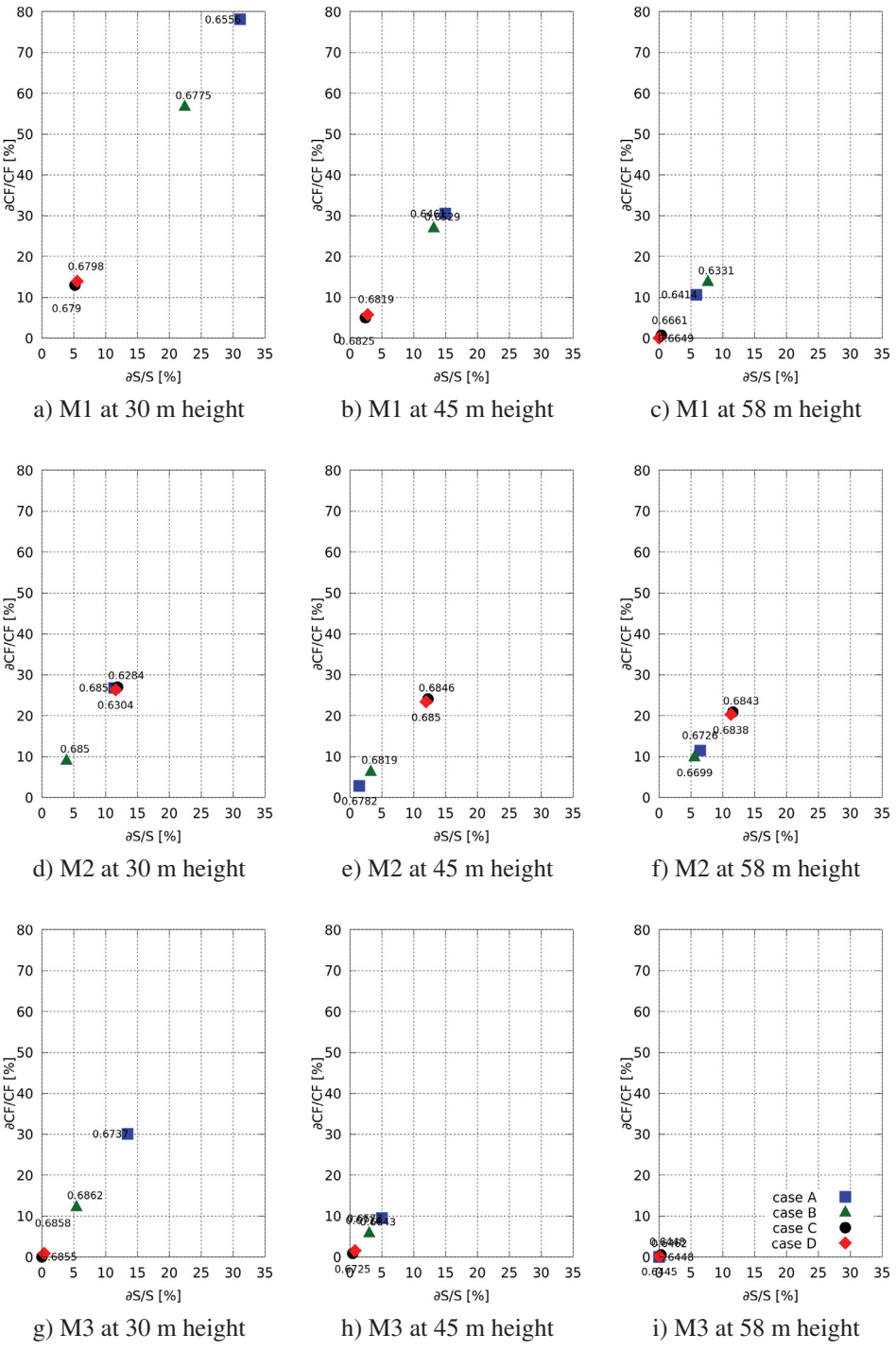
Analogously, this same evaluation was carried out for all model cases which gave the results of $\frac{\partial S}{S}$ (x-axis), $\frac{\partial CF}{CF}$ (y-axis) and $\frac{\partial CF}{\partial S}$ (labels) in figure 4.11.

Position M1. As previously seen, models that resolve the turbulence within the forest (cases *C* and *D*) give more accurate results than models that do not resolve the forest effect (cases *A* and *B*). As such, canopy models give smaller uncertainty in S and CF than no canopy models (from sub-figures 4.11a to 4.11c). This can be seen *e.g.* at 30 m height (sub-figure 4.11a), case *D* gives an error on CF of 13.91% while case *A* gives an enormous CF error of 78.11%. Furthermore, a very small difference is found between cases *C* and *D*.

Position M2. As previously discussed, RANS turbulence models overpredict the effects of topographic wakes. In this case, it helped no canopy models (cases *A* and *B*) to give improved wind speed predictions; thus, smaller uncertainties compared with canopy models (from sub-figures 4.11d to 4.11f). As *e.g.* at 45 m height (sub-figure 4.11e), case *B* gives a CF error of 6.35% while case *D* gives a CF error of 23.38%.

Position M3. Since all speed-up factors are normalized by M3 at 58 m height, the uncertainty for all cases has been reduced (from sub-figures 4.11g to 4.11i). Nevertheless, at 30 m and 45 m height, the canopy models still give better results than no canopy models. For example, at 30 m height (sub-figure 4.11g), CF error is predicted as 0.05% for case C and 0.86% for case D; while 12.25% for case B and 30% for case A.

In general, all cases show ratios of $\frac{\partial CF}{CF} / \frac{\partial S}{S}$ within a range of 1 to $\leq$2. This implies that 1% error in speed-up factor prediction leads to $\sim$2% error in capacity factor for cases with $S < 1$. Furthermore, these results agree with the analysis of chapter 1: the relative error in CF for a unit percent error in S approaches 3-to-1 for cases where $S < 0.5$ and less than 1-to-1 when $S > 1$ (see figure 1.3b). To conclude, it has been shown that the error in capacity factor can be quantified with the error in speed-up factor S; but S depends on the non-linear field U which is difficult to accurately predict. Undoubtedly, better wind flow modelling improves the prediction of U, leading to more accurate S, and thus less uncertainty in energy yield forecasting.



a) M1 at 30 m height b) M1 at 45 m height c) M1 at 58 m height

d) M2 at 30 m height e) M2 at 45 m height f) M2 at 58 m height

g) M3 at 30 m height h) M3 at 45 m height i) M3 at 58 m height

Figure 4.11 Percent error in capacity factor $\frac{\partial CF}{CF}$ for 1% error in speed-up factor $\frac{\partial S}{S}$ for the four model cases. The labels show the values of the the absolute error in CF for an absolute error in S ($\frac{\partial CF}{\partial S}$)

CONCLUSION AND RECOMMENDATIONS

The main objective of this study was to evaluate the uncertainty in wind flow modelling over a moderately complex forested site and to quantify its impact on capacity factor for wind energy predictions, using a RANS model coupled with a modified $k-\varepsilon$ turbulence closure in the open-source software *OpenFOAM* v.2.4.0. To accomplish this endeavor, the present work followed a methodology that aimed to respect the dominant physical processes of the lower atmosphere at a microscale scale with computational methods.

