ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
A MASTER'S DEGREE IN MECHANICAL ENGINEERING
M.Eng.

BY
Eric JEANNOTTE

ESTIMATION OF LIDAR BIAS OVER COMPLEX TERRAIN USING NUMERICAL TOOLS

MONTREAL, JULY 24 2013

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

Prof. Christian Masson, Thesis director
Mechanical Engineering Departement

Prof. Louis Dufresne, Thesis co-director
Mechanical Engineering Departement

Prof. François Morency, Committee president
Mechanical Engineering Departement

Dr. Dariush Faghani, External examiner
GL Garrad Hassan

Dr. Robert Benoit, Examiner
Mechanical Engineering Departement

THIS DISSERTATION WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

ON JUNE 25 2013

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

The completion of this project would have been impossible without the precious support that I received from my director, Christian Masson. His patience and his remarkable capacity of analyzing and synthesizing enlightened me countless times.

I also gratefully acknowledge all the partners who collaborated to this project: GL Garrad Hassan, Technocentre éolien, NRG, Leosphere and Cartier Énergie Éolienne. I would like to specially thank Dariush Faghani, from GL Garrad Hassan, for his guidance and insightful comments about my work. Thank you also to Ana Poenariu from GL Garrad Hassan, your help was really appreciated.

Also, I would like to kindly thank Chérif, Nicolas Lussier-Clément, Mary Bautista, Jörn Nathan, Hugo Olivares, Hajer Ben Younes, Oscar Alejandro, Jonathon Sumner, David Lapointe-Thériault and Marc Pelletier for all the interesting discussions that we had and that contributed positively to this project. It is a real chance that we have to work in such an inspiring environment.

Finally, I must sincerely thank my family and my girlfriend, Evelyne, for their continuous support throughout this project. Your encouragement really gave me hope during tough times.

ESTIMATION OF LIDAR BIAS OVER COMPLEX TERRAIN USING NUMERICAL TOOLS

Eric JEANNOTTE

ABSTRACT

For a few years, a new wind measurement instrument has been competing with standard cup anemometers: wind LiDARs. Despite numerous advantages such as ease of deployment and the possibility to scan at multiple heights simultaneously, the performances of this instrument over complex terrain are still a matter of debate. This is mainly due to the flow homogeneity assumption made by the remote sensor which leads to a positive or a negative bias. The objective of this work was to implement a method to evaluate LiDAR bias over complex terrain using OpenFOAM. To accomplish this task, a CFD model capable of dealing with complex terrain and sparse forest was developed in OpenFOAM v1.7. A RANS approach coupled with a modified k-ε turbulence model accounting for extra turbulence generated by the forest was used. To estimate LiDAR bias, the method proposed by Bingöl *et al.* (2008) was implemented as a post-processing tool.

First, a simple verification of the model was carried out by modeling neutrally stratified boundary layer. Apart from the usual overshoot of k in the near-wall cells, results agreed well with analytical solutions. Then, the modifications brought to the solver to account for the effects of forest were validated. In order to do so, flow over and within a dense forest was modeled and results were compared to experimental data of Amiro (1990) and numerical results of Dalpé and Masson (2008). An innovative top boundary condition totally independent of the forest displacement height developed by Lussier-Clément (2012) was also validated. The experiment showed that, in the presence of forest, imposing fixed values for U, k and ε at the top of the domain is not appropriate. The LiDAR bias post-processing algorithm was also validated for flow over an isolated Gaussian hill. The effects of the scanning height as well as the slope of the hill were investigated. For terrain slopes ranging from $\sim$25% to $\sim$43%, LiDAR bias ranging from 2% up to 10% was observed. The generalization of the method for large areas revealed to be particulary useful at showing the extant of the bias.

Finally, a real case scenario was studied where a LiDAR was sited in the Gaspe peninsula on a complex and densely forested terrain. The assessment of the CFD model for this site firstly revealed the significant impact of both the location and nature of the inlet boundary condition. Then, the LiDAR bias was estimated with the help of OpenFOAM v1.7, MeteoDyn WT 4.0 and WAsP Engineering. Numerical results were compared to experimental data. Despite the presence of terrain complexity up to a distance of eight times the radius of the scanned disc around the remote sensor, very little error was observed, suggesting that the LiDAR is only affected by topographic variations closer to the scanned volume.

Keywords: LiDAR, Complex Terrain, OpenFOAM, CFD

ESTIMATION OF LIDAR BIAS OVER COMPLEX TERRAIN USING NUMERICAL TOOLS

Eric JEANNOTTE

RÉSUMÉ

Depuis quelques années, un nouvel instrument de mesure du vent fait concurrence aux anémomètres à coupelles classiques: le LiDAR. Malgré plusieurs avantages tels que sa facilité à être déployé et sa capacité à mesurer la vitesse du vent à plusieurs hauteurs simultanément, les performances de celui-ci sur terrain complexe sont encore matière à débat. Ceci est principalement dû à l'hypothèse d'homogénéité de l'écoulement que fait le LiDAR pour évaluer la vitesse au centre du disque scanné. L'objectif de ce travail consistait à implanter une méthode pour évaluer le biais LiDAR en terrain complexe à l'aide d'OpenFOAM. Pour accomplir cette tâche, un modèle numérique capable de simuler la couche limite atmosphérique en terrain complexe recouvert de forêt a été mis en place dans OpenFOAM v1.7. Une approche RANS, jumelée à un modèle k-ε modifié qui prend en compte la turbulence générée par la forêt, a été utilisée. Pour estimer le biais du LiDAR, la méthode proposée par Bingöl *et al.* (2008) a également été mise en place et utilisée comme outil de post-traitement.

Dans le but de vérifier le modèle, une couche limite atmosphérique neutre et homogène a tout d'abord été simulée. Excepté une surestimation de k près de la paroi, les résultats furent concluant par rapport à la solution analytique. Par la suite, les modifications apportées dans le but de tenir compte des effets de la forêt ont été validées. Pour ce faire, l'écoulement à l'intérieur et au dessus d'une forêt dense a été simulé et les résultats furent comparés aux mesures expérimentales d'Amiro (1990) et aux résultats numériques de Dalpé and Masson (2008). De plus, une condition frontière dévelopée par Lussier-Clément (2012), indépendante de la hauteur de déplacement de la forêt, fut validée. L'expérience a également démontré que fixer des valeurs pour U, k et ε à la paroi supérieure en présence de forêt était inapproprié. L'outil de post-traitement mis en place fut également validé pour un écoulement au dessus d'une colline Gaussienne. Les effets de la hauteur de scan ainsi que de la pente de la colline furent investigués. Pour des pentes variant de $\sim$25% à $\sim$43%, un biais allant de 2% à 10% fut observé. La méthode fut également généralisée à la quasi totalité de la topographie, ce qui se révéla être un outil utile pour évaluer l'étendu du biais.

Finalement, un cas réél où un LiDAR situé sur un site complexe et recouvert de forêt a été étudié. L'évaluation du modèle numérique a d'abord révélé l'importance capitale de la position ainsi que de la nature de la condition frontière d'entrée. Ensuite, le biais LiDAR fut estimé à l'aide d'OpenFOAM v1.7, Meteodyn WT 4.0 and WAsP Engineering et le tout fut comparé à des mesures expérimentales. Malgré la présence de complexité topographique sur une distance de huit fois le rayon du disque scanné autour du LiDAR, très peu d'erreur fut observée, ce qui laisse envisager que le LiDAR n'est affecté que par la topographie très près du volume scanné.

Mots-clés: LiDAR, Terrain complexe, OpenFOAM, CFD

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

AAV	Anse-à-Valleau
ABL	Atmospheric Boundary Layer
AGL	Above Ground Level
CBL	Convective Boundary Layer
CFD	Computational Fluid Dynamics
DBS	Doppler Beam Swinging
FVM	Finite-Volume Method
LES	Large Eddy Simulation
LOS	Line of Sight
NBL	Neutral Boundary Layer
SBL	Stable Boundary Layer
SST	Shear Stress Transport
TI	Turbulence Intensity
TKE	Turbulent Kinetic Energy
VAD	Velocity-Azimuth-Display

LIST OF SYMBOLS

a,b,c	Parameters for rectified trigonometric function
B	Offset angle between LIDAR North and true North, °
C_d	Drag coefficient of forest
$C_{1\varepsilon}$	Model constant of k-epsilon model
$C_{2\varepsilon}$	Model constant of k-epsilon model
$C_{4\varepsilon}$	Model constant of $k-\varepsilon$ forest model
$C_{5\varepsilon}$	Model constant of $k-\varepsilon$ forest model
C_μ	Model constant of k-epsilon model
c_p	Specific heat at constant pressure, J/kg·K
d	Height of inlet for computations over complex terrain, m
g	Gravitational constant, m/s^2
G_k	Production turbulent kinetic energy, Pa/s
G_{kP}	Production turbulent kinetic energy of near-wall cell-centre
h	Displacement height of vegetation, m
H	Height of Gaussian hill, m
k	Turbulent kinetic energy, m^2/s^2
k_P	Turbulent kinetic energy of near-wall cell-centre, m^2/s^2
l	Characteristic length, m
L	Distance between summit and inflection point of Gaussian hill (half-width), m
LAI	Leaf area index
p	Pressure, Pa
p^*	Non-dimensional pressure
u,v,w	Wind speed components, m/s
U	Velocity field, m/s
U,V,W_{LiDAR}	Wind components as calculated by LiDAR according to Bingöl's linear approach, m/s

u^*	Friction velocity, m/s
u_g^*	Local friction velocity, m/s
S_ε	Source term of $k-\varepsilon$ forest model
SH	LiDAR scanned height, m
S_k	Source term of $k-\varepsilon$ forest model
S_{phi}	Source term of general transport equation
S_u	Source term of forest model
t	Time, s
T	Temperature, K
$\overline{u_i'u_j'}$	Reynolds stresses
U^*	Non-dimensional velocity
U_∞	Free stream velocity, m/s
U_P	Velocity of near-wall cell-centre, m/s
U_w	Velocity at the wall, m/s
$V_{rN,rS,rE,rW}$	Radial wind velocity in North, South,East and West direction, m/s
V_h	Horizontal wind speed, m/s
x^*	Non-dimensional position
z	Altitude, m
z_P	Height of near-wall cell-centre, m
z_0	Roughness lenght, m
α	Leaf area density, m^2/m^3
α_p	Permeability of a porous media, m^2
α_p^*	Non-dimensional permeability
β	Inertial coefficient, m^{-1}
β_d	Model constant of $k-\varepsilon$ forest model
β_p	Model constant of $k-\varepsilon$ forest model
β_p^*	Non-dimensional inertial coefficient

γ	Wind flow inclination angle, °
Γ	Diffusion coefficient of general transport equation
δ_{ij}	Kronecker delta
ε	Dissipation, m^2/s^3
ε_P	Dissipation of near-wall cell-centre, m^2/s^3
θ	Wind direction, °
Θ	Potential temperature, K
κ	Von Karman constant
λ	Ratio of scanning height and distance between summit and inflection point of Gaussian hill
μ	Molecular viscosity, kg/m·s
ν	Kinematic viscosity, m^2/s
ν_t	Turbulent viscosity, m^2/s
ν_{tP}	Turbulent viscosity of wear-wall cell-centre, Pa.s
φ	Ratio of Gaussian hill height and distance between summit and inflection Point of gaussian hill (H/L)
ρ	Air density, kg/m^3
σ_k	Model constant of k-epsilon model
σ_ε	Model constant of k-epsilon model
τ	Shear stress, Pa
ϕ	General flow variable
ϕ'	Fluctuating part of given flow property
ϕ_l	Cone half opening angle, °
ϕ_{FP}	Given flow property
Φ	Steady part of given flow property
Ω	Angular velocity of the Earth, rad/s

INTRODUCTION

Background

Having experienced an impressive growth in the last decade, IEA reported that wind energy was contributing to a non-negligible amount of $\sim$3% of the world's electricity demand at the end of 2011. Canada, for instance, added 1298 MW of wind energy capacity in that same year representing an investment of roughly 3 billion CAD. With a total installed capacity of more than 5200 MW, wind energy contributed to $\sim$2.5% of the country's demand in electricity, thus showing the important role of the wind industry in both the economic and energy sectors.

Implementation of wind farms calls for significant investments which explains the crucial importance of carefully analyzing the financial aspect of each step of a wind project. The quantification of the resource, which comes early in the process, is thus a very important aspect to consider as it has a direct impact on the energy output of the project, which is closely linked to the economic output. To measure wind speed, there exists mainly four types of wind measuring instruments: cup anemometers, sonic anemometers, propeller anemometers and Doppler sensors (SODAR and LiDAR) (Manwell *et al.*, 2009). Apart from Doppler sensors, these instruments require to be mounted on meteorological masts, which represents important investments. Moreover, masts are vulnerable to harsh climate (high winds, precipitations) and their relocalization is complicated. For these reasons, another type of wind measuring instrument has recently come into play: remote sensing devices. Based on the Doppler Effect, these devices are more commonly known as SODAR (Sonic Detection and Ranging) and LiDAR (Light Detection and Ranging). These instruments emit sound or light in the atmosphere and measure the frequency shift of the reflected waves. From this shift in frequency, the apparatus is able to calculate radial wind speeds. A more detailed description of LiDARs functionning will be presented in chapter 1.

SODAR and LiDAR are very interesting options for the industry. For instance, one can think of ease of deployment, the capacity to obtain wind speeds at high altitudes (up to 200m) and the possibility to obtain measurements at multiple heights simultaneously. Over relatively flat terrains, remote sensing devices yield good results when compared to cup anemometers (Briggs *et al.*, 2011; Lindelöw-Marsden, 2009). Despite these advantages, there is one main drawback associated with them. While retrieving the wind speeds, they assume that the measured wind flow is homogenous within the sampled volume, which leads to serious bias when used over complex terrain (Bingöl *et al.*, 2008; Arranz, 2011).

Objectives and methodology

LiDARs used in the wind industry generally rely on the the flow homogeneity assumption to retrieve the wind components and consequently, a positive or negative bias is observed when these instruments are used in situations where the flow is distorted (Lindelöw-Marsden, 2009; Bingöl *et al.*, 2008; Bradley, 2008). Among many suggestions to correct and/or a predic LiDAR bias, one implies modeling the bias with the help of computational fluid dynamics (CFD). The main objective of this work is to implement a method to estimate the LiDAR bias, due to the flow homogeneity assumption, over complex terrain using the open source CFD toolbox OpenFOAM. It was accomplished through the kind collaboration of the following partners: GL Garrad Hassan, Leosphere, NRG Systems and TechnoCentre éolien and Cartier Énergie Éolienne.

Unfortunately, no such thing as an atmospheric model over complex terrain is present in OpenFOAM, however the CFD package offers all the tools necessary to implement what is needed. Therefore, the work can be separated in two specific aspects:

- The implementation of a numerical model for the neutral atmospheric boundary layer over complex terrain covered with forest;
- The implementation of a post-processing algorithm for modeling the LIDAR bias over complex terrain.

Since the two above-mentioned tasks are very distinct, the methodology will be separated in two parts to avoid confusion. In the future, the developed model could then be used to assess the modeled bias.

Thesis organization

This master's thesis is divided in four chapters: introduction, methodology, model validation and verification and finally, application to complex terrain.

Despite the significant presence of CFD throughout this work, the review of litterature is entirely devoted to remote sensing as it is the principal interest. More precisely, remote sensing basic principles are covered and a presentation of the two major types of LiDARs present in the industry is done. It is also explained why the use of LiDARs over complex terrain can be problematic and how the bias can be estimated with the help of various methods.

The methodology describes physical phenomena involved in atmospheric boundary flow and how these are implemened in the mathematical model. The LiDAR bias post-processing algorithm is also explained and described in this section.

The third chapter first presents a basic verification of the model for neutrally stratified boundary layer, followed by a validation of the implementation of the forest model and concludes with a validation of the LiDAR bias post-processing algorithm over an isolated Gaussian hill.

Finally, the last chapter is dedicated to the application of the developed model to a real complex terrain situation, Anse-à-Valleau (AAV), where both the CFD model and the LiDAR bias are evaluated.

CHAPTER 1

REVIEW OF LITTERATURE

1.1 Remote sensing basic principles

Coherent LiDARs operation is based on the Doppler Effect, which is a commonly known phenomenon for acoustic waves and electromagnetic waves. Basically, the Doppler Effect states that a change in perceived frequency will occur if a source and a listener (or observer) are moving relative to each other (Werner, 2005). In the case where the observer and the source are approaching each other, the perceived frequency increases. The opposite happens when they are moving away from each other. Currently, two major types of LiDARs are used in the wind industry: pulsed LIDARs and continuous-wave LIDARs (Hill, 2010). The main differences between both instruments is explained below.

1.1.1 Pulsed LiDARs

As described by Cariou and Boquet (2010), pulsed LiDARs send many pulses of light into the atmosphere. The light is then reflected on naturally present particles in the air, and comes back with a modified frequency. The travel time of the pulses from the moment it leaves the instrument to the moment it comes back is monitored in order to determine the height at which measurements are taken. Signal processing equipment within the device is responsible for selecting only part of the received data that corresponds to the desired measurement heights. From the data obtained after processing, it is then possible to derive the horizontal and vertical components of the wind. For complete explanations of signal processing regarding pulsed LiDARs, the reader may refer to Cariou and Boquet (2010).

In the case of a WINDCUBE [1] for instance, a scanning technique called Doppler beam swinging (DBS) is used. The path followed by the laser beam in each direction is called the line of sight (LOS), and the velocities calculated by the LiDAR along this path are called radial velocities or abbreviated V_{los} or V_r. At approximately every 1.5 seconds, the laser is rotated 90° and a new radial velocity is obtained. After four iterations of this process, four radial velocities located on the same disc are obtained. Considering the coordinate system depicted in figure 1.1, the following equations can be derived

[1] The WINDCUBE is a LiDAR developed by the company LEOSPHERE.

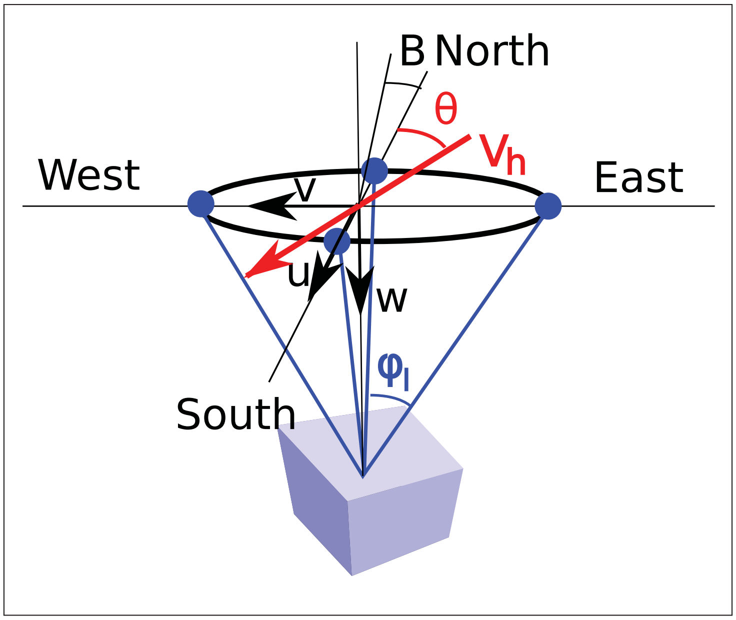


Figure 1.1 Lidar geometry

$$V_{rN} = u \sin\phi_l + w \cos\phi_l, \tag{1.1}$$

$$V_{rE} = v \sin\phi_l + w \cos\phi_l, \tag{1.2}$$

$$V_{rS} = -u \sin\phi_l + w \cos\phi_l, \tag{1.3}$$

$$V_{rW} = -v \sin\phi_l + w \cos\phi_l. \tag{1.4}$$

Where ϕ_l is the cone angle. One can solve for u, v and w and the result is

$$u = \frac{V_{rN} - V_{rS}}{2 \sin\phi_l}, \tag{1.5}$$

$$v = \frac{V_{rE} - V_{rW}}{2 \sin\phi_l}, \tag{1.6}$$

$$w = \frac{V_{rN} + V_{rE} + V_{rS} + V_{rW}}{4 \cos\phi_l}. \tag{1.7}$$

Therefore, the horizontal velocity (V_h) and the direction of the wind (θ) can be obtained as follows

$$V_h = \sqrt{u^2 + v^2}, \tag{1.8}$$

$$\theta = tan^{-1}\frac{u}{v}. \tag{1.9}$$

Note that a direction offset (B) must be included if the instrument is not aligned relative to true North. A complete numerical example showing how to calculate horizontal wind speed and wind direction from WINDCUBE raw data is presented in Appendix I.

1.1.2 Continuous-wave (CW) LiDARs

As opposed to pulsed LiDARs, and as the name implies, the laser beam in this type of instrument is continuous. In order to determine the velocity at a specific height, CW LiDARs focus the laser beam at the desired height. It is important to note that the diameter and the length of the focused zone increase as height increases. Ultimately, at heights above several hundreds of meters, it becomes impossible to establish a focused zone (Hill, 2010). Again, signal processing is done with the help of many electronic parts within the LiDAR and the reader is encouraged to consult Hill (2010) for complete explanations.

The scanning technique used by the ZephIR[2] to retrieve the wind components is called a conical scan or velocity-azimuth-display (VAD) (Hill, 2010). The process is very well documented by Hill (2010) and Werner (2005). One conical scan consists of 50 radial velocities probed at different azimuth angles. At the end of one scan, 50 readings are available. Then a cosine function of this form

$$v_r = |a\cos(\theta - b) + c| \tag{1.10}$$

is fitted to the data obtained, which consists of radial velocities in terms of azimuth angles. To extract the wind components from the trigonometric function, the following relations are used, where ϕ_l is the cone angle

$$V_h = -a/sin(\phi_l), \tag{1.11}$$

$$w = -c/\cos(\phi_l). \tag{1.12}$$

The algorithm has of two solutions separated by 180° for the wind direction, however this issue is resolved with the help of a ground based anemometer. The fitting process can be done with up to 150 samples and the algorithm used is usually a least square fit.

Results can be presented in a number of ways. Among these, a polar plot, often called the “figure of height”, gives useful information to the user. The shape of the latter figure can be used to detect flow inhomogeneity as described in Hill (2010).

[2] The ZephIR is a LiDAR developed by the company QinetiQ.

1.2 LiDAR flow uniformity assumption

The process used by LiDARs to calculate the wind components rely on many assumptions and each step leading to these components introduces uncertainties. This process, used to retrieve wind speed, can be crudely decomposed in three phases: measurement, extraction, and interpolation. In the measurement phase, the uncertainty is mainly due to the hardware that composes the remote sensor. In the extraction process, the bias comes from the difficulty to obtain a representative wind distribution at a given height since the sampling volume is stretched over several meters. Finally, interpolation deals with averaging the wind components over a volume with the help of the techniques described previously (DBS and VAD). Of course, there are many factors that can badly affect these 3 tasks, such as precipitations, clouds, probe volume effects, range resolution, inhomogeneous aerosol scatter distribution, etc. (Lindelöw-Marsden, 2009). Many authors agree (Lindelöw-Marsden, 2009; Bingöl *et al.*, 2008; Bradley, 2008) to say that the interpolation part causes considerable problems over complex terrain. This is due to the assumption of flow homogeneity and it is this latter topic that is investigated in the present work. From this point, the bias caused by the averaging process will be refered to as LiDAR bias.

As explained earlier, the size of this scanned disc varies with height and depends on the opening angle θ_l. The homogeneity assumption shows good results on flat terrain but yields often large bias on complex terrain as shown in table 1.1 on page 16. Before going any further, it is essential to define what will be considered a "complex terrain" in the present work. The definition used will be the one proposed by IEC 61400-12-1 (2005), which states:"terrain [...] that features significant variations in topography and terrain obstacles that may cause flow distortion".

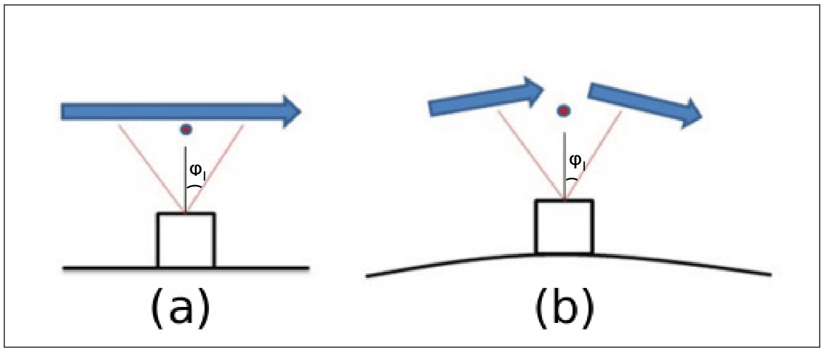


Figure 1.2 Uniform flow VS non uniform flow

Coming back to the interpolation problem, it is interesting to note that the time period over which the average is done varies depending on the method used to retrieve the wind components (VAD of BDS) but is on the order of a few seconds. As figure 1.2 (a) shows, the issue is less problematic when the flow is homogeneous since there is almost[3] no change in horizontal neither vertical wind speed components within the scanned disc. On the other hand, when there is a change in one of the wind components, the flow homogeneity assumption is no longer valid. Bingöl *et al.* (2008) prove it mathematically for a case where the flow can vary linearly in any direction. The conclusions are:

- The calculated horizontal wind components are independent of the cone opening angle. They are however dependent on the horizontal gradient of vertical velocity;
- The vertical component is dependent on the cone angle.

Boquet *et al.* (2010) experimentally tested the effect of the opening angle and concluded that results were best when the angle was comprised between 15 and 30 degrees. The authors explain these results as follows: having a small ϕ decreases the impact of the flow homogeneity assumption since the averaging volume size decreases. However, it decreases the quality of the projection of the wind components on every beam.

1.3 Suggested solutions to LiDAR bias

Many options are available to deal with the bias and much information exists in the literature about this topic. These options will be presented in this section.

1.3.1 Linear approach

The most documented methods are the ones suggested by Bingöl *et al.* (2008). The first one will be called the "linear approach" due to the nature of the method. As briefly discussed in the previous section, this approach assumes that the velocity field $\mathbf{U}(u,v,w)$ varies linearly within the LiDAR scanned volume such that

$$U_i(x) = U_i(0) + x_j dU_i/dx_j. \tag{1.13}$$

[3] Note that turbulence and vertical shear over flat terrain can also introduce a deviation between wind speed measured by a LiDAR and a cup anemometer, especially when the diamater of the scanned disc becomes large (Lindelöw-Marsden, 2009).

x=0 is the center of the scanned disc at scanning height SH. The radial wind speed V_r is then calculated by projecting the velocity field on the direction vectors of each laser beam.

$$\mathbf{n} = (\cos\theta \sin\phi_l, \sin\theta \sin\phi_l, \cos\phi_l) \tag{1.14}$$

Finally, Bingöl *et al.* (2008) fit a trigonometric series of the form a + b$sin\theta$ + c$cos\theta$ to the resulting radial wind speeds to retrieve the cartesian wind components:

$$U_{lidar} = U + SH\frac{\partial W}{\partial x}, \tag{1.15}$$

$$V_{lidar} = V + SH\frac{\partial W}{\partial y}, \tag{1.16}$$

$$W_{lidar} = W - \frac{1}{2}\tan^2(\phi_l)\frac{\partial W}{\partial z}. \tag{1.17}$$

where x, y are the horizontal directions and z is vertical. Notice that the horizontal wind components are independent of the opening angle ϕ_l. The second term on the right of these equations represents the remote sensor bias due to flow homogeneity assumption. Therefore, if one is able to determine this term, it is then possible to make a correction to the estimated wind components. The term is composed of the height at which the measurement is done multiplied by the gradient of the vertical velocity. The gradient of the vertical velocity can be calculated using different techniques. One technique is proposed in IEC 61400-12-1 standard working draft 1. The horizontal gradients are first rearranged in terms of vertical flow inclination angles $\gamma_1$and γ_2. Considering the simple setup shown in figure 1.3, the gradient of vertical velocity can be expressed as

$$\frac{dW}{dx} = \frac{w_2 - w_1}{\Delta x}. \tag{1.18}$$

Thus, equation (1.15) can then be written as

$$\frac{u_{lidar} - u}{u} = SH\frac{w_2 - w_1}{u\Delta x}. \tag{1.19}$$

Then, with the help of the following relations

$$\Delta x = 2SH\tan\phi_l, \tag{1.20}$$

$$w_1 = u\tan\gamma_1, \tag{1.21}$$

$$w_2 = u\tan\gamma_2, \tag{1.22}$$

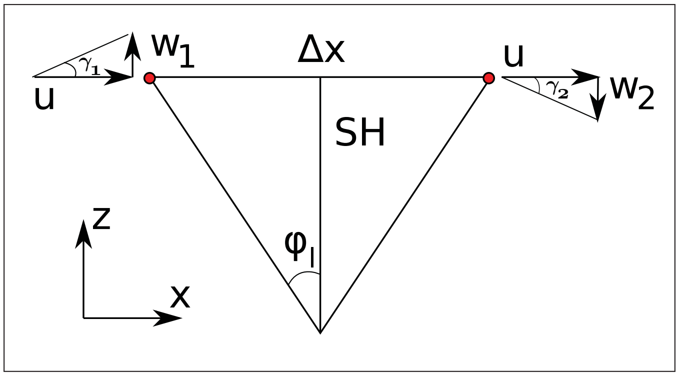


Figure 1.3 2D scanning arrangement

the remote sensing measurement bias can be rearranged only in term of the cone half opening angle and the inflow angle

$$\frac{u_{lidar} - u}{u} = \frac{\tan(\gamma_2) - \tan(\gamma_1)}{2\tan(\phi_l)}. \tag{1.23}$$

In order to determine the vertical flow inclination angle, the IEC standard working draft 1 relies on the terrain inclination. More precisely, it calculates, using a best fit algorithm, the average slope of the terrain in a radius of five times the measurement height around each beam probe. The method was not tested in this work but according to the standard working draft 1, measurement bias ranging from 0.2% to 0.5% were obtained in an area with “gentle slopes but important variations in terrain” (IEC 61400-12-1, 2005, p.117).

Horizontal gradients of vertical velocity can also be calculated with the aid of CFD. Bingöl *et al.* (2008) in their article used WAsP Engineering to validate the method. Meteodyn WT 4.1 was used by Pelletier *et al.* (2011). Since the procedure to calculate the gradients varies from software to software, no method describing the way to get these will be described in this work.

1.3.2 Modeling LiDAR bias: CFD Approach

Another approach also proposed by Bingöl *et al.* (2008) consists of modeling the effects of the homogeneity assumption with the help of numerical flow analysis tools. Basically, a wind flow simulation is run over the terrain where the instrument is deployed. The cartesian wind components (u, v, w) are obtained at the beam focused zones. These are projected on the unit normal vectors of the beams resulting in radial velocities. From these, the horizontal wind speed, the wind direction and the vertical wind speed (all at the center of the LiDAR scanned circle) can be derived using the equations presented in the section 1.1.1 and 1.1.2. The obtained wind velocity, different from the one obtained by the CFD at that same location, thus takes into account the bias caused by the flow homogeneity assumption. Result is usually presented as the ratio of those two velocities. The technique was notably used by Arranz (2011) and Harris *et al.* (2010). As seen in the graph below, the method has the advantage of clearly showing the over- or under-estimation.

