isep Instituto Superior de Engenharia do Porto

Development of procedures for the simulation of atmospheric flows over complex terrain, using OpenFOAM

João Manuel da Silva Azevedo

Thesis submitted for the Master degree in
Mechanical Engineering: Energy and Fluid Dynamics

School of Engineering | Polytechnic of Porto
Mechanical Engineering Department

November, 2013

Relatório da Unidade Curricular Dissertação/Projecto/Estágio do 2º ano do Mestrado em Engenharia Mecânica – Área de Especialização em Energia.

Candidato:
João Manuel da Silva Azevedo, Nº 1050596, 1050596@isep.ipp.pt

Orientação Científica:
Prof. Fernando Aristides da Silva Ferreira de Castro, fac@isep.ipp.pt
Co-orientação Científica:
Prof. José Carlos Pereira Lopes da Costa, loc@isep.ipp.pt

Mestrado em Engenharia Mecânica
Área de Especialização em Energia
Departamento de Engenharia Mecânica

Instituto Superior de Engenharia do Porto

Novembro de 2013

"The greatest good you can do for another
is not just to share your riches,
but to reveal to him his own."

Benjamin Disraeli (1804-1881)

Agradecimentos

Aos meus pais, por um dia me terem convencido a voltar a estudar.

Ao ISEP, por tudo o que me ensinou e pelos amigos que me deu a conhecer.

Ao Prof. Fernando Castro, meu orientador, pela confiança depositada em mim e pelo apoio permanente ao longo de todo este trabalho com o seu enorme conhecimento e dedicação.

Ao Prof. Lopes da Costa, meu co-orientador, pelas valiosas sugestões e comentários.

À Ana, por todos os dias estar presente na minha vida.

Abstract

The main purpose of this work was the development of procedures for the simulation of atmospheric flows over complex terrain, using OpenFOAM. For this aim, tools and procedures were developed apart from this code for the preprocessing and data extraction, which were thereafter applied in the simulation of a real case.

For the generation of the computational domain, a systematic method able to translate the terrain elevation model to a native OpenFOAM format (*blockMeshDict*) was developed. The outcome was a structured mesh, in which the user has the ability to define the number of control volumes and its dimensions. With this procedure, the difficulties of case set up and the high computation computational effort reported in literature associated to the use of *snappyHexMesh*, the OpenFOAM resource explored until then for the accomplishment of this task, were considered to be overwhelmed.

Developed procedures for the generation of boundary conditions allowed for the automatic creation of idealized inlet vertical profiles, definition of wall functions boundary conditions and the calculation of internal field first guesses for the iterative solution process, having as input experimental data supplied by the user. The applicability of the generated boundary conditions was limited to the simulation of turbulent, steady-state, incompressible and neutrally stratified atmospheric flows, always recurring to RaNS (*Reynolds-averaged Navier-Stokes*) models.

For the modelling of terrain roughness, the developed procedure allowed to the user the definition of idealized conditions, like an uniform aerodynamic roughness length or making its value variable as a function of topography characteristic values, or the using of real site data, and it was complemented by the development of techniques for the visual inspection of generated roughness maps. The absence and the non inclusion of a forest canopy model limited the applicability of this procedure to low aerodynamic roughness lengths.

The developed tools and procedures were then applied in the simulation of a neutrally stratified atmospheric flow over the *Askervein hill*. In the performed simulations was evaluated the solution sensibility to different convection schemes, mesh dimensions, ground roughness and formulations of the $k-\varepsilon$ and $k-\omega$ models. When compared to experimental data, calculated values showed a good agreement of speed-up in hill top and lee side, with a relative error of less than 10% at a height of 10 m above ground level. Turbulent kinetic energy was considered to be well simulated in the hill windward and hill top, and grossly predicted in the lee side, where a zone of flow separation was also identified. Despite the need of more work to evaluate the importance of the downstream recirculation zone in the quality of gathered results, the agreement between the calculated and experimental values and the OpenFOAM sensibility to the tested parameters were considered to be generally in line with the simulations presented in the reviewed bibliographic sources.

Keywords: Computational Fluid Dynamics, CFD, Simulation of Atmospheric Flows Over Complex Terrain, OpenFOAM.

Resumo

O presente trabalho teve como principal objetivo o desenvolvimento de procedimentos para a simulação de escoamentos atmosféricos sobre topografia complexa, usando OpenFOAM. Para este fim foram desenvolvidos procedimentos e ferramentas externas a este código, adequados ao pré-processamento e extração de dados, os quais foram posteriormente aplicados na simulação de um caso prático.

Para a geração do domínio computacional foi criado um método sistemático capaz de traduzir o modelo de elevação da topografia em estudo para um formato nativo do OpenFOAM (*blockMeshDict*). O resultado foi a obtenção de uma malha estruturada, na qual o utilizador tem a possibilidade de definir o número e dimensões dos volumes de controlo. Com este procedimento foram consideradas como ultrapassadas as dificuldades de configuração e a necessidade de elevados recursos computacionais reportadas na literatura associadas à utilização da ferramenta *snappyHexMesh*, o recurso do OpenFOAM até aqui explorado para a realização desta tarefa.

Os procedimentos desenvolvidos para geração de condições de fronteira permitiram a automatização da criação de perfis verticais de entrada idealizados, a definição das condições de parede e a determinação de valores para arranque do processo iterativo de resolução, tendo como dados de entrada valores experimentais fornecidos pelo utilizador. A aplicabilidade das condições de fronteira geradas foi limitada à simulação de escoamentos turbulentos, estacionários, incompressíveis e em regime neutralmente estratificado, sempre com o recurso a modelos RaNS (*Reynolds-averaged Navier-Stokes*).

Para modelação da rugosidade de superfície, o procedimento desenvolvido permitiu a definição por parte do utilizador de condições idealizadas, como a definição de um valor de rugosidade uniforme ou a variação deste em função de parâmetros característicos da topografia, ou a utilização de dados reais de superfície, e foi complementado com o desenvolvimento de técnicas para a inspecção visual dos mapas de rugosidade gerados. A não abordagem neste trabalho da possibilidade de inclusão de modelos de floresta limitou a validade deste procedimento a valores de rugosidade considerados como reduzidos.

Os procedimentos e ferramentas desenvolvidos foram aplicados na simulação de um escoamento atmosférico neutralmente estratificado sobre o *Monte de Askervein*. Nas diversas simulações realizadas foi avaliada a sensibilidade da solução a diferentes esquemas convectivos, tamanhos de malha, condições de rugosidade e formulações dos modelos de turbulência $k - \varepsilon$ e $k - \omega$. Quando comparados com dados experimentais, os valores obtidos demonstraram uma boa concordância do *speed-up* no topo e a jusante da topografia, obtendo-se erros inferiores a 10% a uma altitude de 10 m acima do solo. A energia cinética turbulenta foi considerada como bem simulada a montante e no topo da topografia, e prevista de forma grosseira a jusante, local onde foi também identificada uma zona de separação do escoamento. Não obstante a necessidade de mais trabalho para aferir a importânica da presença de uma zona de recirculação a jusante da topografia na qualidade dos resultados, em termos gerais, a concordância da solução obtida com os valores experimentais e a sensibilidade do OpenFOAM aos diversos parâmetros testados foram consideradas como estando em linha com outras simulações presentes na literatura.

Palavras-chave : Mecânica dos Fluídos Computacional, Simulação de Escoamentos Atmosféricos sobre Topografia Complexa, OpenFOAM.

Contents

List of Tables

List of Figures

Symbols and abbreviations

Roman Characters

A_0, A_s	coefficients of the *Realisable* $k-\varepsilon$ model
a_1	constant of the $k-\omega$ model
C_μ, $C_{\varepsilon1}$, $C_{\varepsilon2}$	constants of the $k-\varepsilon$ model
c_p	specific heat at constant pressure
d	zero plane displacement height
F_1, F_2	blending function of the SST $k-\omega$ model
f_j	face j of a computational domain patch
g	modulus of gravity acceleration vector
h	height of canopy elements
k	turbulent kinetic energy
k^*	non-dimensional turbulent kinetic energy
L	length scale
nx, ny, nz	number of coordinates in three orthogonal directions of the computational domain
Nx, Ny, Nz	number of control volumes in three orthogonal directions of the computational domain
P, p	pressure
P_0	pressure reference state
P_ε	production of turbulent kinetic energy dissipation rate
P_k	production of turbulent kinetic energy
P_ω	production of turbulent kinetic energy dissipation frequency
R	ratio of dimensions between the first and last control volumes measured in one orthogonal direction; universal gas constant
r	ratio of dimensions between one control volume and its neighbour measured in one orthogonal direction
Re	Reynolds number
Ro	Rossby number
S	generic source term
S_{ij}	tensor of strain rates
$\Delta\mathrm{S}$	speed-up ratio
T	temperature
t	time
Δt	time interval
U_c, u	characteristic velocity; streamwise velocity
U, u_i	velocity field

U_0	reference speed
$-\overline{u_i'u_j'}$	Reynolds stress
u_*, v_*, w_*	friction velocity
u_i	i component of velocity field
U^*	coefficient of the *Realisable* $k-\varepsilon$ model
ΔU_R	velocity perturbation by ground roughness changes
V	modulus of velocity vector; y component of velocity vector
v	y component of velocity vector
V_h	modulus of horizontal components of velocity vector
V_{ref}	reference speed
W, w	z component of velocity vector
x	component of an orthogonal coordinates system
x^i	i component of a Cartesian coordinates system
x_i	x component of coordinate i
$x_{j,n}$	n x component of face j
y	component of an orthogonal coordinates system
y^+	local Reynolds number
y_i	y component of coordinate i
$y_{j,n}$	n y component of face j
y_p	distance from a wall and adjacent cell centre
z	component of an orthogonal coordinates system
z_0	aerodynamic roughness length
z_i	height of a boundary layer
z_p	distance from a wall and adjacent cell centre
z^+	local Reynolds number
Δz	height of control volume adjacent to ground surface

Greek characters

α,	coefficient of the $k-\omega$ model
β	coefficient of the *RNG* $k-\varepsilon$ model
β, β^*	coefficients of the $k-\omega$ model
β_1, β_2	coefficients of the *SST* $k-\omega$ model
δ	height of a boundary layer
δ_{ij}	Kronecker tensor
ε	rate of destruction of turbulent kinetic energy
η, η_0	coefficients of the *RNG* $k-\varepsilon$ model
κ	von Kármám constant
γ_1, γ_2	constants of the *SST* $k-\omega$ model
Γ_d	adiabatic lapse rate
μ	dynamic viscosity
μ_t	turbulent dynamic viscosity
μ_l	laminar turbulent dynamic viscosity
ν	kinematic viscosity
ν_t	turbulent kinematic viscosity

ν_l	laminar kinematic viscosity
ω	dissipation frequency of turbulent kinetic energy
Ω	angular velocity of earth rotation
ρ	density
σ	standard deviation, coefficient of the $k-\omega$ model
σ_k, σ_ε	coefficients of the $k-\varepsilon$ model
σ^*	coefficients of the $k-\omega$ model
σ_{k1}, σ_{k2}, $\sigma_{\omega1}$, $\sigma_{\omega2}$	coefficients of the $SST\ k-\omega$ model
σ_{ij}	tensor of Reynolds stresses
ϕ	generic scalar; function of the $SST\ k-\omega$ model
ϕ_1, ϕ_2	coefficients set of the $SST\ k-\omega$ model
Φ,	latitude
τ	shear stress
τ_{ij}	tensor of viscous stresses
θ	potential temperature

Operators

$\overline{(\ )}$	average
$(\ )'$	perturbation
$\langle\ \rangle$	filter
∂/∂	partial derivative
d/d	total derivative
Δ	difference
∇	gradient
∇^2	Laplacian
$\mathcal{O}$	order of magnitude

Acronyms

AGL	*Above Ground Level*
ASL	*Above Sea Level*
ASCII	*American Standard Code for Information Interchange*
BST	*British Summer Time*
CBL	*Convective Boundary layer*
CFD	*Computational Fluid Dynamics*
CPU	*Central Processing Unit*
CSV	*Comma Separated Values*
DNS	*Direct Numerical Simulation*
GAMG	*Generalised Geometric-Algebraic Multi-Grid*
LES	*Large-eddy Simulation*
Max, max	maximum
Min, min	minimum
NBL	*Nocturnal Boundary Layer*

PBiCG	*Preconditioned Bi-Conjugate Gradient*
QUICK	*Quadratic Upwind Interpolation for Convective Kinematics*
QUICKV	*Vectorial form of QUICK*
RaNS	*Reynolds-averaged Navier-Stokes*
RNG	*Renormalization Group*
SBL	*Stable Boundary layer*
SRTM	*Shuttle Radar Topography Mission*
SST	*Shear Stress Transport*
STL	*STereoLithography*
TKE	*Turbulent Kinetic Energy*
URaNS	*Unsteady Reynolds-averaged Navier-Stokes*
UTM	*Universal Transverse Mercator*
VTK	*Visualization Toolkit*

Chapter 1

Introduction

1.1 Motivation

The aim of this work was the development of procedures for the simulation of atmospheric flows over complex terrain, using OpenFOAM v.2.1.1, thereby contributing for the use of open source CFD (*Computational Fluid Dynamics*) codes in this specific field of engineering.

On its base were the OpenFOAM lessons in the Computational Fluid Dynamics course lectured by the Mechanical Engineering Department of the School of Engineering of Polytechnic of Porto (DEM-ISEP), where a group of researchers develop Windie, a state of the art code for wind resource assessment.

The goals were the development of tools and procedures for the case preprocessing, like the generation of computational mesh, definition of boundary conditions, roughness mapping and set up of turbulence modelling coefficients, and data extraction of simulation results in a wind resource assessment related form. All of these tools should run apart from OpenFOAM, which by its turn should be tested without code modifications.

Being this work intended also as a first step for code validation, developed tools and procedures would be applied on the simulation of real cases in both neutrally and non-neutrally stratified states. The gathered results would then be the base for the evaluation of the merit of the developed procedures and the OpenFOAM sensibility to convection schemes, grid dimensions, ground roughness conditions and turbulence modelling.

In an advanced stage of its development, the amount of work required for the simulation of non-neutrally stratified flows and for the presentation and discussion of the respective results was considered unable to perform in the available schedule, therefore limiting the applicability of developed tools and procedures to the simulation of neutrally stratified atmospheric flows.

1.2 Thesis content

This thesis is divided in five chapters (including the present one) plus two appendixes.

In chapter 2 is performed a bibliographic review about the atmospheric boundary layer physics and turbulence modelling, and a brief introduction to OpenFOAM case structure.

Chapter 3 presents the tools and procedures developed during this work for the preprocessing and data extraction of atmospheric flows simulations over complex terrain, using OpenFOAM.

In chapter 4 is performed the simulation of a neutrally stratified atmospheric flow over complex terrain recurring to the developed tools and procedures, and the presentation and discussion of gathered results.

Chapter 5 presents the conclusions and suggestions for future work.

Appendix A shows the OpenFOAM case folders structure and its contents on the integration of developed tools and procedures. In appendix B are presented the instructions for preprocessing, processing and data extraction of a typical case.

1.3 State of the art

The study of flow in the ABL (*Atmospheric Boundary Layer*) has been of great interest in engineering applications. Wind resource assessment for turbine sitting, study of pollutant dispersion and forecast of wind loads over buildings have been some of the fields that promoted a continuous development of ABL flow simulation tools over the past few decades.

Although the calculation of ABL flow over complex terrain has always been a difficult task to perform, the costs associated to field experimental campaigns and the difficulty to reproduce all the physics in wind tunnel experiments [Paiva et al. (2009)] make these alternatives infeasible every time real problems need to be solved.

Extensive bibliographical sources attribute to Jackson and Hunt (1975) the first detailed study concerning ABL flows over isolated low-slope hills, where they developed an analytical linear theory for this class of flows. Taylor (1977), Walmsley et al. (1982) and Raithby et al. (1987) among others, performed subsequent numerical studies with recurring modelling improvements, where the latter, using three-dimensional finite volume techniques, is commonly mentioned as the first CFD (*Computational Fluid Dynamics*) study about ABL flows over complex terrain. The experimental field studies of Taylor and Teunissen (1985), Mason and King (1985) and Salmon et al. (1988), and the wind tunnel studies of Teunissen et al. (1987), have been used until nowadays for model validation.