First, chapter 1 establishes the link between wind speed uncertainty and energy uncertainty. It was shown that the absolute error in capacity factor CF is a slight function of speed-up factor S (normalized mean wind speed) for γ (ratio of rated to mean wind speeds) close to 2, with δCF roughly half of δS, but varies greatly for small γ. This analysis showed that low wind speed sites ($S < 1$) lead to a higher uncertainty in CF than high wind speed sites ($S > 1$).

Second, chapter 2 presents the RANS equations under the assumption of a Newtonian fluid and steady, incompressible flow that does not consider Coriolis forces nor thermal effects. While the effects of complex topography were implicitly captured in the RANS equations, the effects of the forest were explicitly calculated with two models: a canopy model and a displacement height model. These models mainly differ in whether or not they modify the governing equations. Then, attention was given in the treatment of the boundary conditions, which agree with the guidelines of Richards and Hoxey (1993) and Hargreaves and Wright (2007) to produce a sustainable atmospheric boundary layer (ABL) when using the $k-\varepsilon$ turbulence model.

Third, chapter 3 validates the canopy model by reproducing the case of Dalpé and Masson (2008) for a black spruce forest. Additionally, given the limitations of obtaining the specific parameters that define the characteristics of a forest, a generic leaf area density (α) distribution was tested. The importance of an accurate leaf area index (LAI) integration to exactly fit the wind shear was shown. However, the generic α distribution allows preliminary approaches.

Since, the model of Dalpé and Masson (2008) limits its use to cyclic conditions for the inlet and outlet boundaries, accommodations were made to use the complete set of boundary conditions recommended by Richards and Hoxey (1993) and Hargreaves and Wright (2007). It was assumed that Dalpé and Masson (2008) feed the case with a fully-developed flow which is independent of the values of friction velocity (u_*) and roughness length at the inlet ($z_{0\text{inlet}}$). Therefore, for cases with defined inlet-outlet boundary conditions, two sensitivity analyses on u_* and $z_{0\text{inlet}}$ were first conducted in order to understand their roles when trying to obtain a fully-developed flow as inlet. These sensitivity studies showed that different values of either u_* or $z_{0\text{inlet}}$ give no significant difference in the vicinity of the forest, but they do at higher altitudes. Particularly, different results of U and k were found at the top of the domain by modifying the velocity gradient through $z_{0\text{inlet}}$. To conclude, normally it is desired to reach a specific value of U at the top (U_{top}) and at a reference height (U_{ref}). The value of the ratio of these values $\frac{U_{top}}{U_{ref}}$ can be achieved with a proper slope in the inlet velocity profile with respect to height (from the ground to the top). It was shown that this particular slope can be obtained by adjusting the value of $z_{0\text{inlet}}$.

Fourth, chapter 4 presents the four model cases that were calculated for a real site located in Quebec, Canada: A) terrain only, B) displacement height, C) canopy model with uniform forest, and D) canopy model with real forest distribution. The results were compared in terms of speed-up factors with two years of measurement data. It is concluded that for a forested terrain, like the present one, it is necessary to account for the forest effect to replicate the measurement data. A forest model that does not resolve the drag effect of the forest (as in the displacement height model) yields encouraging results but is less accurate, notably with overpredicted speed-up factors and incorrect wind shear. The canopy model gives better results than the displacement height model, especially with improved wind shears. This model requires the specific forest parameters: its exact height distribution, its leaf area density distribution α, and its drag coefficients C_D. When these parameters are not detailed and the terrain under study is

densely forested, the assumption of constant forest properties delivers promising results. It is also important to consider the time required for preparation and simulations. With a computer of 12 cores, the preparation time for each model is on the order of several hours, and the simulation time varies from one day for forest models that do not modify the governing equations to approximately seven days for canopy models. Additionally, when there appears to be a topographic wake, the wind speeds may be underpredicted as the turbulence closure exaggerates this effect.

Finally, in the same chapter 4, the uncertainties in the energy calculation in terms of capacity factor CF for the four models were calculated. It was shown that the uncertainty in S plays an important role in the uncertainty in CF; particularly, moderately complex forested sites lead to $S < 1$ that represent low wind speed sites which require more accurate modelling. Better wind flow modelling improves the prediction of S that leads less uncertainty in energy yield forecasting. In this case, the 2.76% uncertainty in speed-up factor associated with the real forest distribution model leads to an uncertainty in the energy calculation of just 5.76%. Additionally it is concluded that regardless of the wind class, the relative error in CF for a unit percent error in S approaches 3-to-1 for cases where S <0.5 and less than 1-to-1 when S >1.

Proposed future work

The present work was destined to evaluate the uncertainty in the wind energy production forecasting, based on modelling the lower atmospheric wind flow using CFD tools. To facilitate the modelling endeavor, this master's thesis only addresses mechanical turbulence due to complex topography and forest. However, there are other parameters that should be taken into account to improve the simulated results.

In line with the objectives and with the limitations of the present work, it is important to consider the following recommendations for future work.

- Obtain the specific physical characteristics of the forest: leaf area density distribution α and drag coefficient C_D. This work assumed one type of α and C_D which reasonably matched the general characteristic of the site; nevertheless, it would be interesting to analyze the results given by more precise values of these quantities at a particular site.

- Analyze non-dense forest sites. These types of cases could prove that a canopy model with the real forest distribution gives advantageous results compared with a canopy model that simply assumes a uniform forest distribution.

- Use other turbulence models. Analyzing the under-predicted wind speed results given by the topographic wake zone, the necessity of a more detailed turbulence model is clear. Respecting the interest of using moderate computational resources, Unsteady Reynolds-Averaged Navier-Stokes (URANS) could be a good proposal.

- Include the effects of thermal stratification. The inclusion of this source of turbulence may give a more precise evaluation of the wind energy potential of a site.