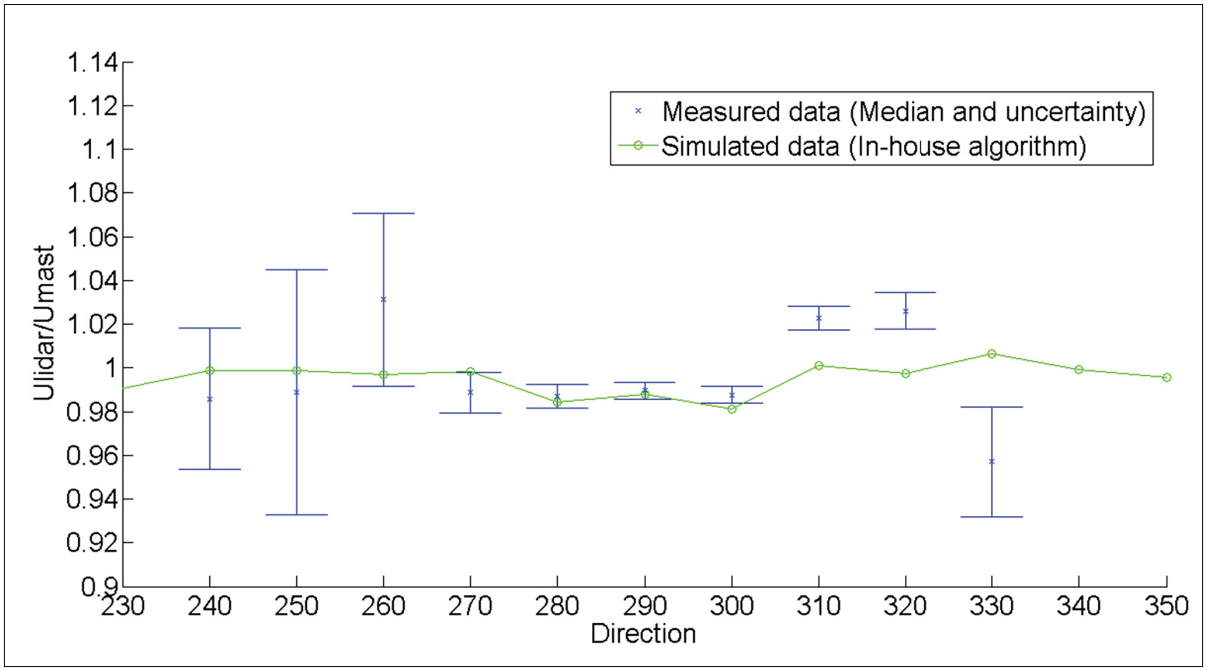


Figure 1.4 Modeled LiDAR bias obtained with Meteodyn (Jeannotte *et al.*, 2011)

1.3.3 Modeling LiDAR bias: WAsP script

The modeling method described in section 1.3.2 was implemented by Bingöl *et al.* (2008) in a WAsP Engineering Script (WES) that is commercially available. Even though the technique is exactly the same, the script uses WAsP Engineering which is a linearized flow model that does not solve the Reynolds Averaged Navier-Stokes (RANS) equations iteratively as done by traditional CFD softwares. Nonetheless, it was tested in this work and results will be presented in chapter 4.7. Despite being user friendly, the script lacks some important features such as an easy way to account for the orientation of the LiDAR with true north. The script does not consider possible spatial separation between mast and LiDAR either.

1.3.4 Meteodyn correction tool

Meteodyn WT also provides a commercial tool sold as an add-on to their CFD package to correct LiDAR bias in complex terrain. Although it is very easy to use for the regular Meteodyn WT users and presents some nice features such as the choice of different forest models and thermal stabilities, there is no access to the correction algorithm. The output is a 10-minute corrected time series that is generated from the LiDAR 1-second time series (raw output files). Regarding the different input parameters (thermal stabilities, forest densities and roughness), Boquet *et al.* (2011) tested their sensitivity on the correction done by Meteodyn correction tool and came to the conclusion that their effect is weak. This can be partly explained by the fact that the correction is done using relative information of opposed beams rather than absolute speed up factors.

1.3.5 LEOSPHERE FCR: Flow Complexity Recognition

LEOSPHERE FCR technology consists of an algorithm directly built in the WINDCUBE LiDAR that calculates flow variations within the sensed volume using radial velocities at all heights simultaneously. An additional 5th vertical laser beam is also used to obtain the vertical wind speed. According to the manufacturer, LiDAR bias is brought down to <1% (Boquet, 2012). Unfortunately, not much details about the algorithm is present in the literature yet. The FCR module was introduced in early 2012 by LEOSPHERE and was not tested in this report.

This method is very promising for many reasons

- Time efficient since corrections are done in realtime;
- Does not make flow homogeneity assumption;
- Eliminates all uncertainties related to CFD (as it does not use CFD).

The last point is interesting, but it is important to mention that other assumptions (undisclosed) are made throughout the process of flow recognition.

1.3.6 Using multiple LiDARs

Mann *et al.* (2009) conducted an experiment, named "The LiDAR Musketeer Experiment", using three LiDARs pointing at the same location, thus eliminating the need of averaging over a given volume. A sonic anemometer was also used to evaluate the remote sensors' measurements. The test was conducted with three WINDCUBE LiDARs. One was collocated to the mast and pointing upward retrieving the vertical wind speed whereas the two others were positioned with an angle of 56° with respect to the vertical. In order to compare the LiDAR measurements to those of the anemometer, the latter were projected onto the directions of the beams of each remote sensor. Results generally show a very good agreement between remote sensors and the anemometer except for the apparatus scanning vertically which was believed to be misaligned. Of course, this setup eliminates the flow homogeneity assumption but the requirement of three LiDARs makes it rather costly. However, it opens the door to precise 3D turbulence analysis around wind turbines as suggested by Mann *et al.* (2009).

1.4 Synthesis of LiDAR performance over flat and complex terrain

In this section, a summary of the performances of different LiDARs over flat and complex terrain is presented. A couple of precisions are needed regarding the content of the following table. For each case, the corresponding test site and its complexity are shown. The complexity is qualified according to the terms employed by the author of the study. The observed bias is presented as a linear fit between LiDAR and cup measurements.

The table also presents the modeled bias, the technique and the software used to obtain these results. As shown in figure 1.4, the modeled bias is calculated per wind direction. In the following summary table, only minimum and maximum bias values are given. The names of the techniques are the same as the ones described in the previous section. Finally, bias after correction is also shown when it applies, again presented as linear fits unless other precisions mentionned. Some LiDARs can be identified by the scanning height, so it was also included when available.

Table 1.1 revealed that only a few parties used numerical tools to model LiDAR bias and even less attempted a correction. However, the results obtained by Harris *et al.* (2010) and Briggs *et al.* (2011) suggest that using CFD might be a promising approach for correcting LiDAR bias.

Table 1.1 Summary table of different institutions/parties that measured and (sometimes) corrected LiDAR measurements

Author	**LiDAR**	**Site**	**Complexity**	**Observed Bias (Measurements)**	**Observed Bias (Modelled)**	**Bias after Correction**
Bingöl *et al.* (2008)	ZephIR (10-150m)	Lavrio (Greece)	Gently Sloping	@32m y=0.94x @76m y=0.96x	Linear Approach: @32m:-6% to 1% @76m:-5% to 1% CFD Approach (WES): @32m:-7% to 3% @76m:-5% to 2%	NA
	ZephIR (10-150m)	Panahaiko (Greece)	Very Complex	@30m y=0.93x @53m y=0.96x	Linear Approach: NA CFD Approach: (WES): @32m:-8% to 0% @76m:-10% to 0%	NA
Foussekis (2011)	ZephIR (soft. 2.0)	CRES Test Station (Greece)	Mild with gentle slopes and low vegetation	@54m y=0.9376x @78m y=0.9408x @100m y=0.9519x	NA	NA
	WINDCUBE with 27.7° prism			@54m y=0.9427x @78m y=0.9457x	NA	NA
	WINDCUBE with 15° prism			@54m y=0.9344x @78m y=0.9457x	NA	NA
	ZephIR (soft. 2.0)	Central mountainous Greece 1500m asl (above Sea level)	Very Complex	@20m y=0.9314x	NA	NA
	WINDCUBE	Coastal Greece 1000m asl	Very Complex	@40m y=0.94x	NA	NA

Continued on next page

Author	LiDAR	Site	Complexity	Observed Bias (Measurements)	Observed Bias (Modelled)	Bias after Correction
Arranz (2011)	ZephIR	Albacete, Spain	Complex (Hills covered with forest)	@79m y=0.976x	CFD Approach (WES): @79m:-8% to -4% CFD Approach (CFDWind2.0)[4]: @79m:-5% to 0%[5]	NA
Wagner (2010)	WINDCUBE	Hosvore, west coast of Denmark	Flat surrounded by farmland	@90m y=1.013x	NA	NA
				@90m y=0.990x	NA	NA
Harris *et al.* (2010)	ZephIR	Southern part of the U.K.	small plateau with steep drops on all sides	@60m y=0.9711x	CFD Approach Ventos:-2%to-4% WES: -2%to-7%	Adjusted with: Ventos y=0.9982x
Briggs *et al.* (2011)	WINDCUBE V2	Northeast region of the United States	Flat (farmland with scattered trees)	@80m y=1.007x	NA	NA
		Northwest region of the United States	Complex(elevated areas divided by deep ravines)	@80m y=0.976x	NA	Adjusted[6] with: WAsP:1% In-House:-2% Meteodyn:0%

[4] CFDWind2.0 is a code derived from Fluent.

[5] Due to time restriction, only the sector [275°-305°] was tested.

[6] These percentages are averages over all sectors, results were not presented as linear fits.

CHAPTER 2

METHODOLOGY

2.1 Introduction

As previously mentionned, modeling LiDAR bias requires a suitable atmospheric flow model, which is not part of the standard v1.7 release of OpenFOAM used in this work. To implement such a model, it is crucial to examine first some characteristics of the site such as terrain complexity, surface roughness, the presence of vegetation etc., in order to reproduce as faithfully as possible the effects of these via the mathematical model.

The first part of the methodology will be dedicated to physical concepts having an effect on atmospheric boundary flow. It will also present the modifications of the mathematical model required to correctly account for these effects. Some information will also be given regarding the numerical method. The second part of the methodology will be devoted to the explanation of the post-processing algorithm used to evaluate LiDAR bias.

2.2 PART I: CFD MODEL

As opposed to a software based on a linearized flow model (WAsP Engineering) such as the one originally used by Bingöl *et al.* (2008) for modeling LiDAR bias, we opted for an open source CFD package, which uses the finite-volume method (FVM) and that contains multiple solvers and turbulence models called OpenFOAM. Before actually proceeding to the presentation of the physical concepts which affect atmospheric flow, it was judged useful to briefly present some advantages of OpenFOAM

- Licences are free, which greatly facilitates massive parallel calculations;
- The code is accessible and thus can be adapted to specific needs;
- It contains non-linear solvers and turbulence models capable of modeling the atmospheric boundary layer (ABL) as opposed to linear models such as WAsP Engineering.

2.2.1 Physical concepts

2.2.1.1 Atmospheric boundary layer structure

The well-known principle of boundary layer also applies for the flow of air at the surface of the earth, it is called the atmospheric boundary layer (ABL). Although, the notion of ABL is covered by numerous authors, the information here comes mainly from Emeis (2006). The size of the ABL is quite variable but can be estimated to reach 1000m to 3000m above the surface of the earth. This variation in height can be caused by topography, vegetation, heat exchange and other phenomena taking place at the surface. On flat terrain, the ABL can be subdivided in three major layers as depicted in figure 2.1. The lowest and smallest part is the roughness sublayer, which starts from the ground and extends to approximately ten times the roughness length. The roughness length (denoted z_0) is a parameter used to characterize the surface roughness. Theoretically, the wind speed should be zero at z_0. Table 2.1 shows different values of z_0 corresponding to various types of surfaces. Being in direct contact with the soil, this layer experiences important gradients of heat and momentum (Emeis, 2006).

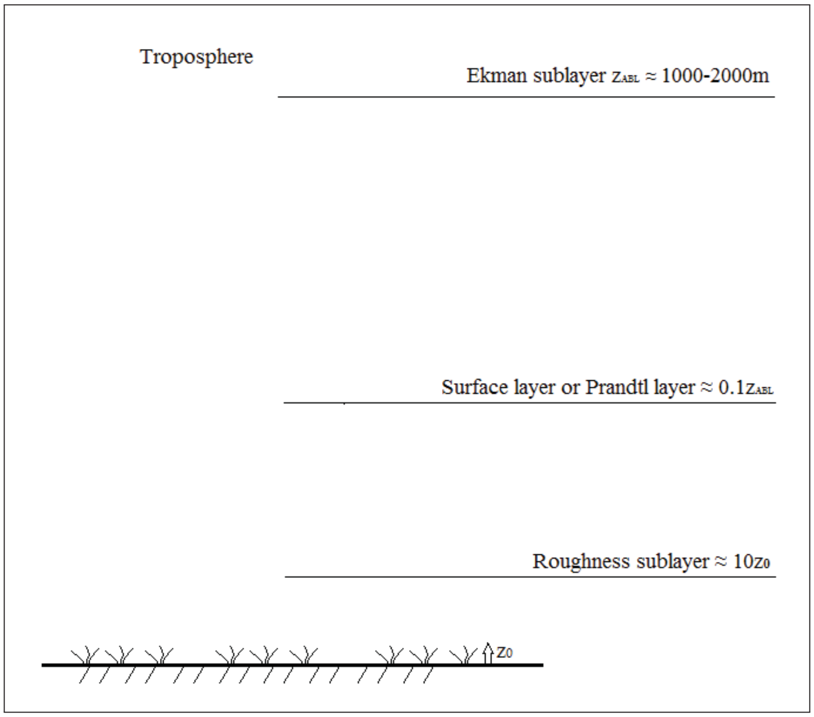

Troposphere
Ekman sublayer $z_{ABL} \approx$ 1000-2000m
Surface layer or Prandtl layer $\approx 0.1z_{ABL}$
Roughness sublayer $\approx 10z_0$
z_0

Figure 2.1 Atmospheric boundary layer structure inspired from (Emeis, 2006)

Table 2.1 Different roughness lengths z_0 for various types of terrain.
Values taken from Manwell *et al.* (2009)

Terrain	$\mathbf{z_0}$[m]
Very smooth ice of mud	0.00001
Lawn Grass	0.008
Crops	0.05
Forest and woodlands	0.5
Centers of cities with tall buildings	3.0

Then, the layer immediately above the roughness sublayer is the Prandtl layer (sometimes called constant-flux or surface layer), which goes up to approximately 10% of the total ABL thickness. In this layer, the effect of the earth's rotation (Coriolis effect) is generally neglected and the turbulent fluxes are taken as constant. If we consider a neutral, horizontally homogeneous and steady surface layer, the wind profile follows a logarithmic expression

$$U(z) = \frac{u^*}{K} ln(\frac{z}{z_0}) \tag{2.1}$$

where K is the Von Karman constant and u^* is the friction velocity. The latter can be expressed as

$$u^* = \sqrt{\frac{\tau_0}{\rho}} \tag{2.2}$$

where τ_0 is the shear stress at the surface and ρ is the density. For a complete derivation of the logarithmic profile, the reader is encouraged to consult Manwell *et al.* (2009)

The last part before the free atmosphere is known as the Ekman layer. In this layer, air motion is the result of an equilibrium between a pressure gradient, the Coriolis force and dissipation frictional forces. As we go up the Ekman layer, the horizontal wind component becomes stronger and stronger until the free atmosphere is reached. In the latter, the wind at this height is called geostrophic wind and can be considered purely horizontal since it results from the equilibrium between pressure and the Coriolis force (Emeis, 2006).

2.2.1.2 Thermal stratification

The movement of the air in the ABL might be considerably affected by the gradients of temperature that occur during the day. The main heat input comes from the sun, which in turn heats up the ground. The ABL can experience three types of scenarios in terms of temperature distribution: convective boundary layer (CBL), stable boundary layer (SBL) and neutral or unstable boundary layer (NBL).

CBL

This kind of ABL forms during the day shortly after sunrise, when the sun starts heating the ground. The heat stored in the ground is transmited to the air in direct contact with it. It results in a vertical motion of the air by natural convection and consequently explains the name Convective Boundary Layer. In the presence of cooler clouds, downward air motion can also occur at the top of the ABL, generating turbulence (Stull, 1988).

NBL

A NBL happens when there is no contribution to motion from buoyancy forces in the ABL. The scenario rarely occurs but is nonetheless plausible. It can be observed in the presence of strong winds and clouds, which prevent the sun from heating up the ground (Stull, 1988).

SBL

Finally, an SBL occurs generally at night, when the ground is colder than ambient air. The flow in proximity of the surface becomes quite slow and turbulence decreases. However, under certain conditions, air a bit farther from the ground can accelerate to form what is called nocturnal jet, thus increasing turbulence at that level (Stull, 1988).

To better understand the temperature distribution within the ABL under the 3 different thermal stratifications cases, let's take a look at the potential temperature distribution as expressed by Kaimal and Finnigan (1994)

$$\frac{\partial \Theta}{\partial z} = (\frac{\partial T}{\partial z} + \frac{g}{c_p}) \tag{2.3}$$

where Θ is the potential temperature, T is the temperature, z is the altitude, g is the gravitational constant and c_p the specific heat at constant pressure. So

- For CBL: $\frac{\partial \Theta}{\partial z} > 0$;
- For NBL: $\frac{\partial \Theta}{\partial z} = 0$;
- For SBL: $\frac{\partial \Theta}{\partial z} < 0$.

These relations can be easily understood considering figure 2.2. Notice the difference in the normalization height. The end of the CBL is easily identifiable, it happens where the profiles

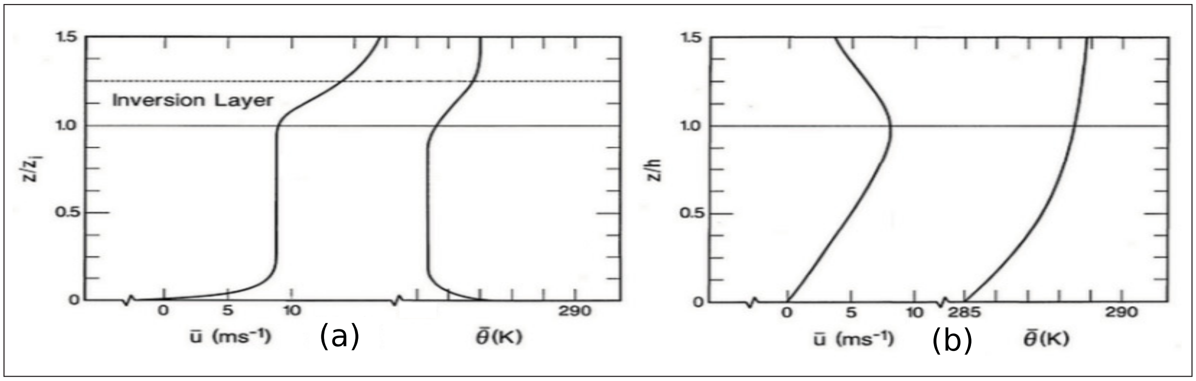


Figure 2.2 Wind speed and potential temperature profiles for the CBL (a) and SBL (b), adapted from Kaimal and Finnigan (1994)

start to drift off their constant values. It marks the beginning of the inversion layer, which was used to normalize the altitude. In the case of the SBL, the transition is not as obvious. The height used to normalize should in theory correspond to the point where turbulence dropped to 5% of the value at the surface. (Kaimal and Finnigan, 1994)

2.2.1.3 Topographic and roughness maps

The shape of the terrain greatly impacts atmospheric flow. In order to properly model the effect of topography, inputs such as topographic and roughness maps are necessary. GLGH provided these maps in a number of formats. They are used as an input for the CFD simulations. The format used in this project is ".xyz", which consists of a list of coordinates X and Y associated to a given height value or roughness value, depending if it is a topographic or roughness map. The resolutions of the topographic and the roughness maps were respectively 20m and 25m.

2.2.2 Mathematical model

2.2.2.1 Navier-Stokes equations

Fluid flow is governed by three fundamental conservation laws: conservation of mass, conservation of momentum and conservation of energy. These three conservation laws contain thermodynamic variables, which are in turn governed by equations of state. Since the flow analyzed in the context of this thesis will be assumed to be independent of the temperature (no heat transfer) and incompressible, the energy equation as well as the equations of state will not be used. The mass and momentum equations remain, respectively[1]

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho U_j)}{\partial x_j} = 0 \tag{2.4}$$

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\delta_{i3} g - 2\varepsilon_{ijk}\Omega_j U_k - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j} \tag{2.5}$$

In these equations, U_j represent the fluid velocity in the three cartesian directions and x_j denote these directions. ρ is the density, g the gravitational constant, p the pressure, Ω the vector of the angular Earth velocity and τ the viscous stresses on the fluid element. These equations can be simplified assuming

- a Newtonian fluid, which simplifies the viscous term (Versteeg and Malalasekera, 1995);
- incompressible flow;
- molecular viscosity is independent of position;
- no effect of gravity;
- no Coriolis effects since LiDAR target heights are monly in the surface layer;
- steady flow.

The continuity and momentum equations can then be written as follow

$$\frac{\partial U_j}{\partial x_j} = 0, \tag{2.6}$$

$$U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j}. \tag{2.7}$$

2.2.2.2 RANS equations

Since we are interested in the mean behavior of the flow within the context of this project, equations (2.6) and (2.7) will be rewritten in their time-averaged form using Reynolds averaging. The principle states that a flow property can be decomposed in a mean part and fluctuating

[1] Equations are presented using Einstein's notation. This notation will be used for the rest of the thesis.

part (Versteeg and Malalasekera, 1995). Mathematically

$$\phi_{FP} = \Phi + \phi' \tag{2.8}$$

$$\Phi = \frac{1}{\Delta t}\int_0^{\Delta t} \phi_{FP}(t)dt \tag{2.9}$$

According to Versteeg and Malalasekera (1995), Δt should approach infinity for the principle of Reynolds averaging to be valid, but considering Δt larger than the longest time scale of the slowest variation of the considered flow system is satisfactory. This yields equations (2.10) and (2.11) where the overbar notation means that the affected term is time-averaged.

$$\frac{\partial U_j}{\partial x_j} = 0 \tag{2.10}$$

$$U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j} \tag{2.11}$$

U_i and P are now averaged values. Notice the apparition of a new turbulent term in the momentum equation, called Reynolds stresses, due to the averaging process applied to the convection term. The bar over this term means that it is time-averaged. This new term adds six unknowns to the system of equations, which complicates its resolution. According to Versteeg and Malalasekera (1995, p. 52), "the terms involve products of fluctuating velocities and consitute convective momentum transfer due to the velocity fluctuations".

2.2.2.3 Modeling turbulence

Turbulence has great influence on ABL flow, especially in complex and forested terrain, and thus it is crucial to model it appropriately. Turbulence is characterized by vortices, also called eddies. Eddies of different sizes and timescales superimpose in the ABL and have direct effects on the mean wind flow. If we look at the wind flow in the ABL for a short period of time, 1-2 hours for example, we will notice that wind speeds vary extensively, mostly due to turbulence. Modeling such variations is not an easy task, and that's why techniques such as Reynolds averaging are used (Stull, 1988).

The system of equations consisting of equations (2.10) and (2.11) involves more unknowns than the number of equations. This problem is known as the "closure problem" and it can be tackled with the help of a turbulence model. A review of the different models will not be done here since it was not part of the objectives of this project, but if the reader is interested in the topic, more information can be found in Versteeg and Malalasekera (1995)

The k-ε model was chosen because it is simple, computionnally cheap and widely used (Versteeg and Malalasekera, 1995). Moreover, good results were obtained for atmospheric wind flow simulation by Alinot and Masson (2005), Dalpé and Masson (2008) and Brodeur and Masson (2008).

The k-ε model uses the well-known Boussinesq approximation to deal with Reynolds stresses. This simplification assumes an anology between the viscous term and the Reynolds stress term as follow

$$\tau_{ij} = -\rho\overline{(u_i'u_j')} = \nu_t \left[\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right] - \frac{2}{3}\rho k \delta_{ij} \tag{2.12}$$

where ν_t stands for the turbulent or eddy viscosity, k is the turbulent kinetic energy (TKE), which can be expressed as

$$k = \frac{1}{3}\overline{u_i'u_j'} \tag{2.13}$$

Notice the similarity with Newton's viscous law where turbulent stresses are proportional to the rate of deformation of the fluid (Versteeg and Malalasekera, 1995). Two additionnal equations are necessary to "close" the problem, one for k and one for the dissipation (ε) respectively (Jones and Launder, 1972)

$$\frac{\partial(\rho k U_i)}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k - \rho\varepsilon + S_k \tag{2.14}$$

$$\frac{\partial(\rho \varepsilon U_i)}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + C_{1\varepsilon}\frac{\varepsilon}{k}G_k - C_{2\varepsilon}\rho\frac{\varepsilon^2}{k} + S_\varepsilon \tag{2.15}$$

Note that the unsteady terms in the above equation are absent due to the assumption of steady flow. The expressions for the turbulent viscosity and production of k (G_k) are:

$$\nu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{2.16}$$

$$G_k = -\rho \overline{(u_i' u_j')} \frac{\partial u_i}{\partial x_j} \tag{2.17}$$

The meaning of each term in equations (2.14) and (2.15) is shown in table 2.2 and all the values of the model constants $\sigma_k, \sigma_\varepsilon, C_{1\varepsilon}, C_{2\varepsilon}$ and C_μ are presented in table 2.3.

S_k and S_ε are source terms for modeling the turbulent effects of forest. They will be further explained in section 2.2.2.4.

Table 2.2 Meaning of each term in k and ε transport equations

Transport by convection	=	Transport by "diffusion"	+	Rate of production	-	Rate of destruction	+	Source

Table 2.3 Model constants for Standard k-ε model

C$_\mu$=0.09	σ $_k$ =1.00	σ $_\varepsilon$ =1.30	C$_{1\varepsilon}$=1.44	C$_{2\varepsilon}$=1.92

2.2.2.4 Modeling the forest

In the final case that will be investigated in this work (chapter 4), the LiDAR is located in a densely forested region, which has an important impact on the flow (Stull, 1988). To model vegetation, a commonly used method relies on imposing a displacement height (h) combined with the appropriate roughness length (z_0) field at the bottom wall of the calculation domain (Dellwik *et al.*, 2006). This method can be implemented using Richards and Hoxey (1993)'s approach. However, selecting proper values for z_0 is not obvious. Therefore, the forest was modeled as a porous media (Dalpé and Masson, 2008, 2009). Even though it was applied only in 1D and 2D over flat terrain, the results presented in (Dalpé and Masson, 2008, 2009) were promising.

When the flow passes through a porous media, it experiences a pressure loss (equation 2.18), which can be taken into account in the momentum equation. Note that the first term on the right side is related to viscous losses whereas the second to inertial losses.

$$\frac{dp}{dx} = -\left(\frac{\mu}{\alpha_p} U_i + \beta \rho U_i |\mathbf{U}|\right), \tag{2.18}$$

where

- μ is molecular viscosity;
- α_p is the permeability [m^2];
- β is the inertial coefficient [m^{-1}] and it can be expressed as the product of C_d and α which in this case refers to a leaf area index. It represents the leaf surface [m^2] per volume of forest [m^3]. It varies with height from one type of forest to another. For more information on how it is calculated and its significance, please refer to Amiro (1990). C_d is a drag coefficient.

Claiming that viscous losses are very small compared to inertial losses in the ABL flow (Dalpé and Masson, 2008), equation (2.18) reduces to:

$$S_u = -\rho C_d \alpha U_i |\mathbf{U}| \tag{2.19}$$

This can be demonstrated by carrying a simple non-dimensional analysis. It is first necessary to establish the five following non-dimensional parameters

$$\alpha_p^* = \frac{\alpha_p}{l^2}, \tag{2.20}$$

$$\beta^* = \frac{\beta}{l}, \tag{2.21}$$

$$x^* = \frac{x}{l}, \tag{2.22}$$

$$U^* = \frac{U}{U_\infty}, \tag{2.23}$$

$$p^* = \frac{p}{\rho U_\infty^2} \tag{2.24}$$

where U_∞ and l are the free stream velocity and a characteristic length respectively. By substituting these in (2.18) and multiplying by

$$\frac{x^*}{U^{*2}},$$

we get

$$\frac{dp^*}{dx^*} = -\frac{1}{Re}\frac{U^*}{\alpha_p^*} + \beta^* U^{*2}, \tag{2.25}$$

which shows that in the presence of a high Reynolds number (as it is the case for ABL flow according to Stull (1988)), the viscous term becomes negligible. Thus,

$$\frac{dp^*}{dx^*} \approx \beta^* U^{*2} \tag{2.26}$$

Expression (2.19) is introduced in the momentum equation as a sink term (S_u) and takes care of properly simulating the pressure loss generated by the forest on the wind flow. Source terms are also needed in the transport equations for k and ε since slightly above the forest, turbulence is increased and so does dissipation within the forest (Masson and Sumner, 2009). Respectively, the source terms for k and ε are

$$S_k = -\rho C_d \alpha \left[\beta_p |\mathbf{U^3}| - \beta_{\mathbf{d}} k |\mathbf{U}|\right], \tag{2.27}$$

$$S_\varepsilon = -\rho C_d \alpha \frac{\varepsilon}{k} \left[C_{\varepsilon 4} \beta_p |\mathbf{U^3}| - \mathbf{C_{\varepsilon 5}} \beta_{\mathbf{d}} k |\mathbf{U}|\right]. \tag{2.28}$$

The above source terms are evaluated only for cells within the forest and are set to zero for all other cells. The standard k-ε constants were modified similarly to Dalpé and Masson (2008). Table 2.4 presents a summary of these constants. These will be used throughout this work in the presence of forest.

Table 2.4 Modified model constants for k-epsilon model with forest

C_μ	σ_k	σ_ε	$C_{1\varepsilon}$	$C_{2\varepsilon}$	$C_{4\varepsilon}$	$C_{5\varepsilon}$	β_p	β_d	κ
0.03	1.00	2.12	1.44	1.92	0.78	0.78	1.0	5.03	0.42

The reader might have noticed that all added source terms are directly proportional to the product $C_d \cdot \alpha$, which demonstrates the importance of these variables on the model. In fact, after much numerical experimentation, Dalpé and Masson (2008) came to the conlusion that what really matters is the product C_d·LAI.

LAI stands for "leaf area index" and is defined by the integral of the leaf area distribution

$$LAI = \int_0^h \alpha dz. \tag{2.29}$$

The LAI will mostly affect the velocity distribution within the forest and the amount of turbulence slightly above the forest. For the case presented in chapter 4, we are interested in velocities much higher than the height of the forest, hence the importance of LAI alone.

Some specific information regarding the treatment of forest in chapter 4 is needed. Since no information was readily available at AAV for the drag coefficient (C_d) nor the leaf area density (α), it was decided to pick a common tree species found in Quebec (NRC, 2011), i.e. blackspruce, and vary the drag coefficient to match experimental results as best as possible. The chosen leaf area distribution, shown in figure 2.3, was part of a set of three distributions analyzed by Amiro (1990). It was non-dimensionalized by the height of the forest.