According to Corbett et al. (2008), the inability of the models developed until then to predicted some physical aspects of the flow, like speed reduction and flow separation on the lee side of hills, motivated the development of more sophisticated techniques with non linear capabilities, often recurring to RaNS (*Reynolds-averaged Navier-Stokes*) models. One of those models was from Castro (1997), that using a terrain-following coordinate system with a two-equation $k-\varepsilon$ turbulence model, greatly improved the prediction of the speed-up on the lee side of the hill, identifying also a zone of flow separation. The benefits and limitations of $k-\varepsilon$ based turbulence models were also studied by Kim and Patel (2000), who tested different turbulence models, Castro et al. (2003), who explored the importance of numerical grid spatial discretization and roughness length parameter, and Kasmi and Masson (2010), who proposed a modified $RNG\ k-\varepsilon$ model.

With the advent of increasingly powerful computational resources, LES (*Large-eddy Simulation*) techniques have become more common in the ABL flows simulation. Lopes et al. (2007) performed large-eddy simulations of a neutrally stratified flow, achieving good agreement for the mean-velocity field and a better prediction of turbulent kinetic energy when compared to earlier RaNS simulations.

The wind resource assessment for turbine sitting is nowadays one of the main applications for ABL flow simulation codes. WAsP, MS-Micro/3, WindSim and Windie are four state of art codes used in the wind energy industry. While the models they incorporate vary from linear models in WAsP and MS-Micro/3, to non linear URaNS (*Unsteady RaNS*) models in WindSim and Windie, their capabilities range from neutrally to fully thermally induced flows simulation, roughness and/or canopy modelling and mesoscale meteorological data coupling.

In the same manner as most of the codes used nowadays for wind resource assessment, open source CFD codes have had their foundations in colleges scientific community. OpenFOAM initially developed at the Imperial College of London and SU^2 from Stanford University are two examples of open source numerical solvers with embedded multi physics models being developed and optimized for CFD applications.

The use of open source CFD codes for atmospheric flows simulation over complex terrain its not yet a current practice in the wind energy industry, being also a matter with very few available reports in literature, some of which will be further addressed.

Chapter 2

Theoretical background

2.1 Atmospheric boundary layer

2.1.1 Its structure

Atmospheric boundary layer is not only the part of the planet earth where we live, but also where lie most of our valuable natural resources.

In the origin of its driving forces are large-scale atmospheric motions set up in response to spatial variations of air pressure and Coriolis force, leading to what is called as *geostrophic winds* [Arya (2001)].

One definition of ABL is given by Jacobson (2005), that defines boundary layer *as the region of the atmosphere between the Earth's surface and a 500 to 3000 meter height that is influenced substantially by energy and moisture from the surface*. According to Stull (1988), its depth is variable in space and time, and frictional drag, evaporation, heat transfer, pollutant emission and terrain orography are the main parameters that influence its structure, wherein in the absence of which, winds would be purely geostrophic.

In a matter of stability, Stull (1988) classifies the ABL as a CBL (*Convective Boundary Layer*) or a stable and typically NBL (*Nocturnal Boundary Layer*), acting both, alternately, on a daily basis (see Figure 2.1). Whereas CBL is usually associated to day time, where solar radiation heat gains at the surface create an unstable and intense mixing layer, in the NBL statically stable air tends to suppress turbulence. Throughout the CBL winds are subgeostrophic (meaning that its mean speed is lower then geostrophic winds) with constant speed and direction along almost all of its height. Their speed starts to decrease near the ground due to a no slip condition, yielding a nearly logarithmic speed profile. Although winds at night become lighter near the ground surface, for typically short periods and at an altitude of approximately 200 meters, their speed in the NBL can become supergeostrophic, inducing wind shear that tends to generate turbulence, a phenomenon which is also called as *nocturnal jets*.

Two more zones of the ABL are considered important to mention: (i) a stable layer at the top of the ABL, also called as *Entrainment Zone*, that acts as a lid, restraining the

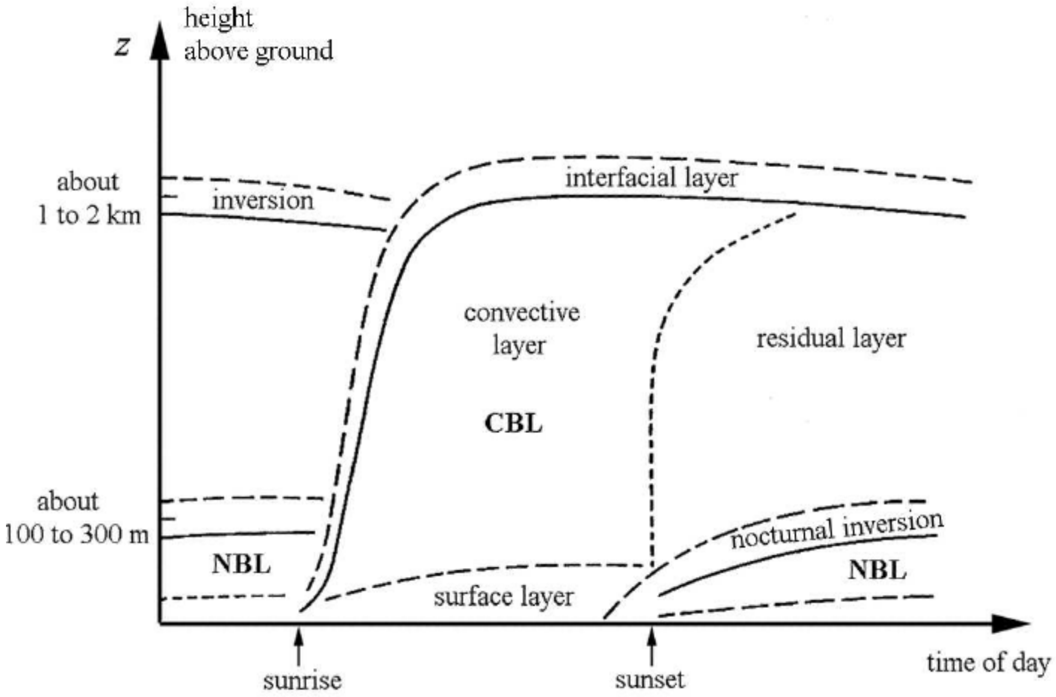


Figure 2.1: Convective Boundary Layer and Nocturnal Boundary Layer on a daily cycle. Unidentified source.

domain of turbulence and (ii) a region in the lower five to ten percent of the ABL, called as *Surface Layer*, in which significant vertical gradients of wind speed, temperature and humidity are observed.

With regarding to the impact of ground surface in the ABL, Garratt (1992) identified two distinct zones: (i) an *outer region* (sometimes referred as *Ekman layer*), where the flow shows little dependence on the nature of surface and Coriolis force due to earth rotation dominates, and (ii) an *inner layer*, with a depth approximately equal to the mentioned *Surface Layer*, where flow is mainly dependent on the surfaces characteristics and little affected by rotation.

2.1.2 Turbulence in the ABL

In the ABL, turbulence refers to fluctuations in wind speed on a relatively fast time scale, typically less than about 10 minutes.

Stull (1988) enumerates three main causes for turbulence in the ABL:

1. Solar heating of ground that causes thermals of warm air to rise in the form of large eddies.

2. Frictional drag over the ground that causes wind shear to develop and become turbulent.

3. Obstacles like trees and buildings that deflect the flow, causing turbulent wakes adjacent and downwind the obstacle.

It is in the lowest depth of the ABL, in the mentioned *Surface Layer*, that turbulence intensity reaches its maximum value due to surface drag, decreasing with height until a nil value in the free atmosphere, making the latter to behave like it can not respond to surface induced changes. In an ABL flow where turbulence is dominated by the wind shear distribution, dissipation of TKE (*Turbulent Kinetic Energy*) also tends to reach it maximum close to the ground, yielding there a near balance between viscous dissipation and shear production.

The different TKE transfer mechanisms in the ABL are identified by Wyngaard (2010), function of its stability:

- In a stable ABL, net rate of buoyant destruction and viscous dissipation yield a loss of TKE. Being mechanical production the only source of turbulence, the latter must extract kinetic energy from the mean flow in order to survive. As a result, the adjustment to geostrophic flow occurs over the depth of the stable ABL.
- In a convective ABL, most of the TKE production is from buoyancy and most of its dissipation occur in the ground (see Figure 2.2). Its much more diffusive behaviour shifts the adjustment to geostrophic flow to the entrainment zone.

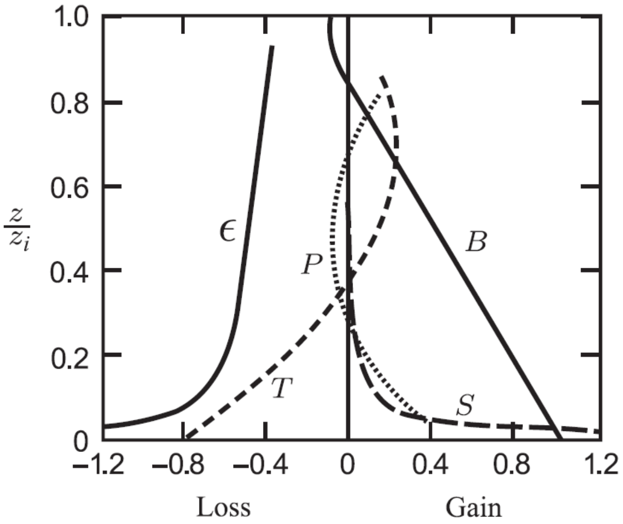


Figure 2.2: The non-dimensionalized (with w_* and z_i) TKE budget in a quasi-steady, horizontally homogeneous CBL. ϵ, viscous dissipation; T, turbulent transport; P, pressure transport; S, shear production; B, buoyant production. From Wyngaard (2010).

2.1.3 Stratification and stability

In a neutrally stratified atmosphere, temperature decreases with height at a ratio called as *adiabatic lapse rate*, which for dry air is given by: [Arya (2001)]

$$\Gamma_d = \frac{\partial T}{\partial z} = -\frac{g}{c_p} \tag{2.1}$$

As the earth surface can be warmer or cooler than the air above, heat transfer between the surface and the air makes temperature profile to deviate from its adiabatic lapse rate, arising what is named as *thermal stratification*. If an air parcel is warmer than the surrounding air, it will feel an upward buoyancy force, making thermals of warm air to rise and its pressure to decrease in response to the surroundings. This will lead to a decrease in temperature and consequently to a decrease in the density of the parcel due to its expansion.

In order to compare air parcels existing at diferent pressures, Oke (1987) defines a more convenient temperature variable, θ, called as *potential temperature*, which is constant during isentropic displacements in the atmosphere:

$$\theta = T\left(\frac{P_0}{P}\right)^{R/c_p} \tag{2.2}$$

It is from potential temperature definition and from its rate of change with height that atmospheres stratification stability can be evaluated:

- If $\partial\theta/\partial z = 0$, the atmosphere is *neutrally stratified*. According to Wyngaard (2010), a pure neutrally stratified ABL is rare in land surface because small potential temperature changes can cause large buoyancy effects.
- If $\partial\theta/\partial z > 0$, the atmosphere is *stably stratified*, which means that if a parcel of air is adiabatically lifted in air, it will be cooler than the surrounding air and more dense and less buoyant, tending to return to its equilibrium position.
- If $\partial\theta/\partial z < 0$, the atmosphere is *unstably stratified*, meaning that if a parcel of air is adiabatically lifted in air, it will be warmer than the surrounding air and more buoyant, making it to accelerate.

Being buoyancy one of the main causes of turbulence in the ABL, stability of its thermal stratification has considerable impacts on wind velocity profile and turbulence behaviour. Oke (1987) and Salby (1996) list some of these impacts (see also Figure 2.3):

- In a neutrally stratified boundary layer, turbulence is entirely of mechanical origin and depends solely on the surface friction and vertical distribution of wind shear.
- Under unstable stratification, turbulent vertical velocities w' are reinforced by buoyancy and therefore vertical movements of eddies are enhanced. Near the surface, mechanical effects continue to dominate, but at greater heights, thermal effects become increasingly more important. This results in a progressive vertical stretching of eddies and a reduction of the wind gradient.
- Under stable stratification, vertical movement and turbulent vertical velocities w' are damped, progressively compressing the eddies and making the vertical component of velocity to contain small fluctuations when compared to other stratification states.

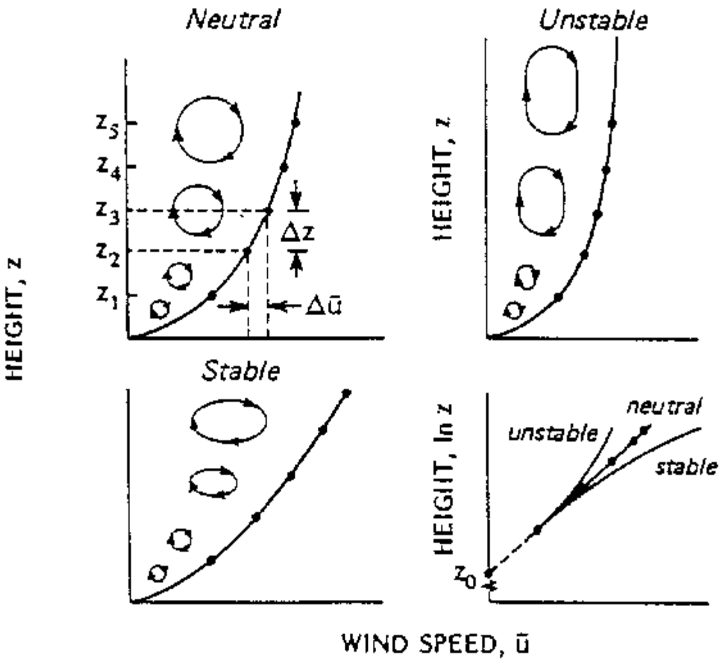


Figure 2.3: The effect of thermal stability on the wind speed profile. Adapted from Oke (1987).

Irrespectively of its stability, mechanically induced turbulence decreases with height until a near null value in the upper atmosphere. Field measurements also showed that the turbulence intensity decreases with increasing wind speed and at low wind speeds is strongly dependent on the atmospheric stability [see Gasch and Twele (2002)].

2.1.4 Roughness and displacement height

As mentioned in Section 2.1.1, wind speed profile ranges from its free stream velocity at some height above the surface, to zero at the ground surface, yielding a logarithmic profile. According to extensive literature sources, and for a neutrally stratified ABL, this profile can be approximated by

$$u(z) = \frac{u_*}{\kappa} \ln\left(\frac{z}{z_0}\right), \tag{2.3}$$

where u_* is a frictional velocity who relates air density with shear stress caused by surface drag by $\tau = \rho {u_*}^2$.

The parameter z_0 in Eq. 2.3 is the ground surface roughness parameter, also called as *aerodynamic roughness length*, which defines the mean level where momentum is absorbed by a canopy [Raupach and Thom (1981)], representing also the height where wind flow approaches zero. Looking at equation 2.3, it can be concluded that as bigger the roughness parameter z_0, the smaller the wind velocity for the same height z. This effect is also mentioned by Oke (1987), as schematically represented in Figure 2.4.

Also Elliot (1958) studied the impact of ground roughness changes on wind profile, defining a variable ΔU_R that he called *velocity perturbation*, concluding that ΔU_R will be negative for flow from a smooth to a rough surface and positive for the rough to smooth case.

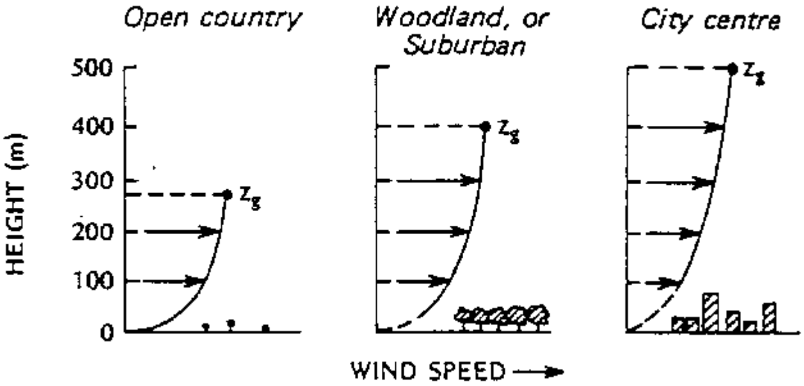


Figure 2.4: The effect of ground roughness on the wind velocity profile. Adapted from Oke (1987).