APPENDIX I

PRETREATMENT OF TOPOGRAPHY AND ROUGHNESS MAPS

Previously, it was mentioned that when working with real topographies it is necessary to treat both topography and roughness maps to respect the structure of the mesh and the boundary conditions. In order to do so, the topography map file is extruded with the values of z_0. In this work, the Geographic Information System software *QGIS* (Quantum GIS Development Team, 2002) was utilized, but there exist other GIS softwares that can be used for the same purpose. Case *D* is taken as example to explain the treatment; mainly, because it additionally modifies the original rugosity map to adequate it to the canopy model.

1. Firstly, two files are combined: the original rugosity map and the real forest map (distribution of tree's heights). Both files in raster format (.XYZ extension) should be aligned and contain the same numbers of columns and rows. This can be done in the *QGIS* raster calculator: select the smallest raster map and select "current layer extent" then select the biggest raster and set an output file and save.

2. Secondly, the rugosity map will be modified to adequate the desired z_0 distribution. This will be done by setting a logic in the *QGIS* raster calculator. In the presence of forest with heights between 10 and 15 meters height [8], a value of z_0=0.05 m [9] is set, and in the absence of forest, the z_0 of the original rugosity map will be used. This logic in *QGIS* will look like:

   $$(((\text{"}forestMap@1\text{"} >= 10)AND(\text{"}forestMap@1\text{"} <= 15))OR(\text{"}rugoMap@1\text{"} > 0.7)) * 0.05 + \text{"}rugoMap@1\text{"} * (((\text{"}forestMap@1\text{"} > 15)OR(\text{"}forestMap@1\text{"} < 10))AND(\text{"}rugoMap@1\text{"} < 0.8)).$$

[8] Range used for this study.

[9] Value used for the black spruce forest in the canopy model.

A new file will be generated with the correct z_0 distribution.

3. Thirdly, the original topography map and the new rugosity map, will be cropped such a manner that they are aligned (same delimited extensions) and that they contain the same numbers of columns and rows (as done in step 1).

4. Finally, a new topography map will be created as a sum of the foregoing maps. This is simply done in the *QGIS* raster calculator as adding the raster files (topograph + rugosity). The final raster formats of both files have to be .XYZ in order to be used in *ZephyTOOLS*.

APPENDIX II

CASE SETUP IN OPENFOAM

A simple structure of a case in OpenFOAM is shown in figure II-1

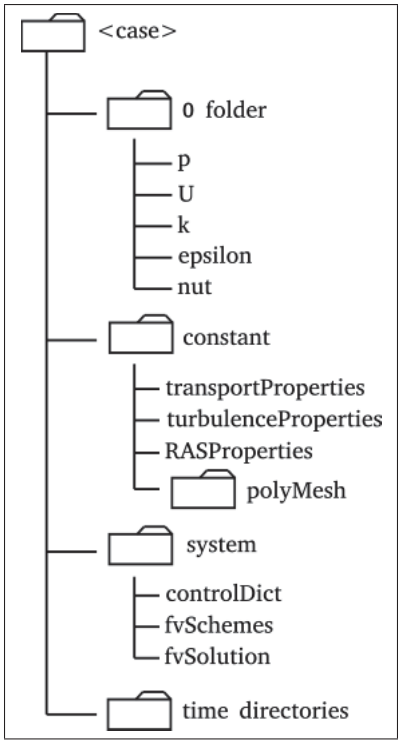


Figure-A II-1
OpenFOAM structure
for cases

1. 0 folder

It contains the boundary conditions of all the variables involved.

2. Constant folder

As its name implies, it contains all the parameters that won't change, which mainly (for this work) are the coefficients of the turbulence closure and the mesh. Several files will detail the

properties of the turbulence closure, and the mesh is contained in a folder called "polyMesh". Additionally, when working with the canopy model, the forest as a defined volume, will be contained in this constant folder.

3. System folder

It contains several files that define how the simulation will be carried out. Particularly, three main files are present: controlDict, fvSchemes and fvSolution.

3.1 system/controlDict file

Mainly, it defines the start and end parameters with all the time sets.

3.2 system/fSchemes file

The computationally schemes that were previously described will be set in this file.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default          steadyState;
}

interpolationSchemes
{
    default          linear;
}

fluxRequired
{
    default          no;
    p;
}
```

```
32
33 gradSchemes
34 {
35     default         cellMDLimited Gauss linear 0.5;
36 }
37
38 laplacianSchemes
39 {
40     default         Gauss linear limited 0.3;
41 }
42 snGradSchemes
43 {
44     default         limited 0.3;
45 }
46
47 divSchemes
48 {
49     default                         none;
50     div(phi,U)                      bounded Gauss linearUpwind grad(U);
51     div(phi,k)                      bounded Gauss upwind;
52     div(phi,epsilon)                bounded Gauss upwind;
53     div((nuEff*dev(T(grad(U)))))    Gauss linear;
54     div((nuEff*dev(grad(U).T())))   Gauss linear;
55 }
56
57 // ************************************************************************* //
```

3.3 fSolution

All the algorithm controls will be set here as the equation solvers, convergence parameters, relaxation factors, among others.

```
1  /*--------------------------------*- C++ -*----------------------------------*\
2  | =========                 |                                                 |
3  | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
4  |  \\    /   O peration     | Version:  2.4.0                                 |
5  |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
6  |    \\/     M anipulation  |                                                 |
7  \*---------------------------------------------------------------------------*/
8  FoamFile
9  {
10     version     2.0;
11     format      ascii;
12     class       dictionary;
13     object      fvSolution;
14 }
15 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
16
17 cache
18 {
19     grad(U);
20 }
21
22 solvers
23 {
24     p
```

```
25      {
26          solver                  GAMG;
27          tolerance               1e-9;
28          relTol                  1e-01;
29          smoother                GaussSeidel;
30          nPreSweeps              0;
31          nPostSweeps             0;
32          cacheAgglomeration      on;
33          agglomerator            faceAreaPair;
34          processorAgglomerator   masterCoarsest;
35          nCellsInCoarsestLevel   50;
36          mergeLevels             1;
37      }
38      U
39      {
40          solver                  smoothSolver;
41          tolerance               1e-9;
42          relTol                  1e-01;
43          smoother                GaussSeidel;
44          nSweeps                 1;
45      }
46      k
47      {
48          solver                  smoothSolver;
49          tolerance               1e-9;
50          relTol                  1e-01;
51          smoother                GaussSeidel;
52          nSweeps                 1;
53      }
54      epsilon
55          {
56          solver                  smoothSolver;
57          tolerance               1e-9;
58          relTol                  1e-01;
59          smoother                GaussSeidel;
60          nSweeps                 1;
61      }
62  }
63
64  SIMPLE
65  {
66      nNonOrthogonalCorrectors    0;
67
68      residualControl
69      {
70          p               1.00e-6;
71          U               1.00e-6;
72          k               1.00e-6;
73          epsilon         1.00e-6;
74      }
75      pRefCell        0;
76      pRefValue       0;
77  }
78
79  relaxationFactors
80  {
81      fields
82      {
83          p               0.30;
84      }
```