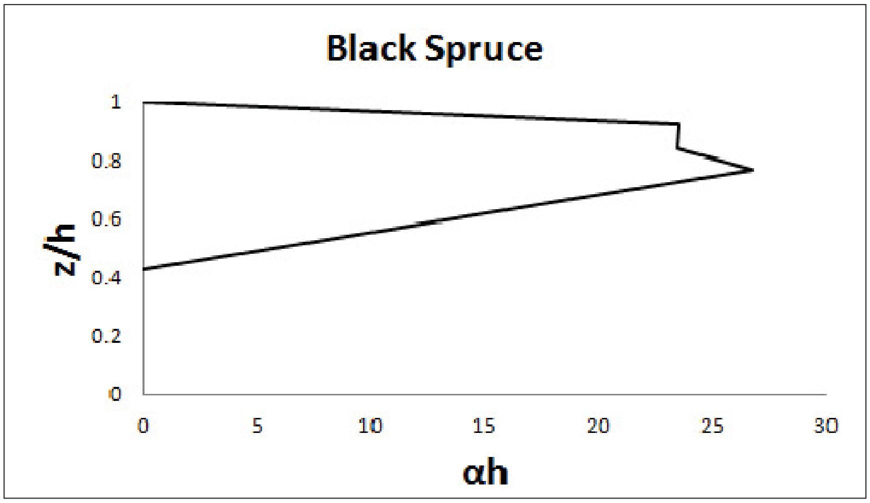


Figure 2.3 Leaf area density distribution is from Amiro (1990)

The location and height of the forest in the computational domain were determined with vector maps provided online by the "Ministère des ressources naturelles du Québec". The classification is shown in table 2.5. The value in the middle of the class was used. Unfortunately, a few areas of AAV contained no information about the forest height. A mean value based on the available data was used for these areas. In order not to model the effects of forest twice (through the roughness length and the porous media approach), a roughness length of 0.05m, which corresponds to a terrain covered with crops according to table 2.1, was applied everywhere forest was present. The roughness and forest input files used by OpenFOAM are generated using a custom C++ utility. Example of this utility's output is shown in figure 2.4.

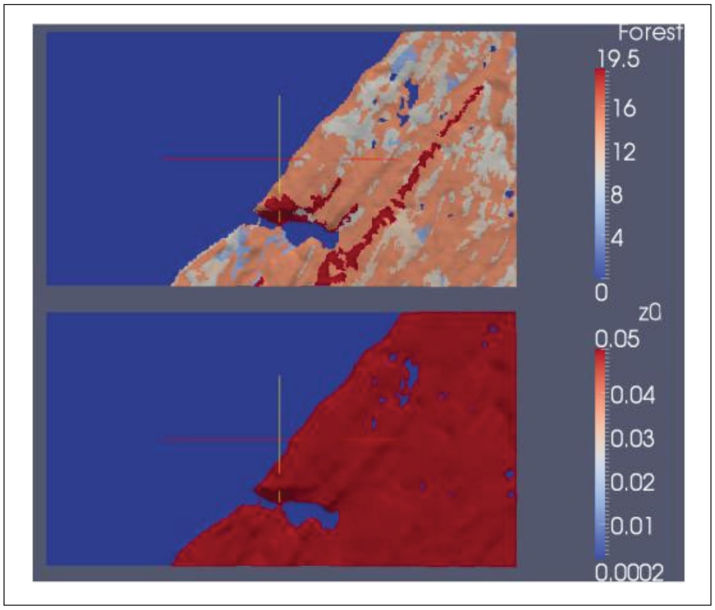


Figure 2.4 Example of roughness length (bottom) and forest height (top) maps used as inputs for simulations

Table 2.5 Forest height classes

Forest Height Class	Height (m)
1	22 +
2	[17,22[
3	[12,17[
4	[7,12[
5	[4,7[
6	[2,4[
7	[0,2[

Note that this approach also assumes that leaf area distributions are valid for trees of different heights. This implies that the leaf area density is higher for small trees and smaller for large trees. However, as shown in figure 2.4, the forest height at AAV6 is mostly close to 10m, i.e. the height at which the blackspruce distribution used in this work was designed for.

The idea of using the roughness maps to determine the location and height of the forest, as done by Meteodyn WT 4.0, was considered. The approach consists of multiplying the local roughness by a factor of 30 to get the height of the forest. When the height of a cell is greater than $30z_0$, this cell is considered to be within the forest. The method was not retained because first, the value of 30 was questionnable, and secondly because regardless of the value of z_0, it was possible to have forest provided that the first cell was small enough, which hardly makes sense.

2.2.3 Numerical method

2.2.3.1 Discretization

Finding an analytical solution to the Navier-Stokes equations is a difficult task partly due to the interdependence of the variables. It is at this point that computational power comes into play. In order to solve these equations, we use a technique called finite volume discretization, which consists of decomposing the computational domain into many control volumes (CVs) and solving the discretized transport equations in each of these. The discretization process of the transport equations is presented in this section. The partial differential equations presented in the last section can all be written in a more general form where ϕ represents a general variable and Γ a diffusion coefficient (Versteeg and Malalasekera, 1995). The expressions *div* and *grad* stand for divergence and gradient respectively in the following equations.

$$\frac{\partial(\rho\phi)}{\partial t} + div(\rho\phi u) = div(\rho\Gamma grad\phi) + S_\phi. \tag{2.30}$$

Integrating this equation over a control volume and neglecting the first term when dealing with steady state phenomena, one gets

$$\int_{CV} div(\rho u\phi)dV = \int_{CV} div(\rho\Gamma grad\phi)dV + \int_{CV} S_\phi dV \tag{2.31}$$

Each term of equation (2.31) has the meaning presented in table 2.6. Then, by subdividing the computational domain in several control volumes and using Gauss' theorem, it is possible to discretize each term of equation (2.31). Centers of control volumes are called nodes and quantities are calculated at these locations. In figure 2.5, they are represented by the points W (West), P and E (East). S stands for "surface normal vector" and has magnitude of face area, f for "face" and d is the vector joining two adjacent nodes. The convection, diffusion and source

Table 2.6 Meaning of each term of general transport equation.

Rate of increase ϕ of fluid element	+	Net rate of flow of ϕ out of fluid element	=	Rate of increase of of ϕ due to diffusion	+	Rate of increase of ϕ due to sources

terms can be respectively discretized as follow (Jasak, 1996)

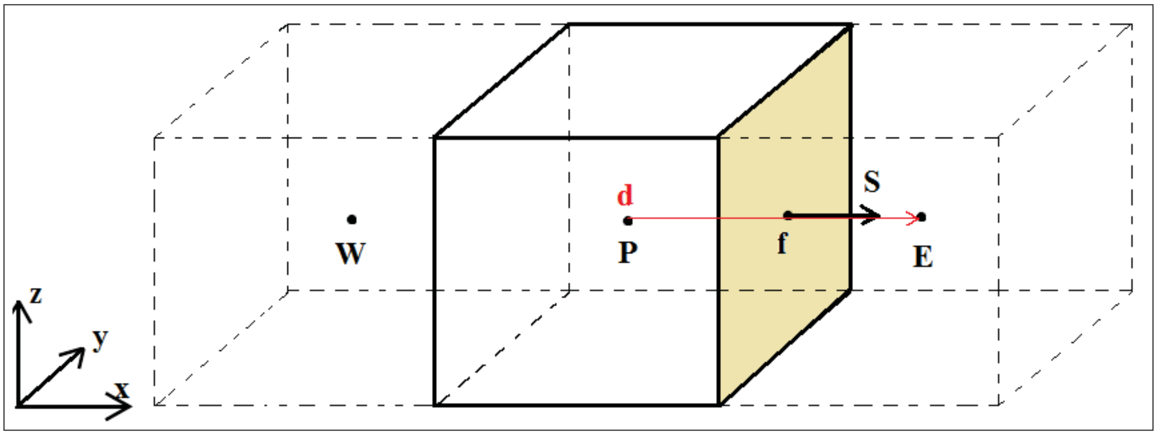


Figure 2.5 Example of control volumes

$$\int_{CV} div(\rho u \phi) dV = \sum_f S.(\rho u \phi)_f \tag{2.32}$$

$$\int_{CV} div(\rho \Gamma grad \phi) dV = \sum_f S.(\rho \Gamma grad \phi)_f \tag{2.33}$$

$$\int_{CV} S_\phi(\phi) dV = SuV_P + SpV_P\phi_p \tag{2.34}$$

Note that the source term was linearized as follow to ensure boundedness of the solution (Versteeg and Malalasekera, 1995):

$$S_\phi(\phi) = Su + Sp\phi \tag{2.35}$$

There exists a variety of schemes to evaluate ϕ and its gradients at the face locations. Only the schemes used in the context of this study will be presented here. For convection, the first order scheme "upwind" was chosen. This scheme takes into account the direction of the flow and is unconditionally stable. The expression for ϕ at the face is given by the following relations:

$$\phi_f = \phi_P \qquad \text{(flow from } W \text{ to } E\text{)} \tag{2.36}$$

$$\phi_f = \phi_E \qquad \text{(flow from } E \text{ to } W\text{)} \tag{2.37}$$

However, in certain circumstances where the flow is considerably misaligned with the grid, the upwind scheme tends to lose accuracy and creates what we call false diffusion (Jasak, 1996).

The scheme used for the diffusion is "Gauss linear corrected". It uses a standard central differencing scheme to calculate the gradient but also takes into account mesh non-orthogonality. This is done by decomposing the surface normal vector into two components: one parallel with d and the other orthogonal with the same vector. In an orthogonal mesh, the product of S and the gradient of ϕ is expressed as follows:

$$S.(grad\phi)_f = |S|\frac{\phi_E - \phi_P}{|d|} \tag{2.38}$$

In the case of non-orthogonal mesh, the above expression is decomposed as follow where the terms on the right-hand side represent orthogonal contribution and non-orthogonal correction respectively

$$S.(grad\phi)_f = \Delta.(grad\phi)_f + k.(grad\phi)_f \tag{2.39}$$

The surface normal vector S can be decomposed as

$$S = \Delta + k. \tag{2.40}$$

Figure 2.6 shows an example of surface normal vector decomposition. This example minimizes the non-orthogonal correction.

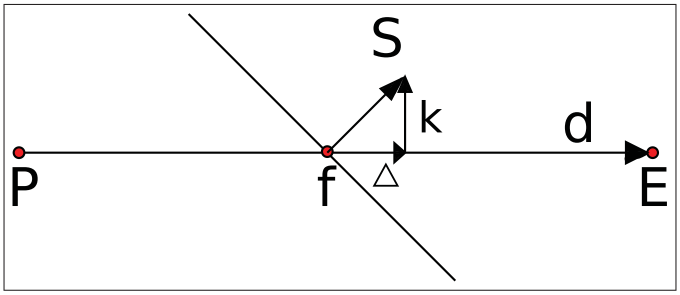


Figure 2.6 Minimum-correction approach for mesh non-orthogonality, adapted from Jasak (1996)

There exist several ways of decomposing the surface normal vector. For more information about this topic and discretization in general used in OpenFOAM, the user is encouraged to consult Jasak (1996).

2.2.3.2 Solving and convergence

The velocity/pressure coupling was solved using the SIMPLE algorithm. The solver using this algorithm in OpenFOAM is *simpleFoam*. The simulations were considered as converged when the normalized residuals reached at least 10e-6 or below for significant quantities. Please consult APPENDIX II for more details about the calculation of residuals in OpenFOAM.

2.2.3.3 Mesh

Not only the transport equations need to be discretized, the computational domain as well. Dividing the domain into many control volumes, or cells, is a process known as meshing. It is a very important step as it can directly influence the solution. OpenFOAM offers two main mesh generation utilities: *blockMesh* and *snappyHexMesh*. A short description of both tools as well as a complete description of the technique used to mesh complex topography can be found in Appendix III. The meshes realized in this work can be divided in two categories: meshes over flat terrain and meshes over complex terrain.

For flat terrain, the *blockMesh* utility was used. The result is a structured and orthogonal mesh composed exclusively of hexahedra.

For the complex terrain case, a more sophisticated method was developed. The result is a structured and curvilinear mesh composed exclusively of hexahedra. On top of using only tools available in OpenFOAM, the procedure is fast and simple. There are however some drawbacks associated with this method and the reader is encouraged to consult Appendix III for more information.

Images of the meshes created with both techniques will be presented throughout this work.

2.2.3.4 Boundary conditions and initialization

Boundary conditions and problem initialization are really problem-dependent. For this reason, they will be described for each case presented in this work.

2.3 PART II: LIDAR POST-PROCESSING ALGORITHM

In this second part of the methodology, it is described how the LiDAR bias is evaluated.

2.3.1 Modeling LiDAR bias over complex terrain

It was chosen to implement Bingöl *et al.* (2008)'s LiDAR modeling approach because among the strategies presented in the literature review, it is very well documented and does not make the assumption of flow linearity as opposed to the linear approach (which calculates flow angles based on terrain slopes as presented in section 1.3.1). The latter is also interesting, but assuming linear variation of the flow over complex terrain may be inappropriate as the sampled volume increases. Moreover, it also assumes that the flow is parallel to the topography to determine the inflow angles, which is not always realistic over complex terrain.

The strategy consists of calculating cartesian velocity components with the help of numerical flow simulations and projecting these in the directions of the laser beams of the remote sensor leading to radial velocities. Then, simple trigonometry is used in conjunction with the flow homogeneity assumption to retrieve the wind components at the center of the measurement disc.

More precisely, refering to figure 2.7, the cartesian velocity components (U_x,U_y and U_z) obtained with the CFD at locations P1,P2,P3,P4 are projected onto the lines of sight of the LiDAR with the help of the direction unit vector *n* of each laser beam. The result is a set of 4 radial velocities. Using equations (1.6) and (1.5) (page 6), the horizontal velocity at the center of the scanned disc ($U_{LiDAR'}$) can be calculated, assuming homogeneous flow within the sampled volume.

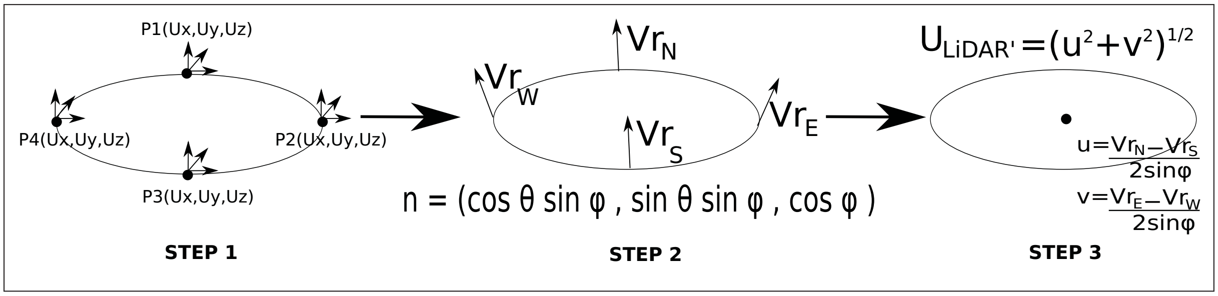

P1(Ux,Uy,Uz)
P4(Ux,Uy,Uz)
P2(Ux,Uy,Uz)
P3(Ux,Uy,Uz)
STEP 1
Vr_N
Vr_W
Vr_S
Vr_E
$n = (\cos\theta \sin\varphi , \sin\theta \sin\varphi , \cos\varphi)$
STEP 2
$U_{LiDAR'} = (u^2+v^2)^{1/2}$
$u = \frac{Vr_N - Vr_S}{2\sin\varphi}$
$v = \frac{Vr_E - Vr_W}{2\sin\varphi}$
STEP 3

Figure 2.7 Conversion process of cartesian CFD velocities to artificially calculated horizontal LiDAR velocity

To estimate LiDAR bias alone, the above procedure is enough. However, in practice, like in the real case presented in chapter 4, the LiDAR might be located several meters away from the mast. Thus, it is interesting to estimate how much of a difference to expect due to the distance between both instruments.

For instance, the LiDAR deployed at AAV (chapter 4) is located roughly 79m away from the meteorological mast called AAV6. On top of the bias caused by the LiDAR's homogeneity assumption, a difference between LiDAR and cup measurements is expected caused by the distance between both instruments. The situation is presented in figure 2.8. Thus, three sets of velocity ratios will be analysed. Refering to figure 2.8, the ratios are

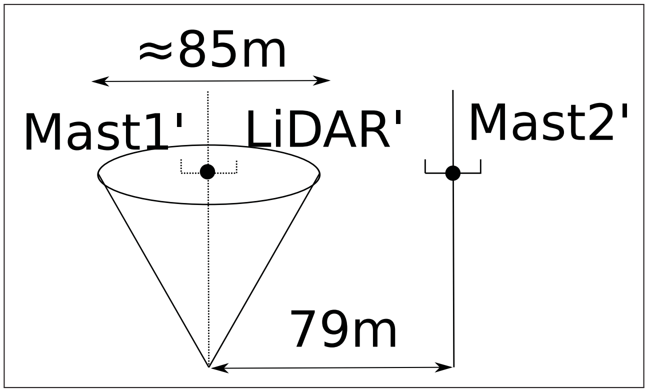


Figure 2.8 Locations of CFD defined points of interest. The prime(') means "CFD calculated".

1. The LiDAR bias caused by flow homogeneity assumption (LiDAR'/Mast1'). The points Mast1' and LiDAR' occupy exactly the same location but are calculated differently. Mast1' is simply the CFD calculated velocity whereas LiDAR' is the velocity obtained with the modeling procedure that was just described;

2. The effects of the distance between the LiDAR and the mast (Mast1'/Mast2'). Mast2' is the CFD calculated point corresponding to AAV6;

3. The combined effect of the LiDAR bias and distance between the instruments (LiDAR'/Mast2'). This ratio will be compared to experimental data of LiDAR versus AVV6 data at 80m.

CHAPTER 3

MODEL VERIFICATION AND VALIDATION

Modeling the atmospheric boundary layer over complex terrain covered with forest presents many challenges. Before considering the complete model, it was decided to analyze each aspect step-by-step. Hence, the following steps were carried out:

- Modeling neutral homogeneous atmospheric boundary layer over flat terrain and verification with analytical solutions;
- Modeling ABL flow over flat terrain covered with forest and validation with experimental data (Amiro, 1990) and computational results (Dalpé and Masson, 2008);
- Modeling ABL flow over an isolated Gaussian hill and validation of LiDAR bias post-processing tool.

3.1 Modeling homogeneous neutral boundary layer

First, it was important to make sure that an homogeneous and neutrally stratified boundary layer could be modeled in OpenFOAM. The goal is simple: maintain logarithmic wind profiles on flat terrain throughout the whole computational domain. The work accomplished in this section basically revisits the findings of Sumner and Masson (2010) without adding corrections for discretization error.

3.1.1 Mesh

It is not straightforward to simulate one-dimensional flow in OpenFOAM, so a two-dimensional simulation was set-up as shown in figure 3.1. The mesh consists of a 5000m long and 500m high domain. There are 50 vertical cells and 500 cells in the direction of the flow. A cell-to-cell vertical expansion of 1.08311 was used, which leads to a first cell height of 0.78m and a top cell of 39.09m approximately.

3.1.2 Boundary conditions

Conditions imposed at the boundaries of a computational domain are crucial since they have a direct impact on the solution of the problem being resolved. The computational domain

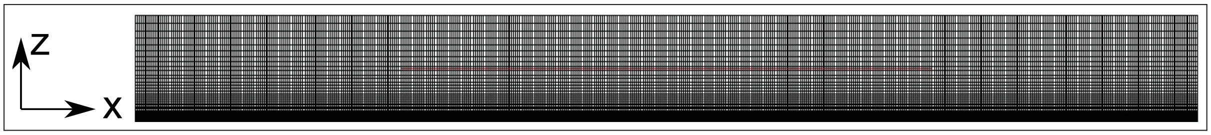


Figure 3.1 Mesh used to simulate homogeneous and neutrally stratified boundary layer.

considered here consists of 4 boundaries. Please refer to figure 3.2 for the name and location of each boundary. As shown in figure 3.3, the region between ground and z_0 is not solved. This is the case for all computations presented in the present work.

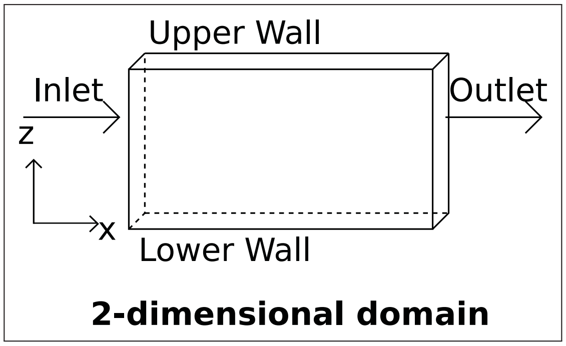


Figure 3.2 Name and location of boundaries on computational domain.

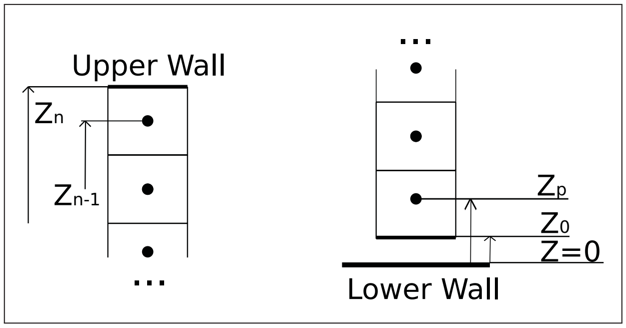


Figure 3.3 Domain upper and lower boundary

Inlet/Outlet: A cyclic condition was applied between inlet and outlet for all quantities (U,P k and ε). This boundary condition recycles all the quantities from the outlet and imposes them at the inlet.

Upper Wall: Since the terrain was flat with no obstacles, the standard Richards and Hoxey (1993) procedure was followed. The authors state that a constant shear stress should be applied in the direction of the flow in order to obtain a homogeneous ABL (along with the appropriate bottom boundary condition). The shear stress is defined as

$$\tau = \rho u^{*2}. \tag{3.1}$$

By imposing a given friction velocity at the top of the domain, it is possible to determine the corresponding velocity

$$U(z) = \frac{u^*}{\kappa} \ln\left(\frac{z}{z_0}\right). \tag{3.2}$$

Then, k and ε values to prescribe at the top can be obtained with the following relations respectively

$$k = \frac{u^{*2}}{C_\mu}, \tag{3.3}$$

$$\varepsilon = \frac{u^{*3}}{\kappa z}. \tag{3.4}$$

This boundary condition will be refered to as "Fixed Value".

Lower Wall: First, note that the region between the wall and z_0 is not resolved. Again, Richards and Hoxey (1993) approach was used at this boundary. It suggests that the production of kinetic energy, which correspond to the source (G_k) in the transport equation of k, is set equal to the dissipation ε only when k_p satisfies the following equation

$$k_p = \frac{u_g^{*2}}{\sqrt{C_\mu}} \tag{3.5}$$

where the local friction velocity u_g^* is calculated as follow

$$u_g^* = \frac{\kappa U_p}{\ln(z_p/z_0)}. \tag{3.6}$$

In order to satisfy the above statement (G_k=ε and equation (3.5)), the near-wall cell-averaged production of k and the average dissipation rate must be calculated respectively as follow

$$\overline{G_{k,p}} = \frac{u_g^{*3}}{2\kappa(z-z_0)} \ln\left(\frac{2z_p - z_0}{z_0}\right), \tag{3.7}$$

$$\overline{\varepsilon_p} = \frac{\sqrt{C_\mu} k_p u_g^*}{2\kappa(z-z_0)} \ln\left(\frac{2z_p - z_0}{z_0}\right). \tag{3.8}$$

The near-wall cell-center value of dissipation must be set to

$$\varepsilon_p = \frac{\sqrt{C_\mu} k_p u_g^*}{\kappa z_p}. \tag{3.9}$$

Richards and Hoxey (1993) also state that a retarding shear stress should be applied. This can be implemented through the prescription of the appropriate turbulent viscosity in the near-wall cell-centre. A non-slip condition (U=0) is used at z=z_0 and the velocity profile from z_0 to z_p is assumed to be logarithmic. The shear stress at the wall can be expressed as

$$\tau = u_g^{*2} = \nu_t \frac{\partial U}{\partial z}. \tag{3.10}$$

Equation 3.5 can be rearranged such that

$$u_g^{*} = C_\mu^{1/4} k_p^{1/2}. \tag{3.11}$$

The vertical gradient of U can be approximated with the help of finite differencing

$$\frac{\partial U}{\partial z} \approx \frac{(U_p - U_w)}{(z_p - z_0)}. \tag{3.12}$$

After substituting (3.11) and (3.12) in (3.10), one gets

$$u_g^{*2} = \nu_t \frac{(U_p - U_w)}{(z_p - z_0)}. \tag{3.13}$$

By setting U_w=0, substituting U_p by (3.2) and isolating the turbulent viscosity, the result is

$$\nu_{tp} = \frac{C_\mu^{1/4} k_p^{1/2} \kappa (z_p - z_0)}{\ln(\frac{z_p}{z_0})}. \tag{3.14}$$

3.1.3 Initialization

In order to accelerate convergence, analytical solutions for U, k and ε were prescribed using equations (3.2), (3.3) and (3.4) respectively.

3.1.4 Input parameters

The model constants for this validation are the ones used by Alinot and Masson (2005) and are presented in table 2.3. The parameters used for the calculation of the shear stress imposed at the upper wall are shown in table 3.1.

3.1.5 Results and discussion

As it can be seen in figure (3.4), very good agreement between numerical and analytical solutions was observed except for k in the near-wall cells, which is a common issue in ABL modeling. These profiles were considered as satisfying but it is important to point out that

Table 3.1 Flow parameters used for calculation of upper boundary shear stress

Reference height, z_{ref}	6 m
Roughness length, z_0	0.01 m
Reference mean wind speed, U_{ref}	10 m/s
Von Karman constant, κ	0.40

solutions exist to correct this problem. An approach is presented by Sumner and Masson (2010).

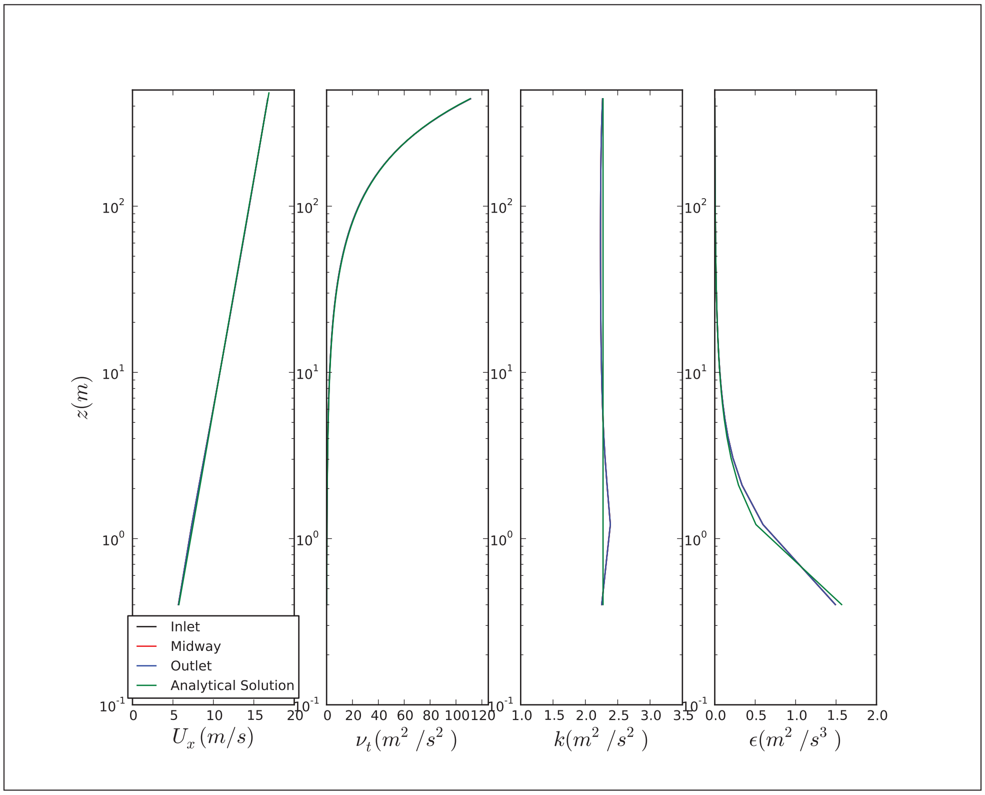


Figure 3.4 Comparison between analytical and numerical profiles of U, ν_t, k and ε at inlet, outlet and halfway.

3.2 Validation of forest model

As explained earlier, a porous media approach was used to model the effect of the forest on the flow. By default, *simpleFoam* is not suited for taking into account the effect of the vegetation. Therefore, the additions of source terms in the momemtum and turbulence transport equations were needed (see section 2.2.2.4 for details). In order to verify that these modifications were added correctly, an attempt was made to reproduce the results obtained by Dalpé and Masson (2008).

The experiment consists of modeling fully developed turbulent flow over and within three types of forest: aspen, blackspruce and pine. Numerical results are then compared to experimental data of Amiro (1990) and computational results of Dalpé and Masson (2008).

3.2.1 Mesh

The dimensions of the 2D mesh are as follow: 100m long by 800m high. The domain is decomposed in 192 and 10 cells in the vertical and flow direction respectively (see figure 3.5) . A vertical cell-to-cell expansion ratio of 1.03683 is used. The near-wall bottom and top cells measure respectively ∼0.03m and ∼28m. Cells are all hexahedra and it was generated using *blockMesh*.

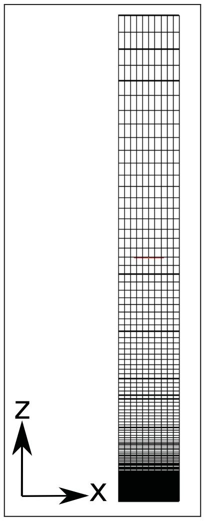


Figure 3.5
Mesh used for forest validation.

3.2.2 Boundary conditions

The boundary conditions are the same as the ones described in 3.1.2. An additionnal top boundary condition developed by Dalpé and Masson (2008), which was derived to deal with the presence of forest, was tested. The idea behind this particular boundary condition is to avoid the assumption of logarithmic profiles from the top of the domain all the way down to the bottom wall. Instead, a logarithmic profile is assumed only at the top of the domain. This is done by firstly imposing the velocity at z_n (refer to figure 3.3). For each iteration, a friction velocity is calculated using equation (3.15), which then allows to calculate and prescribe k_n and ε_n using equation (3.16) and (3.17) respectively. Note that the positions *n* and *n*-1 were set as shown in figure 3.3 and not as suggested by (Dalpé and Masson, 2008). The author defines these positions as being the two higest nodes of the domain.

$$u^*_{top} = \frac{(u_n - u_{n-1})\kappa}{\ln(z_n/z_{n-1})}, \tag{3.15}$$

$$k_n = \frac{u^{*2}_{top}}{\sqrt{C_\mu}}, \tag{3.16}$$

$$\varepsilon_n = \frac{u^{*3}_{top}}{\kappa z_n}. \tag{3.17}$$

Equation (3.15) was obtained by assuming logarithmic profiles at z_n and z_{n-1} such that

$$u_n = \frac{u^*}{\kappa}\ln\left(\frac{z_n}{z_0}\right) = \frac{u^*}{\kappa}(\ln(z_n) - \ln(z_0)), \tag{3.18}$$

$$u_{n-1} = \frac{u^*}{\kappa}\ln\left(\frac{z_{n-1}}{z_0}\right) = \frac{u^*}{\kappa}(\ln(z_{n-1}) - \ln(z_0)). \tag{3.19}$$

Subracting both equations leads to

$$u_n - u_{n-1} = \frac{u^*}{\kappa}\ln\left(\frac{z_n}{z_{n-1}}\right). \tag{3.20}$$

Rearranging for u^* gives equation (3.15).