For surfaces where $z_0 > 0.1$ m, Taylor and Lee (1987) suggest the introduction of a zero plane displacement height d, replacing z by $(z - d)$ in Eq. 2.3, in order to match observed profiles to the logarithmic form, as sketched in Figure 2.5.

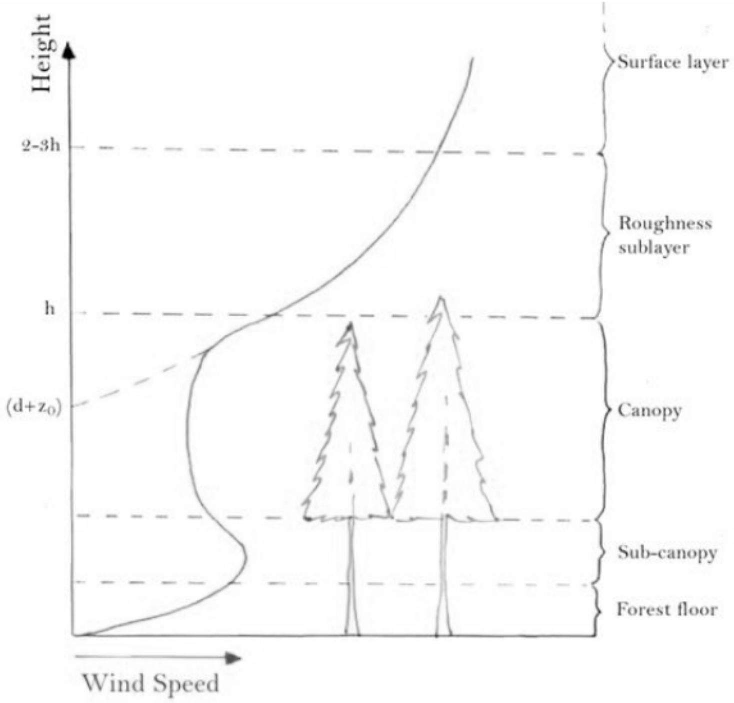


Figure 2.5: Wind velocity profile of flow over forest canopy. From Gardiner (2004).

According to Shaw and Pereira (1982), in forest canopies, d and z_0 are dependent on the height of the canopy, the structure and flexibility of individual plants, the size and arrangement of plant parts, and the planting density: d and z_0 are directly proportional (but not equal) to the height of canopy elements h. While z_0 increases with elements density in sparse canopies and decreases with density in dense canopies, d is only dependent on canopies height and density, whether or not they are in dense areas.

Although roughness parameters can be obtained numerically by fitting the wind velocity profile to experimental data, some analytical models have been developed for d and z_0 estimations. Shaw and Pereira (1982), Choudhury and Monteith (1988) and Raupach (1994) are three examples of these models, all accounting for forest canopy height and density on d and z_0 calculations. As a rule of thumb, Garratt (1992) suggests that z_0 can be estimated as 1/10 of roughness height h and Oke (1987) proposes 2/3 for the ratio d/h. Table 2.1 shows typical d and z_0 parameters for non-urban homogeneous terrain.

Surface type	**z_0 [m]**	**d [m]**
Sea, sand and snow	0.0002	-
Flat desert	0.0002-0.0005	-
Short grass	0.008-0.03	0.025-0.05
Long grass	0.02-0.06	0.1-0.3
Low crops	0.04-0.09	0.2-0.9
High crops	0.12-0.18	0.6-1.5
Continuous bush land	0.35-0.45	1.8-2.4
Mature pine forest	0.8-1.6	3.5-5.6

Table 2.1: Typical roughness parameters for non-urban homogeneous terrain. Wieringa (1980). Adapted from Tieleman (2003).

Oke (1987) mentions also changes in direction of moderate to strong winds due to transitions between zones with different roughness, identifying a convergence of wind direction over a rough surface with wind direction over a smooth surface, and a divergence of wind direction over a smooth surface when it contacts a rough one. These changes will be less noticed in unstable and greater in stable conditions when compared to near neutral conditions.

2.1.5 Length scales and time scales

The physical phenomenons in the ABL have orders of magnitude of time and space that range from a second to several days and from a few millimetres to hundreds of kilometres.

Stull (1988) classifies the horizontal spatial scales in microscales and mesoscales, where mesoscale refers to orders of magnitude above some few kilometres. Hurricanes, nocturnal low level jets and thunderstorms, are typical mesoscale phenomenons, with time scales from one hour to several days. In the microscale range are wakes, plumes and mechanical turbulence, whose time scales range from a few minutes to just a few seconds. For the particular case of turbulence, eddies size can range from near to the boundary layer depth itself, to just a few millimetres, making it difficult to model with respect to the required computational resources.

Being ABL flow always a balance between pressure gradients, Coriolis force and surface drag, the length scale of the modelled phenomenons will dictate the relative importance of involved parameters [see Kantha and Clayson (2000)]. Rossby number, defined as

$$Ro = \frac{U_c}{L2\Omega sin\Phi} \tag{2.4}$$

(where U_c, L, Ω and Φ are respectively, a characteristic velocity, length scale, angular velocity of earth rotation and the latitude of the studied phenomenon), is a measure of the importance of Coriolis accelerations arising from earth rotation. If $\mathcal{O}(Ro) \leq 1$, it means that the system is strongly affected by Coriolis forces. For systems where $\mathcal{O}(Ro) \gg 1$, inertial forces dominate and therefore Coriolis effects can be neglected.

In what concerns the simulation of ABL flows over complex terrain, Kim et al. (2000) state that typical mesoscale (also know as *global circulation*) models are not suitable for that purpose, not only for being based on a hydrostatic approximation that neglects local inertial effects, but also because they are not able to resolve variations in topography in the vertical direction that are important to predict local wind patterns, which typically involve flow separation and recirculation eddies on surfaces with variable roughness.

Castro et al. (2010) combined the benefits of these distinct modelling approaches. They used the WRF-ARW code with historical meteorological data as boundary conditions to simulate a first mesoscale domain. Results were then used as boundary conditions for a URaNS microscale model (Windie code) instead of typical idealized boundary conditions. With this technique, prediction of velocity ratios showed much smaller mean errors than those obtained with mesoscale modelling or uncoupled microscale modelling with idealized boundary conditions.

2.1.6 Topography and turbine sitting

Wind resource assessment for turbine sitting tries to predict a spatial and temporal wind flow pattern in such a way that the profitability of a wind farm life time can be estimated with a pre-established level of uncertainty.

For wind farm evaluation is not only important to predict the quantity of wind during a certain period, but also, the quality of the wind, where small scale topographic variations have a considerable impact [see Palma et al. (2008)]. Santos (2012) lists the most important quantities to assess during wind farm studies, in which topographic effects are here addressed:

Horizontal speed

As energy content of the wind varies with the cube of the average wind speed [Gasch and Twele (2002)], wind turbines profitability will be strongly dependent on small wind speed changes. Surface elevations cause an increase of speed, also called *speed-up*, that Taylor and Lee (1987) attribute to the fact that the whole boundary layer is displaced up and over the hill, and flow acceleration due to the air having to travel faster through the partial constriction caused by terrain. Taylor and Lee (1987) also mention the possibility that speed-up can be reduced by up to about 30% relative to the neutral case for unstable stratification and increased up to 50% with stable stratification.

As mentioned is Section 2.1.4, roughness and forest canopy have a considerable impact on wind speed profile. Figure 2.6 shows schematically this impact on a wind turbine.

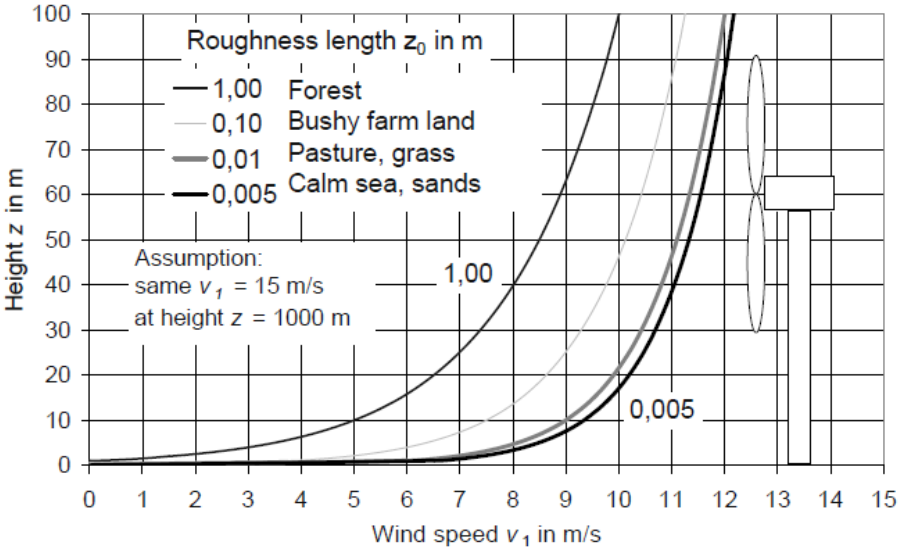


Figure 2.6: Vertical wind profile for different aerodynamic roughness lengths z_0, assuming a geostrophic wind speed of 15 m/s. From Gasch and Twele (2002).

Turbulent intensity

Turbulence intensity is extensively defined in literature as the ratio of standard deviation σ_i to the mean speed $\overline{u_i}$. Together with blade weight, is the main cause of fatigue, producing alternating loads on the blade, root and drive train, alike to reduce wind turbines life time. Gasch and Twele (2002) identified two distinct sources of turbulence affecting wind turbines. One is a *natural turbulence*, caused among other by obstacles, surface roughness and terrain inclination. Figure 2.7 shows a phenomenon likely to happen on steep slope hills: turbulent flow separation on the lee side with a length of influence sufficient to affect the hill top.

The other source of turbulence is the wind turbine wake, that causes an *induced turbulence*, problematic when wind farm layout dictates too small distances between the wind turbines. Induced turbulence means not only alternating loads, but also a reduction

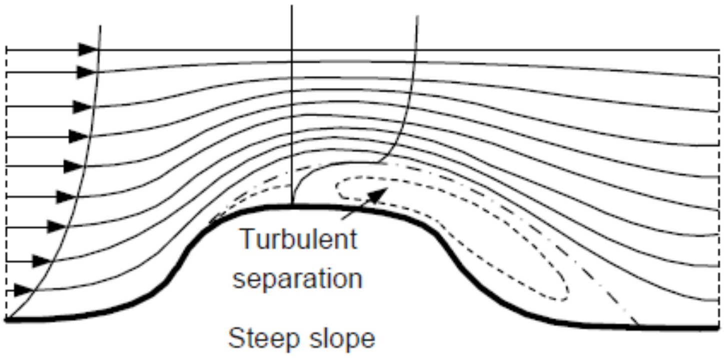


Figure 2.7: Turbulent flow separation on a steep slope hill. From Gasch and Twele (2002).

of wind speed, decreasing the available wind energy for the downstream turbines. According to Santos (2012), the bigger the natural turbulence, the faster it vanishes the wake behind a wind turbine.

Vertical slope

Depending on the site orography, wind profile may deviate from horizontal direction, yielding an oblique flow. According to Gasch and Twele (2002), this deviation is caused by an inclination of the terrain, whose influence decreases with increasing height above ground. The oblique direction of wind makes the blades to be exposed to an alternating inflow angle, leading to increased loads at the blade root and rotor shaft to be stressed by bending, as schematically shown in Figure 2.8.

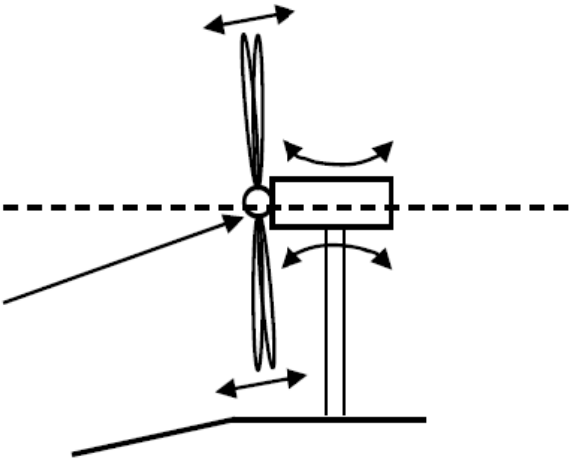


Figure 2.8: Loads on a wind turbine due to oblique flow. From Gasch and Twele (2002).

Wind shear

Gasch and Twele (2002) defined wind shear as the *change of wind speed per meter of height.* It not only causes the blades to experiment alternating loads during every revolution, but also makes wind speed gradient to increase, reducing wind speed for a given height (see

Figure 2.9). Main influences on wind shear behaviour have already been mentioned and are the terrain inclination, obstacles, small distance between turbines and atmospheric stability.

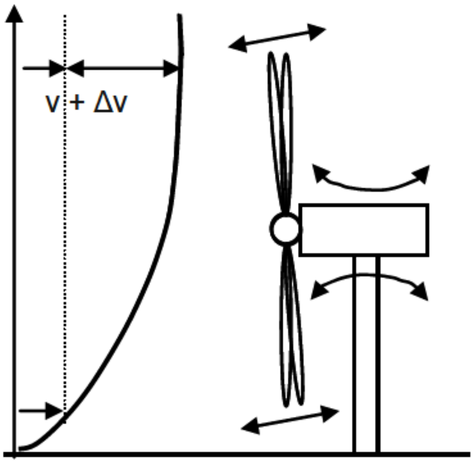


Figure 2.9: Loads on a wind turbine due to wind shear. From Gasch and Twele (2002).

2.2 Turbulence modelling

2.2.1 Introduction

In what concerns the simulation of turbulent flows, Pope (2000) distinguishes between *turbulent simulation* and *turbulence model*. Whereas in a *turbulent simulation* equations are solved for a time-dependent velocity field that, to some extent, is representative of the velocity field $U(x,t)$, in a *turbulence model* equations are solved for some mean quantities, like average speed and dissipation rate.

Two *turbulent simulation* approaches are DNS (*Direct Numerical Simulation*) and LES. In a DNS, all length scales and time scales of the Navier-Stokes equations are solved in order to determine the velocity field $U(x,t)$ for one realization of the flow. Because computational costs increase as Re^3 (*Reynolds number*), DNS is usually restricted to low to moderate Re flows. In LES, equations are solved for a filtered velocity field $\langle U \rangle (x,t)$ representative of the larger scales of turbulent motions. Because small scale motions are not directly solved, involved equations need to include a model to represent the influence of those small scales in the larger scales.

A *turbulence model* involves the averaging of the Navier-Stokes equations, technique also known as RaNS. Due to this averaging, extra terms appear due to the interaction between various turbulent fluctuations. These terms are the so called *Reynolds stresses* $-\overline{u_i' u_j'}$ [see Versteeg and Malalasekera (2007)] and can be evaluated following two different approaches:

1. From a turbulent viscosity model, which can be based on a *mixing length model*

[see Pope (2000)], or can be obtained from quantities such as turbulent kinetic energy k or its dissipation rate ε, for which specific transport equations are solved.

2. Or from modelled transport equations, also know as *Reynolds stress models*, that are solved directly to obtain Reynolds stresses.

According to Versteeg and Malalasekera (2007), RaNS models computing resources required for reasonably accurate flow computations are modest, which made this approach the mainstay of engineering flow calculations over the last few decades.

2.2.2 Computed RaNS models

In this work, turbulence modelling was limited to two-equation RaNS models where *Reynolds stresses* are calculated following the concept of turbulent viscosity. Those computed models will be hereafter briefly presented. The presented transport equations are on its steady-state and incompressible formulation and do not account for buoyancy contributions to the turbulent kinetic energy production term.