```
85      equations
86      {
87          U               0.70;
88          k               0.30;
89          epsilon         0.30;
90      }
91  }
92
93
94  // ************************************************************************* //
```

4. Time directories

It contains the results of all variables for different analyzed times.

APPENDIX III

CANOPY MODEL

The developed of the code for the canopy model was achieved by creating an application in OpenFoam. The basic structure of a new application in OpenFoam is shown in figure III-1. In this work, *createCdAlpha* was built to account for the leaf area density α per unit of volume multiplied by the drag coefficient C_D.

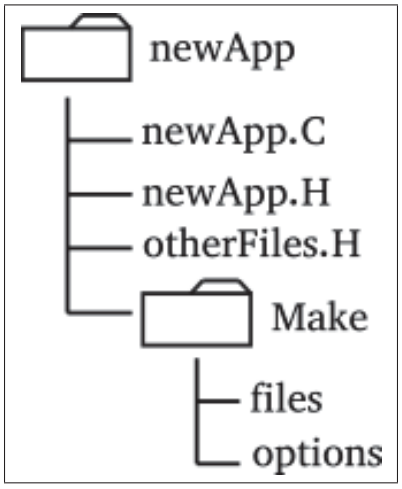


Figure-A III-1
OpenFOAM structure
for applications

0.1 createCdAlpha.C

Class definition file that sets the instructions. This file represents the exact α distribution. It can be used whether for a uniform forest distribution (as in case *C*) and to read the real forest heights (as in case *D*).

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
```

```
9       This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or modify it
12      under the terms of the GNU General Public License as published by
13      the Free Software Foundation, either version 3 of the License, or
14      (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
17      ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
18      FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
19      for more details.
20
21      You should have received a copy of the GNU General Public License
22      along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.
23
24  Application
25      createCdAlpha
26
27  Description
28      Utility to generate CdAlpha field for use with kEpsilonDenseForest
29      turbulene model and forest source.  Assumes a triangular leaf area density
30      function.
31
32  Notes
33      - Current implementation for a uniform forest is based on wallDist.H
34      - A better implementation would vertically offset the ground boundary by H
35        instead of offsetting normally.
36      - Non-uniform forest implementation does this!
37
38  \*---------------------------------------------------------------------------*/
39
40  #include <iostream>
41  #include <fstream>
42  #include "fvCFD.H"
43  #include "singlePhaseTransportModel.H"
44  #include "RASModel.H"
45  #include "wallDist.H"
46
47  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
48
49  int main(int argc, char *argv[])
50  {
51
52      argList::addOption
53      (
54    "H",
55    "scalar",
56    "Set (uniform) forest height in metres"
57      );
58
59   argList::addOption
60   (
61    "LAI",
62    "scalar",
63    "Leaf area index"
64   );
65
66   argList::addOption
67   (
68    "y0",
```

```
 "scalar",
 "Non-dimensional height of forest bottom"
);

argList::addOption
(
 "ymax",
 "scalar",
 "Non-dimensional height of maximum leaf density"
);

argList::addOption
(
 "y2",
 "scalar",
 "Non-dimensional height"
);

argList::addOption
(
 "y3",
 "scalar",
 "Non-dimensional height"
);

    argList::addOption
    (
 "Cd",
 "scalar",
 "Uniform drag coefficient"
    );

    argList::addOption
    (
        "map",
        "word",
        "Name of map file"
    );

    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "createFields.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

    volVectorField C = mesh.C();

    // (1) Uniform forest
    if ( !useMap )
    {
        scalar Alpha0=0.00;
        scalar AlphaMax=2.66;
        scalar Alpha2=2.33;
        scalar Alpha3=2.33;
        scalar Alpha4=0.00;

        forAll(CdAlpha, celli)
        {
            if (Y[celli]/H.value() <= y0)
```

```
129             {
130                 CdAlpha[celli] = 0.0;
131             }
132
133             if ((Y[celli]/H.value() > y0) && (Y[celli]/H.value() <= ymax))
134             {
135                 CdAlpha[celli] = (((AlphaMax - Alpha0)/(ymax - y0)) * ((Y[celli]/H.value() - y0))) + Alpha0;
136             }
137
138             if ((Y[celli]/H.value() > ymax) && (Y[celli]/H.value() <= y2))
139             {
140                 CdAlpha[celli] = (((Alpha2 - AlphaMax)/(y2 - ymax)) * ((Y[celli]/H.value() - ymax))) + AlphaMax;
141             }
142
143             if ((Y[celli]/H.value() > y2) && (Y[celli]/H.value() <= y3))
144             {
145                 CdAlpha[celli] = (((Alpha3 - Alpha2)/(y3 - y2)) * ((Y[celli]/H.value() - y2))) + Alpha2;
146             }
147
148             if ((Y[celli]/H.value() > y3) && (Y[celli]/H.value() < 1.0))
149             {
150                 CdAlpha[celli] = (((Alpha4 - Alpha3)/(1.0 - y3)) * ((Y[celli]/H.value() - y3))) + Alpha3;
151             }
152
153             if (Y[celli]/H.value() >= 1.0)
154             {
155                 CdAlpha[celli] = 0.0;
156             }
157
158             Info << C[celli] << "\t" << Y[celli] << "\t" << CdAlpha[celli] << endl;
159         }
160
161         CdAlpha *= Cd;
162
163     }
164     else
165
166     // (2) Non-uniform forest
167     {
168         // Process raster to get H field
169         // This is (usually) a very large file. To avoid loading the entire file, the raster should be written on a square
      cartesian grid and the file formatted to have lines of constant byte length. C++ i/o can bytecount forward to the
      required line.
170
171         // Raster info. It is obtained from the original forest map raster.
172         scalar dx = 93.734; // Size of the rows
173         scalar dy = -69.1972; // Size of the columns
174         int nx = 2075; // Number of rows
175         int ny = 3860; // Number of columns
176         scalar x0Raster = 0.0; // Physical domain x0 corner in raster coordinates. Not given for confidential reasons
177         scalar y0Raster = 0.0; // Physical domain y0 corner in raster coordinates. Not given for confidential reasons
178
179         // Byte count per line
180         int lineSize = 31;
181
182         // Physical domain centre in raster coordinates. Not given for confidential reasons.
183         scalar x0Mesh = 0.0;
184         scalar y0Mesh = 0.0;
185
186         // Open map file
```

```
187         std::ifstream rasterFile("test.xyz");
188         if (!rasterFile.is_open()) {
189             FatalError
190             << "Raster file could not be opened"
191             << exit(FatalError);
192         }
193
194         std::streampos readByte = rasterFile.tellg();
195         std::streampos begin,end;
196
197         begin = rasterFile.tellg();
198         rasterFile.seekg (0, std::ios::end);
199         end = rasterFile.tellg();
200
201         int fileSize = end-begin;
202
203         Info << "File size = " << fileSize << endl;
204
205         // Get list of wall patches
206         labelList wallPatches = mesh.boundaryMesh().findIndices("wall");
207
208         Info << "Wall patches: " << wallPatches << endl;
209
210         bool debug(false);
211
212         scalar xData, yData;
213         int count = 0;
214         int totalFaces = 0;
215         vector xHat(1,0,0);
216
217         forAll(wallPatches, patchI)
218         {
219             const fvPatchVectorField& faceCentres = mesh.C().boundaryField()[wallPatches[patchI]];
220
221             // Create field to store forest height data
222             scalarField Hfield = 0*mag(faceCentres);
223
224             Info << faceCentres.size() << " faces on patch" << endl;
225             totalFaces += faceCentres.size();
226
227             // Read raster into Hfield
228             forAll(faceCentres,faceI)
229             {
230                 Info << "Patch " << faceI+1 << " of " <<faceCentres.size() << endl;
231
232                 scalar x = faceCentres[faceI].x() + x0Mesh - x0Raster;
233                 scalar y = faceCentres[faceI].y() + y0Mesh - y0Raster;
234
235                 // Check that point exists in raster
236                 if (x < 0 || y > 0 || x > nx*dx || y < ny*dy ) {
237                     Hfield[faceI] = 0;
238                     count++;
239                     Info << "Mesh point outside raster file" << endl;
240                     Info << endl;
241                 }
242                 else
243                 {
244                     // Nearest line number
245                     int lineNo = static_cast<int>(y/dy)*nx + static_cast<int>(x/dx);
246
```

```
247                     readByte = lineNo * lineSize;
248
249                     // Check that byte exists in file
250                     if (readByte < begin || readByte > end)
251                             Info << "Error: Trying to read outside of file! " << static_cast<int>(readByte) << endl;
252
253                     rasterFile.seekg(readByte);
254
255                     // Read
256                     rasterFile >> xData;
257                     rasterFile >> yData;
258
259                     // Double check that raster point is close to mesh point
260                     if (mag(x-xData)>mag(dx)) {
261                         Info << "Not closest point!  " << x << " " << xData << endl;
262                     }
263                     if (mag(y-yData)>mag(dy)) {
264                         Info << "Not closest point!  " << y << " " << yData << endl;
265                     }
266
267                     rasterFile >> Hfield[faceI];
268
269                     if (Hfield[faceI] > 998)
270                         Hfield[faceI] = 0;
271
272                     if (debug)
273                     {
274                         Info << "Byte no =   " << static_cast<int>(readByte) << endl;
275                         Info << "File size = " << fileSize << endl;
276                         Info << "dx = " << dx << ", dy = " << dy << endl;
277                         Info << "xMesh = " << faceCentres[faceI].x() + x0Mesh << ", yMesh = " << faceCentres[faceI].y() +
    y0Mesh << endl;
278                         Info << "xRaster = " << xData + x0Raster << ", yRaster = " << yData + y0Raster << endl;
279                         Info << "H = " << Hfield[faceI] << endl;
280                         Info << endl;
281                     }
282
283                 }
284
285             }
286
287             Info << "\nCalculating CdAlpha..." << endl;
288
289             const fvPatch& currPatch = mesh.boundary()[wallPatches[patchI]];
290
291             forAll(currPatch,faceI)
292             {
293                 //faceI = 8929;
294
295                 // Get vertex of face
296                 label currentCell = currPatch.faceCells()[faceI];
297                 scalar zGround = faceCentres[faceI].z();
298                 scalar h = C[currentCell].z() - zGround;
299
300                 if(debug) {
301                     Info << "zGround = " << zGround << endl;
302                     Info << "Current cell = " << currentCell << endl;
303                     Info << "C[currentCell] = " << C[currentCell].z() << endl;
304                     Info << "h = " << h << endl;
305                     Info << "Hfield[faceI] = " << Hfield[faceI] << endl;
```

```
306                 }
307
308                 // Only modify CdAlpha field if forest height is larger than specified value
309                 if (Hfield[faceI] > 9.0)
310                 {
311
312                 while ((Hfield[faceI] - h) > 0)
313                 {
314                     scalar AlphaMax=2.0*LAI/Hfield[faceI]/(1-y0);
315
316                     // Set CdAlpha
317                     if (h/Hfield[faceI] < y0)
318                     {
319                         CdAlpha[currentCell] = 0.05; // Add a minimum resistance to avoid under canopy jet
320                     }
321
322                     if ((h/Hfield[faceI] > y0) && (h/Hfield[faceI] < ymax))
323                     {
324                         CdAlpha[currentCell] = AlphaMax * (h/Hfield[faceI] - y0)/(ymax - y0);
325                     }
326
327                     if ((h/Hfield[faceI] > ymax) && (h/Hfield[faceI] < 1))
328                     {
329                         CdAlpha[currentCell] = AlphaMax * (1.0 - (h/Hfield[faceI]-ymax)/(1-ymax));
330                     }
331
332                     // Get all neighbours
333                     labelList neighbours = mesh.cellCells()[currentCell];
334
335                     Info << "Face: " << faceI << endl;
336                     Info << "Neighbour cells: " << neighbours << endl;
337                     Info << "Hfield[faceI] = " << Hfield[faceI] << endl;
338                     Info << "h = " << h << endl;
339
340                     // Get vertical neighbour
341                     scalar search = 1e-1;
342                     label newCurrentCell = 0;
343                     forAll(neighbours,cellI)
344                     {
345                         if(debug) {
346                             Info << "C[neighbours[cellI]]: " << C[neighbours[cellI]] << endl;
347                             Info << "C[currentCell]: " << C[currentCell] << endl;
348                         }
349
350                         //if(mag((C[neighbours[cellI]] - C[currentCell]) & xHat) < TOL)
351                         if(C[neighbours[cellI]].z() > C[currentCell].z()) {
352                             if(mag((C[neighbours[cellI]] - C[currentCell]) & xHat) < search) {
353                                 search = mag((C[neighbours[cellI]] - C[currentCell]) & xHat);
354                                 newCurrentCell = neighbours[cellI];
355                             }
356                         }
357                     }
358
359                     currentCell = newCurrentCell;
360
361                     // Re-calculate h
362                     h = C[currentCell].z() - zGround;
363                     if(debug) {
364                         Info << "h = " << h << endl;
365                         Info << "Current cell = " << currentCell << endl;
```

```
                    }
                }

                }
            }
        }

        rasterFile.close();

        CdAlpha *= Cd;

        Info << "Maximum CdAlpha = " << max(CdAlpha) << endl;
    }

    CdAlpha.write();

    Info<< nl << "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
        << "  ClockTime = " << runTime.elapsedClockTime() << " s"
        << nl << endl;

    Info<< "End\n" << endl;

    return 0;
}


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

0.2 createCdAlpha.H

Header file that contains the names of the classes and their functions.

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

\*---------------------------------------------------------------------------*/