3.2.3 Initialization

k and ε were set to $\approx$10e^{-3} and U was set to 0.

3.2.4 Input parameters

The input parameters regarding the forest are the same as the ones used by Dalpé and Masson (2008) and are summarized in figure 3.6 and table 3.2.

Table 3.2 Characteristics of aspen, blackspruce and pine forests

	Blackspruce	Jack pine	Aspen
Height, h(m)	10	15	10
Leaf area index, LAI	9.19	2.22	3.57
Drag Coefficient, C_d	.15	.45	0.4
Roughness height, z_0(m)	0.05	0.01	0.1

Table 3.3 Flow parameters for flow within and above forest

	Friction velocity u^*	Upper boundary velocity U(m/s)
Fixed Value	0.42	(10 0 0)
Dalpé and Masson (2008)	Calculated locally	(10 0 0)

The friction velocity for the "Fixed Value" boundary condition was calculated based on the roughness length and the velocity at the top of the domain using neutrally stratified logarithmic wind profiles (equations (3.2), (3.3) and (3.4)).

3.2.5 Results and discussion

The objective was to obtain similar results to those of Dalpé and Masson (2008) in order to validate the modifications brought to the solver. This was successfully achieved as shown in figure 3.7. The latter shows experimental measurements (Amiro, 1990) as well as computational results obtained with two different upper boundary conditions ("Fixed Value" and Dalpé and Masson (2008)). In general, the model predicts reasonnably well U and u^* but underestimates k. This might be considered a weakness of the method as it was observed both by Katul *et al.* (2004) and Dalpé and Masson (2008). The results are normalized at z/h=1.21, z/h=1.13 and z/h=1.31 for blackspruce, jackpine and aspen respectively.

At low heights, both top boundary conditions seem to perform well and yield approximately the same results, but it is not the case as we go up the domain. As figure 3.8 depicts, a significant

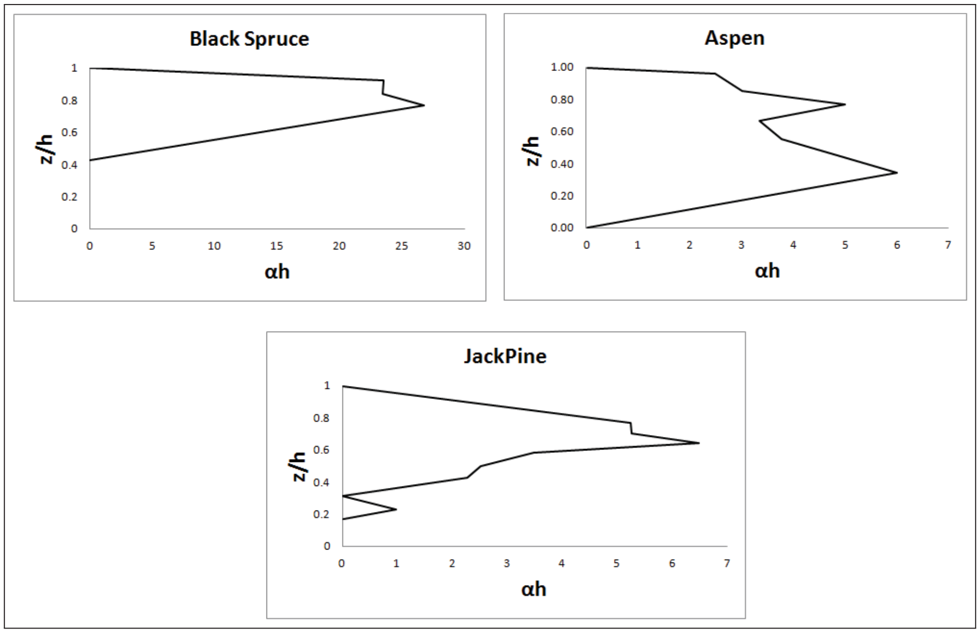


Figure 3.6 LAI distributions (normalized by forest height) used for validation of forest model.

difference is present between the k and U profiles. While forcing values for U, k and ε, the displacement height created by the forest is neglected. Thus, it forces the solution to be entirely logarithmic while in fact the flow in the first few meters near the bottom wall does not follow a logarithmic profile.

Even though this behavior was predictable, it was believed that placing the upper boundary high enough would damp the effects of such a boundary condition ("Fixed Value"), which is obviously not the case.

Furthermore, when looking closely at Dalpé and Masson (2008)'s approach (equations (3.18) and (3.19)), the displacement height created by the forest is also neglected. Also, one must be careful when using this boundary condition over complex terrain as the distances from z_n and z_{n-1} with respect to the lower wall are variable. Therefore, a more appropriate boundary condition for flow over complex terrain covered with forest is needed.

3.2.6 Appropriate upper boundary condition for flow over complex terrain covered with forest

The challenge here is partly due to the unknown displacement height created by the forest, which makes the velocity and turbulent quantities impossible to fix at the upper boundary of

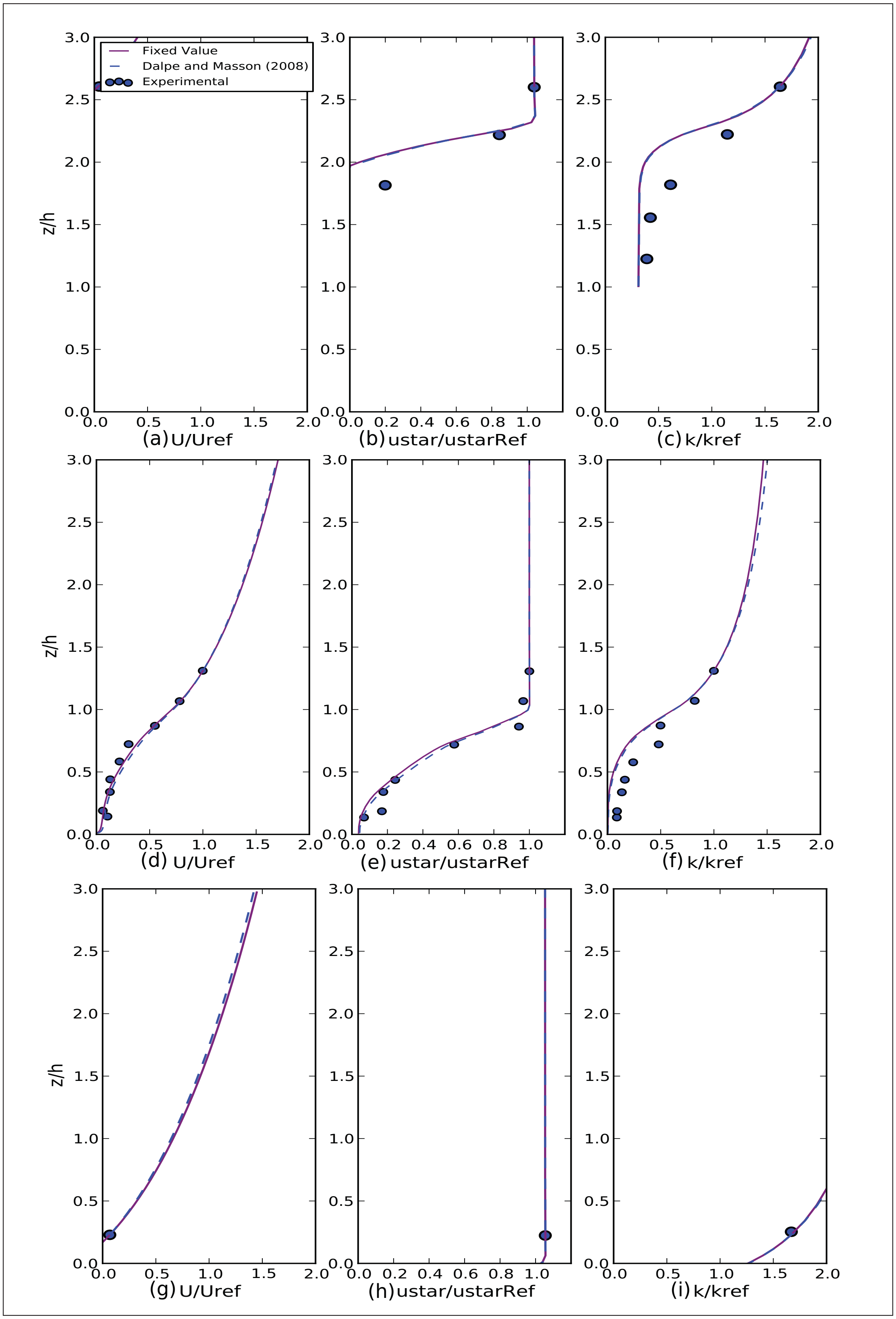


Figure 3.7 Near-wall velocity, friction velocity and turbulent kinetic energy profiles for the three types of forests (starting from top): BlackSpruce forest, Aspen forest and Jack Pine forest.

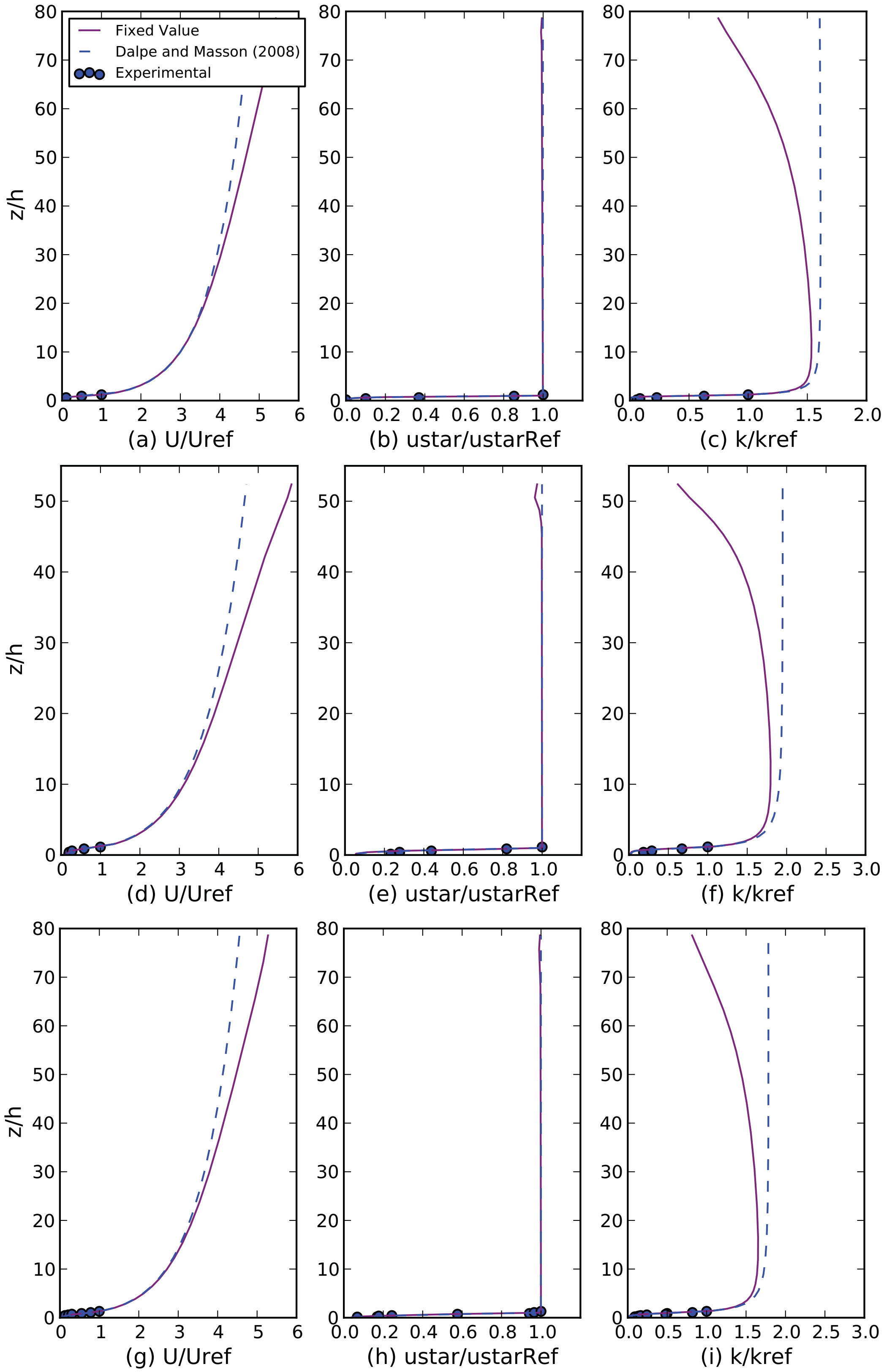


Figure 3.8 Same as fig. 3.7 but complete domain.

the domain. An alternative approach is to fix a shear stress through a friction velocity. It is assumed that the friction velocity above the forest is constant. Then, it is possible to fix the gradient of velocity using the turbulent viscosity from the previous iteration

$$\frac{\partial U}{\partial z} = \frac{\rho u^{*2}}{\nu_t}. \tag{3.21}$$

As opposed to both conditions tested previously, this approach implies a different velocity for each cell near the boundary, which makes sense when considering that the logarithmic expression for the velocity is height dependent. The dissipation is expressed as

$$\varepsilon_n = \frac{u_{top}^{*3}}{\kappa(z_n - h)}, \tag{3.22}$$

h being the unknown displacement height created by the vegetation. In order to determine its value, equation (3.15) can be rewritten to include h

$$u_{top}^{*} = \frac{(u_n - u_{n-1})\kappa}{\ln \frac{z_n - h}{z_{n-1} - h}}. \tag{3.23}$$

The displacement height can then be isolated easily

$$h = \frac{z_n - z_{n-1} f(u^*)}{1 - f(u^*)} \tag{3.24}$$

where

$$f(u^*) = \exp\left(\frac{\kappa}{u^*}(u_n - u_{n-1})\right). \tag{3.25}$$

Substituting equation (3.24) into equation (3.22), one can prescribe a value or a gradient for the dissipation at the upper boundary. Another possibility suggested by Lussier-Clément (2012) is to express ε in terms of the turbulent viscosity, which avoids the need to calculate the displacement height. The author decided to prescribe a gradient but the idea is the same if one prefers a fixed value

$$\frac{\partial \varepsilon}{\partial z} = \frac{u^{*3}}{\kappa(z_n - h)^2}. \tag{3.26}$$

If one substitutes the logarithmic expressions for k and ε (equations (3.16) and (3.22) respectively) into the equation of the turbulent viscosity (2.16), the result is

$$\nu_t = u^* \kappa (z_n - h). \tag{3.27}$$

Substituting equation (3.27) into equation (3.26), a gradient of ε, independent of h, can be prescribed

$$\frac{\partial \varepsilon}{\partial z} = \frac{u^{*3} \kappa}{\nu_t^2}. \tag{3.28}$$

Next section validates this approach over flat terrain.

3.2.6.1 Comparison with Lussier-Clément(2012) upper boundary condition

Both the approach of Lussier-Clément (2012) and Dalpé and Masson (2008) were tested in the same conditions (same mesh, same input parameters, etc) as the simulation realized for forest validation in section 3.2. Model constants are the one presented in table 2.4. Forest and flow parameters are summarized in table 3.4. The friction velocity was calculated based on the results obtained with Dalpé and Masson (2008) when setting the upper boundary velocity to 10 m/s.

Table 3.4 Forest and flow parameters

	Lussier-Clément (2012)	Dalpé and Masson (2008)
Friction velocity u^*	0.75	Calculated locally
Upper boundary velocity U(m/s)	Calculated locally	10
Forest height, h(m)	10	
Leaf area index, LAI	9.19	
Drag Coefficient, C_d	0.15	

As expected, both conditions proved to yield essentially the same results on flat terrain (see figure 3.9).

However, the user should be careful of not setting other velocity components to zero when using Dalpé and Masson (2008)'s approach as it would prevent mass from exiting the domain and potentially create artificial acceleration. This is not an issue over flat terrain as U_y and U_z are often close to zero. In OpenFOAM, Dalpé and Masson (2008)'s condition could be used, for

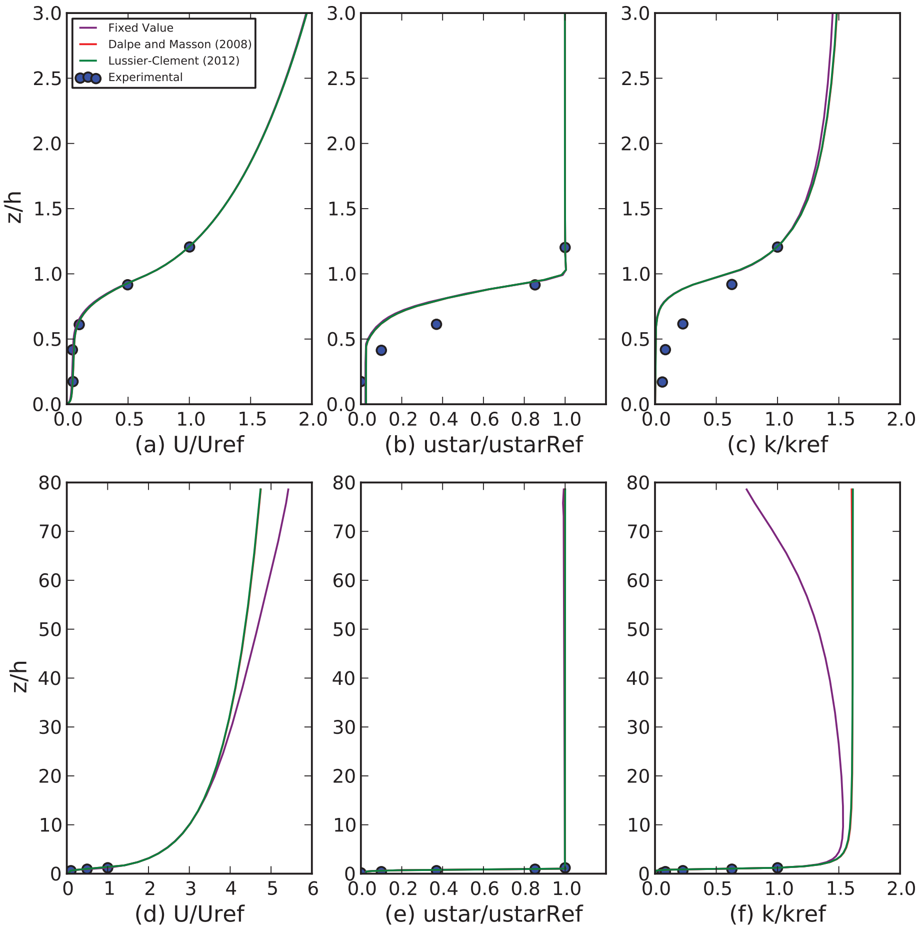


Figure 3.9 2D comparison of upper boundary condition: velocity, friction velocity and *k* profiles for the near-wall region (first row) and full domain (second row).

example, in conjunction with zero normal gradient for other transverse component of velocities (U_y and U_z). The use of fixed values and gradient conditions implies a mixed boundary condition. Lussier-Clément (2012)'s condition does not require such a mixed boundary condition as only gradients are imposed.

Despite the very similar results obtained, Lussier-Clément (2012) was considered more appropriate since it is completely independent of the forest displacement height and there is no need of finding the distance between top cells and lower boundary, which can be challenging over complex terrain. This implies finding lower wall cells that are aligned perpendicularly with top cells.

3.3 Modeling LiDAR bias on a theoretical case

3.3.1 Objective

Last step before considering the real case scenario consists of validating the implementation of the LiDAR post-processing algorithm.

It was proposed by Hill (2010) to evaluate the LiDAR bias in different situations with the help of computer modeling. A number of parameters such as atmospheric stability, vegetation and topographic effects can impact LiDAR performance and they all deserve attention. In fact, any parameter that could potentially make the flow inhomogeneous would be interesting to investigate as it directly influences LiDAR measurements. In this chapter, it was decided to concentrate on topographic effects only. More precisely, the following experiment had as an objective to validate the implementation of the LiDAR post-processing algorithm described in section 2.3.1.

This was realized by running a wind flow simulation over several gaussian hills in order to estimate LiDAR bias for each case. It was chosen to use an opening angle of 28°, which falls in the recommendations of Boquet *et al.* (2010), and four beams separated by 90°, just as the WINDCUBE in the last chapter.

The experiment was separated in two subtasks:

1. First, directional LiDAR bias was evaluated only at one location where maximum bias was expected, which is at the top of the hill. Due to symmetry of the geometry, only one computation per case was necessary to simulate all wind directions;
2. Secondly, the methodology was carried for a large area of the computational domain. This was conducted only when the wind comes from West.

The idealized complex terrain considered for this experiment consists of an isolated hill described by a Gaussian function where the height of the hill varies based on the (X,Y) position as shown in figure 3.10. Parameter L specifies the distance from center of the hill to the point of inflection (point where the slope is maximum). H represents the maximum height of the hill.

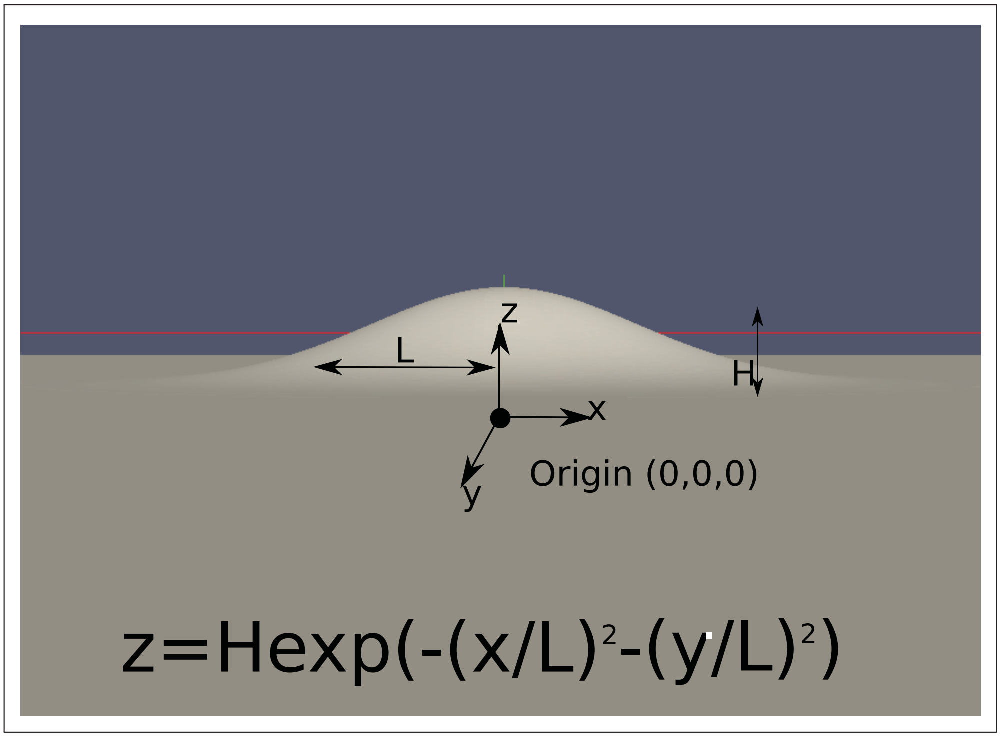


Figure 3.10 Gaussian hill with important dimensions.

It was decided to non-dimensionalize the results so that they can be used as a preliminary tool for positioning LiDARs on real sites. The quantity used to non-dimensionalize is L. The LiDAR bias was assumed to be dependent only on the following quantities

- The (X,Y) location of the remote sensor with respect to the hill;
- The scanning height of the instrument SH;
- The height of the hill H.

The non-dimensionalized scanning height SH/L and hill height H/L will be refered to as λ and φ respectively.

3.3.2 Mesh

The domain is 1400m x 1400m x 1000m. The hill is located exactly in the middle of the bottom boundary. A close-up of the mesh near the hill is shown in fig 3.11. Mesh is structured and contains only hexahedra (note that the diagonal line in each cells in fig. 3.11 is only due to a display issue with the visualization application). Parameters are identical for all three cases and are summarized in table 3.5.

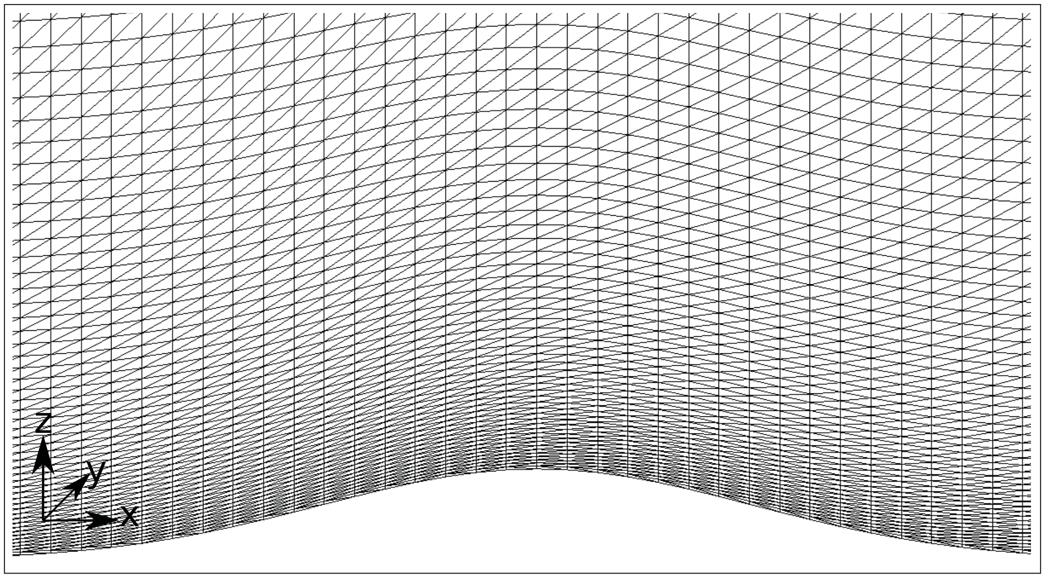


Figure 3.11 Close-up of mesh near Gaussian hill.

3.3.3 Boundary conditions

The boundary conditions are identical as those used for simulation with forest (section 3.2) except for the inlet where logarithmic wind profiles were used and the outlet where a zero normal gradient was used for all variables. Forest is absent in this set of computations. The roughness is constant throughout the whole domain. Flow parameters are presented in table 3.5. The terrain is flat all around the hill up to the boundaries.

3.3.4 Initialization

Again, k and ε were set to $\approx$10e^{-3} and U was set to 0.

3.3.5 Results and discussion

First, in order to evaluate the LiDAR post-processing algorithm, it is necessary to have an idea of the behavior of the flow surrounding the hill. To do so, we will refer to figure 3.12, which shows a slice of the flow field taken exactly at y=0.

Two sets of results are presented in both sections 3.3.5.1 and 3.3.5.2. The first set shows the impact of λ when φ is kept constant, and the second set shows the opposite.

LiDAR Characteristics			
Non-dimensional scanning height, $\lambda = SH/L$	0.5	0.8	1.5
Cone opening angle	28°		

Hill Characteristics			
Non-dimensional hill height, $\varphi = H/L$	0.3	0.4	0.5
Max slope(%)	25.7	34.4	42.8
Length, L (m)	100		

Mesh Characteristics	
Distance from boundaries to start of hill (flat zone) (m)	≈ 500
Horizontal resolution (m)	10
Number of vertical cells	85
Vertical cell-to-cell expansion ratio	1.0468
Near-wall cell height (m)	≈ 1

Flow Characteristics	
Reference height (m)	80
U_{ref} (m/s)	≈ 7.95
Roughness length (m)	0.002

Table 3.5 This table summarizes geometric parameters of the Gaussian hill as well as important mesh and flow characteristics used in this experiment.

3.3.5.1 Directional LiDAR bias

Considering first directional results presented in figure 3.13, it is interesting to note that when the LiDAR is located on top of the hill ("LiDAR2" on figure 3.12), it underestimates the horizontal velocity for all tested scenarios. The latter phenomenon can easily be explained by the acceleration of the flow as it travels above the hill. Since the remote sensor shoots its beams with a half-opening angle of 28°, radial velocities used to retrieve the wind components are taken upstream and downstream ("c" and "d" respectively on figure 3.12) where the velocity is slightly less than the one calculated at the summit. Vertical components of velocities are also more important at these locations than the actual vertical component at the center of the scanned disc.

The periodic pattern observed for each case is due to projection of the wind components on the lines of sight, which varies depending on the wind direction.

A LiDAR can scan the wind at multiple heights ranging from a few tens of meters to a couple of hundreds of meters. Figure 3.13 (a) shows that as the scanning height becomes dominant with respect to the size of the hill, the quality of the results improves. Again, this is not surprising

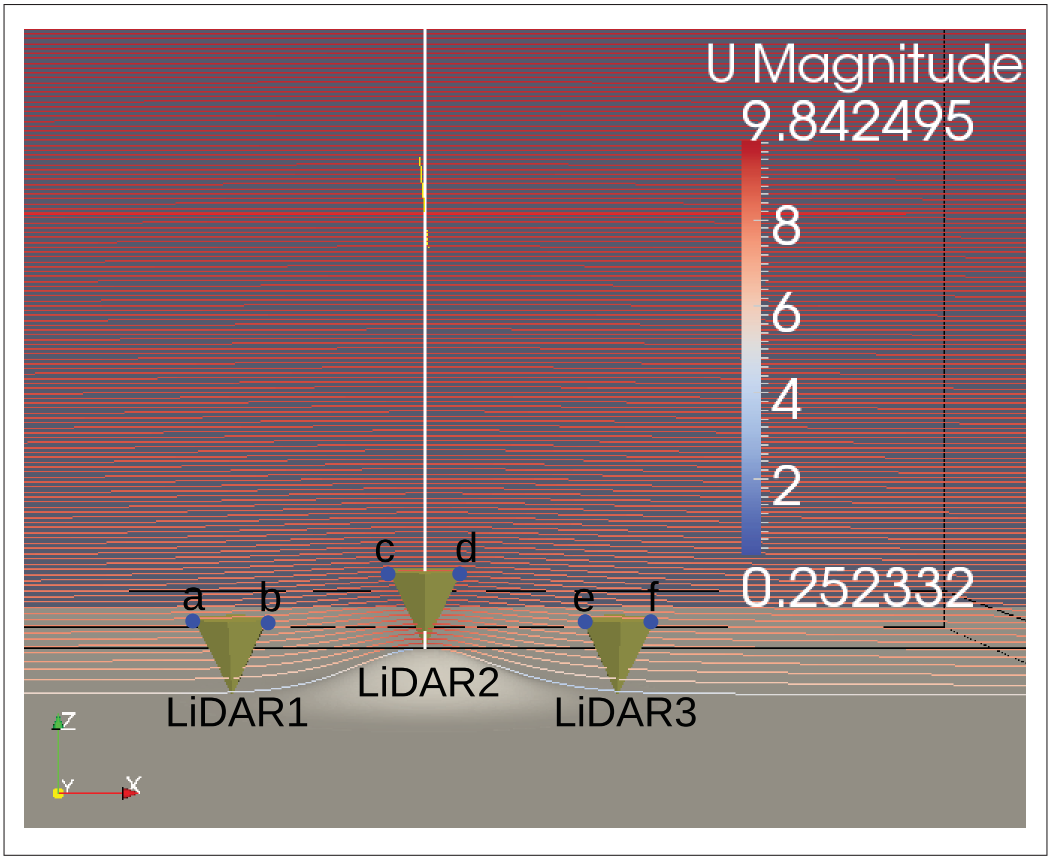


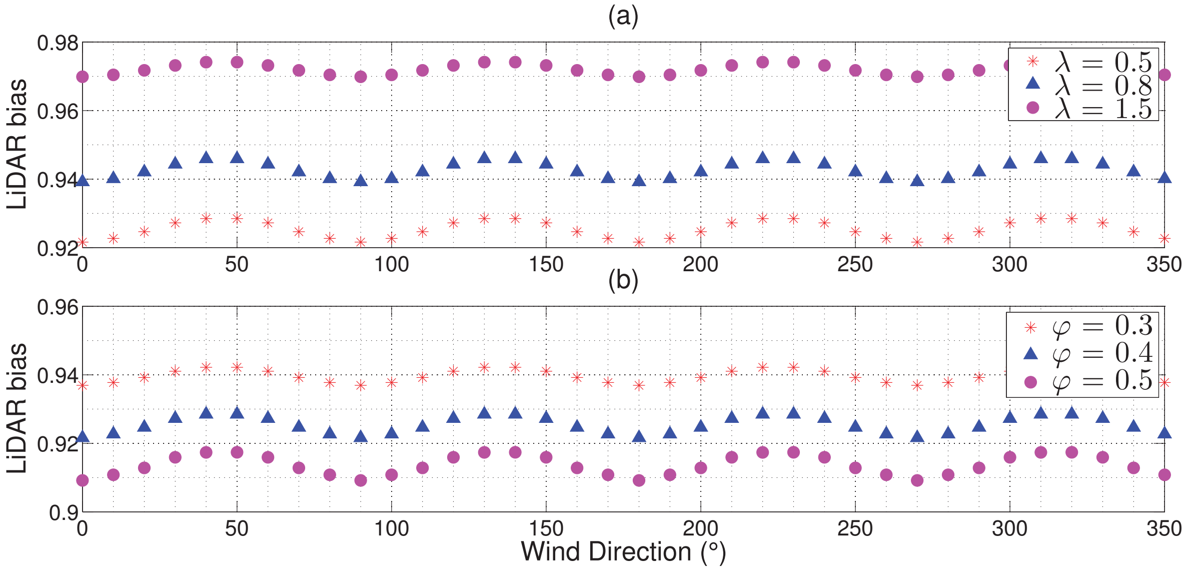
Figure 3.12 Flow field around Gaussian hill where λ=0.8 and ϕ=0.5. Flow goes from left to right.

since the flow far away from the obstacle is less disturbed, thus diminishing the impact of the homogeneity assumption made by the instrument.