The $k - \varepsilon$ model

To Launder and Spalding (1974) is often credited what is called as the standard $k - \varepsilon$ model, in which turbulent viscosity μ_t is computed by

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon}, \tag{2.5}$$

being C_μ a dimensionless constant that relates turbulent shear stress with turbulent kinetic energy. The scalars turbulent kinetic energy k, defined as

$$k = \frac{1}{2}\left(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}\right) \tag{2.6}$$

and its dissipation rate ε are obtained solving two extra transport equations:

$$\rho \frac{\partial \overline{u_j} k}{\partial x^j} = \frac{\partial}{\partial x^j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x^j}\right] + P_k - \rho\varepsilon, \tag{2.7}$$

$$\rho \frac{\partial \overline{u_j} \varepsilon}{\partial x^j} = \frac{\partial}{\partial x^j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x^j}\right] + \frac{C_{\varepsilon 1}\varepsilon}{k}P_k - \frac{C_{\varepsilon 2}\rho\varepsilon^2}{k}, \tag{2.8}$$

where the term P_k represents the production of mechanical turbulence, given by

$$P_k = \sigma_{ij}\frac{\partial \overline{u_i}}{\partial x^j} \tag{2.9}$$

and Reynolds stresses σ_{ij} are computed using the Boussinesq relationship [see Versteeg and Malalasekera (2007)] as follows:

$$\sigma_{ij} = -\rho\overline{u_i'u_j'} = -\frac{2}{3}\rho k\delta_{ij} + \mu_t\left(\frac{\partial\overline{u_i}}{\partial x^j} + \frac{\partial\overline{u_j}}{\partial x^i}\right) \tag{2.10}$$

Prandtl numbers σ_k and σ_ε connect the diffusivity of k and ε to the turbulent viscosity and the dimensionless constants $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$ allow for the correct proportionality between the terms in Eq. 2.8.

The dimensionless constants C_μ, σ_k, σ_ε, $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$ are obtained empirically, tuning the model. Table 2.2 shows two typical sets of values for these constants: the *Standard* set, obtained by Launder and Sharma (1974) and often associated with industrial flows, and the *Atmospheric* set, proposed by Beljaars et al. (1987) for a better description of atmospheric flows.

	C_μ	σ_k	σ_ε	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$
Standard	0.090	1.00	1.30	1.44	1.92
Atmospheric	0.033	1.00	1.85	1.44	1.92

Table 2.2: Typical coefficients of the $k - \varepsilon$ model.

The *Realisable* $k - \varepsilon$ model

According to Pope (2000), the $k - \varepsilon$ model although is said to perform well in two-dimensional thin shear flows with mean streamline curvature and mean pressure gradients, tends to fail in three-dimensional boundary layer flows with strong pressure gradients. To circumvent this deficiency, Shih et al. (1995) proposed a new formulation for both the turbulent dissipation rate equation and turbulent viscosity, arising what is named as the *Realisable* $k - \varepsilon$ model.

In this new formulation, the coefficient C_μ in Eq. 2.5 is no longer a constant but a function of local strain rate and vorticity. Based on values of C_μ obtained experimentally for different type of flows and on what is called as *realisability conditions* [see Versteeg and Malalasekera (2007)], a new formulation for the C_μ coefficient is proposed:

$$C_\mu = \frac{1}{A_0 + A_s U^* \frac{k}{\varepsilon}}, \tag{2.11}$$

where A_0 is a constant and A_s and U^* calculated as proposed by Shih et al. (1995b).

The main difference in the turbulent dissipation rate equation when compared to the standard Eq. 2.8 is the substitution of term P_k, which was dependent on the Reynolds stresses, by a source term S, yielding what the authors claim to be a more robust formulation in what concerns to vortex stretching and dissipation description. This new equation is then defined as,

$$\rho\frac{\partial\overline{u_j}\varepsilon}{\partial x^j}=\frac{\partial}{\partial x^j}\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial\varepsilon}{\partial x^j}\right]+C_1S\rho\varepsilon-\frac{C_2\rho\varepsilon^2}{k+\sqrt{\nu\varepsilon}}, \tag{2.12}$$

where,

$$C_1=max\left[0.43;\frac{\eta}{\eta+5}\right],\ \eta=S\frac{k}{\varepsilon},\ S=\sqrt{2S_{ij}S_{ij}},\ S_{ij}=\frac{1}{2}\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right) \tag{2.13}$$

Table 2.3 presents the model coefficients where the dimensionless constants C_2, σ_k and σ_ε were determined experimentally for a series of different flows.

A_0	C_μ	σ_k	σ_ε	C_1	C_2
4.0	Eq. 2.11	1.00	1.20	Eq. 2.13	1.90

Table 2.3: Typical coefficients of the *Realisable* $k-\varepsilon$ model.

The *RNG* $k-\varepsilon$ model

This model was developed using a mathematical technique called *renormalization group theory*, that according to Versteeg and Malalasekera (2007), removes the small scales of motion from the governing equations by expressing their effects in terms of larger scale motions.

While turbulent kinetic energy transport equation remains the same as in Eq. 2.7, to the turbulent dissipation rate equation is added an additional source term R_ε, that according to Kasmi and Masson (2010), acts as a sink, reducing both k and μ_t and making this model more responsive to the effect of streamline curvature than the standard $k-\varepsilon$ model, and more appropriate for flows over complex terrain. Kim and Patel (2000), comparing different turbulence models, also mention an overestimation of turbulent viscosity by the standard $k-\varepsilon$ model in the upper atmosphere, where wind shear is weaker.

The ε transport equation yields then:

$$\rho\frac{\partial\overline{u_j}\varepsilon}{\partial x^j}=\frac{\partial}{\partial x^j}\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial\varepsilon}{\partial x^j}\right]+\frac{C_{\varepsilon1}\varepsilon}{k}P_k-\frac{C_{\varepsilon2}\rho\varepsilon^2}{k}-R_\varepsilon, \tag{2.14}$$

where,

$$R_\varepsilon=\frac{C_\mu\rho\eta^3(1-\eta/\eta_0)}{1+\beta\eta^3}\frac{\varepsilon^3}{k}, \tag{2.15}$$

being η calculated as in Eq. 2.13, while β and η_0 are dimensionless constants.

Speziale and Thangam (1992) compared two different formulations for this model coefficients: a first one, from Yakhot and Orszag (1986) and named as the *Original RNG* $k-\varepsilon$ model where the coefficients are all constants calculated explicitly from the renormalization group theory, and a second, a *Modified RNG* $k-\varepsilon$ model from Yakhot et al. (1992), where $C_{\varepsilon 1}$ is no longer a constant.

Kasmi and Masson (2010) also proposed a different formulation of the turbulent dissipation rate equation, intended for better description of a neutral ABL flow, obtaining this way a new set of model coefficients.

Table 2.4 resumes the different formulations for the *RNG* $k-\varepsilon$ model coefficients. The set named as *Atmospheric* refers to the one proposed by Kasmi and Masson (2010).

	β	η_0	C_μ	σ_k	σ_ε	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$
Original	0.015	4.38	0.085	0.7179	0.7179	1.063	1.72
Modified	0.015	4.38	0.085	0.7179	0.7179	$1.42 - C_{1R}$	1.68
Atmospheric	0.012	4.38	0.0333	1.00	1.30	0.47	1.68

Table 2.4: Typical coefficients of the *RNG* $k-\varepsilon$ model. $C_{1R} = \eta(1-\eta/\eta_0)/(1+\beta\eta^3)$.

The $k-\omega$ model

Wilcox (1988), identifying also the mentioned limitation of the standard $k-\varepsilon$ model to correctly describe strong pressure gradients, proposed a new two-equation turbulent viscosity model. In this new model, while turbulent viscosity and turbulent kinetic energy quantities are kept, turbulent dissipation rate is replaced by a dissipation frequency ω, related to k and ε by

$$\omega = \frac{\varepsilon}{\beta^* k}, \tag{2.16}$$

where β^* is a dimensionless constant with the same physical meaning as C_μ in the $k-\varepsilon$ model. Putting together Eqs. 2.5 and 2.16, and considering $\beta^* = C_\mu$, turbulent viscosity yields:

$$\mu_t = \rho \frac{k}{\omega} \tag{2.17}$$

With this new quantity ω defined, Wilcox (1988) postulated the following two equations for turbulent kinetic energy and dissipation frequency, respectively:

$$\rho \frac{\partial \overline{u_j} k}{\partial x^j} = \frac{\partial}{\partial x^j} \left[(\mu + \sigma^* \mu_t) \frac{\partial k}{\partial x^j} \right] + P_k - \beta^* \rho \omega k, \tag{2.18}$$

$$\rho\frac{\partial\overline{u_j}\omega}{\partial x^j} = \frac{\partial}{\partial x^j}\left[(\mu+\sigma\mu_t)\frac{\partial\omega}{\partial x^j}\right] + \frac{\alpha\omega}{k}P_\omega - \beta\rho\omega^2, \tag{2.19}$$

where P_k is calculated as in Eq. 2.9 and P_ω calculated using also this same equation, but replacing k by ω in Boussinesq relationship (Eq. 2.10).

The dimensionless coefficients of this model are resumed in Table 2.5 [see Wilcox (1988) for coefficients determination].

β^*	σ^*	σ	α	β
0.090	0.5	0.5	5/9	0.075

Table 2.5: Typical coefficients of the $k-\omega$ model.

The *SST* $k-\omega$ model

Menter (1992) identified some benefits and limitations of the original $k-\omega$ model: although it can achieve an enhanced behaviour in strong pressure gradients when compared to the standard $k-\varepsilon$ model, it still shows some difficulty to correctly describe adverse pressure gradients with flow separation and a too high sensitivity to inlet free stream turbulence properties as boundary conditions.

These observations led Menter (1992) to propose a hybrid model: a $k-\omega$ model in the near wall region that switches to a standard $k-\varepsilon$ model in the free region far from the wall. While turbulent kinetic energy equation keeps the same formulation as in Eq. 2.18, to the dissipation frequency equation is added an extra source term. Due to a change in nomenclature in k equation, both are here presented:

$$\rho\frac{\partial\overline{u_j}k}{\partial x^j} = \frac{\partial}{\partial x^j}\left[(\mu+\sigma_k\mu_t)\frac{\partial k}{\partial x^j}\right] + P_k - \beta^*\rho\omega k, \tag{2.20}$$

$$\rho\frac{\partial\overline{u_j}\omega}{\partial x^j} = \frac{\partial}{\partial x^j}\left[(\mu+\sigma_\omega\mu_t)\frac{\partial\omega}{\partial x^j}\right] + \gamma P_\omega - \beta\rho\omega^2 + 2\rho(1-F_1)\sigma_{\omega 2}\frac{\mu_t}{k}\frac{\partial k}{\partial x^j}\frac{\partial\omega}{\partial x^j} \tag{2.21}$$

Transition from $k-\omega$ to $k-\varepsilon$ behaviour is assured by a function ϕ that determinates the set of constants to be used, where

$$\phi = F_1\phi_1 + (1-F_1)\phi_2, \tag{2.22}$$

being ϕ_1 and ϕ_2 a $k-\omega$ and $k-\varepsilon$ constant respectively, and F_1 a blending function that allows a smooth transition from the $k-\omega$ behaviour near the wall, to the $k-\varepsilon$ model at free stream.

To the turbulent viscosity formulation is added a limiting function in order to accurately capture flow separation under adverse pressure gradients:

$$\mu_t = \frac{\rho a_1 k}{max(a_1\omega, SF_2)}, \tag{2.23}$$

where F_2 is also a blending function and S calculated as in Eq. 2.13.

Based on the experience from general purpose applications of the model, Menter and Esch (2001) introduced a series of modifications in order to improve its performance, mostly in what concerns to blending functions formulation and model coefficients. Menter et al. (2003b,a) revisited these improvements, from which proposed the model coefficients listed in Table 2.6 [see also Menter and Esch (2001) for blending functions formulation].

ϕ_1 set	$\beta^* = 0.090$	$\beta_1 = 0.0750$	$\sigma_{k1} = 0.85$	$\sigma_{\omega 1} = 0.500$	$\gamma_1 = 0.5532$	$a_1 = 0.31$
ϕ_2 set	$\beta^* = 0.090$	$\beta_2 = 0.0828$	$\sigma_{k2} = 1.00$	$\sigma_{\omega 2} = 0.856$	$\gamma_2 = 0.4403$	–

Table 2.6: Typical coefficients of the $SST\ k - \omega$ model.

2.3 OpenFOAM

2.3.1 Introduction

OpenFOAM (the acronym for Open Field Operation and Manipulation) is a C++ based software package comprising multi-physics numerical solvers and complementary utilities. Its development started in the late 1980s at the Imperial College of London, being released as an open source code in 2004. It is nowadays maintained and distributed under the GNU General Public License by the OpenFOAM Foundation.

In its package are included preprocessing utilities for mesh generation and manipulation, tools for the decomposition and reconstruction of computational domains for parallel processing, and postprocessing utilities, like paraFoam, for data extraction and visualization. Numerical capabilities include the RaNS, LES and DNS simulation of incompressible, compressible and buoyancy driven flows, and solvers for molecular dynamics and combustion problems, among others.

Experience with OpenFOAM has shown that its learning curve is conditioned by the shortage of detailed user guides. Template cases supplied with every release with important code comments act as tutorials, giving the user a basis to adapt a typical case to its specific problem. The open source feature also allows the code to be better understood by the user and if intended, modified for a specific case or application.

2.3.2 Case structure

Three main folders comprise a typical OpenFOAM case:

1. `0` folder

   In this folder are stored the files that define the computational domain boundary conditions. Each involved quantity (pressure, velocity, turbulent kinetic energy, etc.) is defined by one file, and in each file assigned the conditions for each boundary, often termed patch.

2. `constant` folder

   This folder contains the files that define the case geometry and its computational domain, and some constants to be used in the transport equations, like modelled transport properties. Typical subfolder is `polyMesh`.

3. `system` folder

   In this folder are kept the files that define solvers convergence criteria and numerical schemes, domain decomposition for parallel processing and postprocessing data extraction, among others.

An example of the structure and content of an OpenFOAM case is presented in Figure 2.10. It concerns the simulation of an adiabatic and incompressible flow using the $k - \varepsilon$ turbulence model and parallel processing.

```
<Case>
|-- 0
|   |-- epsilon
|   |-- k
|   |-- nut
|   |-- p
|   `-- U
|-- constant
|   |-- polyMesh
|   |   `-- blockMeshDict
|   |-- RASProperties
|   `-- transportProperties
`-- system
    |-- controlDict
    |-- decomposeParDict
    |-- fvSchemes
    `-- fvSolution
```

Figure 2.10: Example of an OpenFOAM case folder structure and its content.

2.3.3 OpenFOAM and the simulation of ABL flows

A bibliographic research about the simulation of atmospheric flows using OpenFOAM led to the conclusion that this is not yet a current practice in the wind energy industry and its use and development seams to be limited to academic community.

Three examples of the application of OpenFOAM on the simulation of atmospheric flows over complex terrain were found: the Master Thesis of Tapia (2009) and Martinez (2011) and the research study of Sumner et al. (2010). The main conclusions drawn from this studies are that, although OpenFOAM can be a promising tool for the simulation of atmospheric flows over complex terrain, mesh generation from topographical data is still a difficult task to perform, missing also some important features, like a canopy model and the account for Coriolis force.

Chapter 3

Developed procedures

3.1 Introduction

In this chapter are presented the developed preprocessing and postprocessing tools and procedures for the simulation of atmospheric flows over complex terrain, using OpenFOAM. All tests and simulations were conducted in OpenFOAM v2.1.1, which was used *as is*, meaning that no code modifications were made. Developed tools were coded in FORTRAN and compiled with gfortran under a Linux distribution.

3.2 Model limitations

The present work was developed having on its basis the following assumptions and simplifications:

- **Absence of buoyancy and gravity force**

  Tools and procedures were developed for the simulation of steady state and idealized neutrally stratified flows, where buoyancy effects are absent (see Section 2.1.3). Hydrostatic pressure due to gravity was also neglected.

- **Absence of a canopy model**

  In OpenFOAM, modelling of surface roughness is limited to the effect of aerodynamic roughness length z_0 which by its turn is taken in account in ground patch *wall function*. Zero plane displacement height and its effects as mentioned in Section 2.1.4 are here neglected, limiting its applicability to low roughness lengths.

- **Absence of Coriolis force**

  As far as it was possible to determine, OpenFOAM transport equations do not have the ability to take in account for Coriolis force effects. Evaluation of the impacts of this simplification can be made for each simulation case as mentioned in Section 2.1.5. For code transformation in order to embed Coriolis force see Tapia (2009).