```

```
26      Info<< "Creating field CdAlpha" << endl;
27      volScalarField CdAlpha
28      (
29          IOobject
30          (
31              "CdAlpha",
32              runTime.constant(),
33              mesh,
34              IOobject::NO_READ
35          ),
36          mesh,
37          dimensionedScalar("CdAlpha", dimensionSet(0,-1,0,0,0,0,0), 0.0),
38          zeroGradientFvPatchScalarField::typeName
39      );
40
41      Info<< "Calculating wall distance field" << endl;
42      volScalarField Y(wallDist(mesh).y());
43
44      // Initial parameters
45      word map = "";
46      bool useMap(false);
47      dimensionedScalar H("H", dimLength, 0);
48      scalar Cd = 0.0;
49      scalar LAI = 0.0;
50      scalar y0 = 0.0; // Y/H!
51      scalar ymax = 1.0;
52      scalar y2 = 0.0;
53      scalar y3 = 0.0;
54
55      // Set simulation type based on arguments
56      if (args.optionFound("map"))
57      {
58          map = args.optionRead<word>("map");
59          useMap = true;
60      }
61      else
62      {
63          // Set forest depth
64          if (args.optionFound("H"))
65          {
66              H.value() = args.optionRead<scalar>("H");
67          }
68          else
69          {
70              FatalErrorIn(args.executable())
71              << "Forest height must be specified"
72              << exit(FatalError);
73          }
74      }
75
76      // Set drag coefficient
77      if (args.optionFound("Cd"))
78      {
79          Cd = args.optionRead<scalar>("Cd");
80      }
81      else
82      {
83          FatalErrorIn(args.executable())
84          << "Drag coefficient must be specified"
85          << exit(FatalError);
```

```
86      }
87
88      // Set leaf area index
89      if (args.optionFound("LAI"))
90      {
91          LAI = args.optionRead<scalar>("LAI");
92      }
93      else
94      {
95          FatalErrorIn(args.executable())
96          << "Leaf area index must be specified"
97          << exit(FatalError);
98      }
99
100     if (args.optionFound("y0"))
101     {
102         y0 = args.optionRead<scalar>("y0");
103     }
104
105     if (args.optionFound("ymax"))
106     {
107         ymax = args.optionRead<scalar>("ymax");
108     }
109
110     if (args.optionFound("y2"))
111     {
112         y2 = args.optionRead<scalar>("y2");
113     }
114
115     if (args.optionFound("y3"))
116     {
117         y3 = args.optionRead<scalar>("y3");
118     }
119
120
121 // ************************************************************************* //
```