The opposite is also true when looking at λ=0.5 (figure 3.13 (a)) where the flow is more affected by the obstacle, which directly impacts the wind speed calculated by the remote sensor. Note that at lower scanning height, the volume swept by the beams is smaller, reducing the effects of the averaging process. Still, the flow is highly non-homogeneous in this volume and a large bias is observed.

Figure 3.13 (b) shows the effect of varying φ, which is equivalent to changing the slope of the hill. The relation between the maximum slope of the Gaussian hill (always located at L by definition) and φ is the following

$$maxslope = \sqrt{\frac{2}{e}}\varphi. \tag{3.29}$$



fbox

Figure 3.13 Directional LiDAR bias is shown in (a) for λ = 0.5, 0.8 and 1.5 and φ = 0.4 and in (b) when λ = 0.5 and and φ = 0.3, 0.4, and 0.5. Wind directions 0°, 90°, 180° and 270° correspond to W,S,E and N respectively.

Results show that the slope of the hill definitely has an impact on the bias. When λ remains constant, as the slope increases, the bias also increases.

3.3.5.2 LiDAR bias map

When the position of the LiDAR is already determined on a given site, it is legitimate to proceed to an in-depth local performance study of the instrument as the one just realized in the previous section. However, this approach can be generalized for an entire site, as it will be presented in this section. As mentionned earlier, it is a known fact that LiDAR's performances over complex terrain are still debatable. Keeping that in mind, it is however possible to mitigate these effects by carefully chosing the location of the instrument prior to its deployment.

In order to do so, one must be aware of the expected LiDAR bias for every position of the site of interest. Such a "bias map" can be generated by reproducing the previously used procedure for all the grid points of the topographic map. This means that for each grid point, coordinates in space of LiDAR sample points need to be calculated. The list of coordinates can then be used to sample the velocities in the simulation. Finally, a script can be used to estimate LiDAR bias at every grid point based on the sampled velocities. All these steps were performed on the Gaussian hill for different values of λ and φ. Results are presented in figures 3.14 and 3.15

where each contour line denotes a 1% variation of the LiDAR bias.

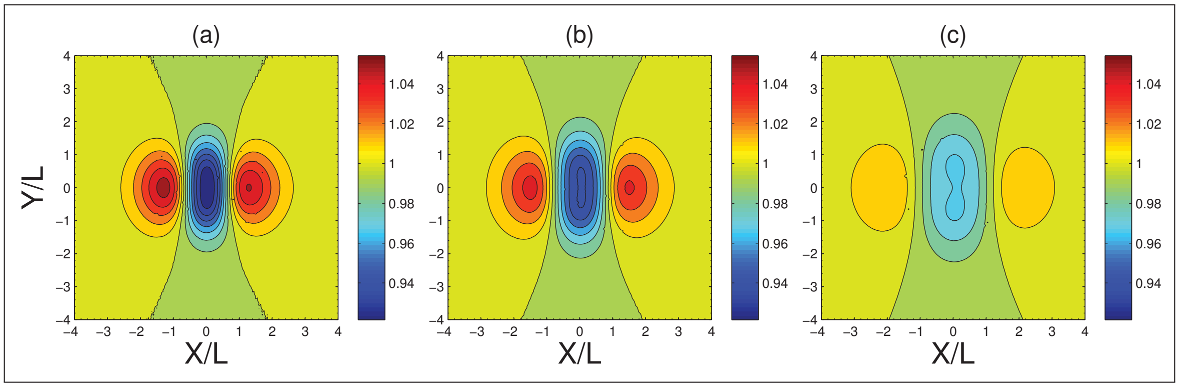


Figure 3.14 (a), (b) and (c) show the LiDAR bias calculated with a resolution of 5mx5m with $\lambda = 0.5$, 0.8 and 1.5 respectively and $\varphi = 0.4$. Wind comes from West (left).

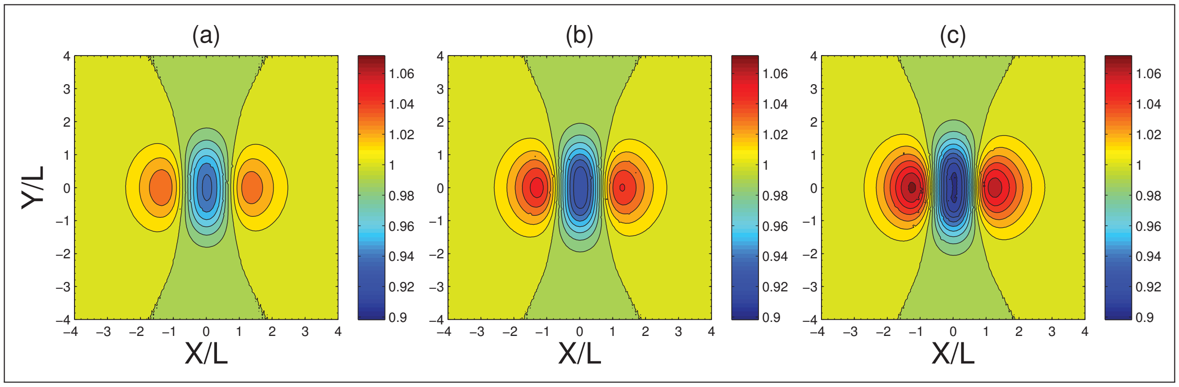


Figure 3.15 (a), (b) and (c) show the LiDAR bias calculated with a resolution of 5mx5m with $\lambda = 0.5$ and and $\varphi = 0.3$, 0.4, and 0.5 respectively. Wind comes from West (left).

At first glance, general comments can be made for both figures 3.14 and 3.15. It can be seen that placing the LiDAR upstream and downstream ("LiDAR1" and "LiDAR3" on figure 3.12) the hill causes an overestimation of the horizontal velocity. Again, this is due to one of the beam sampling the velocity near the summit of the hill where the velocity is higher as compared to the location of the instrument ("b" and "e" on figure 3.12). Notice however the small difference in shape of both overestimation regions caused by the asymetrical flow solution. Then, as we move towards the top of the hill, underestimation is observed for the reasons presented in the previous section. At sufficient upstream and downstream distance from the hill, no bias is present, as expected since the flow is unaffected by the hill. However, it can be seen that a bias of magnitude <1% on the sides (North and South) of the hill propagates as far as $4L$, which shows how

Looking closer at figure 3.14 shows that the magnitude of the bias decreases as λ increases, whereas the extent of this bias slightly increases.

Figure 3.15 demonstrates the effect of increasing φ, which results in an increase of the magnitude of the bias. The bias zone also expands slightly. Table 3.6 summarizes the maximum overestimation and underestimation of all scenarios tested in this experiment as well as the approximate radius of the zone affected by more than |1%| bias.

Tested configurations	φ=0.4			λ=0.5		
	λ=0.5	λ=0.8	λ=1.5	φ=0.3	φ=0.4	φ=0.5
Range of LiDAR bias(%)	[-7.8, 5.4]	[-6.1, 4.4]	[-3.44, 2]	[-6.3, 3.8]	[-7.8, 5.4]	[-10.2, 7.2]
Radius zone affected by > |1%|	$\approx 2.2L$	$\approx 2.3L$	$\approx 2.3L$	$\approx 2L$	$\approx 2.2L$	$\approx 2.3L$

Table 3.6 This table summarizes the magnitude of the LiDAR bias for the cases tested in this experiment.

As an attempt to compare these recommendations with IEC 61400-12-1 working draft 1, the latter recommends considering terrain of up to five times the scanning height ($5SH$)around each beam probe volume of the remote sensor to proceed to the calculations of the inflow angles required by the linear approach (see section 1.3.1). For a LiDAR operating with an opening angle of 28°, the radius of the scanned disc is $\approx 0.5SH$, so the range to be considered is $\approx 5.5SH$

If we take for example the case where $\lambda = 0.5$ and $\varphi = 0.3$ (figure 3.15 (a)), a distance of $2L$ was recommended, which corresponds to 200m. The scanned height for this case was 50m, so in terms of the scanning height, the results obtained suggest a range of $4SH$. The same analysis was carried for all tested cases and results are presented in table 3.7.

Tested configurations	φ=0.4			λ=0.5		
	λ=0.5	λ=0.8	λ=1.5	φ=0.3	φ=0.4	φ=0.5
Radius zone affected by > |1%|	$\approx 4.4SH$	$\approx 2.875SH$	$\approx 1.93SH$	$\approx 4SH$	$\approx 4.4SH$	$\approx 4.6SH$
IEC 61400-12-1 working draft 1	$5.5SH$					

Table 3.7 This table compares the results of the theoretical case to the recommendations of IEC 61400-12-1 working draft 1.

It can be seen from table 3.7 that working draft 1 is conservative compared to the results obtained in this experiment, especially for high values of λ.

3.3.6 Conclusion

Considering the flow field shown in figure 3.12 and the distribution of LiDAR bias observed in this experiment, it can be said that the implementation of Bingöl *et al.* (2008)'s approach seems appropriate. Results also agree with Bradley (2008) who conducted a similar analysis using a potential flow approach on quasi Gaussian shape. Bradley (2008) also found zones of over estimations upstream and downstream of the hill as well as major underestimation when the remote sensor is located on top of the hill.

It is interesting to note that these results could be used as a rule of thumb for siting a four beam LiDAR with an opening angle of 28°. It is believed that running simulations over many configurations of simple geometries (ridges, plateau, etc.) such as the one investigated here could be useful for remote sensors preliminary siting investigations. Using these results could be a first step towards more sophisticated methods.

On a more complex geometry, it would be interesting to proceed to a directional analysis. The zones showing large bias would certainly move depending on the wind direction. An averaging process could then be carried to determine optimal siting positions considering the complete wind rose.

Even though the generalization of Bingöl *et al.* (2008)'s method was realized only for a simple geometry, it revealed to be particularly useful at showing the extent of LiDAR bias caused by topography. Of course, the results are closely linked to the CFD model employed and this might be considered a limitation of the method. For instance, the model used in this experiment makes use of wall functions which might not be appropriate for very complex geometries where recirculation could appear. However, in order to benefit from such computationally inexpensive models, these problematic zones where topography is highly complex could be identified with the help of contour and slope maps prior to CFD analysis. It could also be interesting to run the same experiment with different turbulent models like $k-\omega$ Shear Stress Transport (SST) for example to assess the impact of turbulence modeling on LiDAR bias modeling. An unsteady approach, such as LES (Large Eddy Simulation), would also reproduce more faithfully the turbulent fluctuations of the flow measured by the LiDAR but would at the same time require more computionnal resources.

CHAPTER 4

APPLICATION TO COMPLEX TERRAIN: ANSE-À-VALLEAU

This final chapter is dedicated to estimating LiDAR bias for a wind field over a real complex terrain. However, before actually proceeding to this task, the performances of the CFD model over complex and forested terrain will be evaluated with the help of both experimental and numerical data. Meteodyn WT 4.0 will be used for the numerical comparison.

Secondly, the numerical results obtained for the CFD validation will be used to evaluate LiDAR bias. This bias will also be compared to experimental data as well as numerical data. The numerical comparison will be done both with Meteodyn WT 4.0 and WAsP Engineering.

Note that the above mentionned tasks use different sets of experimental measurements for their validation and each set will be described independently.

4.1 Experimental setup

The experimental data available for the CFD validation corresponds to wind speeds recorded by calibrated anemometers at 60m from March 14 2007 to December 29 2009 in Anse-à-Valleau (AAV) in Gaspe peninsula, Quebec, Canada. The meteorological masts on which the equipment is installed are called AAV3 and AAV6 and are approximately 4km away from each other. Their position can be seen in figure 4.2. Data was quality controlled by GL Garrad Hassan and all suspicious data was eliminated. Data between 15 and 199° was removed due to turbine wake at AAV6. Only concurrent data was kept, which means that whenever data was unavailable for one mast, the data sample was neglected for both masts. The resulting wind roses for both masts are presented in figure 4.1. For each available wind direction, both the experimental and simulated ratios of horizontal wind speed $U_{horAAV3}$ / $U_{horAAV6}$ will be compared.

4.2 Domain and mesh

4.2.1 Domain sensitivity analysis

The size of the domain and the position of the boundaries are very important aspects in CFD. Much care must be taken when choosing the position of the boundaries. The objective is to place the boundaries far enough from the zone of interest to prevent them from having an effect

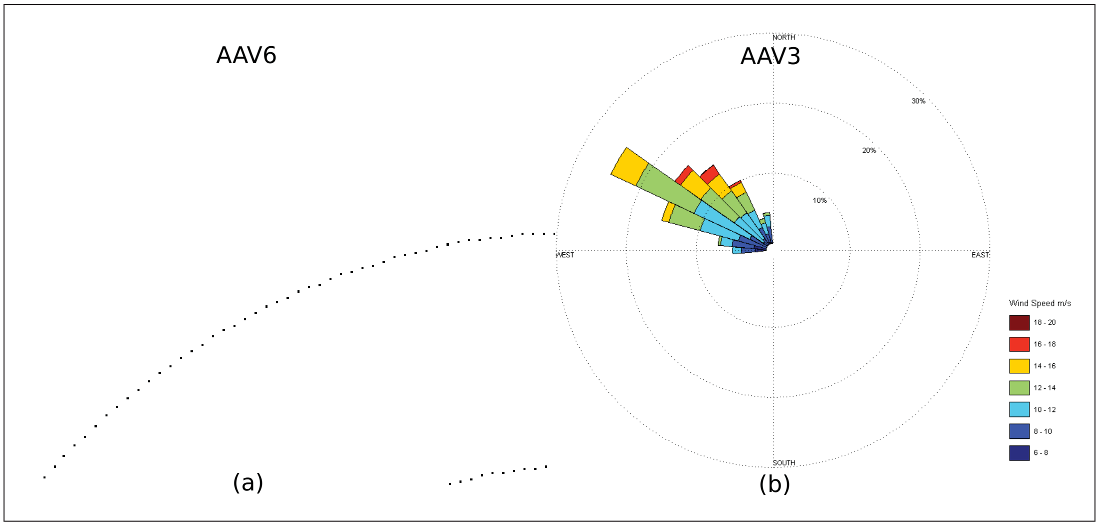


Figure 4.1 (a) and (b) show the wind distributions for masts AAV6 and AAV3.

on the solution. It must be kept in mind that a large domain means high computational cost. Therefore, in order to get the best compromise, an iterative approach was adopted. The steps of this approach go as follows

1. Define a domain enclosing both masts and chose a reasonnable distance from masts to boundaries. Run a simulation on this domain;
2. Starting from the domain defined in the previous step, move the inlet a couple nodes away from the masts;
3. Repeat step 2 until solution shows a monotonic behavior and results do not vary significantly relative to the previous simulation;
4. Repeat step 2 and 3 for all lateral boundaries.

The above procedure is illustrated in figure 4.2 (black rectangles). It is crucial to keep the cells' dimensions constant to maintain the same nodal positions for AAV3 and AAV6 and thus avoid interpolation when sampling results.

The top boundary was placed at 3500m, which is about ten times the height of the highest obstable in the domain. The purpose of such a high value is to avoid blockage effects. Note that it could probably be optimized using the above mentionned procedure.

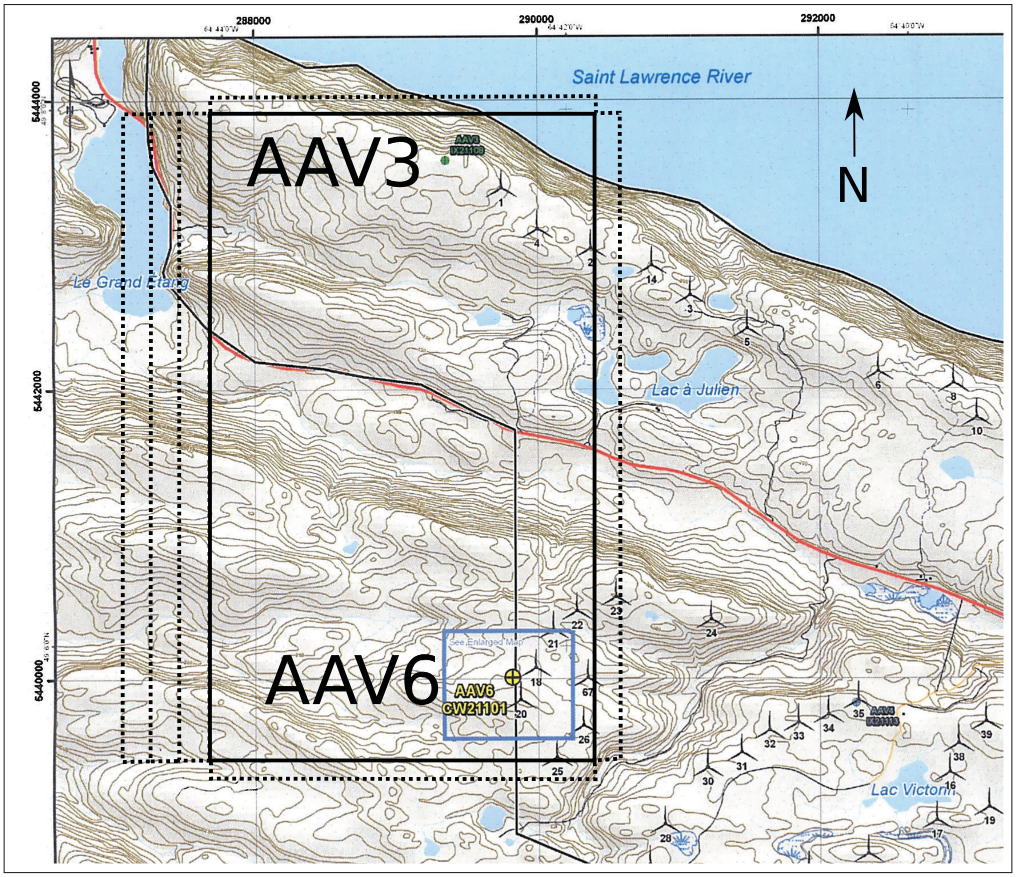


Figure 4.2 Map of Anse-à-Valleau adapted from Jeannotte *et al.* (2011) with positions of masts AAV3 and AAV6 illustrating domain sensitivity analysis.

It was decided to proceed with the domain sensitivity analysis using the wind directions where the topography is the most complex for both masts. Again, this is a compromise between accuracy and computational cost. The best approach would be to perform this analysis for all directions since every direction is a new "problem". The wind roses presented figure 4.1 show that the wind blows in sectors 270° to 350°. Thus, the most complex and out-of-wake wind direction in terms of topography for both masts corresponds to 270° since for all other directions, wind comes from the water for AAV3.

The results of the above-mentionned procedure are presented in table 4.1.

The initial domain dimensions (highlighted in grey in table 4.1) were chosen such that both masts were included and that a sufficient distance upstream was present to let the flow develop before getting to the points of interest. Of course, these dimensions are a first guess and this

Table 4.1 Domain sensitivity analysis

# of nodes (Δx≈ Δy≈35m)				**AAV3(m/s)**			**AAV6(m/s)**			$\frac{U_{horAAV3}}{U_{horAAV6}}$	**Variation of**
West	**East**	**North**	**South**	U_x	U_y	U_{hor}	U_x	U_y	U_{hor}		$\frac{U_{horAAV3}}{U_{horAAV6}}(\%)$
105	30	75	75	6.97	0.57	6.99	7.29	-0.11	7.29	0.95	-
135	30	75	75	6.69	0.58	6.71	7.18	-0.09	7.18	0.935	-2.50
165	30	75	75	6.82	0.59	6.85	7.42	-0.11	7.42	0.922	-3.89
195	30	75	75	6.81	0.62	6.84	7.19	-0.14	7.19	0.952	-0.79
165	**45**	75	75	6.86	0.56	6.88	7.43	-0.06	7.43	0.927	0.49
165	30	**90**	75	6.83	0.56	6.85	7.42	-0.13	7.42	0.923	0.09
165	30	75	**90**	6.84	0.60	6.87	7.45	-0.09	7.45	0.921	-0.08

study should reveal the optimal locations. The results of the smallest domain were used to calculate the variations (last column) for the cases where only the West boundary is moving. For these cases, the boundary is moved away by 30 nodes each time, which corresponds to a displacement of roughly 1km. As expected, the position of the inlet has a noticeable impact on the solution. Unfortunately, it was not possible to obtain a descending monotonic variation of the ratio of horizontal velocities of AAV3 and AAV6. By moving the inlet from 105 to 195 nodes (3.6km to 6.8km), variations were kept below 4%, which was considered sufficient. Finally, it was decided to use the simulation with 135 (highlighted in green in table 4.1) nodes because it includes the major topographic obstacles.

The last tested case for the location (195 nodes) of this boundary contains over ten million cells, an amount of cells that starts to be impractical in the industry if no cluster is available. Eventually, it is believed that pushing away the boundary even more would help achieve a monotonic behavior. However, this calls for additional computational ressources. Using a simple geometric expansion of the grid at the entrance of the domain could be envisaged. A combination of micro- and meso-scale could also be considered. The micro-scale part would contain the zone of interest (the masts in this case) and the meso-scale part could reach the surrouding topography. The meso-scale portion could potentially generate more appropriate boundary conditions for the micro-scale portion.

For the three remaining side boundaries (East, North and South), a displacement of 15 nodes was tested. The reason for using half the number of nodes as for the West boundary is explained by behavior of the streamlines which tend to stay parallel to the undisturbed velocity over the site of AAV as depicted in figure 4.3. The reference case from which are calculated the variations of $U_{horAAV3}$ / $U_{horAAV6}$ for these boundaries is highlighed in gray in table 4.1. None of these revealed a significant variation of the results, thus initial dimensions were kept for further simulations.

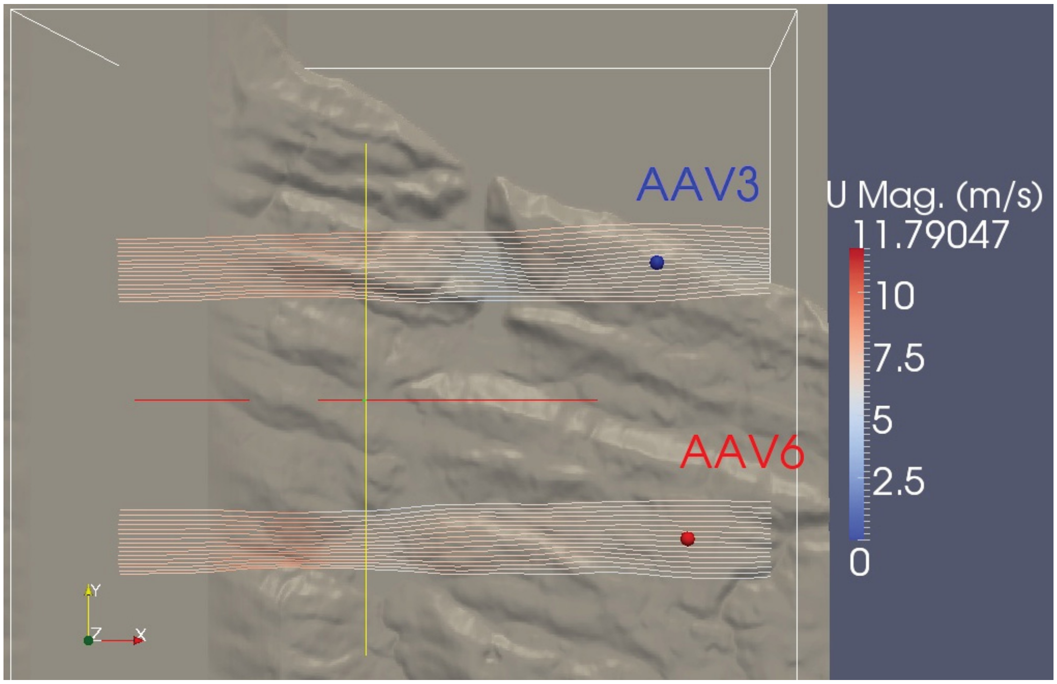


Figure 4.3 Streamlines surrounding both masts for wind direction 270° (from West to East). Streamlines were plotted at 60m above ground level (ABL) using XY-coordinates of masts as seeding points.

4.2.2 Mesh properties

In total, nine meshes were created: one for each direction ranging from 270° to 350° in 10°. For all cases, the mesh was aligned with the wind. Even though the number of nodes from the masts to the boundary is kept constant for every direction, (as shown in figure 4.4) the area between both masts changes. Consequently, the total number of cells is direction dependent.

In order to correctly capture the gradients near the bottom patch, a vertical expansion ratio of 1.045 was used. This results in 113 cells ∼1m to ∼160m high, from lower to upper boundary respectively.

Horizontal refinement was treated differently. Considering that the remote sensor is located roughly 80m from AAV6, it was decided that using the same meshes for both the assessment of

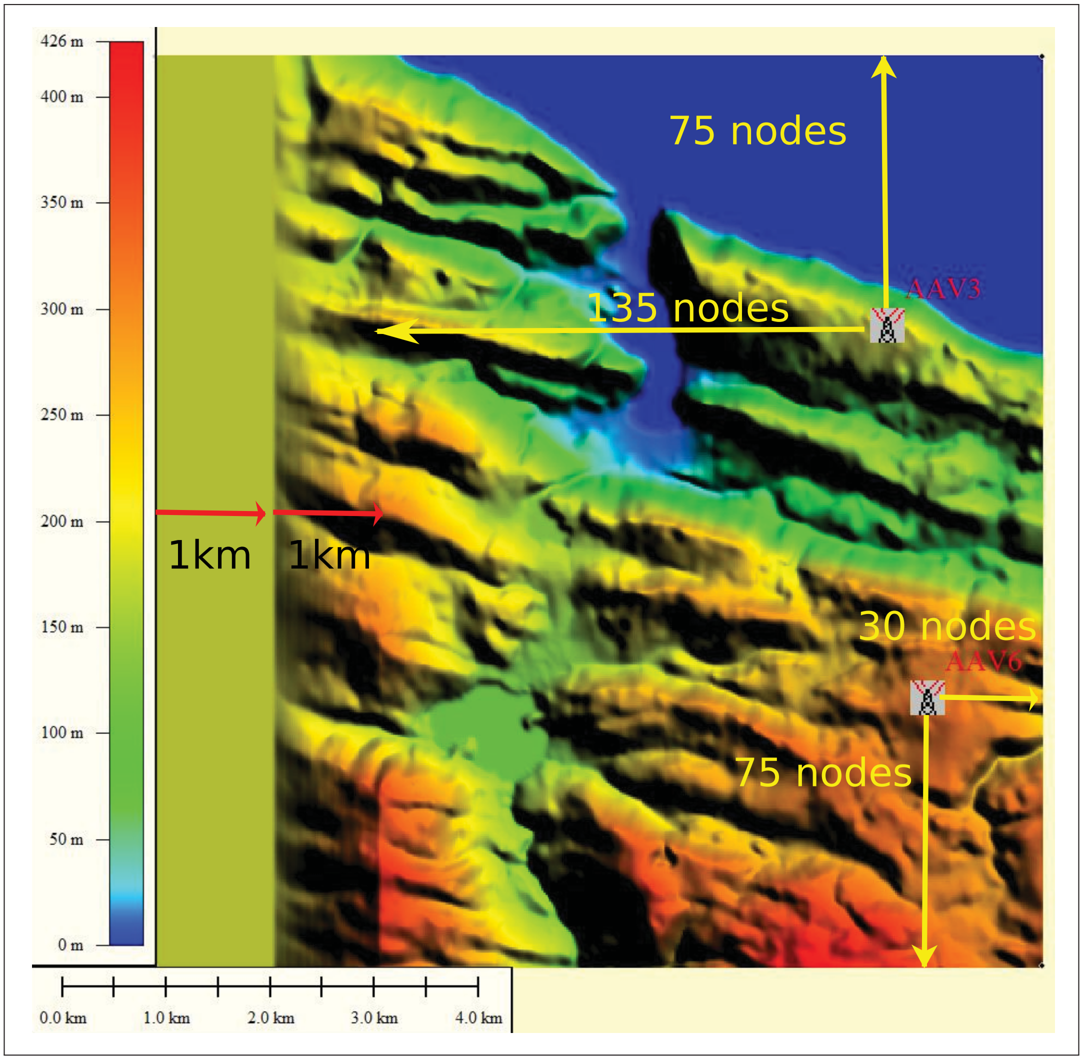


Figure 4.4 Computational domain for wind direction 270° showing distances from masts to boundaries.

the model and the reproduction of the LiDAR behavior would be a convenient approach. This route however must be taken with care when it comes to refinement. The refinement suggested in the literature for modeling the LiDAR should be around 10m (Boquet *et al.*, 2011). In order not to distort the results for the model assessment, it was decided to refine around both masts up to a radius of 300m in such a way that the LiDAR is included in the refinement. From 300m to 150m relative to a mast, cells are split once in the x and y directions. The same operation is repeated for cells in a radius of 150m relative to a mast. To bring down the horizontal resolution smaller or equal to 10m, cells are split twice in total. This local refinement was realized with a tool called *refineMesh*, which basically splits in half selected hexadra cells in selected directions (x and/or y and/or z). The intersection between the background mesh and the newly cut cells is filled with polyhedra cells. This is done to avoid having two faces of a given cell connecting only one face of another cell.