- **Incompressible and dry air**

  Air was modelled in its dry and incompressible form, which according to reviewed bibliographic sources is a current practice in the simulation of neutrally stratified ABL flows and whose impacts in final results can be neglected.

3.3 Preprocessing procedures

3.3.1 Mesh generation

Introduction

For numerical mesh generation of a complex terrain two OpenFOAM utilities were preliminary tested: (i) *snappyHexMesh* and (ii) *extrudeMesh*, whose main results and conclusions are here briefly presented:

snappyHexMesh

This is an OpenFOAM utility that sculpts a previously generated mesh block according to the *xyz* information of a STL (*STereoLithography*) file.

Prior to the STL file generation, *xyz* terrain data was manipulated with *gsurf*, an ISEP *in-house* code for UTM (*Universal Transverse Mercator*) coordinate transformation, that allows for coordinate grid rotation, grading of nodal spacing in a geometric expansion basis and flattening of domain borders for calculations smoothing. Terrain data used during this stage of the work concerns to *Serra do Cabeço da Rainha*, a hill in the centre of Portugal near the city of Castelo Branco. Figure 3.1 shows the terrain surface generated by *gsurf* in the VTK (*Visualization Toolkit*) format. A 20 × 20 km domain was meshed in a 60 × 60 coordinates surface, with a central coordinates distance in both x and y directions of 200 m that expands towards to the edges.

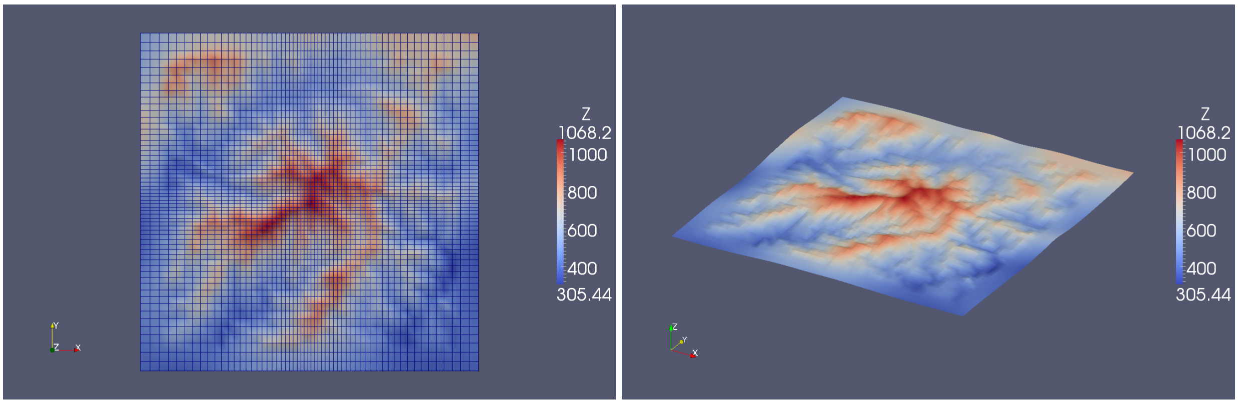
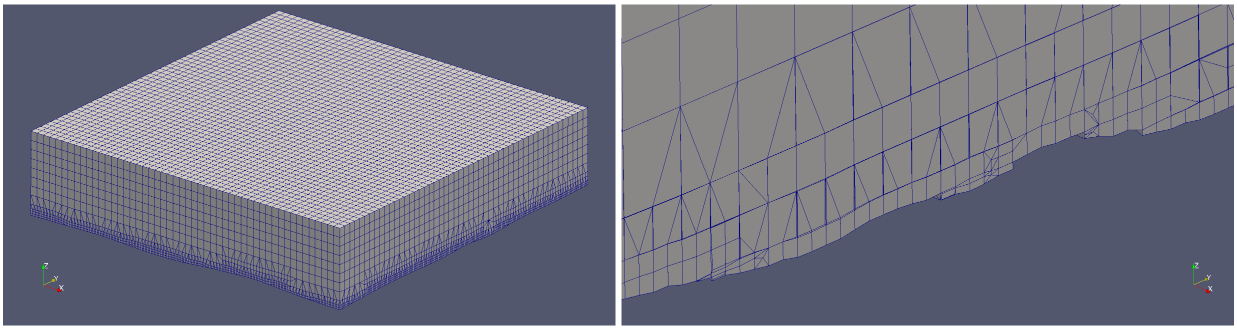


Figure 3.1: VTK surface of *Serra do Cabeço da Rainha*. North points to x negative.

STL file was generated from *xyz* VTK file data recurring to a GNU Octave script. Mesh block for *snappyHexMesh* application was modelled with the same *xy*



(a) (b)

Figure 3.2: *snappyHexMesh* test with *Serra do Cabeço da Rainha* terrain data. (a) Domain overview. (b) Cross section plane normal to x at domain centre.

dimensions as STL surface and with a height of 4000 m.

Configuration of *snappyHexMesh* was performed having on its basis existing `snappyHexMeshDict` tutorial files and its code comments, always following a trial and error procedure. Although STL surface was successfully morphed in the mesh block, results showed a highly unstructured aspect near the terrain surface (see Figure 3.2) and the inability to accurately control the total amount of cells in final mesh.

The work of Martinez (2011) showed that, although mesh quality can be improved near the ground surface, *snappyHexMesh* output is highly dependent on the available computational resources and on the user skills to tune the set up parameters in `snappyHexMeshDict` file for each topography.

extrudeMesh

This OpenFOAM utility allows the user to extrude a previously generated STL surface, giving rise to a shaped mesh block.

STL surface used in this test was the same as for *snappyHexMesh*. Parameters in `extrudeMeshDict` file were set up in order to extrude 20 cells with an expansion ratio r (the ratio of dimensions between one control volume and its neighbour) equal to 1.1 along a distance of 4000 m towards z positive direction. Figure 3.3 shows the mesh obtained with this procedure.

Main conclusions from the conducted test were that, although near ground mesh aspect was highly improved when compared to *snappyHexMesh* procedure, the total amount of cells obtained in final mesh was exactly twice as expected considering the original VTK surface file data. This difference between expected and obtained cells number was attributed to the intrinsic triangulated structure of the STL format, in which each rectangular face of the VTK surface was split in two triangles during STL file generation.

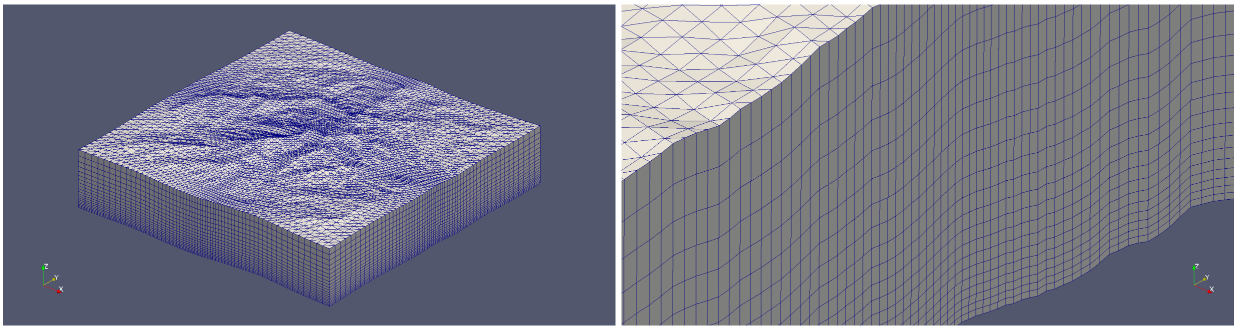

(a) (b)

Figure 3.3: *extrudeMesh* test with *Serra do Cabeço da Rainha* terrain data. (a) Domain overview. (b) Cross section plane normal to x at domain centre.

Developed procedure

Gathered results with *snappyHexMesh* and *extrudeMesh* utilities motivated the development of a more systematic tool (here termed *write_blockMeshDict*) which could assure an accurate control over the total amount of cells that a domain is decomposed in, keeping also an adequate aspect ratio near the ground surface.

Followed approach for mesh generation was the development of a code able to read the VTK file created by *gsurf* and write its xyz data in the OpenFOAM `blockMeshDict` format. With this procedure, and upon run of OpenFOAM *blockMesh* utility, each face from the VTK surface would originate an hexahedron with bottom face shaped to ground surface z data and an horizontal top face at an user defined height. These hexahedrons would also be split in a predefined number of cells, in which each cell z dimensions would expand at a ratio R in z positive direction.

Figure 3.4 shows the vertex referral scheme and domain patch decomposition followed for `blockMeshDict` file generation. Each vertex is coded in the form $n = (x_i, y_j)$, where n is the vertex number and (x_i, y_j) the xy coordinate pair for vertex n. Acronyms nx and ny stand respectively for the number of coordinates in x and y directions that the domain was decomposed in upon VTK file generation. Figure 3.6 shows the definition of the first and last pair of haxahedrons of a domain in which $nx = ny = 60$.

Domain boundaries were decomposed in six patches: *ground* to which was assigned the `wall` attribute, and *sky*, *inlet*, *outlet*, *front* and *back*, all defined as generic `patch`. To nodes belonging to *ground* patch were assigned the respective z coordinates of VTK file data and for nodes in *sky* patch z was set equal to an user defined domain height.

Mesh obtained with this procedure is shown in Figure 3.5. In its base was the same VTK file used in *snappyHexMesh* and *extrudeMesh* preliminary tests. A domain of $20 \times 20 \times 4$ km was decomposed in $59 \times 59 \times 20$ cells expanding at a ratio R (the ratio of dimensions between the first and last control volumes measured in one orthogonal direction) equal to 20 in z positive direction. Total number of cells was exactly as expected (69620) and near ground surface shows a fully structured mesh following terrain

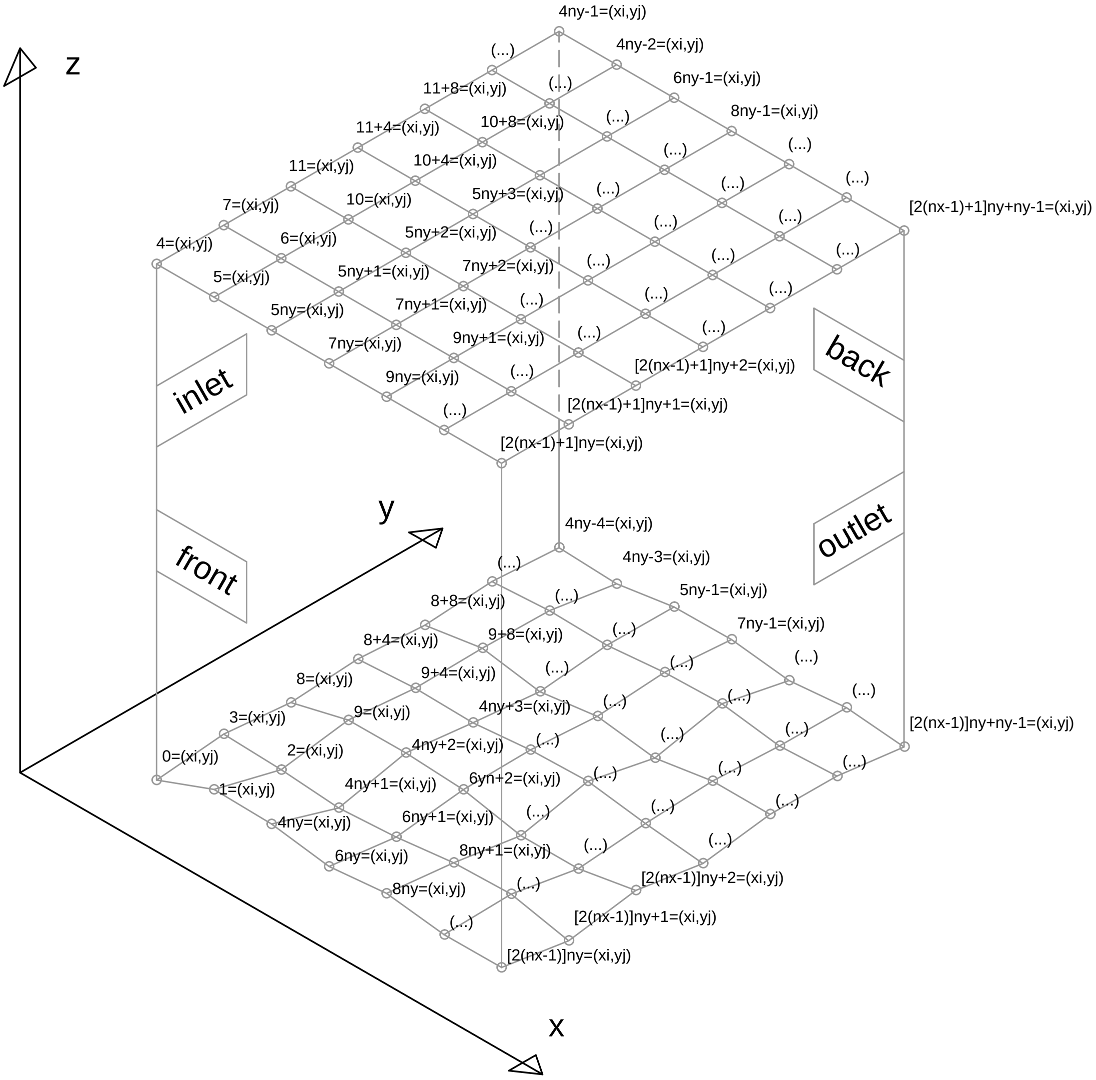
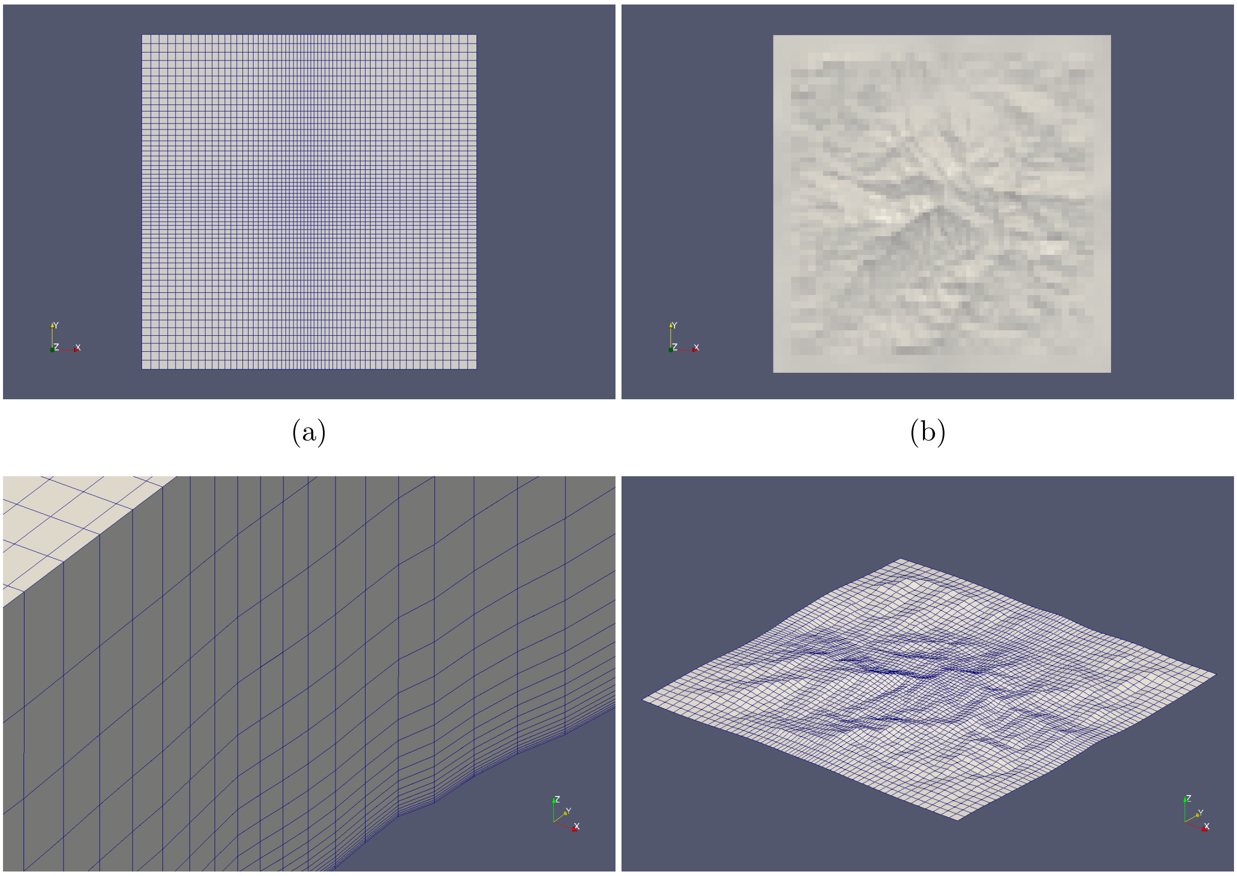


Figure 3.4: Vertex coding scheme and domain patch decomposition for `blockMeshDict` file generation. *Ground* patch below *sky* patch.

z data. Coordinates grid manipulation with *gsurf* took approximately 30 seconds to run and compiled *write_blockMeshDict* code less then 1 second to run and write `blockMeshDict` file. Clock time of *blockMesh* utility was approximately 1/20 when compared to *snappyHexMesh* and almost equal to *extrudeMesh* clock time. Developed procedure was successfully tested on both rectangular (i.e., $nx \neq ny$) and square coordinates grids.