BIBLIOGRAPHY

Amiro, B. D. (1990). Comparison of turbulence statistics within three boreal forest canopies. *Boundary-layer meteorology*, 51(1-2), 99–121.

Apsley, D. D. & Castro, I. P. (1997). A limited-length-scale k-ε model for the neutral and stably-stratified atmospheric boundary layer. *Boundary-layer meteorology*, 83(1), 75–98.

Arroyo, R. C., Rodrigo, J. S. & Gankarski, P. (2014). Modelling of atmospheric boundary-layer flow in complex terrain with different forest parameterizations. *Journal of physics: Conference series*, 524, 012119.

Ayotte, K. W. (2008). Computational modelling for wind energy assessment. *Journal of wind engineering and industrial aerodynamics*, 96(10), 1571–1590.

Bardina, J. E., Huang, P. G. & Coakley, T. J. (1997). *Turbulence Modeling Validation, Testing, and Development* (Report n°NASA-TM-110446). Moffett Field, CA United States: Nasa Technical Memorandum, 100.

Bautista, M. C. (2015). *Turbullence modelling of the atmospheric boundary layer over complex topography*. (Ph. D. thesis, École de technologie supérieure, Montreal, QC. Canada).

Bechmann, A., Sørensen, N. N., Berg, J., Mann, J. & Réthoré, P. E. (2011). The Bolund Experiment, Part II: Blind Comparison of Microscale Flow Models. *Boundary-layer meteorology*, 141(2), 245–271.

Bechmann, A., Berg, J., Courtney, M. S., Hans, E. J., Mann, J. & Sørensen, N. N. (2009). *The Bolund Experiment : Overview and Background* (Report n°Risø-R1658(EN)). Roskilde, Denmark: Risø DTU.

Ben Younes, H. (2016). *Simulation de la couche limite atmosphérique sur un couvert forestier en terrain avec orographie*. (Ph. D. thesis, École de technologie supérieure, Montreal, QC. Canada).

Berg, J., Mann, J., Bechmann, A., Courtney, M. S. & Jørgensen, H. E. (2011). The Bolund Experiment, Part I: Flow Over a Steep, Three-Dimensional Hill. *Boundary-layer meteorology*, 141(2), 219–243.

Boudreault, L.-É., Bechmann, A., Sørensen, N., Sogachev, A. & Dellwik, E. (2014). Canopy structure effects on the wind at a complex forested site. *Journal of physics: Conference series*, 524, 012112.

Boudreault, L. É., Bechmann, A., Tarvainen, L., Klemedtsson, L., Shendryk, I. & Dellwik, E. (2015). A LiDAR method of canopy structure retrieval for wind modeling of heterogeneous forests. *Agricultural and forest meteorology*, 201, 86–97.

Bradley, E. F. (1980). An experimental study of the profiles of wind speed, shearing stress and turbulence at the crest of a large hill. *Quarterly journal of the royal meteorological society*, 106(447), 101–123.