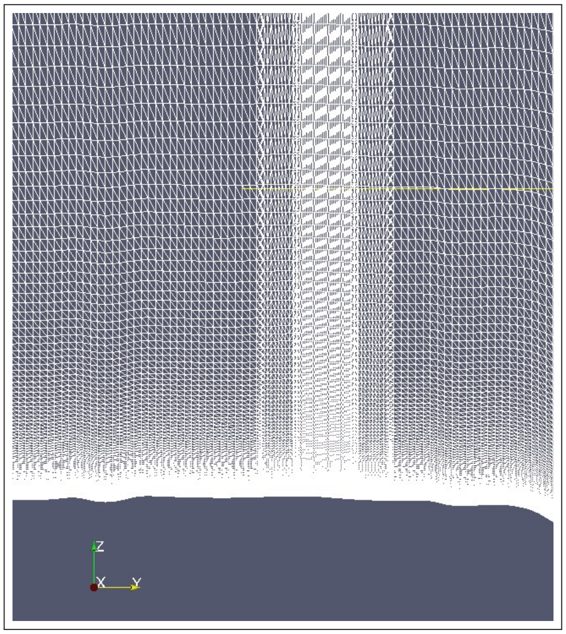


Figure 4.5 Mesh refinement over complex terrain (front view).

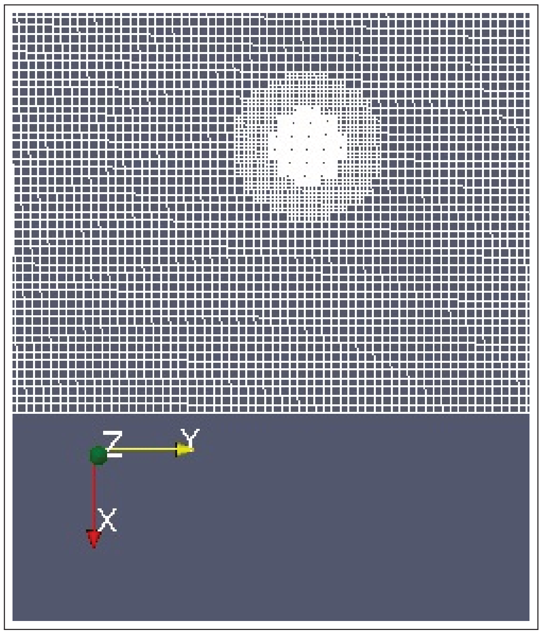


Figure 4.6 Mesh refinement over complex terrain (top view).

This brings up another important aspect: the size of the background mesh. The goal here is to minimize the discretization error while minimizing the number of cells. There exists a test called the "Grid Convergence Index" (GCI) (Celik *et al.*, 2008) that can help to quantify this discretization error. The test was conducted using the *x*-component of velocity at AAV6 at 60m for a wind direction of 270° (same as for domain sensitivity analysis). The size of the considered meshes are 1,525,997 cells, 3,237,951 cells and 7,153,126 cells. The grid refinement factor *r*, as suggested by the author , was kept such that ≈1.3. This factor is expressed as

$$r = \frac{h_{fine}}{h_{coarse}} \tag{4.1}$$

where

$$h = \frac{1}{N}\left[\sum_{i=1}^{N}(\Delta V_i)\right]^{1/3}. \tag{4.2}$$

N and V are the number of cells and the total volume of a given mesh respectively. The GCI revealed a discretization error of 3.86% for the finest mesh and an apparent order of the method to be ≈0.35. The fine mesh has a horizontal resolution of ≈35m outside of refinement zones. This resolution was kept for other directions.

4.3 Boundary conditions

The set of boundary conditions used for this set of simulations is presented in table 4.2. For details about the expressions presented, please refer to sections 3.1.2 and 3.2.6. The expression "0-Grad" is a condition that fixes the normal gradient to zero.

Table 4.2 Summary of boundary conditions used over complex terrain

	U	p	k	ε	ν_t
Lower Wall	(0 0 0)	0-Grad.	0-Grad.	$\varepsilon_P = \frac{\sqrt{C_\mu} k_P u_g^*}{\kappa z_P}$	$\nu_{tP} = \frac{C_\mu^{1/4} k_p^{1/2} \kappa (z_P - z_0)}{\ln(z_P / z_0)}$
UpperWall	$\frac{\partial U_x}{\partial z} = \frac{\rho u^{*2}}{\nu_t}$	0-Grad.	0-Grad.	$\frac{\partial \varepsilon}{\partial z} = \frac{u^{*3} \kappa}{\nu_t^2}$	Calculated
Inlet	$U(z) = \frac{u^*}{\kappa} \ln\left(\frac{z}{z_0}\right)$	0-Grad.	$k = \frac{u_g^{*2}}{\sqrt{C_\mu}}$	$\varepsilon = \frac{u^{*3}}{\kappa z}$	Calculated
Outlet	0-Grad.	0	0-Grad.	0-Grad.	Calculated
Front Wall	0-Grad.	0-Grad.	0-Grad.	0-Grad.	Calculated
Back Wall	0-Grad.	0-Grad.	0-Grad.	0-Grad.	Calculated

Note that logarithmic solutions for U, k and ε are prescribed at the inlet. It is of course an approximation for lack of experimental results over complex terrain. However, in order to limit the effects of such an approach, like recirculation for example, an artificially smooth transitional zone was added between the inlet and the real topography as depicted in figure 4.7.

Section "A" consists of a 1km-long plateau. The height (d) of this section corresponds to the average height of the site. Section "B" is also 1km-long and smoothly introduces the topography. It cannot be seen in figure 4.7 but as we approach section "C", more and more topographic features are added to section "B" until section "C" is reached where full topography is present. Note that the distance "A+B" is added to the distance found in the domain sensitivity analysis. An exception was made to this procedure for directions 340° and 350° since the inlet was falling completely in the water. For these two cases, the coast really acts as a large obstacle to the flow. Therefore, no smoothing zones were added and the height (d) was set to sea level. The 2km buffer was nonetheless kept for consistency with other directions. Figure 4.8 shows the computational domain for the case where the wind comes from 350°.

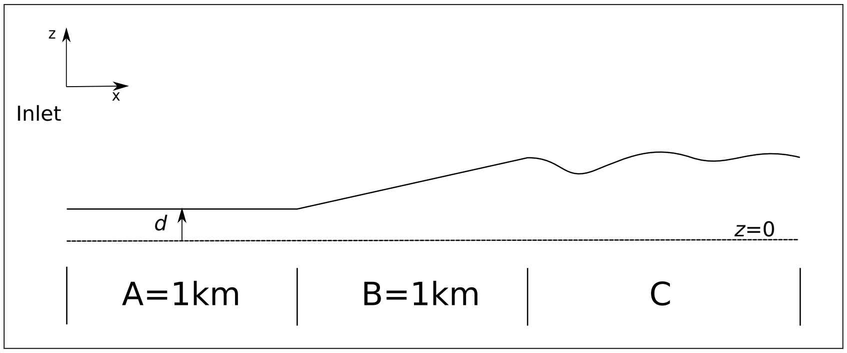


Figure 4.7 Smoothing process for application of logarithmic profiles at inlet

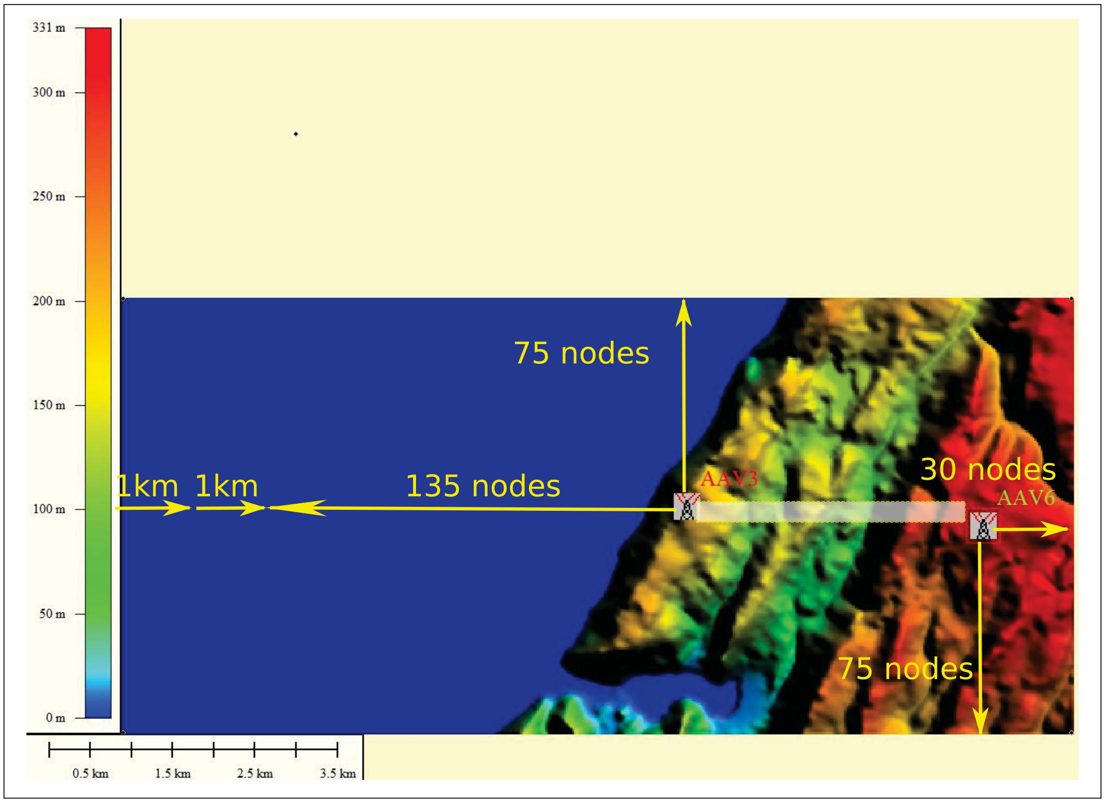


Figure 4.8 Computational domain for wind direction 350° showing distances from masts to boundaries.

For the set simulations realized in this chapter, the average elevation of the site was ranging from 100 to 200m. The reason why such a variation is observed is due to the rotation of the computational domain for each direction, which changes the region considered for the simulations. Of course, the presence of the water for certain wind directions pulls down the average elevation.

When considering the expression for the velocity at the inlet, there are two variables for which a decision must be made: the roughness length (z_0) and the friction velocity (u*). In the absence of experimental data, it is still possible to make educated guesses.

To determine a reasonable value for the roughness length, results obtained in section 3.2 for flow above forest over flat terrain were revisited. The profiles obtained for U, ε and k represent a fully developed flow over a dense forest; a flow that could be compared to one observed at AAV when the wind blows solely from the land. Thus, a roughness length value was calculated from this theoretical case using the standard logarithmic velocity profile expression. A value of $z_0 = 2.6$m was found, which is very high but expected since a dense forest has a great impact on the flow near the surface. The applicability of this value is more or less meaningful when the inlet is located partially on the ground and the water. For these cases, which corresponds to directions 290° to 320°, the value for z_0 was calculated following a simple linear approximation between $z_0 = 0.002$m and $z_0 = 2.6$m using the ratio

$$\frac{\text{Length of inlet on ground}}{\text{Total length of inlet}} \tag{4.3}$$

For 330°, only a few meters of the inlet were on the ground, so the roughness length was assigned as if it was only water. For the last directions, 340° and 350°, $z_0 = 0.002$m was used since the wind blows entirely from the water.

Concerning the friction velocity, a tuning approach was adopted. In order to obtain reasonnable velocity values at AAV6, u^* was tuned until the calculated horizontal velocity roughly matched the experimental value at 60m.

Finally, all computations were run with both variables and constant forest height. The constant forest height scenario was run with a forest drag coefficient of C_d=0.15, which falls in the recommended range for most vegetation in the literature (Katul *et al.*, 2004), and a forest height of 10m. Regarding the variable forest height case, three values of C_d were tested: 0.15, 0.015 and 0.0015. Testing different drag coefficient values was considered important since no information was provided for AAV about the type of forest. All these tests were performed with the same leaf area distribution, i.e. blackspruce.

4.4 Initialization

The solutions of these simulations were totally unkown. However, as an attempt to accelerate convergence, a forest displacement height was guessed and logarithmic solutions were prescribed above this height for U, k, and ε. For some cases, this strategy did not work and the solutions diverged. Hence, in order to reach convergence for these scenarios, k and ε were set to $\approx$10e^{-3} and U was set to 0.

4.5 Meteodyn parameters

Meteodyn is a commercial FVM based CFD package used in the wind industry. It will be used as a reference for the model developed in OpenFOAM. The control over the input parameters is of course much more limited than OpenFOAM. For this reason, only this section will be dedicated Meteodyn's configuration.

It is important to mention that the licence allowed the software to run only on one processor and 2GB of RAM, as compared to previous simulations realized with OpenFOAM where multiple processors and much larger amount of memory were available. Therefore, in order to keep computational time acceptable, the size of the mesh was roughly decreased by 40% relative to OpenFOAM simulations. It is however difficult to comment about the effect of such a reduction of the mesh size since the order of the method used by Meteodyn is unknown. Considering the results of the GCI conducted previously for OpenFOAM computations (section 4.2.2), it seems that reducing the amount of cells in OpenFOAM to the same amount as for the computations realized with MeteoDyn would not have significantly improved the results.

Furthermore, the refinement was done differently since the construction of unstructured meshes is not possible in Meteodyn. To refine a particular location, one must define a result point in the domain, a desired minimum horizontal cell dimension and an horizontal expansion factor. Then, the software starts from this location with the mentionned cell dimension and contructs the mesh up to the boundaries following the given horizontal expansion. The result is a fully structured and orthogonal curvilinear mesh containing only hexahedra. Note that the mesh is always aligned with the wind flow.

Again, the mesh was done so that simulations will be used for model assessment and LiDAR treatment. Thus, refinement was done around both the masts and the LiDAR. An effort was also made to keep approximately the same dimensions from masts to boundaries as those found in the domain sensitivity analysis presented in table 4.1. However, Meteodyn assigns the same distance from domain center to all boundaries. The domain center was then set as the center point between both masts. A distance equal to the one found between inlet and masts in section 4.2.1 was imposed, which correspondonds to 165 nodes (135 upstream + 30 for transitional zone B in fig. 4.7)*35m$\approx$6km. An extra 1km buffer was added to compensate for the distance from center point to masts, as illustrated in figure 4.9. Table 4.3 also summarizes important data about the mesh.

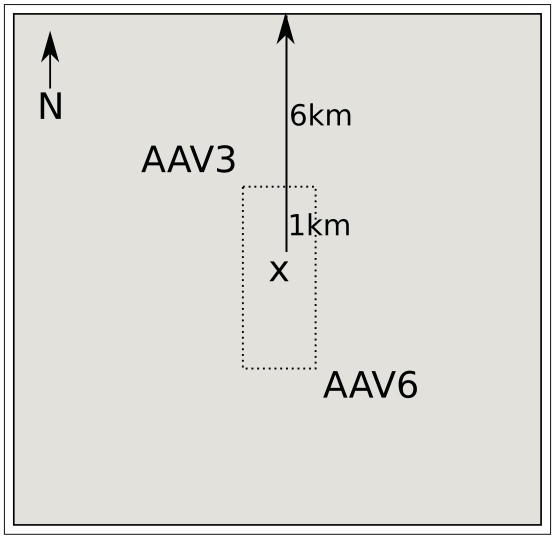


Figure 4.9 Computational domain for wind direction 270° showing distances from domain center "x" to boundaries.

Table 4.3 Meteodyn mesh parameters

Minimum horizontal resolution (m)	10
Minimum vertical resolution (m)	1
Horizontal expansion factor	1.045
Vertical expansion factor	1.15
Distance from domain center to boundary (m)	7000

The fundamental difference between Meteodyn and the model developed in OpenFOAM resides in the turbulence model. On top of the RANS equations, Meteodyn solves a transport equation for the turbulent kinetic energy that varies according to the turbulent lenght scale, which in turn depends on the mixing length. The latter is calculated based on the presence of forest. As opposed to the turbulence model used in OpenFOAM, there is no dissipation equation. Sink term accounting for the forest in the *k*-equation is also different. For further details, the reader is encouraged to consult Meteodyn's user manual.

There are also some differences in the boundary conditions used. At the inlet, Meteodyn also uses logarithmic profiles for U and k within the surface layer. Above the surface layer, k gradually decreases up to the upper boundary and Coriolis effect is considered for generating the velocity profile. Friction velocity was not tuned to match experimental velocity. The ground boundary conditions are also similar except for the treatment of the forest, which is explained in section 2.2.2.4. Symmetry conditions are used for lateral boundaries. Homogeneous conditions are applied at outlet and upper boundary for the pressure.

Finally, the solver used by Meteodyn is MIGAL. It solves simultaneously the velocity and the pressure. (Ferry, 2000)

4.6 Results and discussion

Figure 4.10 shows the ratio of horizontal velocity between masts AAV3 and AAV6. The OpenFOAM horizontal velocity was calculated using

$$U_h = \sqrt{U_x^2 + U_y^2} \tag{4.4}$$

whereas it was explicitly given in Meteodyn's output and experimental masts data files. The blue forks indicate one standard deviation of the ratio for each direction. For a better understanding of the results and discussion, figure 4.11 shows the computational domains for wind directions 270° and 350°.

For directions 270° to 290°, both sofwares perform very similarly and are at the limit of one standard deviation below the average ratio. This could be explained by inappropriate inflow boundary conditions. For these directions, the wind comes almost entirely from the land. The logarithmic profiles used at the inlet in both cases, even with a smoothing procedure, seem not perfectly suited for forested and complex terrain. Then, for 300° to 350°, results considerably improve, especially for Meteodyn. This seems to be due to the inlet boundary moving more and more in the water, which makes the inflow boundary condition for both models more appropriate. Meteodyn better reproduces experimental data for 290° to 320°. It was found that Meteodyn uses topographic and roughness data up to 2km outside the computational domain to define its boundary conditions. Consequently, for directions 290° to 310°, Meteodyn would then use a wind profile coming from the water, which would explain the better results.

Still refering to figure 4.10, the variable forest height approach does not seem to significantly improve the results when compared to the constant height approach. The results presented here for a constant forest height (black markers) were obtained using a mean height of 10m. Note

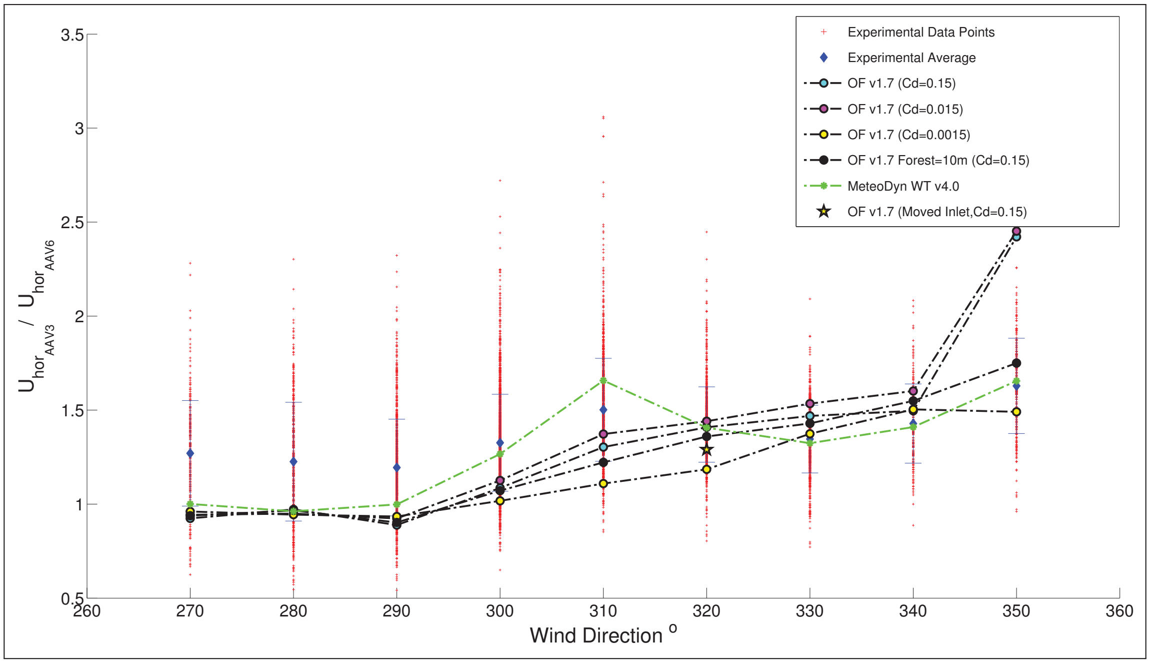


Figure 4.10 Horizontal velocity ratios of AAV3/AAV6 per direction obtained from OpenFOAM and Meteodyn accompanied by experimental data.

that the distribution of forest on the site of AVV is quite uniform, which may explain the similar performances of both scenarios. However, using accurate forest height data would probably be crucial on a site with highly heterogeneous forest distribution.

It is difficult to comment about the effect of the forest drag coefficient when the wind comes from the land (270° to 290°) for both masts as the results are presented in the form of velocity ratios. It is believed that regardless of the drag coefficient used, the wind deceleration is proportional for both masts, thus yielding similar horizontal velocity ratio. The effect of C_d can be better observed when the wind starts to originate from the water. AAV3, being located very close to the coast, is thus much less affected by the presence of forest than AAV6. As expected, low values of C_d led to lower velocity ratios since the flow is able to maintain its momentum and thus the wind speed seen at AAV6 is slightly higher. Surprisingly, not much difference can be observed between C_d=0.15 and C_d=0.015. For wind directions 330° to 350°, where we are most confident about the inlet boundary condition, C_d=0.0015 is in very good agreement with experimental data and Meteodyn. The variations observed at 350° seem to be due to a local discontinuity in forest height near AAV3 which mostly affects the cases where C_d=0.15 and C_d=0.015. The tuning process used to determine a suitable forest drag coefficient for the site of AAV yielded a value of C_d=0.0015.

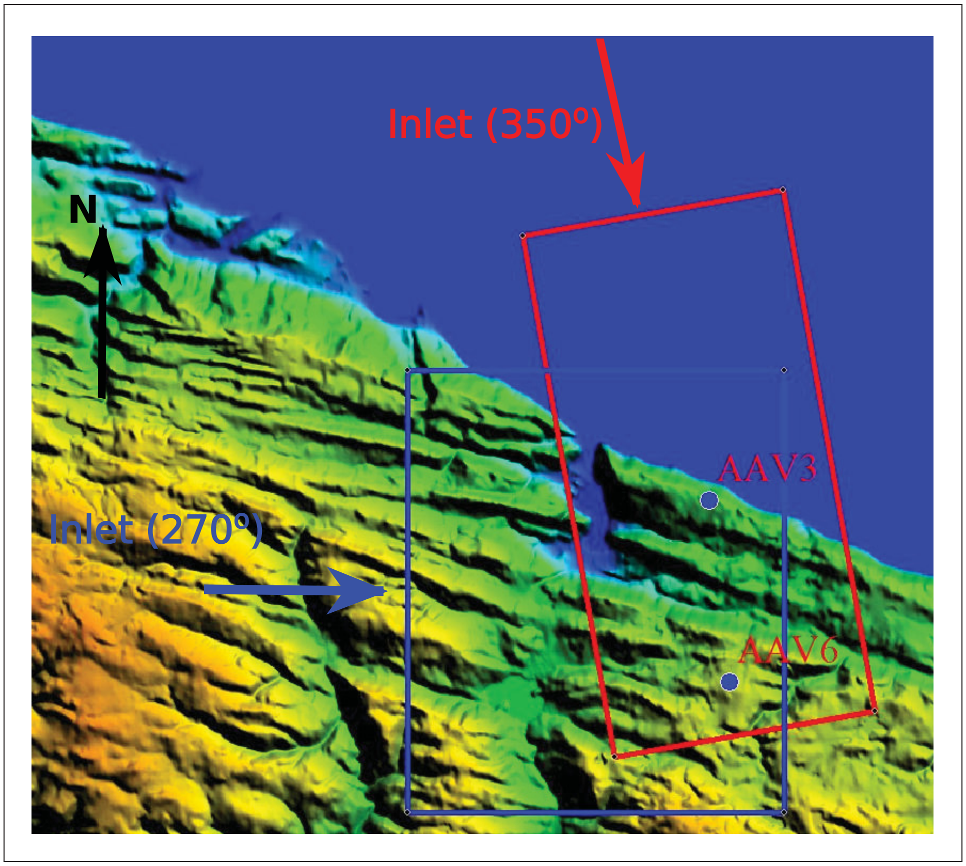


Figure 4.11 Computational domains for wind directions 270° and 350° in a non-rotated frame.

Another good indicator of flow behavior is the turbulence intensity (TI), which is the ratio of the velocity fluctuation to mean velocity. It can be expressed as

$$TI = \frac{\sqrt{\frac{2}{3}k}}{\overline{U}}. \tag{4.5}$$

Representative average TI values provided by GLGH for the site of AAV were compared to calculated TI values. Results are presented in figure 4.12 and table 4.4.

The transition of the wind from the land to the water is very well shown in figure 4.12 (a). It translates into a significant reduction of turbulence intensity at AAV3. It is interesting to see that forest has little to no impact for directions 330° to 350° as the wind comes exclusively from the water. A drag coefficient of 0.0015 produces average experimental TI values very close to experimental averages for both masts, especially for AAV6, further supporting the validity of this value when used with the blackspruce leaf area distribution.

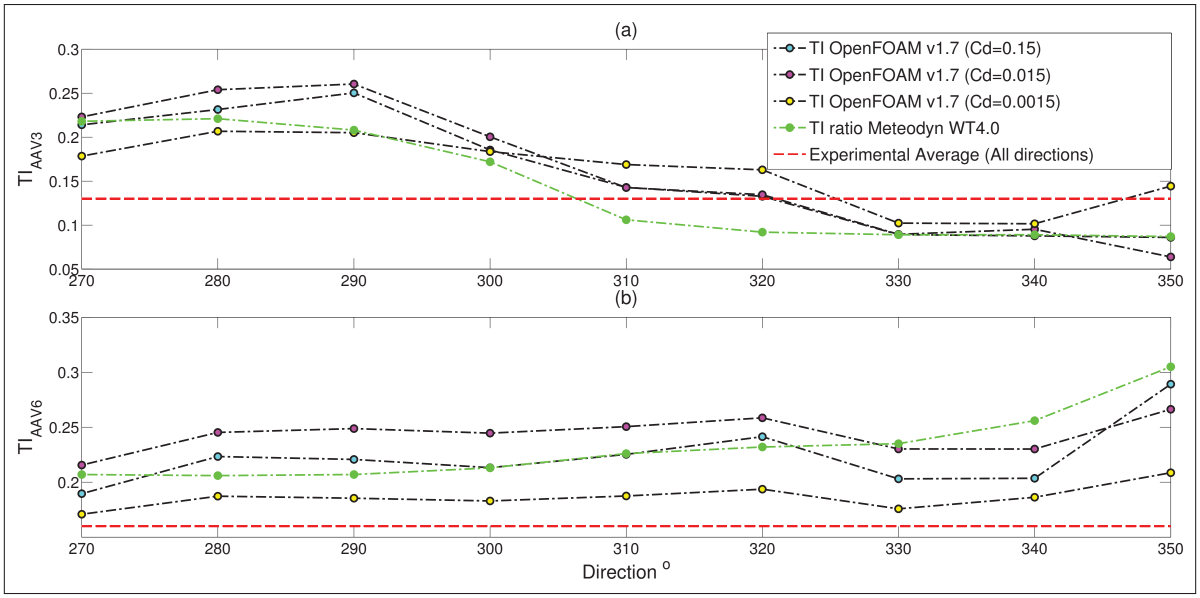


Figure 4.12 Tuburlence intensity per direction for (a) AAV3 and (b) AAV6 obtained from OpenFOAM and Meteodyn accompanied by experimental averages.

Table 4.4 Average TI values for different forest drag coefficient

C_d	AAV3	AAV6
0.15	0.158	0.223
0.015	0.163	0.243
0.0015	0.161	0.186
Experimental Average	**0.13**	**0.16**

Indeed, the forest drag coefficient does influence turbulence but its effect is obviously not linear. Surprisingly, the intermediate value C_d=0.015 presents, in general, higher TI values than C_d=0.15. To further analyze this aspect, $|U|$ and k were studied independently at AAV6. Profiles of both variables are presented in figure 4.13. The friction velocities presented in figure 4.13 are the ones imposed at the top boundary to match the experimental velocity at 60m and are also used to calculate k prescribed at inlet. The reason why the TI is highest when C_d=0.015 is simply due to the high value of k given as inlet condition, which is in turn calculated based on the required friction velocity necessary to match the horizontal wind speed at AAV6. Hence, the real interrogation is why does the case of C_d=0.015 require a higher shear stress than the case where C_d=0.15? It can be explained by looking at the source term introduced in the

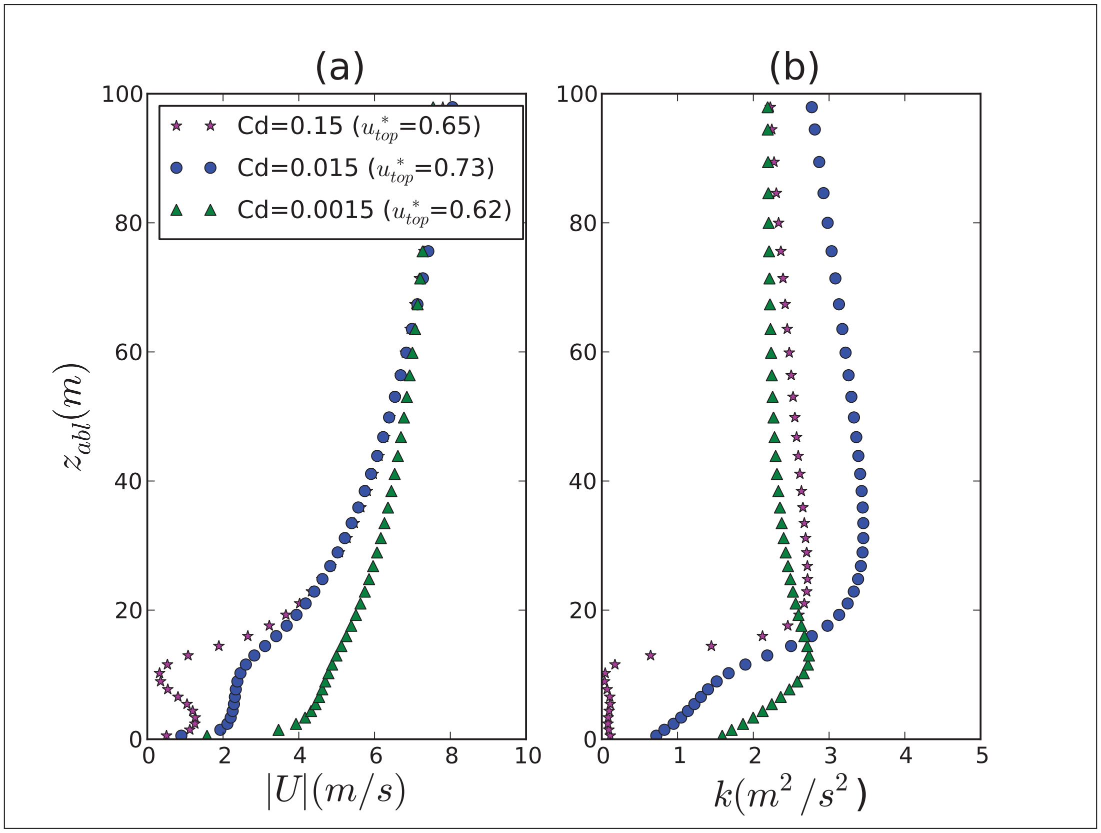


Figure 4.13 Velocity (a) and TKE (b) profiles at AAV6 when the wind comes from 270°. The height of the forest at this location is 14.5m.

momentum equation (eq. 2.18):

$$S_u = -\rho C_d \alpha u_i |u|$$

The only forces acting on flow are the drag force created by the forest and the force induced by the shear stress applied at the top boundary. The effect of surface roughness is neglected as it is very small compared to the drag generated by the forest. Thus, when increasing C_d by a factor of ten, the velocity should decrease by a factor $\sqrt{10}$ to maintain a constant drag force. Taking for example the cases where C_d=0.015 and C_d=0.15, the velocity decreases "faster" than the drag coefficient, resulting in a lower drag force. Therefore, a lower shear force is required at upper boundary to achieve to desired velocity at AAV6.