For *write_blockMeshDict* set up, three parameters have to be defined by the user (see Appendix B for further information): (i) domain height at *sky* patch, (ii) required number of cells in z direction and (iii) its expansion ratio R.



(a) (b)

(c) (d)

Figure 3.5: Mesh obtained for *Serra do Cabeço da Rainha* terrain data with *write_blockMeshDict* procedure. (a) Top view of *sky* patch. (b) Top view of *ground* patch. (c) Cross section plane normal to x at domain centre. (d) Isometric view of *ground* patch. Domain borders were flattened in order to smooth later calculations.

```
blocks
(
        hex (0 1 2 3 4 5 6 7) (...)
        hex (3 2 9 8 7 6 10 11) (...)
        (...)
        hex (7017 7137 7138 7018 7077 7197 7198 7078) (...)
        hex (7018 7138 7139 7019 7078 7198 7199 7079) (...)
);
```

Figure 3.6: Example of application of the vertex referral scheme for `blockMeshDict` file generation. `(...)` represents the omission of the remaining code.

3.3.2 Boundary conditions

Introduction

As mentioned in Section 3.2, development of tools and procedures in this work were based on the prior assumption that those same tools would be applied only to the simulation of idealized incompressible and neutrally stratified flows. Due to this limitation, the quantities that have to be defined in OpenFOAM case `0` folder are restricted to: pressure p, velocity U, turbulent kinetic energy k, turbulent kinematic viscosity ν_t, and turbulent kinetic energy dissipation rate ε or its dissipation frequency ω, whether the computed turbulence model is a $k-\varepsilon$ or a $k-\omega$ based model.

Being turbulent viscosity directly dependent on the surface aerodynamic roughness length, set up of ν_t boundary conditions will be addressed in Section 3.3.3, concerning roughness modelling. In the remaining quantities, pressure is the only that can be considered as having static conditions, i.e., its boundary conditions can be defined irrespectively of site data or any other involved quantities. Table 3.1 shows the boundary conditions assigned to pressure field in `p` file to each of patches that the domain was decomposed in. Additionally to the set up of boundary conditions for user defined patches, OpenFOAM requires a first guess for computational domain internal field.

Patch/Field	Attribute	Value	First guess
ground, inlet	`zeroGradient;`	-	-
sky, front, back	`symmetryPlane;`	-	-
outlet	`fixedValue;`	`uniform 0;`	-
internal	-	-	`uniform 0;`

Table 3.1: Boundary conditions for pressure field set up in `p` file.

The set up of `symmetryPlane` attribute in *sky*, *front* and *back* patches had on its basis the definition of a no-flux condition through these boundaries in velocity field. In that same way, to *ground* patch was also assigned a null gradient condition, given that patch was already defined as a no-slip and impermeable `wall` in `blockMeshDict` file. For *inlet* and *outlet* patches was followed the prescription of OpenFOAM Programmer's Guide (2012) for incompressible flows: to assign a `zeroGradient` condition for *inlet* patch and a `fixedValue` to *outlet* patch, in which the latter will act as a reference state for the remaining pressure field calculation.

Developed procedure

Being U, k, ε and ω boundary conditions dependent on the studied site data, the followed approach was to developed a tool (termed *write_bCs*) able to write `U`, `k`, `epsilon` and `omega` files based on user defined parameters.

In this work, influence of site data on domain boundaries was limited to *inlet* and *ground* patches. For *sky*, *front* and *back* patches was assigned the `symmetryPlane` attribute in order to follow the mentioned no-flux condition, which is physically achieved by setting it in `U` file, making velocity to be tangential to these boundaries. Boundary conditions for *outlet* patch follow the Versteeg and Malalasekera (2007) advice: to assign a null gradient condition for scalars k, ε and ω, and a mass conservation condition for U, which in OpenFOAM is assured by the `inletOutlet` attribute.

Inlet boundary conditions for U, k and ε followed the ones proposed by Castro et al. (2010), in which

$$u = \begin{cases} u_*/\kappa \ln(1+z/z_0), & z < \delta \\ u_*/\kappa \ln(1+\delta/z_0), & z \geq \delta, \end{cases} \tag{3.1}$$

$$k = \begin{cases} u_*^2/C_\mu^{1/2}(1-z/\delta), & z \leq 0.99\delta \\ u_*^2/(100C_\mu^{1/2}), & z > 0.99\delta, \end{cases} \tag{3.2}$$

$$\varepsilon = \begin{cases} C_\mu^{3/4}k^{3/2}/(\kappa z), & z \leq 0.95\delta \\ C_\mu^{3/4}k^{3/2}/(0.95\delta\kappa), & z > 0.95\delta. \end{cases} \tag{3.3}$$

ω profile was derived from Eq. 2.16, also on a height dependence basis.

At *ground* patch, U was set equal to zero in order to define a no-slip condition. To avoid the need to integrate turbulence model equations through the wall, which would highly increase computational effort [Pope (2000)], *wall functions* were assigned to each of the turbulent quantities k, ε and ω, as follows:

- k

  For turbulent kinetic energy was used the OpenFOAM `kqRWallFunction`, which as mentioned in its C++ file header, assigns a zero gradient condition at the wall. According to Pope (2000), this boundary condition has on its basis a commonly used assumption of a constant value of k along the near wall region. This assumption was also extensively found in the reviewed literature concerning the simulation of ABL flows over complex terrain: see e.g. Kim et al. (2000) or Castro et al. (2003) who set $k_{wall} = u_*^2/C_\mu^{1/2}$.

- ε

  OpenFOAM wall function for ε, named as `epsilonWallFunction`, follows the *standard* near wall treatment prescribed by Launder and Spalding (1974), in which $\varepsilon_{wall} = u_*^3/(\kappa z_p)$, being z_p the distance between ground face and adjacent cell centre. Bibliographic research showed two main formulations for ε wall function in the simulation of ABL flows over complex terrain: the standard formulation used, e.g., by Kim and Patel (2000) or Hargreaves and Wright (2007), or the account for aerodynamic roughness length in the form $\varepsilon_{wall} = u_*^3/[\kappa(z_p + z_0)]$ in Richards and Hoxey (1993) or Castro et al. (2003). The inconsistency between OpenFOAM wall function (who does not take in account for z_0) and the alternative ε_{wall} formulation, although identified here, will not be a matter of a more detailed study in this work.

- ω

  Wall function built in OpenFOAM for ω, named as `omegaWallFunction`, follows the Menter and Esch (2001) formulation, in which $\omega_{wall} = \sqrt{{\omega_{vis}}^2 + {\omega_{log}}^2}$, being $\omega_{vis} = 6\nu/(0.075 z_p)$ and $\omega_{log} = u_*/(0.3\kappa z_p)$. In the bibliographic research performed during this work was not possible to identify any specific formulation for ω near wall treatment in the simulation of ABL flows over complex terrain.

First guesses for k and ε were based in the Richards and Hoxey (1993) boundary conditions, in which:

$$k = \frac{u_*^2}{C_\mu^{1/2}}, \tag{3.4}$$

$$\varepsilon = \frac{u_*^3}{\kappa(z + z_0)}, \tag{3.5}$$

setting $z = 0$ in Eq. 3.5 for *wall* patch and equal to an user defined value for internal field.

Prior to the generation of `U`, `k`, `epsilon` and `omega` files, ***write_bCs*** reads the geometrical data of *inlet* patch. This data is gathered from files generated upon run of OpenFOAM *blockMesh* utility: in `boundary` file are identified the faces that match *inlet* patch, in `faces` file the points that define each of the faces, and in `points` file the xyz data of each point. From z coordinates of each point, ***write_bCs*** builds height dependent profiles for u, k, ε and ω, setting height as the distance between *ground* patch and the averaged z coordinate of each set of four points that define a face. These profiles are then written in a raw ASCII file, being read by OpenFOAM solver following an `#include` instruction [see OpenFOAM User Guide (2012)] in each of the `U`, `k`, `epsilon` and `omega` files. Coded boundary conditions in ***write_bCs*** are summarized in Table 3.2.

Figure 3.7 shows the *inlet* boundary conditions generated with this procedure for computational domain in Figure 3.5. Height of first face is approximately 27.6 m, a much higher value when compared to simulation cases found in literature, just set here for the purpose of code testing.

User set up of ***write_bCs*** concerns the definition of six parameters (see Appendix B for further information): u_*, z_0, δ, κ, C_μ or β^* required by OpenFOAM for *wall functions* irrespectively of defined constants for turbulence model, and z for a first guess of ε internal field in Eq. 3.5.

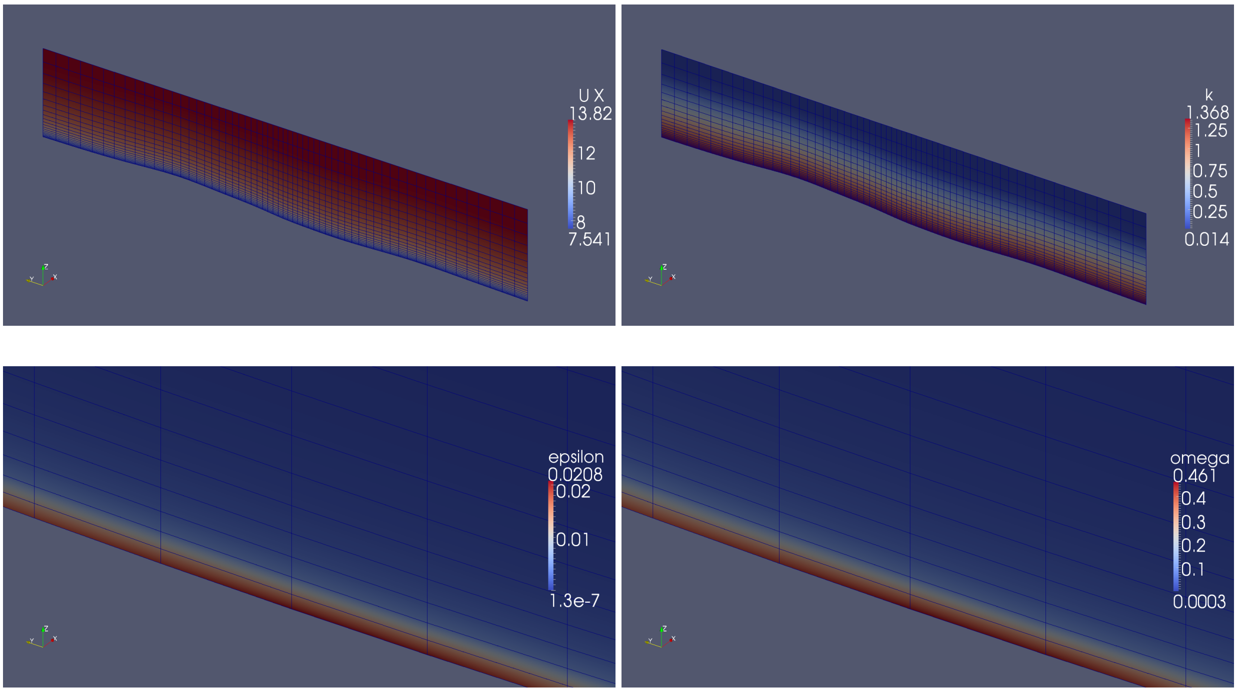


Figure 3.7: Generated boundary conditions at *inlet* patch for *Serra do Cabeço da Rainha* computational domain: $u_* = 0.5$ m/s, $\kappa = 0.41$, $z_0 = 0.03$ m, $\delta = 2500$ m and $\beta^* = C_\mu = 0.033$.

Patch/Field	Attribute	Value	First guess
U			
ground	`fixedValue;`	`uniform (0 0 0);`	-
sky, front, back	`symmetryPlane;`	-	-
inlet	`fixedValue;`	Eq. 3.1	-
outlet	`inletOutlet;`	`$internalField;`	-
internal	-	-	`uniform (0 0 0);`
$\boldsymbol{k}$, $\boldsymbol{\varepsilon}$, $\boldsymbol{\omega}$			
ground	Respective wall function	-	Eq. 3.4, 3.5, 2.16
sky, front, back	`symmetryPlane;`	-	-
inlet	`fixedValue;`	Eq. 3.2, 3.3, 2.16	-
outlet	`zeroGradient;`	-	-
internal	-	-	Eq. 3.4, 3.5, 2.16

Table 3.2: Boundary conditions for U, k, ε and ω coded in *write_bCs*.

3.3.3 Roughness modelling

Introduction

In OpenFOAM, terrain roughness is modelled in `nut` file as a boundary condition of kinematic turbulent viscosity, being aerodynamic roughness length a parameter of `nutkAtmRoughWallFunction`, a *wall fuction* that calculates ν_t in the form

$$\nu_t = \nu_l \left[\frac{z^+ \kappa}{\ln[max(1 + z/z_0, 1 + 10^{-4})]} - 1 \right], \tag{3.6}$$

in which ν_l is the fluid kinematic viscosity.

Although performed bibliographic research has not allowed to validate this function, the following procedure led to conclude that it is based on a shear stress near wall balance from a logarithmic velocity profile as in Eq. 3.1:

1. Considering effective kinematic viscosity $\nu = \nu_l + \nu_t$ [Gorlé et al. (2009)] and using the shear stress definition $\tau = \mu \ \partial u / \partial z = \rho u_*^2$, ν_t can be given by:

$$\nu_t = \nu_l \left(\frac{u_*^2}{\nu_l \partial u / \partial z} - 1 \right) \tag{3.7}$$

2. Discretizing then $\partial u / \partial z \approx \Delta u / \Delta z = (u - u_{wall})/(z - z_{wall})$ and making $z_{wall} = 0$ and $u_{wall} = 0$ due to the no slip condition, Eq. 3.7 takes the form:

$$\nu_t = \nu_l \left(\frac{u_*^2 z}{\nu_l u} - 1 \right) \tag{3.8}$$

3. Finally, writing $u = u_*/\kappa \ln(1 + z/z_0)$ as in Eq. 3.1 and taking $z^+ = u_* z / \nu_l$ [White (1999)], Eq. 3.8 takes the form as in Eq. 3.6, but the *max* function, interpreted here as an artefact to avoid $\ln(1) = 0$ in fraction bottom for $z = 0$.

In the remaining computation field, ν_t keeps being calculated as in Eq. 2.5. According to Hargreaves and Wright (2007), in CFD codes based on the concept of turbulent viscosity, effective kinematic viscosity $\nu = \nu_t + \nu_l$ replaces the laminar ν_l in *momentum* equation.

Developed procedure

Followed approach for roughness modelling was the development of a tool able to write `nut` file based on user defined data. This tool (here termed *write_z0*) should map site roughness data in three different modes: (i) considering an uniform aerodynamic roughness length over all terrain surface, (ii) following an user defined equation, which e.g., would describe roughness on a ground height dependent basis, and (iii) based on real site data. Implementation of each of these modes is here described:

1. **Uniform roughness**

   In uniform roughness mode, *write_z0* simply sets a `z0 uniform $z0u` attribute in `nut` file for *ground* patch, in which `$z0u` is the uniform user defined value for aerodynamic roughness length over terrain surface.