Burton, T., Sharpe, D., Jenkind, N. & Bossanyi, E. (2001). *Wind Energy Handbook*. Baffins Lane, England: John Wiley & Sons, Ltd, 643.

CanWEA. (2016). Wind energy installed capacity. Consulted at http://canwea.ca/wind-energy/installed-capacity/.

Casella, L., Langreder, W., Fischer, A., Ehlen, M. & Skoutelakos, D. (2014). Dynamic flow analysis using an OpenFOAM based CFD tool: Validation of Turbulence Intensity in a testing site. *Itm web of conferences*, 2, 4002.

Castro, F. A., Palma, J. M. L. M. & Silva Lopes, A. (2003). Simulation of the Askervein Flow. Part 1: Reynolds Averaged Navier–Stokes Equations (k–ε Turbulence Model). *Boundary-layer meteorology*, 107(3), 501–530.

Chen, J. M. & Cihlar, J. (1996). Retrieving leaf area index of boreal conifer forests using Landsat TM images. *Remote sensing of environment*, 55(2), 153–162.

Dalpé, B. & Masson, C. (2008). Numerical study of fully developed turbulent flow within and above a dense forest. *Wind energy*, 11(5), 503–515.

Desmond, C. J., Watson, S. J., Aubrun, S., Ávila, S., Hancock, P. & Sayer, A. (2014). A study on the inclusion of forest canopy morphology data in numerical simulations for the purpose of wind resource assessment. *Journal of wind engineering and industrial aerodynamics*, 126, 24–37.

Emeis, S., Courtney, M., Højstrup, J. & Jensen, N. (1993). *Hjardemål experiment data report* (Report n°Risø-M-2289(EN)). Roskilde, Denmark.

Emeis, S. (2011). *Surface-Based Remote Sensing of the Atmospheric Boundary Layer*. Dordrecht, Netherlands: Springer.

ETIPWind. (2016). *Strategic research and innovation agenda 2016* (Report n°ETIPWind-SRIA-2016). Brussels, Belgium: European Technology and Innovation Platform on Wind Energy. Consulted at http://etipwind.eu.

Ferziger, J. H. & Peric, M. (2002). *Computational Methods for Fluid Dynamics* (ed. 3rd). Berlin, Germany: Springer-Verlag.

Gorlé, C., van Beeck, J., Rambaud, P. & Van Tendeloo, G. (2009). CFD modelling of small particle dispersion: The influence of the turbulence kinetic energy in the atmospheric boundary layer. *Atmospheric environment*, 43(3), 673–681.

Gouvernement du Québec. (2017). *Géoboutique Québec*. Énergie et Ressources naturelles Québec. Consulted at http://geoboutique.mern.gouv.qc.ca/edel/pages/recherche/critereRechercheEdel.faces.

Grant, E. R., Ross, A. N. & Gardiner, B. A. (2016). Modelling Canopy Flows over Complex Terrain. *Boundary-layer meteorology*, 161(3), 417–437.

GWEC. (2016). *Global wind report 2015* (Report n°GLOBAL WIND REPORT 2015). Brussels, Belgium: Global Wind Energy Council. Consulted at http://www.gwec.net/global-figures/wind-energy-global-status/.

Hargreaves, D. M. & Wright, N. G. (2007). On the use of the k–ε model in commercial CFD software to model the neutral atmospheric boundary layer. *Journal of wind engineering and industrial aerodynamics*, 95(5), 355–369.

IEA. (2016a). *Energy, Climate Change and Environment 2016 Insights* (Report n°ECCE2016). Paris, France: OECD/IEA. Consulted at https://www.iea.org/publications/.

IEA. (2016b). *World Energy Outlook 2016 (executive summary)* (Report n°WEO2016). Paris, France: OECD/IEA. Consulted at https://www.iea.org/publications/.

IEA. (2016c). *Medium-Term Renewable Energy Market Report 2016 (executive summary)* (Report n°MTREMR2016). Paris, France: OECD/IEA. Consulted at https://www.iea.org/publications/.

IEC 61400-1. (2005). *Wind turbines - Part 1: Design requirements* (ed. 3rd). Geneva, Switzerland: International Electrotechnical Commission.

International Electrotechnical Commision. (2005). Wind turbines - Part 12-1 : Power performance measurements of electricity producing wind turbines. In *IEC 61400-12-1* (pp. 1–8).

Jasak, H. (1996). *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. (Ph. D. thesis, University of London, London, England).

Jeannotte, E. (2013). *Estimation of LIDAR bias over complex terrain using numerical tools*. (M.Ing thesis, École de technologie supérieure, Montreal, QC. Canada).

Jones, W. P. & Launder, B. E. (1972). The Prediction of Laminarization With aTwo-Equation Model of Turbulence. *International journal of heat and mass transfer*, 15(2), 301–314.

Kaimal, J. C. & Finnigan, J. J. (1994). *Atmospheric boundary layer flows : their structure and measurement*. New York, USA: Oxford University Press.

Katul, G. G., Mahrt, L., Poggi, D. & Sanz, C. (2004). ONE- and TWO-Equation Models for Canopy Turbulence. *Boundary-layer meteorology*, 113(1), 81–109.

Kim, H. G. & Patel, V. C. (2000). Test of turbulence models for wind flow over terrain with separation and recirculation. *Boundary-layer meteorology*, 94(1), 5–21.

Kim, H. G., Patel, V. C. & Lee, C. M. (2000). Numerical simulation of wind flow over hilly terrain. *Journal of wind engineering and industrial aerodynamics*, 87(1), 45–60.

Landberg, L., Myllerup, L., Rathmann, O., Petersen, E. L., Jørgensen, B. H., Badger, J. & Mortensen, N. G. (2003). Wind resource estimation - an overview. *Wind energy*, 6(3), 261–271.

Lange, M. & Focken, U. (2006). *Physical approach to short-term wind power prediction*. Berlin, Heidelberg: Springer-Verlag.

Launder, B. E. & Sharma, B. I. (1974). Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in heat and mass transfer*, 1(2), 131–137.

Launder, B. E. & Spalding, D. B. (1972). *Lectures in mathematical models of turbulence*. New York, USA: Academic Press London.

Launder, B. & Spalding, D. (1974). The numerical computation of turbulent flows. *Computer methods in applied mechanics and engineering*, 3(2), 269–289.

Leonard, B. P. (1979). A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering*, 19(1), 59–98.

Lira, A., Rosas, P., Araújo, A. & Castro, N. (2016). *Uncertainties in the estimate of wind energy production* (Report n°Technical Report). Rio de Janeiro, Brazil.