4.6.1 Direction 320° revisited

It was mentionned in the last section that we expect better simulation results when the inlet boundary is in the water. This is mainly supported by the fact that the true wind flow coming from the water is probably very close to a logarithmic wind profile. To verify this hypothesis, direction 320° with C_d=0.0015 was revisited. The inlet, which was initially halfway between the water and the land, was pushed away completely in the water, resulting in a 16km-long domain containing more than eleven millions cells. Refering to figure 4.10, results are convincing. The ratio of horizontal velocity moved inside the uncertainty range, which leads to the conclusion that inlet boundary conditions have a significant impact on the solution. More precisely, not only the location of this boundary is important, but the actual values that are prescribed as well. The result still does not agree perfectly with Meteodyn neither experimental results, but it is important to keep in mind that models use completely different turbulence models, which might benefits Meteodyn in this particular context. For example, Katul *et al.* (2004) showed that one-equation closure models were sometimes more appropriate than higher order models in the presence of forest and that the higher level of complexity of the latter was not bringing much advantages.

The location of the inlet does have an important impact, but other factors such as boundary condition and turbulence modeling probably play a great role as well.

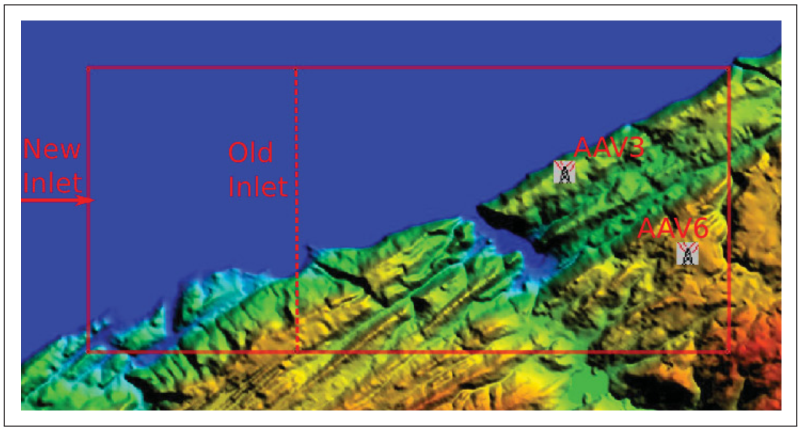


Figure 4.14 New position of inlet for wind direction 320°.

It also raises a worrying issue about how inlet should be treated when flow analysis must be done over a complex terrain where there is no flat terrain in the surroundings neither experimental results to use as input conditions. Meteodyn's approach of using topography and roughness outside the computational domain is an interesting solution to this problem. As mentionned in section 4.2.1, in OpenFOAM one could consider a larger computational domain by using coarse geometric expansion for cells outside the zone of interest.

4.7 Modeling LiDAR bias

The CFD model has shown to perform relatively well over complex terrain and in the presence of forest according to results obtained in section 4.6. Thus, these numerical results were used to evalute LiDAR bias. Only directions where concurring LiDAR and mast data was available (280° to 330°) were used. The OpenFOAM case with C_d=0.0015 was chosen as it seemed to perform better.

Before presenting the LiDAR bias, it is first important to describe where exactly the instrument was located and discuss about the surrounding topography.

4.7.1 Site and equipment

The LiDAR was also sited at AAV, nearby an operating wind farm in a densely forested area in complex terrain and harsh winter conditions as depicted in figure 4.15. It was located 79m Southwest of AAV6 as shown in figure 4.16. The measuring campaign officially started on February 17 2011 and ended March 28 2011. Both instruments recorded 1-second data and then produced 10-minute averages. The data was measured at the same altitude for both instruments ($\approx$ 80m). Wind was coming mainly from Northwest, as it can been seen on the windrose presented in figure 4.16. Other wind directions were available but filtered out to avoid turbine wake.

4.7.2 Terrain complexity classification

Pelletier *et al.* (2011) investigated the complexity of the area surrounding the LiDAR at AAV. Terrain complexity was decomposed in three distinct aspects: orography, ruggedness due to surrounding vegetation, and ruggedness due to roughness variation of the surroundings. Ruggedness (both for vegetation and roughness variations) was characterized based on Dorval (2011) whereas orography was analyzed using IEC 61400-12-1 and IEC 61400-12-2 standards. Terrain was analyzed up to a range of 340m around the remote sensor, which corresponds to eight times the radius of the scanned disc when the instrument shoots at 80m (see figure 4.17).

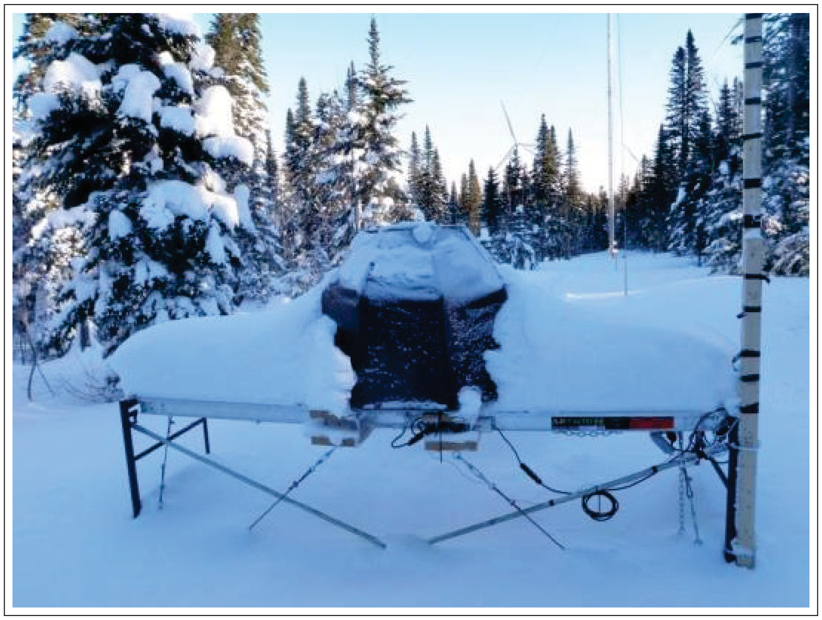


Figure 4.15 LiDAR on 2.5m platform deployed at AAV (Jeannotte *et al.*, 2011).

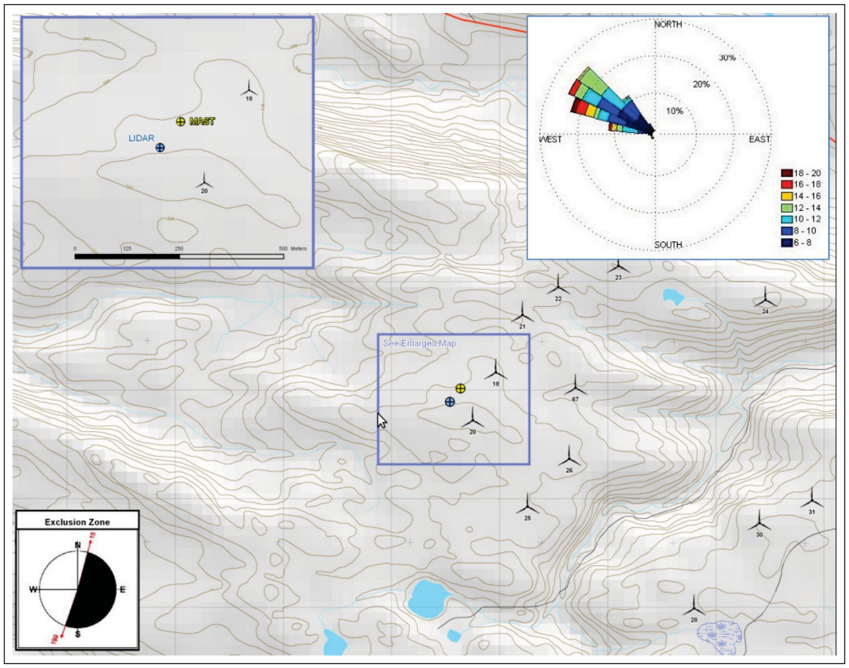


Figure 4.16 AAV Topographic map showing AAV6 (yellow dot) and the LiDAR (blue dot) (Jeannotte *et al.*, 2011).

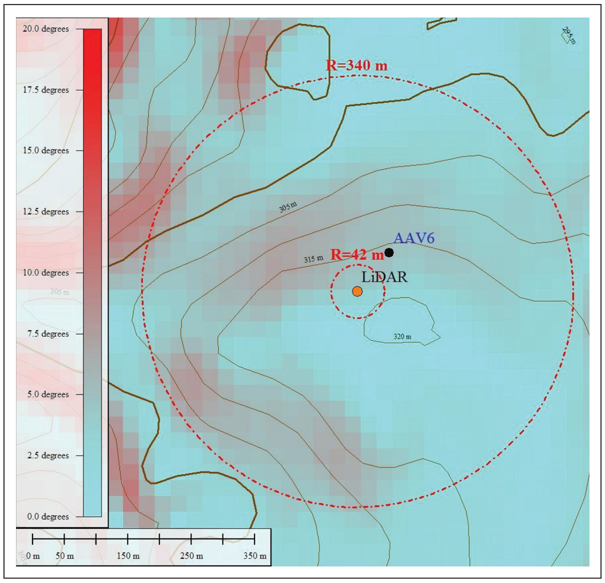


Figure 4.17 Positions of AAV6 and LiDAR on actual topography with 5m contour lines and slopes (color).

The shaded area in figure 4.18 represents the in-wake wind sectors determined based on IEC 61400-12-1. For the out-of-wake wind sectors, the terrain complexity can be classified as follows

- Orography complexity can be classified as being moderately (Class 2) to highly (Class 4) complex;
- Roughness complexity due to the surrounding vegetation can be classified as being moderately (Class 3) to highly (Class 4) complex;
- Roughness complexity due to the surrounding roughness variation can be classified as being lightly (Class 1) to moderately (Class 2) complex.

For more information about the terrain complexity classification, please refer to Dorval (2011) and Pelletier *et al.* (2011).

4.7.3 Data quality control

In order to eliminate all data potentially affected by equipment failure or any other anomalies, three quality control procedures are applied to the LiDAR data. The first filtering level is performed automatically by the LiDAR based on returned signal features and quality. The

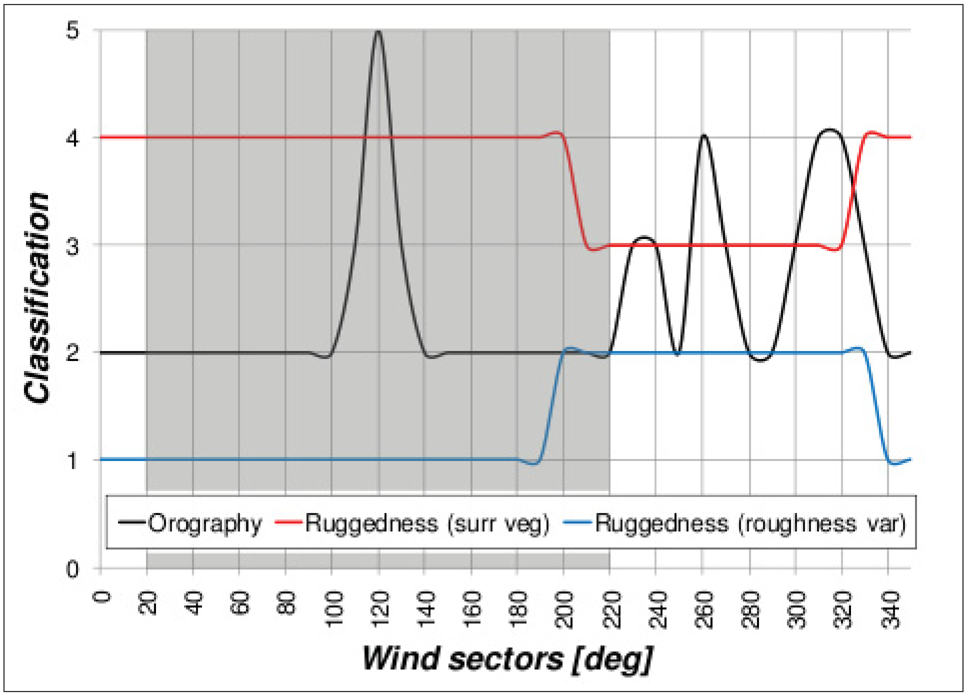


Figure 4.18 Terrain Complexity Classification for AAV terrain from Pelletier *et al.* (2011).

second filtering level is performed with a software using the following data quality control criteria

- 80% availability in 10 minutes;
- Horizontal wind speed within (0 to 60 m/s) and wind directions within (0 to 360°);
- Vertical wind speed within -2 to 2 m/s;
- Horizontal and vertical wind speed standard deviation validation (<5 m/s);
- Off-level Tilt-X or Tilt-Y less than 1°;
- Carrier to Noise Ratio (CNR)<-22 dB.

Whenever any of the above criteria was not respected, data was removed from further analysis. Finally, the last LiDAR quality control is a manual check to insure all readings are sensible.

Concerning the mast, all data subjected to icing and mast wake were removed. The details regarding how these events were identified by GL Garrad Hassan was however not available. Lastly, all the data comprised between 15 and 199° were removed due to turbine wakes.

Note that a global linear regression analysis was done between LiDAR and cup measurements. However, the graph showed a slope very close to one (≈ 0.99) and a very strong coefficient of determination (R^2), which did not reveal much information about LiDAR bias. Therefore, it was decided to compare LiDAR and cup measurements per wind direction.

4.7.4 LiDAR bias

As mentionned in the methodology, three sets of velocity ratios will be presented. The first set will show the bias caused by the flow homogeneity assumption, the second one will show the effect of spatial separation between LiDAR and mast and finally the last set is the combination of these.

For the two first sets of ratios, OpenFOAM results are compared to those of Meteodyn WT 4.0. For the last set of ratios, results are also compared to WAsP engineering results and experimental data.

Please refer to figure 4.19 for the location of Mast1', Mast2' and LiDAR'. It is reminded that Mast1' and Mast2' are horizontal wind speeds calculated from the CFD calculated wind components directly whereas LiDAR' is a horizontal wind speed calculated in post-processing assuming homogenous flow within the LiDAR scanned disc.

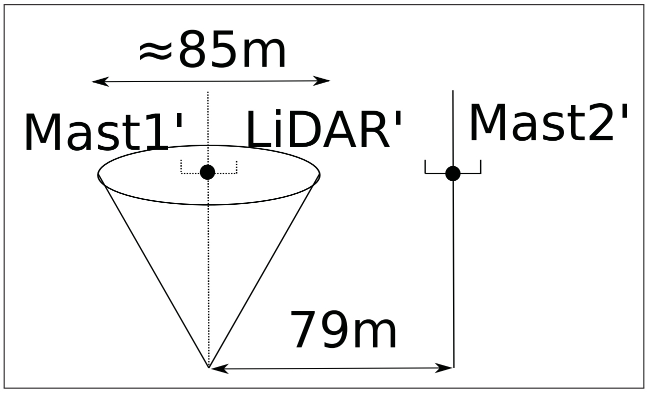


Figure 4.19 Locations of CFD defined points of interest. The prime(') means "CFD calculated".

What is considered here is the bias resulting from the flow homogeneity assumption. Graph 4.20 shows very little deviations from the unity ratio. Again, both softwares are within 1% agreement except for [310°-320°] where Meteodyn shows a bias slightly less than 0.5%.

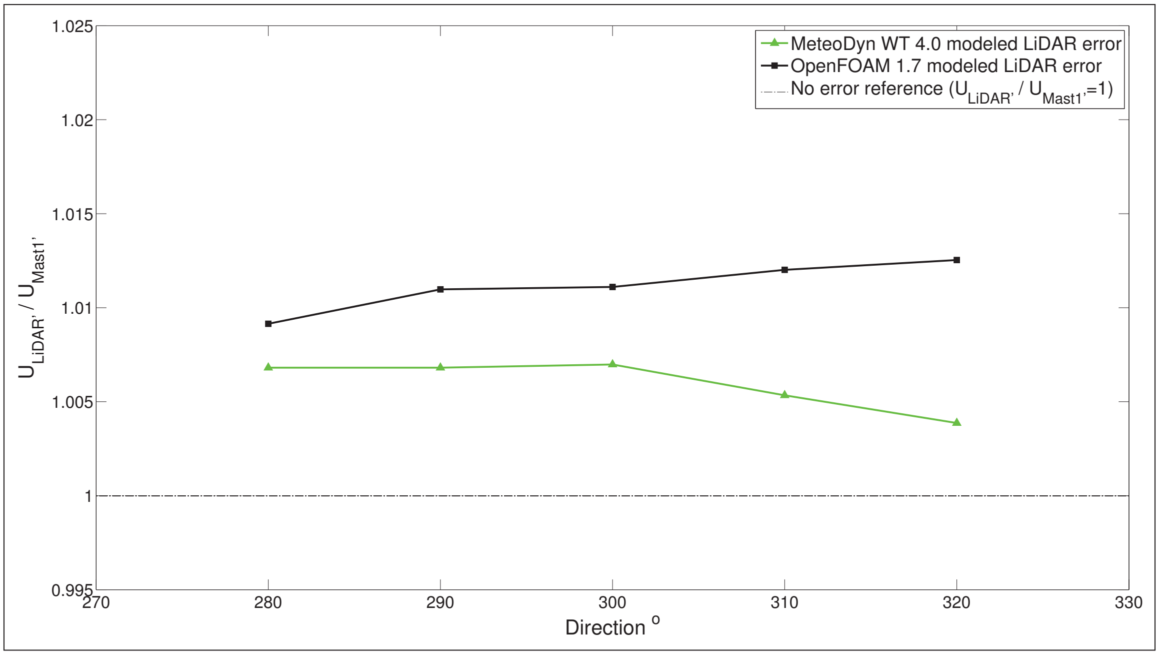


Figure 4.20 Modeled LiDAR bias caused by flow homogeneity assumption.

4.7.5 Terrain bias

Graph 4.21 shows the ratio of horizontal velocity of Mast2' over Mast1' per direction. Note that Mast1' is virtual and not present on the site of AAV. It was defined in the simulations to assess the influence of the spatial distance between the remote sensing instrument and AAV6. For this reason, it is impossible to compare these results with experimental data. Results show that even a distance of 80m causes flow variations, which demonstrates the importance of having collocated instruments for comparison purposes. Since there is no major obstacle in the surroundings of the mast and the LiDAR as shown in figure 4.17, the small deviations (0-1.5%) from a ratio of 1 were expected. Both softwares show quite good agreement especially for the interval [300°-320°].

4.7.6 Terrain and LiDAR bias combined

Finally, both graphs presented earlier are combined in order to compare with experimental results (fig. 4.22). Results obtained from the WAsP Engineering script developed by Bingöl *et al.* (2008) are also included. Domain used in WAsP Engineering is 3km x 3km and centered on LiDAR location. Horizontal resolution was set to 20m, which is the same used by the author of the method. The script was edited to take into account LiDAR offset with respect to true North.

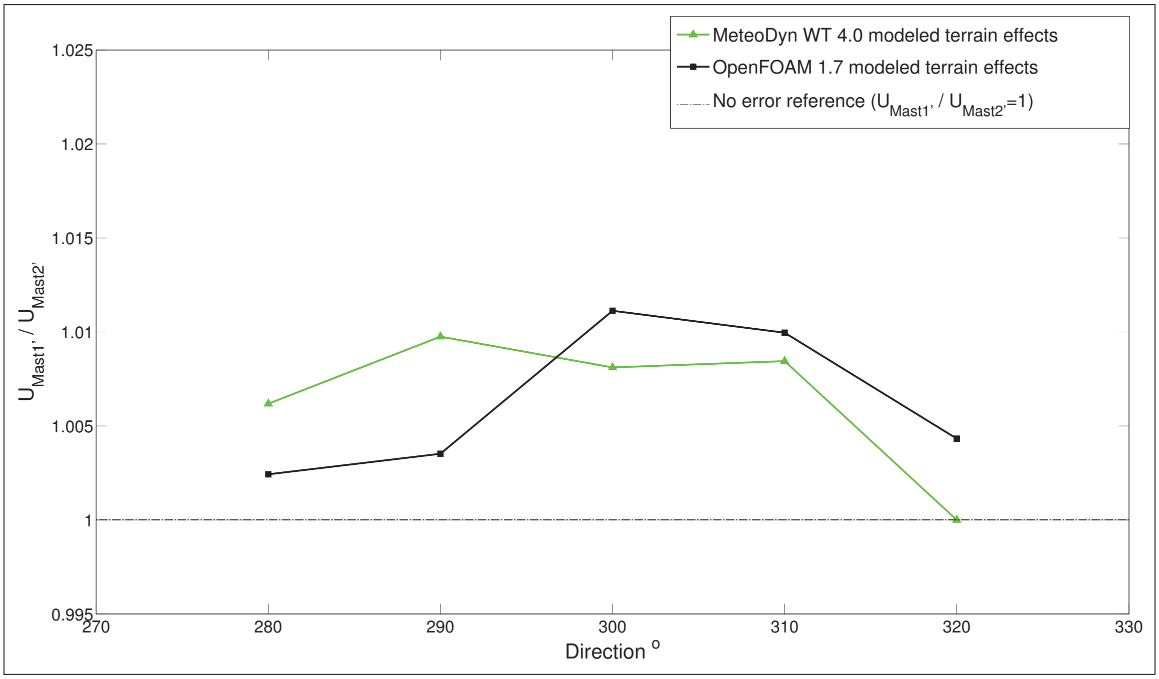


Figure 4.21 Modeled Terrain effects due to the distance between remote sensor and meteorological mast.

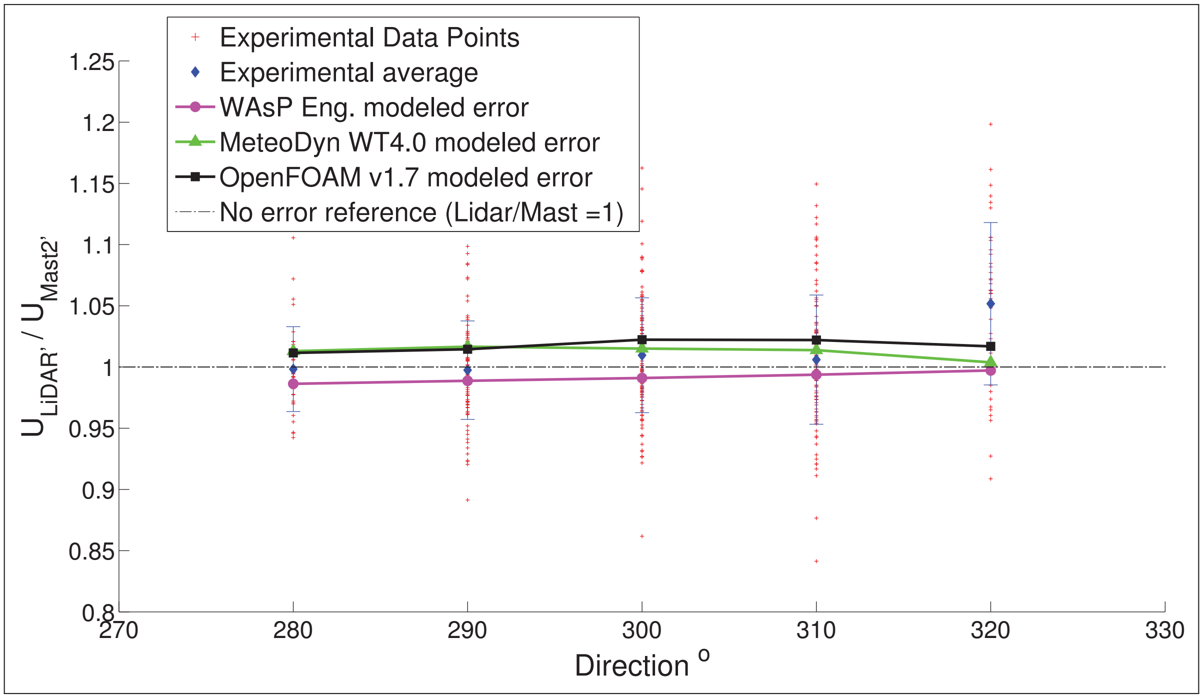


Figure 4.22 Terrain effects and LiDAR bias combined

OpenFOAM and Meteodyn tend to slightly overestimate the bias whereas the opposite is observed with WAsP Engineering. Despite a small difference between modeled and experimental bias for direction 320°, all softwares are well within uncertainty of experimental data. It is now obvious that the choice of flow model directly influences the predictions of LiDAR bias. In fact, the quality of the bias prediction is directly linked to the model's performances in modeling the atmospheric boundary layer over the given terrain.

Even though terrain was classified as moderately to highly complex for these directions, resulting LiDAR bias is very low, which questions both the validity of the terrain classification and the numerical simulations. Since all softwares led to very similar results, the first aspect might deserve more attention. The classification was made using topography up to eight times the radius of the scanned disc away from the LiDAR. When looking at figure 4.17, there is indeed considerable topographic changes up this radius. However, when considering only the actual area covered by the disc when the sensor scans at 80m, which is a circle of diameter ≈85m, terrain shows less than 5m variation in elevation and slopes smaller than 5°.This might explain the comparable results between WAsP engineering and the CFD. Therefore, it suggests that the flow seen by the instrument might not be affected by topography as far as eight times the radius of the scanned disc.

CONCLUSION

The main objective of this study was to implement a method to evaluate LiDAR bias over complex terrain, due the flow homogeneity assumption, with the help of the open source software OpenFOAM. As seen at the beginning of this thesis, LiDAR bias is caused by the flow homogeneity assumption made by the instrument within the volume enclosed by the four beams projected by the sensor. Bingöl *et al.* (2008) demonstrated that the bias mainly comes from the horizontal gradient of vertical velocity. Different techniques were presented to estimate and even correct LiDAR measurements. Among these, Bingöl *et al.* (2008)'s modeling approach was retained. In order to use this technique, an adequate atmospheric flow model needed to be developed in OpenFOAM.

First, simple 2D simulations were carried to make sure homogeneous neutrally stratified wind profiles could be obtained in OpenFOAM.

Then, the forest model already tested in Fluent by Dalpé and Masson (2008) was implemented and validated again in 2D. On top of that, a boundary condition proposed by Lussier-Clément (2012) was tested in the presence of forest and very good results were obtained. The advantage of this boundary condition is its total independence of the vertical position and thus forest displacement height created by forest.

Third, chapter 3.3 was dedicated to evaluating LiDAR bias over a theoretical case in order to evalute the post-processing tool. The independent impact of two non-dimensional parameters (λ and φ) was investigated first in a directional analysis with the LiDAR located on top of a Gaussian hill and secondly over an area covering most of the computational domain including of course the Gaussian hill. The experiment showed that as λ increases, the magnitude of the bias decreases. The extent of the bias however increases. An increase of φ, which corresponds to an increase in the slope of the hill, leads to an increase of both the the magnitude and the extent of the bias. The cases tested in this experiment revealed that placing the LiDAR $3L$ (L being the distance between the center of the hill and the point of inflection) away from the hill would prevent most of the bias. Such simple test cases can be used for preliminary LiDAR siting studies.

In chapter 4, the model was evaluated on the terrain of AAV, which is a complex terrain covered with sparse forest. This was done using Meteodyn WT 4.0 and experimental data. The effect of variable forest height as opposed to constant forest height was investigated. Since the distribution of forest is quite homogeneous on the site studied, no significant difference was observed between both approaches. A tuning process was used to determine the most appropriate value of drag coefficient to be used as no experimental data was available for AAV. Among 3 tested values, 0.0015 yielded best results. Results also showed the significant impact of both the location and the nature of the inlet boundary condition. An improvement of the results was noticed as the inlet moved over water due to the validity of the logarithmic profiles imposed. It was further validated by moving the inlet of case 320° completely in the water.

Finally, the flow model developed was used to evalute LiDAR bias. Results obtained with OpenFOAM were compared to Meteodyn WT 4.0, to the WAsP engineering script developed by Bingöl *et al.* (2008) and also to experimental data. Even though terrain was classified as moderately to highly complex, resulting LiDAR bias was very low and all softwares agreed well with each other. The LiDAR at AAV does not seem to be affected by topographic changes as far as eight times the radius of the scanned disc away from its location, but more by topography very close to the scanned volume.

In order to conclude, one could say that using LiDAR in complex terrain is not necessarily a bad idea depending on the location where the instrument is deployed. Proceeding to a terrain classification is certainly recommended to make a decision, but the distance to be considered must be adequately chosen, which is not an easy task. It depends for sure on the scanning height but also on the overall complexity of the terrain. Therefore, running a wind flow simulation reveals to be a relatively fast approach for selecting an appropriate location for the remote sensor. Implementing the generalization of Bingöl *et al.* (2008)'s method requires some work, but once completed can be used very easily as a post-processing tool. For quick estimation of LiDAR bias, simple cases such as those in chapter 3.3 can also be considered. Of course, performance depends on the flow model and thus requires a critical judgement of the results. Moreover, boundary conditions also proved to be a critical aspect which must be dealt with care. As long as LiDARs will be carying an averaging process to determine cartesian velocity components, emphasis should be put on identifying regions of flow non-homogeneity rather than precisely quantifying the bias.

Main contributions

- Based on the work already initiated by Masson and Sumner (2009) and Dalpé and Masson (2008), modifications were brought to both *simpleFoam* and *k*-ε, respectively the solver and the turbulence model in OpenFOAM used in this study, to account for the effects of sparse and height-variable forest over complex terrain. This led to the creation of two C++ utilities used to generate the necessary forest fields

  - *writeRoughness*: This tool reads a forest height map in .xyz format and creates the corresponding scalar field (called "Forest") for OpenFOAM. It also edits the roughness lenght field "z_0" to assign 0.05m values for locations where forest is present;
  - *CreateSandAlpha*: Based on the new scalar field "Forest", this tools create two other scalar fields called "S" and "alpha", which respectively tells to the solver if forest is present in a given cell and if this is the case, what is the corresponding leaf area density.