2. **Equation based roughness map**

   In this mode, prior to roughness mapping, *write_z0* acquires *ground* patch xyz data from `boundary`, `faces` and `points` files following the same procedure as mentioned in Section 3.3.2 for *write_bCs* and *inlet* patch. For a ground height dependent roughness condition, the value of z_0 for each face is determined from an user defined equation built in *write_z0* code, being ground height the average z value of the each of four points that defined a face. Each of the z_0 values are written in `nut` file preceded by a `z0 nonuniform List<scalar>` instruction [see OpenFOAM User Guide (2012)].

3. **Real site data roughness map**

   For real site data roughness mapping, terrain roughness data in the xyz_0 format is previously manipulated with a modified version of *gsurf* in order to grade and rotate this data in the same form as xyz terrain data for *write_blockMeshDict* application (see Section 3.3.1). With the generated VTK file, *write_z0* matches each of the xy points that define *ground* patch computational domain with each of the xy points of VTK roughness map, assigning the respective z_0 to each of the matched xy pair. The aerodynamic roughness length for each face will be the averaged z_0 of the four points that define that same face. These z_0 values are written in `nut` file recurring to the already mentioned `z0 nonuniform List<scalar>` instruction.

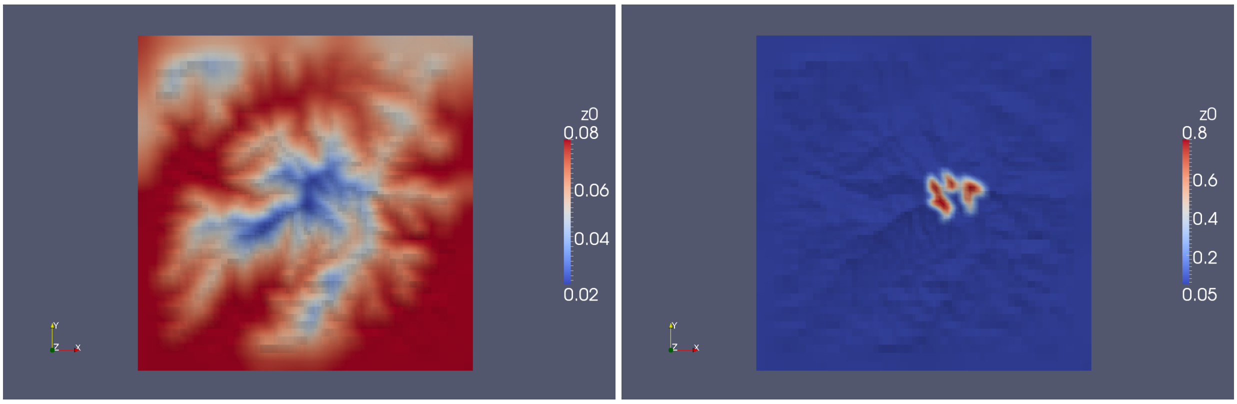
For visual inspection of generated roughness maps an additional file was created in `0` folder, named `z0`, forcing paraFoam to load an extra scalar field with the same visualization capabilities as for remaining scalar quantities in that folder. This `z0` file calls roughness data written by *write_z0* in a raw ASCII file via an `#include` instruction.

Figure 3.8 shows the roughness maps obtained with this procedure for equation based mode and real site data mode, applied to computational domain of *Serra do Cabeço da Rainha* (see Figure 3.5). Equation based roughness mapping mode describes a random canopy condition, just set here for the purpose of code testing. Boundary conditions for turbulent kinematic viscosity are summarized in Table 3.3.

Patch/Field	Attribute	Value	First guess
ground	`nutkAtmRoughWallFunction;`	-	`uniform 0;`
sky, front, back	`symmetryPlane;`	-	-
inlet, outlet	`calculated;`	-	`uniform 0;`
internal	-	-	`uniform 0;`

Table 3.3: Boundary conditions for ν_t coded in *write_z0*.

User set up of *write_z0* concerns the definition of the required roughness mapping mode, including the value of z_0 for uniform roughness mode and the equation for equation based mapping mode (see Appendix B for further information). Like other OpenFOAM wall functions, `nutkAtmRoughWallFunction` requires the definition of C_μ irrespectively of defined turbulence model constants in order to calculate friction velocity in the form $u_* = C_\mu^{1/4}\sqrt{k}$.



(a) (b)

Figure 3.8: Roughness maps for *Serra do Cabeço da Rainha* computational domain. (a) Equation based roughness mapping mode ($z_0 = 0.08$ m for $z \leq 500$ m; linear decay in the form $0.08 > z_0 > 0.02$ m for $500 < z \leq 1050$ m; $z_0 = 0.02$ m for $z > 1050$ m). (b) Real site data close to the point of interest (in domain centre).

3.3.4 Turbulence properties

Introduction

In OpenFOAM, configuration of RaNS model is performed in `RASProperties` file, defining the required turbulence model and its coefficients set.

Additionally to models and formulations mentioned in Section 2.2.2, and based in the work of Martinez (2011), it was also considered a Calibrated $k-\varepsilon$ formulation, in which C_μ is adjusted according to measured sited data: calculating k as in Eq. 2.6 with field measured mean velocity fluctuations $\overline{u'}$, $\overline{v'}$ and $\overline{w'}$ and estimating u_* from an adjusted logarithmic velocity profile, C_μ is then calculated from Eq. 3.4 and σ_ε adjusted following the Beljaars et al. (1987) formulation, in which

$$\sigma_\varepsilon = \frac{\kappa^2}{\sqrt{C_\mu}(C_{\varepsilon 2} - C_{\varepsilon 1})} \tag{3.9}$$

Developed procedure

The aim of the developed tool (here termed *write_turbulenceProperties*) was to automate the configuration of the required turbulence model, writing `RASProperties` file according to an user defined option concerning the required turbulence model and its formulation (see Appendix B for further information).

Coded models in *write_turbulenceProperties* are summarized in Table 3.4. For the *Realisable* $k-\varepsilon$ and Modified *RNG* $k-\varepsilon$ models, η first guess was calculated as proposed by Yakhot et al. (1992), according to which, in the logarithmic region of a turbulent boundary layer $\eta = 1/\sqrt{C_\mu}$.

Model description	Formulation	Coefficients set
Standard $k-\varepsilon$	Launder and Sharma (1974)	Table 2.2
Atmospheric $k-\varepsilon$	Beljaars et al. (1987)	Table 2.2
Calibrated $k-\varepsilon$	Beljaars et al. (1987), Martinez (2011)	Site data + Table 2.2
Realisable $k-\varepsilon$	Shih et al. (1995)	Table 2.3
Original *RNG* $k-\varepsilon$	Yakhot and Orszag (1986)	Table 2.4
Modified *RNG* $k-\varepsilon$	Yakhot et al. (1992)	Table 2.4
Atmospheric *RNG* $k-\varepsilon$	Kasmi and Masson (2010)	Table 2.4
$k-\omega$	Wilcox (1988)	Table 2.5
SST $k-\omega$	Menter and Esch (2001)	Table 2.6

Table 3.4: Turbulence models coded in *write_turbulenceProperties*.

3.4 Postprocessing procedures

The postprocessing procedures developed here were based in the OpenFOAM *sample* utility, a routine for the sampling of scalar and vector fields according to user defined coordinates, allowing for data extraction in a height AGL (*Above Ground Level*) basis. To accomplish this, was developed a tool (here termed *write_sample*) able to write these coordinates in three different modes:

1. **Along a surface at a predefined height above ground level**

   This surface is defined by a square matrix of coordinates generated according to three user defined parameters: (i) the lower and upper xy boundaries in computational domain, (ii) the height AGL of coordinates z component and (iii) the matrix dimensions.

2. **Along a line at a predefined height above ground level**

   For the generation of this line of coordinates, the following parameters are required: (i) the xy position of line middle point in computational domain, (ii) line length, (iii) the angle of rotation referenced from north, (iv) the height AGL of coordinates z component and (v) the number of evenly distributed points along its length.

3. **Along a vertical line**

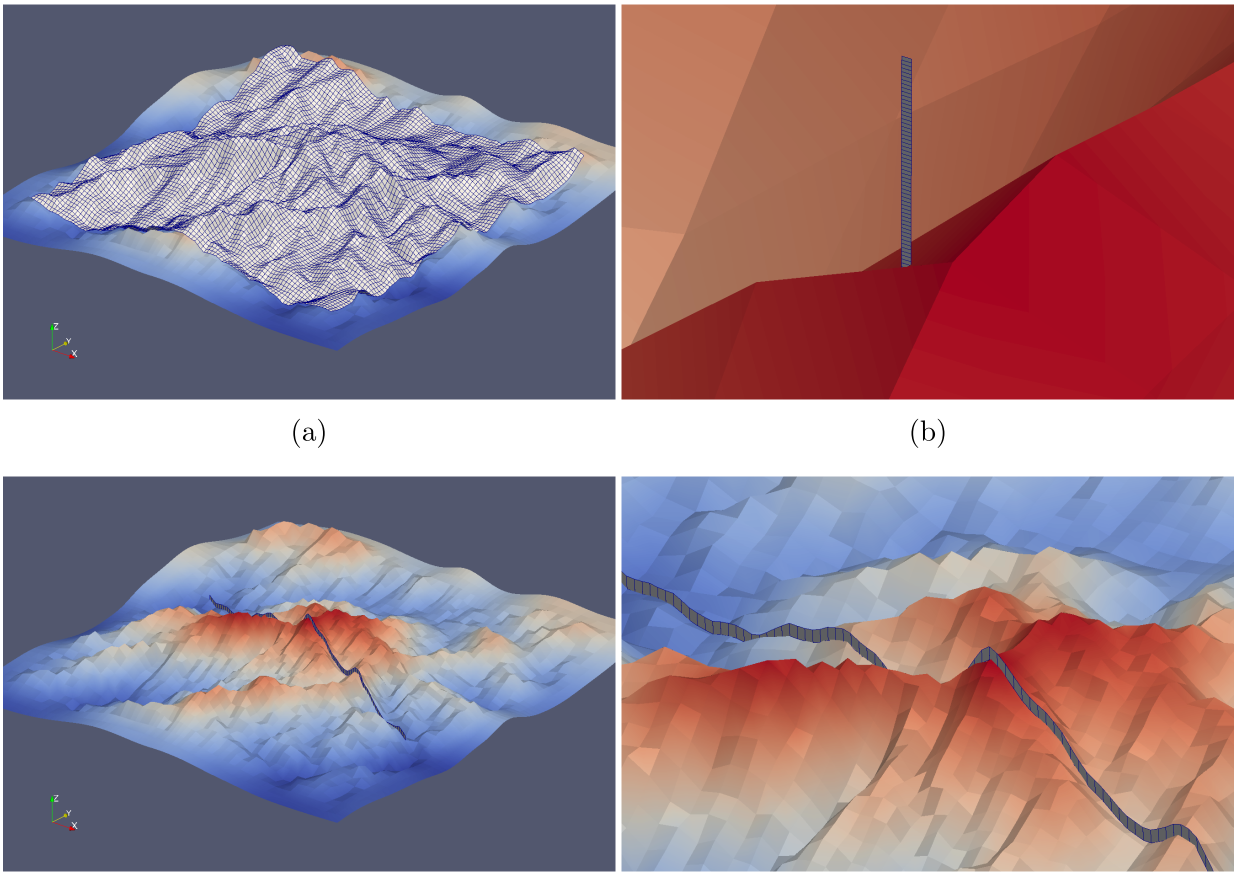
   This vertical line is defined by three parameters: (i) its xy position in computational domain, (ii) the height AGL at profile top, and (iii) the vector dimension, in which points will be evenly distributed between ground level at xy defined position and profile top.

For coordinates generation, *write_sample* first acquires *ground* patch geometric data as described in Section 3.3.3 for *write_z0* and matches each of the xy pair of the generated coordinates with a face belonging to *ground* patch that comprises the condition

$$(x_i, y_i) \in f_j : x_i > \min(x_{j,n}) \wedge x_i < \max(x_{j,n}) \wedge y_i > \min(y_{j,n}) \wedge y_i < \max(y_{j,n}), \tag{3.10}$$

in which (x_i, y_i) is the generated xy coordinate pair, f_j the matched face and $(x_{j,n}, y_{j,n})$ each of the four pair of coordinates that define a face. Ground level z component of each generated coordinate is determined by a bilinear interpolation from the xyz data of the respective face that the coordinate pair was matched with, adding then the predefined AGL value to each of the calculated z components. Coordinates sets are saved in a raw ASCII file and read by OpenFOAM *sample* utility via an `#include` instruction set up in `sampleDict` file as described in Section B.3.

For visual inspection of each of the generated coordinate sets, its xyz data is dumped in three VTK format files. For mode number 2 a surfaced defined by the AGL and ground level lines is created, and for mode number 3 a VTK surface is defined by an offset of the coordinates that describe vertical line in both the x positive and negative directions. Figure 3.9 shows the VTK surfaces generated from coordinates sets concerning each of the three described modes for computational domain of *Serra do Cabeço da Rainha* (see Figure 3.5). In Appendix B complementary information is presented concerning set up of coordinates generation and its procedure.



(a) (b)

(c) (d)

Figure 3.9: Postprocessing coordinates in the VTK format for computational domain of *Serra do Cabeço da Rainha*. (a) Surface positioned 25 m AGL. (b) Vertical profile at domain centre. (c) and (d) Line 25 m AGL rotated 160° from north counter clockwise. *Ground* patch and generated VTK surfaces are magnified 4 times in z direction.

Chapter 4

Simulation of a neutrally stratified ABL flow over the Askervein hill

4.1 Introduction

In this chapter, the simulation of a neutrally stratified atmospheric flow over the *Askervein hill* is performed, applying the preprocessing and postprocessing procedures described in Chapter 3. The *Askervein hill* is located in the west coast of the island of South Uist in the Outer Hebrides of Scotland and field measurement campaigns in the 1980's and the several simulation reports available in literature, some of which already mentioned during this work, make *Askervein* a good starting point for code validation.

Various simulations were conducted in order to evaluate OpenFOAM sensitivity to convection schemes, mesh dimensions, terrain roughness and turbulence models, and the merit of developed tools and procedures in the simulation of atmospheric flows over complex terrain.

4.1.1 Site description

The *Askervein hill* has a nearly elliptical shape in plan view, with a minor axes of approximately 1 km, a major axis of 2 km oriented along a NW−SE line and a maximum height of 116 m above surrounding terrain (126 m above sea level). The gentle slope and its relatively isolated location with a $3 - 4$ km plain topography towards coastline, put *Askervein* on the verge to be classified as a complex terrain.

Technical report of Taylor and Teunissen (1985) concerning a field measurement campaign in 1983 was the base for model calibration and evaluation of gathered results. In this campaign, during the months of September and October, data was collected concerning mean wind speed, turbulent intensities and wind direction, among other. A topographic map of *Askervein hill* is presented in Figure 4.1. Most measuring masts were located along lines A−A and AA−AA, oriented 47^o from North and crossing the hill at positions HT (hill top) and CP (centre point), respectively. Additionally, data was also collected at RS (reference site), a measuring point where wind flow can be considered as undisturbed, located approximately 3 km away from hill major axis in SSW direction, as can be seen in Figure 4.1. Present work focused in runs performed in October 3, when moderate to strong neutrally stratified winds at a dominant 210° direction (just 13° counter clockwise from lines A−A and AA−AA) were observed. Ground cover at the time of field measurement campaign was mostly grass, heather and some flat rocks, making Taylor and Teunissen (1987) to suggest an uniform value of $z_0 = 0.03$ m for aerodynamic roughness length over all the terrain surface.