Liu, J., Chen, J. M., Black, T. A. & Novak, M. D. (1996). E-ε modelling of turbulent air flow downwind of a model forest edge. *Boundary-layer meteorology*, 77(1), 21–44.

Lopes, a. S., Palma, J. M. L. M. & Lopes, J. V. (2013). Improving a Two-Equation Turbulence Model for Canopy Flows Using Large-Eddy Simulation. *Boundary-layer meteorology*, 149(2), 231–257.

Lopes da Costa, J. C., Castro, F. a., Palma, J. M. L. M. & Stuart, P. (2006). Computer simulation of atmospheric flows over real forests for wind energy resource evaluation. *Journal of wind engineering and industrial aerodynamics*, 94(8), 603–620.

Manwell, J. F., McGowan, J. G. & Rogers, A. L. (2009). *Wind energy explained: theory, design and application* (ed. 2nd). Chichester, U.K.: Wiley.

Mason, P. J. & King, J. C. (1985). Measurements and predictions of flow and turbulence over an isolated hill of moderate slope. *Quarterly journal of the royal meteorological society*, 111(468), 617–640.

Milashuk, S. & Crane, W. A. (2011). Wind speed prediction accuracy and expected errors of RANS equations in low relief inland terrain for wind resource assessment purposes. *Environmental modelling & software*, 26(4), 429–433.

Monin, A. S. & Obukhov, A. M. (1954). Basic laws of turbulent mixing in the atmosphere near the ground. *Tr. akad. nauk sssr geofiz. inst*, 24, 163–187.

Morales Garza, V., Nathan, J., Sumner, J. & Masson, C. (2017a). Evaluating the accuracy of RANS wind flow modeling over forested terrain. Part 2: Impact on capacity factor for moderately complex topography. *Manuscript in preparation*.

Morales Garza, V., Nathan, J., Sumner, J. & Masson, C. (2017b). Evaluating the accuracy of RANS wind flow modeling over forested terrain. Part 1: Canopy model validation. *Manuscript in preparation*.

Mortensen, N. G., Heathfield, D. N., Rathmann, O. & Nielsen, M. (2011). *Wind Atlas Analysis and Application Program: WAsP 10 Help Facility*. Denmark: DTU Wind Energy.

Murphy, A. H. (1993). What Is a Good Forecast? An Essay on the Nature of Goodness in Weather Forecasting. *Weather and forecasting*, 8(2), 281–293.

Nakićenović, N. (1996). Freeing Energy from Carbon. *Daedalus*, 125(3), 95–112.

Obukhov, A. M. (1971). Turbulence in an atmosphere with a non-uniform temperature. *Boundary-layer meteorology*, 2(1), 7–29.

Omasa, K., Hosoi, F. & Konishi, A. (2007). 3D lidar imaging for detecting and understanding plant responses and canopy structure. *Journal of experimental botany*, 58(4), 881–898.

Panofsky, H. & Ming, Z. (1983). Characteristics of wind profiles over complex terrain. *Journal of wind engineering and industrial aerodynamics*, 15(1), 177–183.

Patankar, S. (1980). *Numerical heat transfer and fluid flow*. Washington, D.C.: Hemisphere.

Petersen, E. L., Mortensen, N. G., Landberg, L., Højstrup, J. & Frank, H. P. (1998). Wind power meteorology. Part II: siting and models. *Wind energy*, 1(2), 55–72.

Pinson, P. (2006). *Estimation of the uncertainty in wind power forecasting*. (Ph. D. thesis, École Nationale Supérieure des Mines de Paris, Paris, France).

Pope, S. B. (2000). *Turbulent Flows*. Cambridge, U.K.: Press, Cambridge University.

Prospathopoulos, J. & Voutsinas, S. G. (2006). Implementation Issues in 3D Wind Flow Predictions Over Complex Terrain. *Journal of solar energy engineering*, 128(4), 539.

Quantum GIS Development Team. (2002). Quantum GIS Geographic Information System. Version 2.18. Open Source Geospatial Foundation Project. Consulted at http://qgis.osgeo.org.

Raithby, G. D., Stubley, G. D. & Taylor, P. A. (1987). The Askervein hill project: A finite control volume prediction of three-dimensional flows over the hill. *Boundary-layer meteorology*, 39(3), 247–267.

Raupach, M. R. (1994). Simplified expressions for vegetation roughness length and zero-plane displacement as functions of canopy height and area index. *Boundary-layer meteorology*, 71(1-2), 211–216.

Reynolds, O. (1895). On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion. *Philosophical transactions of the royal society of london. (a.)*, 186, 123–164.

Richards, P. & Hoxey, R. (1993). Appropriate boundary conditions for computational wind engineering using the k-epsilon turbulence model. *Journal of wind engineering and industrial aerodynamics*, 46 & 47, 145–153.

Salmon, J. R., Teunissen, H. W., Mickle, R. E. & Taylor, P. A. (1988). The Kettles Hill Project: Field observations, wind-tunnel simulations and numerical model predictions for flow over a low hill. *Boundary-layer meteorology*, 43(4), 309–343.

Stull, R. B. (1988). *An Introduction to Boundary Layer Meteorology*. Dordrecht, Netherlands: Springer Science+Business Media B.V.

Sumner, J. & Masson, C. (2012). k - ε simulations of the neutral atmospheric boundary layer: analysis and correction of discretization errors on practical grids. *International journal for numerical methods in fluids*, 70(6), 724–741.

Sumner, J., Watters, C. S. & Masson, C. (2010). CFD in Wind Energy: The Virtual, Multiscale Wind Tunnel. *Energies*, 3(5), 989–1013.

Svensson, U. & Haggkvist, K. (1990). Two-equation turbulence model for canopy flows. *Journal of wind engineering and industrial aerodynamics*, 35(1), 201–211.

Taylor, P. A. & Teunissen, H. W. (1987). The Askervein Hill project: Overview and background data. *Boundary-layer meteorology*, 39(1-2), 15–39.

Verhoef, A., McNaughton, K. G. & Jacobs, A. F. G. (1997). A parameterization of momentum roughness length and displacement height for a wide range of canopy densities. *Hydrology and earth system sciences*, 1(1), 81–91.

Versteeg, H. K. & Malalasekera, W. (2007). *An introduction to computational fluid dynamics: the finite volume method* (ed. 2nd). Edinburgh, England: Pearson Education Limited.

Weller, H. G. & Tabor, G. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12(6), 620–631.

Zephy-Science. (2012). ZephyTOOLS. Open-Source CFD Wind Farm Design Software. Consulted at http://www.zephy-science.com/.