- A boundary condition first developed by Lussier-Clément (2012) was tested and validated for flow above and within forested areas. The particularity of this boundary condition is its complete independence of the forest displacement height.
- An innovative procedure was developed for meshing over complex terrain using only built-in utilies of OpenFOAM.
- A special investigation was conducted regarding the effect the inflow boundary over complex terrain. Results showed that placing the boundary in the water and prescribing a logarithmic wind profile considerably improved the results, which demonstrates the importance of this boundary on the solution.
- Bingöl *et al.* (2008)'s LiDAR modeling approach was generalized for large areas. It was tested on a theoretical case which demonstrated the impact and extent of the bias caused by the flow homogeneity assumption depending on the instrument location.

Future work

The main objective was about implementing method to evaluate LiDAR bias over complex terrain with the help of numerical tools, but that implies the development of a suitable atmospheric flow model capable of tackling this task. The latter is not a simple task for flow over complex and forested terrain. The model implemented successfully met these two requirements. Nonetheless, results open the door for further improvements both of the CFD model and the LiDAR evaluation approach. First, the quality and uncertainty of the results obtained with the implemented method could be assessed in a future study.

Of course adding the effects of thermal stability as well as the Coriolis force might improve the model, but more basic elements could be considered first especially concerning how forest is modeled. For instance, no information was available regarding the appropriate leaf area (α) distribution to be used at AAV and thus a blackspruce distribution from Amiro (1990) was used. Moreover, it was assumed that the distribution was valid for variable forest height, which might not be the case. Therefore, it would be interesting to investigate whether or not the leaf area distribution is dependent on the height of the vegetation. The investigation could also be carried with the drag coefficient (C_d), which was also assumed to be constant with forest height. Only one type of forest was used for the entire site, however there are probably more than one species of trees present in AAV. The assumption might have had an impact on the results so testing a heterogeneous forest distribution, using experimentally calculated leaf area distributions, could be envisaged.

Results also clearly demonstrated the impact of both the location of the inlet and what is actually prescribed at this boundary. Over complex terrain, unless experimental data is available, it is quite difficult to determine appropriate values to impose for velocity and turbulence. A common practice is to impose standard logarithmic profiles while placing the inlet sufficiently far from the points of interest to minimize the impact of erroneous profiles. This technique was used in this study combined with a constant horizontal mesh resolution. Using a grid expansion could allow placing the inlet boundary even further, thus incorporating more topography, while keeping the amount of cells to a reasonnable level. The latter is very similar to Meteodyn's approach which yielded acceptable results. Mixing a micro- and meso- scale models could also be envisaged.

Concerning the evaluation of LiDAR bias, the same experiment could be carried on a more complex site where the instrument would without a doubt yield incorrect measurements. It would be interesting to compare the real bias and the one reported by the CFD. Note that this was initiated but due to time constraints, it was put aside. Also, the distance between meteorological mast and remote sensor should be minimized as much as possible to avoid any uncertainty due to terrain variation. A higher resolution of topographic maps would certainly improve the modeling of the flow within the LiDAR sample volume.

Finally, it is believed that much efforts should be done towards developing an algorithm that uses radial velocities in order to recognize the shape of the flow within the scanned volume. The new version of LiDAR WINDCUBE uses a fifth laser beam pointing vertically in the center of the LiDAR scanned disc, which allows a direct acquisition of the vertical component of velocity. Four beams remain to determine the two other components. The FCR proposed by Leosphere seems to have adopted this path. Remote sensors offer many advantages as compared to traditionnal meteorological masts and thus deserve attention to make them work reliably over complex terrain.

APPENDIX I

WINDCUBE NUMERICAL EXAMPLE

The following calculations show how to obtain the components of velocity, the magnitude of horizontal velocity and the wind direction (u,v,w,U_h, θ) starting with WINDCUBE radial velocities and azimuth angles presented in table I-1.

a. First step is to calculate the horizontal velocity components u and v. In this case, ϕ_l (cone opening angle) is 27° and β (LiDAR offset w/r to true North) is 0°.

$$v = \frac{V_{rE} - V_{rW}}{2\sin\phi} = \frac{2.58 - -2.86}{2\sin(27 * \pi/180)} = 5.99 m/s$$

$$u = \frac{V_{rN} - V_{rS}}{2\sin\phi} = \frac{-3.24 - 1.51}{2\sin(27 * \pi/180)} = \text{-}5.23 m/s$$

b. Then, the magnitude of horizontal velocity and the wind direction can be calculated:

Notice that in an x-y frame, u becomes 5.23 and v becomes -5.99

$$\begin{aligned} V_h &= \sqrt{u^2 + v^2} = \sqrt{(5.99^2) + (\text{-}5.23)^2} = 7.95 m/s \\ \theta &= \text{Modulo}(360 + \arctan 2(u,v) + \beta, 360) \\ &= \text{Modulo}(360 + \arctan 2(5.23, \text{-}5.99) + 0, 360) \\ &= \text{Modulo}(360 + -48.875 + 0, 360) \\ &= \text{Modulo}(311.125, 360) \\ &= 311.125^\circ \end{aligned}$$

Tableau-A I-1 Example of WINDCUBE output

Timestamp	Azimuth Angle (degree)	Radial Velocity (V_r) (m/s)
00:00:00:000	180	1.51
00:00:01:500	270	-2.86
00:00:03:000	0	-3.24
00:00:04:500	90	2.58
00:00:06:000	180	1.72
...	...	...

c. Finally, the vertical component of velocity, *w*:

$$
\begin{aligned}
w &= \frac{(V_{rN} + V_{rE} + V_{rS} + V_{rW})}{4cos\phi_l} \\
w &= \frac{-3.24 + 2.58 + 1.51 + -2.86}{4cos(27)} \\
w &= -0.56m/s
\end{aligned}
$$

APPENDIX II

INVESTIGATION ABOUT THE CALCULATION OF RESIDUALS IN OPENFOAM

Judging of convergence with residuals monitoring is common practice in CFD. However, one must remain cautious since residuals might not always be an appropriate sign of convergence. At the beginning of this work, simple simulations were done to reproduce neutrally stratified ABL. After an important number of iterations, the residuals for most quantities were brought down several orders of magnitudes whereas other were staying much higher. And yet, computational results showed very good agreement with analytical results. After a bit of research, it was found that residuals in OpenFOAM are normalized by a factor called "normFactor". The norm factor is calculated as follow:

For a matrix:

$$Ax = b \tag{II.1}$$

$$Residuals = Ax - b \tag{II.2}$$

And:

$$x_{Ref} = gAverage(x) \tag{II.3}$$

$$w_A = Ax \tag{II.4}$$

$$p_A = Ax_{Ref} \tag{II.5}$$

$$matrix.small = 1e - 20 \tag{II.6}$$

Thus,

$$normFactor = gSum(cmptMag(w_A - p_A) + cmptMag(source - p_A)) + matrix.small \tag{II.7}$$

"gSum" is a sum and "gAverage" is the average. When used in parallel, operations are done over all processors. "matrix.small" is only a constant to avoid division by zero.

It was then believed that this norm factor would boost the normalised residuals if quantities being solved approach 0. To verify this, a simple experiment was conducted. A switch was turned on in OpenFOAM to allow the display of the normalisation factor. The results are presented below. The case is similar to the one in section 3.1

First, the problematic quantities are the perpendicular components (U_y and U_z) of velocity and the pressure (P). In this particular case, it is interesting to note that the expected solutions for these variables are uniform fields of value 0. The plot showing the norm factor reveals indeed that the normalisation factor becomes very small for these fields, thus dramatically increasing the normalised residuals. In fact, as depicted in figure II-3, the "absolute" residuals of these particular variables are in the same orders of magnitude as other quantities considered as converged when looking at normalised residuals.

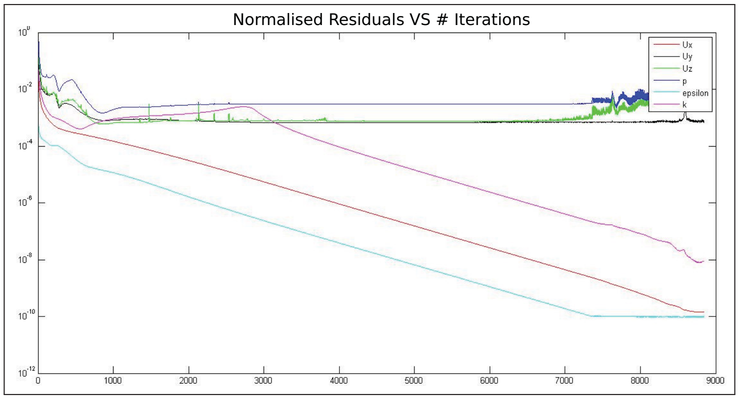


Figure-A II-1: NormFactor VS iterations

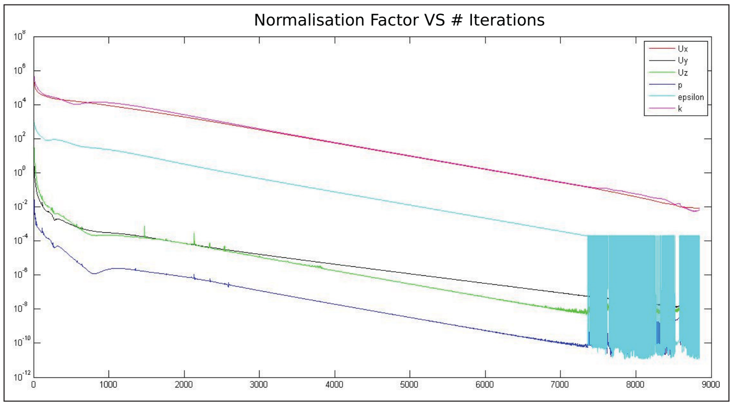


Figure-A II-2: NormFactor VS iterations

The conclusion is that normalised residuals can be misleading to evaluate convergence in some circumstances. Looking at the non-normalised residuals give a good idea about how well the system of equations is respected but on top of that, a CFD user should always take a critical look on the obtained solution.

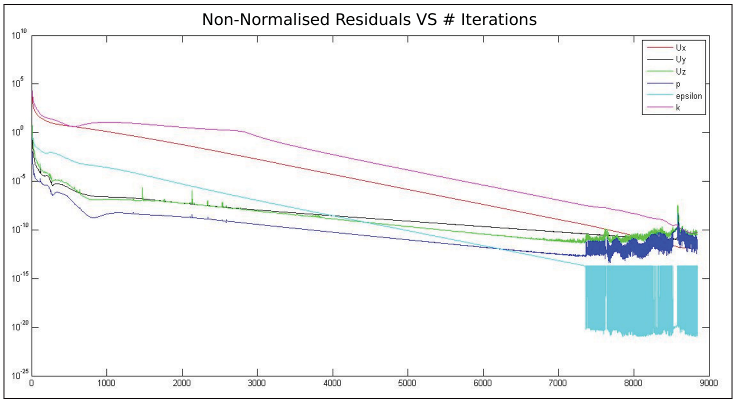


Figure-A II-3: Non normalised residuals VS iterations

APPENDIX III

MESHING COMPLEX TOPOGRAPHY WITH OPENFOAM

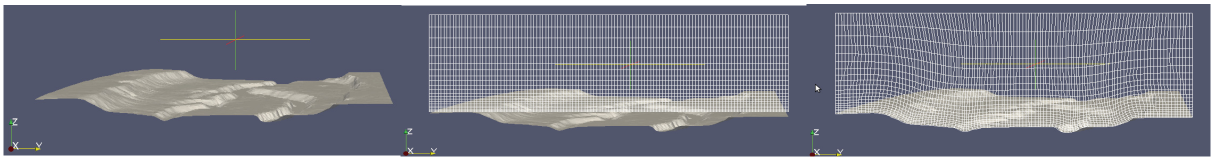
In this section, the reader will find more information on the tools available in OpenFOAM for meshing as well as the process developed for creating mesh over complex terrain used in this master's thesis. A small assessment of the method is also included here.

blockMesh
As the name implies, this tool allows the user to decompose the domain in one or many blocks by specifying the coordinates of the vertices of each block. The blocks can then be cut in multiple cells with the possibility of using grading (cell expansion). *blockMesh* is a very powerful tool but meshing complex 3D structures, such as a complex topography, becomes a real challenge. The difficulty arises in combining the topographic data with the required input format of *blockMesh*.

snappyHexMesh
Starting with a background mesh, usually an hex-structured mesh generated with *blockMesh*, snappyHexMesh allows the user to include a 3D object in Stereolithography (STL) format. It will automatically (with the help of snappyHexMeshdict dictionary) adapt the background mesh to the geometry by refining iteratively. Since the whole process (except for the generation of the background mesh) is done with no interaction from the user, the final result is hard to predict. Of course, it is possible to tune the *snappyHexMesh* dictionary until an acceptable mesh is obtained, but this can be a very long and fastidious process.

OpenFOAM puts several tutorials at the user's disposal to test the different utilities. Among them, one is called *snakeRiverCanyon*, which introduces the use of *moveDynamicMesh*. As the name suggests, this utility is used to move/deform a mesh during a simulation. In this tutorial however, it is used to deform a hex-background mesh onto a complex topography as depicted in the figure III-1. The final mesh is then used for the simulation. At first glance, it seems promising, but it presents some shortcomings

- Limited control over first few cells height;
- Aspect ratios of cells in background mesh must be close to one in order to avoir collapsing cells;



Topography (STL format)

Background Mesh + Topography

Background Mesh fitted to topography

Figure-A III-1: Mesh generation with *moveDynamicMesh*

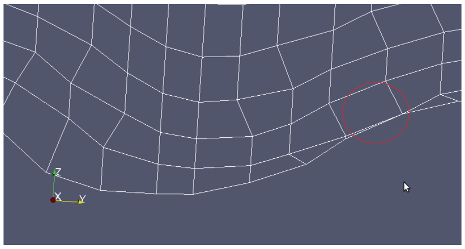


Figure-A III-2: Collapsing cell
due to high aspect ratio

- *moveDynamicMesh* may take several iterations to reasonnably fit complex topographies.

For these reasons, a more appropriate way of meshing complex topographies was required. The desired utility or meshing procedure that we were looking for had to include some important features

- A control over the first cell height;
- Capable of generating large aspect ratio cells (∼50) for large domain simulations;
- Must pass *checkMesh*[1].

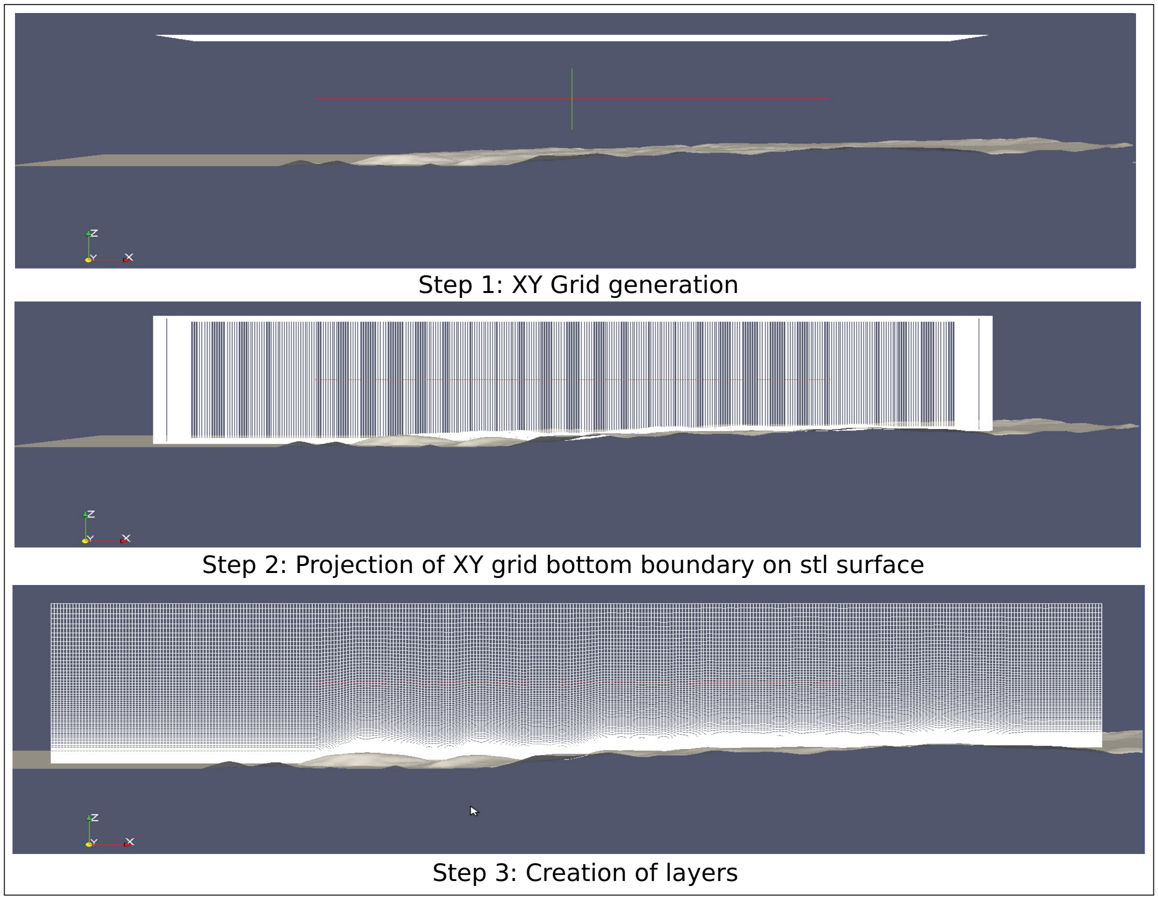
The procedure developed relies on three utilities available in OpenFOAM: *blockMesh*, *moveMesh* and *refineWallLayer*. Another tool available online for free called *meshSpace* was used. Starting with the ".xyz" topographic map described in section 2.2.1.3, it is converted to STL to work with OpenFOAM. A grid with desired X and Y resolution is defined above the STL surface. This grid is then projected onto the topographic map resulting in very elongated rectangular cells. These long cells are then cut following a certain vertical expansion. Here are the steps for mesh generation using the developed method

I. First step is to create an XY grid located above the topography using *blockMesh*. The distance between this grid and the topography is important as it will correspond to the domain height;

II. Secondly, *moveMesh* is used to project the bottom boundary of the grid created in step 1 on the topography. This is done by specifying the direction, displacement and the patch to be moved in a file called *pointDisplacement* located in the case directory. This results in a mesh composed of many elongated cells as shown in the figure III-3;

III. Finally, *refineWallLayer* is used to cut these large cells. The cuts are done respecting a certain vertical cell height expansion. When executed, *refineWallLayer* makes only

[1] *checkMesh* is a utility in OpenFOAM that verifies the quality of given mesh by computing several parameters such skewness,orthogonality,min/max cell volume etc. For more information, please refer to OpenFOAM manual.



Step 1: XY Grid generation

Step 2: Projection of XY grid bottom boundary on stl surface

Step 3: Creation of layers

Figure-A III-3: Developed meshing procedure

one cut, so it was included in a script that was designed to make it run for a certain number of times, depending how many vertical cells are needed. As input parameters, *refineWallLayer* requires <patchName> and <edgeWeight>. The first input tells the utility to cut the cells in the normal direction relative to the given patch. The second input corresponds the location where to make the cut and it is expressed as "distance from patch divided by the total height of the cell". The <edgeWeight> must calculated before each cut in the script mentionned earlier.

In order to determine the expansion factor based on the desired first cell height and total height of the domain, an open source tool called *meshSpace* was used. It uses the total height and any two of the following:number of cells, total expansion ratio, cell-to-cell expansion ratio, start cell size and end cell size. Then, it calculates the remaining quantities. In our case, the first cell height, the final cell height and the total domain height were given as inputs.

The resulting mesh easily passes the *checkMesh* utility. It was also easy to build meshes with large cells aspect ratios (>50). Finally, the method allows a relatively good control over the first cell height. The first cell height dimension cannot be more precise than the ratio of the largest

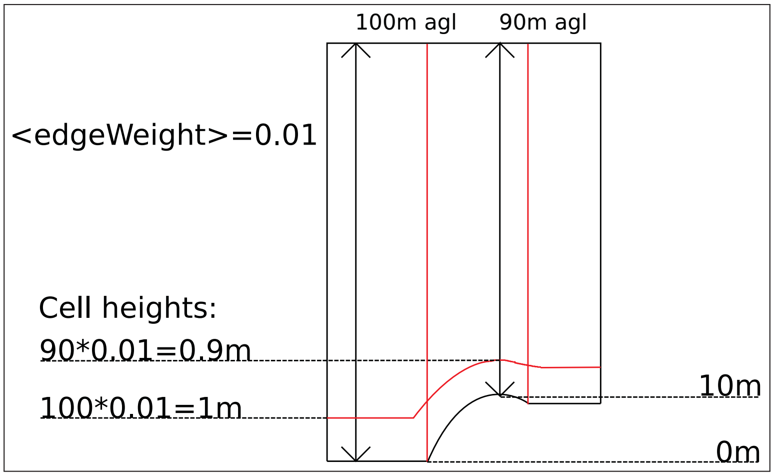


Figure-A III-4: Variation of first cell height

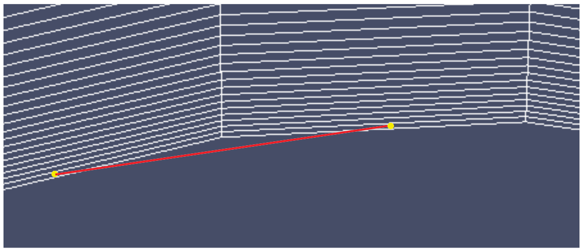


Figure-A III-5: Examples of mesh non-orthogonality near bottom boundary

obstacle in the domain over the total domain height. In order to determine the <edgeWeight> parameter of *refineWallLayer*, a base point on the patch must be chosen. In our case, this point was always chosen to be the lowest one in the domain. It can be better understood when looking at figure III-4.

In order to improve this aspect, an option would be to modify the <edgeWeight> parameter depending on the height of the topography during each layer addition. It might be feasible by editing the code of the utility but this was out of the scope of this work. For all the simulations run within the context of this project, the max error was of the order of 10%. Concretely, we were targeting a first cell height of one meter everywhere, but at some locations in the domain the actual cell height could reach 0.9m, which is not critical.

Another downside of this method concerns the respect of mesh orthogonality. Even though no errors were reported by *checkMesh*, it could create problems on more complex topographies. As opposed to *moveDynamicMesh*, the edges normal to the bottom patch are always parallel to the z-axis and do not necessarily intersect the STL surface at an angle of 90°. This issue can be observed in figure III-5.

REFERENCES

Alinot, C. and C. Masson. 2005. “k- model for the atmospheric boundary layer under various thermal stratifications”. *Transactions of the ASME. Journal of Solar Energy Engineering*, vol. 127, n° 4, p. 438 - 443.

Amiro, B. D. 1990. “Comparison of turbulence statistics within three boreal forest canopies”. *Boundary-Layer Meteorology*, vol. 51, p. 99-121.

Arranz, Paula Gómez. 2011. *Deliverable d6.6.2 measurements in complex terrain using a lidar*. Technical report. Project UpWind.

BAPE, (Bureau d’audiences publics sur l’environnement). 2005. “Projet de parc éolien à L’Anse-à-Valleau”. <http://www.bape.gouv.qc.ca/sections/mandats/eole-valleau-sables/documents/liste_documents_valleau.htm>. Consulted May 21st 2013.

Bingöl, Ferhat, J. Mann, and D. Foussekis. 2008. *Deliverable d6.6.1 modeling conically scanning lidar error in complex terrain with wasp engineering*. Technical report. Project UpWind.

Boquet, M., R. Parmentier, JP Cariou, L. Sauvage, and A. Alergel. 2010. “Innovative Solutions for Pulsed Wind Lidar Accuracy in Complex Terrain (ISARS2010)”. In *June 2010, ISARS Conference*. (LEOSPHERE, 76 Rue Monceau 75008 Paris, France 2010).

Boquet, M., R. Parmentier, JP Cariou, L. Sauvage, and A. Alergel. 2011. “Sensitivity of the CFD based LIDAR Correction”. In *2011, EWEA Conference*. (METEODYN, 14 bd Churchill 44100 Nantes, France 2011).

Boquet, Matthieu. 2012. “Direct Lidar Measurement in Complex Terrain”. Online. <http://www.leosphere.com/file/2012_ewea_po79_direct_measurement_in_complex_terrain.pdf>.

Bradley, S. 2008. “Wind speed errors for LIDARs and SODARs in complex terrain”. (UK 2008), p. 012061 (7 pp.) -.

Briggs, Katy, Tony Rogers, and Mark Young. 2011. “Methods for Correcting Lidar Measurements in Complex Flow Conditions”. In *March 2011, EWEA Conference*. (DNV Renewables (USA) Inc. 1809 7 Avenue, Suite 900 Seattle, WA 98101 USA 2011).

Brodeur, Philippe and Christian Masson. 2008. “Numerical Site Calibration Over Complex Terrain”. p. 12.

Cariou, Jean-Pierre and Matthieu Boquet. 2010. *Deliverable d6.1.1 leosphere pulsed lidar principles contribution to upwind wp6 on remote sensing devices*. Technical report. Project UpWind.

Celik, Ismail B., Urmila Ghia, Patrick J. Roache, Christopher J. Freitas, Hugh Coleman, and Peter E. Raad. 2008. “Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications”. *Journal of Fluids Engineering*, 130(7):078001.

Dalpé, B. and C. Masson. 2008. “Numerical study of fully developed turbulent flow within and above a dense forest”. *Wind Energy*, vol. 11, n° 5, p. 503 - 15.

Dalpé, B. and C. Masson. 2009. “Numerical simulation of wind flow near a forest edge”. *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 97, n° 5–6, p. 228 - 241.

Dellwik, Ebba, Lars Landberg, and Niels Otto Jensen. 2006. “WAsP in the Forest”. *Wind Energy*, vol. 9, n° 3, p. 211–218.

Dorval, Jonathan. 2011. “Évaluation des méthodes d’estimation du productible des parcs éoliens”. Master’s thesis, École de Technologie Supérieure.

Emeis, Stefan, 2006. *Surface-Based Remote Sensing of the Atmospheric Boundary Layer*, p. 174. Springer Netherlands.

Ferry, M. 2000. “The MIGAL solver”.

Foussekis, Dimitri. 2011. *Deliverable d6.6.1b remote sensing – cres activities: Measurements and inter-comparisons in complex terrain*. Technical report. Project UpWind.

Harris, M., I. Locker, N. Douglas, R. Girault, A. Abiven, and O. Brady. 2010. “Validated adjustment of remote sensing bias in complex terrain using CFD”. In *EWEC 2010*. (The Old Barns,Fairoaks Farm, Hollybush,Nr. Ledbury HR8 1EU,England, U.K. 2010).

Hill, Chris. 2010. *Deliverable d6.1.1-b remote sensing (upwind wp6) qinetiq lidar measurement report*. Technical report. Project UpWind.

IEC 61400-12-1. 2005. “Wind turbines - Part 12-1: Power performance measurements of electricity producing wind turbines”. Online. <http://www.iec.ch/>.

IEC 61400-12-2. 2013. “Wind turbines - Part 12-2: Power performance measurements of electricity producing wind turbines based on nacelle anemometry”. Online. <http://www.iec.ch/>.

Jasak, Hrvoje. 1996. “Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows”. PhD thesis, University of London.

Jeannotte, E., D. Faghani, M. Pelletier, M. Boquet, R. Dexter, B. Boucher, E. Osler, and C. Masson. 2011. “LiDAR Measurement in Complex Terrain under Harsch Winter Conditions”. In *2011, CanWEA Conference*.

Jones, W.P and B.E Launder. 1972. “The prediction of laminarization with a two-equation model of turbulence”. *International Journal of Heat and Mass Transfer*, vol. 15, n° 2, p. 301 - 314.

Kaimal, J. C. and John J. Finnigan, 1994. *Atmospheric Boundary Layer Flows: Their Structure and Measurement*, p. 290. Oxford University Press, 200 Madison Avenue, New-York, New-York 10016.

Katul, Gabriel G., Larry Mahrt, Davide Poggi, and Christophe Sanz. 2004. “ONE- and TWO-Equation Models for Canopy Turbulence”. *Boundary-Layer Meteorology*, vol. 113, p. 81-109.

Lindelöw-Marsden, Petter. 2009. *Deliverable d6.1.3 uncertainties in wind assessment with lidar*. Technical report. Project UpWind.

Lussier-Clément, Nicolas. 2012. “Dévelopement d’outil, par simulation numérique, d’aide à la décision pour le positionnement des éoliennes en terrain complexe”. Master’s thesis, École de technologie supérieure.

Mann, J., J-P Cariou, M.S. Courtney, R. Parmentier, T. Mikkelsen, R. Wagner, P. Lindelöw, M. Sjöholm, and K. Enevoldsen. 2009. “Comparison of 3D turbulence measurements using three staring wind lidars and a sonic anemometer”. *Meteorologische Zeitschrift*, vol. 18, p. 135-140.

Manwell, J.F., J.G. McGowan, and A.L. Rogers, 2009. *WIND ENERGY EXPLAINED Theory, Design and Application*. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, ed. Wiley.

Marcolla, B., A. Pitacco, and A. Cescatti. 2003. “Canopy Architecture and Turbulence Structure in a Coniferous Forest”. *Boundary-Layer Meteorology*, vol. 108, p. 39-59.

Masson, C. and J. Sumner. 2009. “On the use of the k-epsilon turbulence model for atmospheric boundary layer flow and wind turbine wakes”. In *proceedings of 2009 International Conference on Computational Methods for Energy Engineering and Environment: ICCM3E*. (Sousse, Tunisia 2009).

NRC, (Natural Resources Canada). 2011. “Trees, insects and diseases of Canada’s forests”. <http://tidcf.nrcan.gc.ca/trees/factsheet/39>. Consulted March 21st 2013.

Pelletier, Marc, Dariush Faghani, Matthieu Boquet, Reesa Dexter, Bruno Boucher, Christian Masson, and L. Landberg. 2011. *Lidar validation in complex terrain*. Technical report. ETS (École de technologie supérieure), GL Garrad Hassan, Leosphere SAS, Technocentre Éolien, NRG Systems.

Richards, P.J. and R.P. Hoxey. 1993. “Appropriate boundary conditions for computational wind engineering models using the k- turbulence model”. (Netherlands 1993), p. 145 - 153.

Stull, Roland B., 1988. *An Introduction to Boundary Layer Meteorology*, p. 669. Kluwer Academic Publishers, 3300 Dordrecht, The Netherlands.

Sumner, Jonathon and Christian Masson. 2010. “k - simulations of the neutral ABL: Achieving horizontal homogeneity on practical grids”. (Orlando, FL, United states 2010).

Versteeg, H K and W Malalasekera, 1995. *An introduction to Computional Fluid Dynamics*, p. 257. Longman Scientific and Technical, Longman House, Burnt Mill, Harlow Essex CM20 2JE, England.

Wagner, Rozenn. 2010. *Deliverable d6.15.1 accounting for the speed shear in wind turbine power performance measurement*. Technical report. Project UpWind.

Werner, Christian. 2005. Doppler wind lidar. Weitkamp, C., editor, *Lidar*, volume 102 of *Springer Series in Optical Sciences*, p. 325-354. Springer Berlin / Heidelberg. ISBN 978-0-387-40075-4. 10.1007/0-387-25101-4-12.