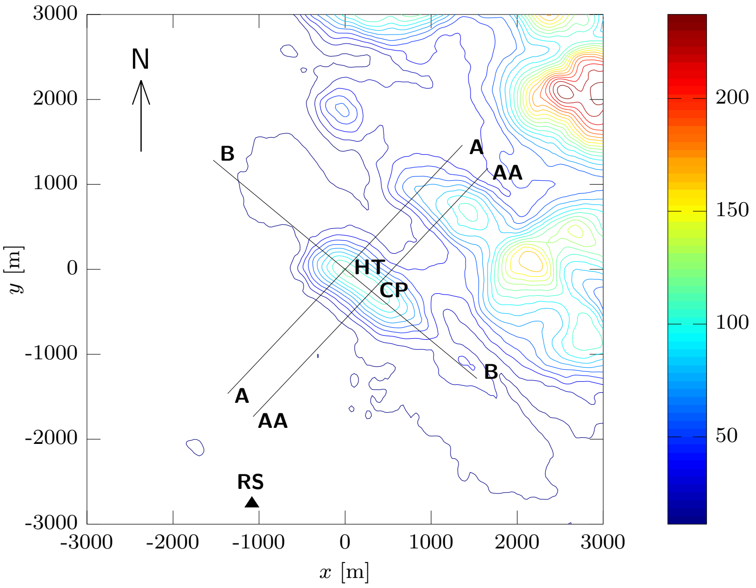


Figure 4.1: *Askervein hill* topographic map centred in HT. Contour plots are coloured according to ground height above sea level.

4.2 Preprocessing

4.2.1 Numerical mesh

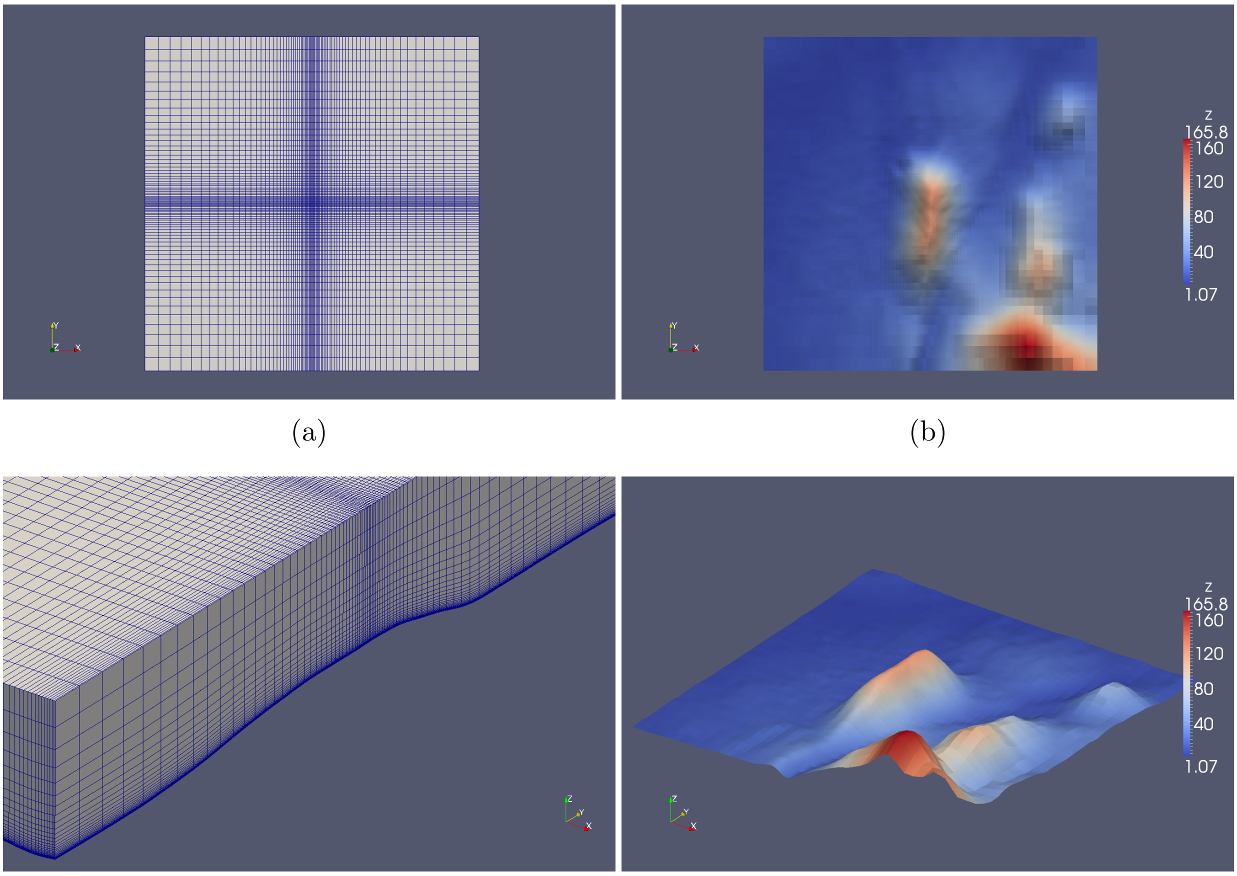
Topographic data for mesh generation was obtained from SRTM (*Shuttle Radar Topography Mission*) elevation model with a resolution of 3 arc-seconds (approximately 90 m), the same data as used for contour plots in Figure 4.1. Gathered elevation data showed a ground height ASL (*Above Sea Level*) in HT of approximately 120 m, 6 m lower when compared to site description in field campaign report.

Mesh dimensions for mesh sensitivity analysis are shown in Table 4.1 and were based in the work of Castro et al. (2003). Topographic data was manipulated with *gsurf* in order to grade ground surface nx and ny dimensions and align *inlet* patch with prevailing wind direction. A computational domain with 4000 × 4000 m^2 and top boundary at 700 m was centred in HT, with a central coordinates distance of 10 m in both x and y directions that expands towards to the edges. Cells dimensions expand at a predefined ratio R in z positive direction in order to yield a height of control volumes in ground surface approximately equal to Δz (see Table 4.1).

Grid	$Nx \times Ny \times Nz$	Total	R	Δz $[m]$
1	$56 \times 56 \times 30$	94080	194.8	0.6
2	$76 \times 76 \times 30$	173280	443.0	0.3
3	$76 \times 76 \times 30$	173280	194.8	0.6
4	$116 \times 116 \times 30$	403680	194.8	0.6
5	$154 \times 154 \times 30$	711480	194.8	0.6

Table 4.1: Grid dimensions for *Askervein hill* computational domain. Ni stand for the number of control volumes in direction i.

Figure 4.2 shows the generated mesh for grid 3 ($76 \times 76 \times 30$) with OpenFOAM *blockMesh* utility recurring to the developed tool *write_blockMeshDict* as described in Section 3.3.1. *Ground* patch is coloured according to the scalar z, the average height ASL of each of the four points that define a face. For all grids, the generated mesh was checked with the OpenFOAM *checkMesh* utility which returned no errors or warnings.



(a) (b)

(c) (d)

Figure 4.2: Computational domain of *Askervein hill* for grid 3 ($76 \times 76 \times 30$). (a) Top view of *sky* patch. (b) Top view of *ground* patch. (c) Cross section plane at domain centre. (d) Isometric view of *ground* patch. *Ground* patch is magnified 5 times in z direction. Direction of wind flow is aligned with x positive.

4.2.2 Boundary conditions

Boundary conditions at computational domain *inlet* were tuned with experimental data registered during run TU-03B conducted between 1400−1700 BST in October 3. Mean horizontal wind speed $\bar{V}_h$ and standard deviations σ_i used for model calibration are shown in Table 4.2.

Location	$\boldsymbol{z}$ **[m]**	$\boldsymbol{\bar{V}_h}$ **[m/s]**	$\boldsymbol{\sigma_u}$ **[m/s]**	$\boldsymbol{\sigma_v}$ **[m/s]**	$\boldsymbol{\sigma_w}$ **[m/s]**
RS	10	8.6	1.223	0.704	0.413

Table 4.2: *Askervein hill* experimental data used for model calibration. Vertical gill UVW anemometer placed at RS 10 m AGL during run TU-03B.

Friction velocity at *inlet* patch was determined solving Eq. 3.1 in order to u_*, taking $u = 8.6$ m/s, $z = 10$ m, $z_0 = 0.03$ m and $\kappa = 0.41$. Calibrated $k-\varepsilon$ turbulence model formulation was tuned calculating k in the form

$$k = \frac{1}{2}\left(\sigma_u^2 + \sigma_v^2 + \sigma_w^2\right) \tag{4.1}$$

and solving Eq. 3.4 for C_μ.

Figure 4.3 shows the non-dimensional horizontal velocity and turbulent kinetic energy profiles at domain inlet following the boundary conditions in Castro et al. (2010) with the calculated values of $u_* = 0.607$ m/s and $C_\mu = 0.116$ and a prescribed ABL height $\delta = 1000$ m. These profiles were compared with field measured values at RS during run TU-03B. Computed horizontal velocity profile closely matched all experimental values until a height AGL of $z \approx 50$ m and k^* presented a maximum deviation of 40% when compared to sonic anemometers at $z \approx 10$ m and a good agreement with remaining measures at higher heights, matching the measured reference value for model calibration at $z = 10$ m (vertical gill UVW). Magnitude of presented deviations between computed and field measured values were considered to be in line with reviewed sources in literature [see e.g. Kim and Patel (2000)].

Relative importance of Coriolis effects was evaluated as in Section 2.1.5. With a characteristic wind speed of 10 m/s, a domain length of 4000 m, an angular velocity of earth rotation equal to 7.29×10^{-5} rad/s [Kantha and Clayson (2000)] and a latitude of 54.18° N, Rossby number yields approximately equal to 20, leading to conclude that, although inertial forces dominate, Coriolis force still has a significant presence in physical domain. Being $z = 10$ m, the predominant height for the evaluation of results in present work, estimated to be inside or near the *inner layer* [Walmsley and Taylor (1996)], flow can be considered to be dominated by *surface layer* dynamics, where Coriolis are small when compared to the other terms of the momentum equation.

Pressure field boundary conditions were previously defined in `p` file according to Table 3.1 and OpenFOAM `U`, `k`, `epsilon` and `omega` files generated recurring to the

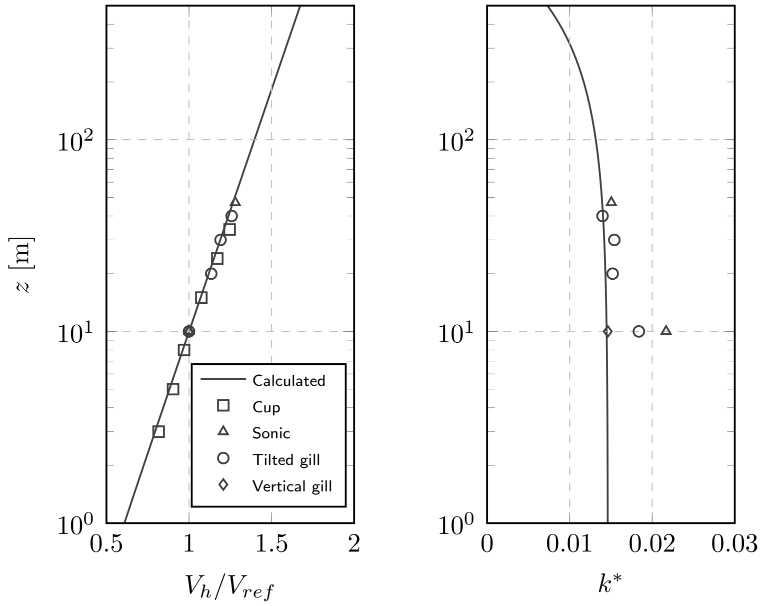


Figure 4.3: Calculated vertical profiles at domain inlet and experimental data measured at RS. V_{ref} is the measured horizontal velocity 10 m AGL. $k^* = k/V_{ref}^2$. Calculated profiles with $u_* = 0.607$ m/s, $\delta = 1000$ m, $C_\mu = 0.116$, $z_0 = 0.03$ m and $\kappa = 0.41$.

developed tool ***write_bCs*** as described in Section 3.3.2 and following the boundary conditions in Table 3.2. Although combination of friction velocity and grid dimensions yield an estimated local Reynolds number z^+ far beyond the commonly suggested range of validity of the mentioned wall functions [see e.g. Menter et al. (2003a) and Versteeg and Malalasekera (2007)], being first node above ground surface in the log region of the turbulent boundary layer, wall functions were therefore considered to be fully applicable.

4.2.3 Roughness mapping

Ground surface aerodynamic roughness length was modelled in two different modes:

1. Considering an uniform value of $z_0 = 0.03$ m along all terrain surface, as proposed in Taylor and Teunissen (1987).
2. According to the non-uniform roughness mode mentioned in Castro et al. (2003), in which roughness decreases linearly from $z_0 = 0.03$ m at $z = 60$ m ASL, to a minimum value of 0.01 m in HT, approximately 120 m ASL in gathered elevation data.

Figure 4.4 shows the generated roughness map for non-uniform roughness mode. Roughness data and turbulent kinematic viscosity boundary conditions were defined in `nut` file, recurring to the developed tool ***write_z0*** as described in Section 3.3.3.

4.2.4 Turbulence models

Simulations were conducted for all turbulence models and formulations mentioned in Table 3.4. For the Calibrated $k - \varepsilon$ formulation, σ_ε was adjusted according to Eq. 3.9,

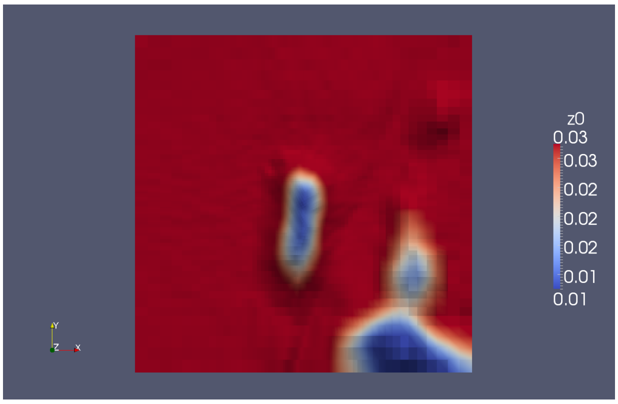


Figure 4.4: Roughness map for *Askervein hill*. Equation based roughness mapping mode ($z_0 = 0.03$ m for $z \leq 60$ m; linear decay in the form $0.03 > z_0 > 0.01$ m for $60 < z \leq 120$ m; $z_0 = 0.01$ m for $z > 120$ m). *Ground* patch is magnified 5 times in z direction.

yielding the coefficients set shown in Table 4.3.

`RASProperties` files for the bulk of the simulations were generated with *write_turbulenceProperties* tool as described in Section 3.3.4.

C_μ	σ_k	σ_ε	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$
0.116	1.00	1.03	1.44	1.92

Table 4.3: Coefficients set for the Calibrated $k - \varepsilon$ model, tuned with field measured data.

4.2.5 Convection schemes

In this work, three interpolation schemes for the determination of convective fluxes at the control volumes faces were tested: (i) *upwind*, (ii) *linear* and (iii) QUICK (*Quadratic Upwind Interpolation for Convective Kinematics*), which according to Ferziger and Perić (2002) are respectively, and per definition, first, second, and third order accurate in terms of a Taylor series truncation error.

In what concerns to QUICK scheme, OpenFOAM has two different formulations: (i) a standard QUICK and (ii) QUICKV, a vectorial form of the standard scheme, both with a limiting function to assure that the interpolated value is between the neighbour upstream and downstream values. Preliminary tests with Atmospheric $k - \varepsilon$ showed a much higher efficiency of the QUICKV formulation, with the standard QUICK presenting approximately twice the initial residual for the velocity field after 200 iterations and a continuous oscillating residual for pressure field, without a notorious diminishing of initial residual when compared to starting iterations.

The test of *linear* and QUICK convection schemes for the flux of scalars k and ε revealed also some instability issues, observing continuous negative values on both k and

ε. This phenomenon was attributed to a numerical instability intrinsic to the *linear* scheme formulation and to an inability of the limiting function built in OpenFOAM QUICK scheme to avoid some undershoot phenomenons due to its quadratic formulation, as described in Versteeg and Malalasekera (2007).

For the presented simulations, convection schemes for scalars k, ε and ω were set to *upwind* and for velocity field U defined as in Table 4.4. These schemes were assigned in `divSchemes` field in file `fvSchemes`.

4.3 Processing

The bulk of the performed simulations are described in Table 4.4, including the iterations number for the accomplishment of convergence criteria and the respective CPU (*Central Processing Unit*) time. All simulations ran on a mainstream laptop with an `Intel Core i5 M430 @ 2.27 GHz` CPU without parallelization.

The OpenFOAM solver *simpleFoam* was used, which according to OpenFOAM User Guide (2012) is indicated for steady-state and incompressible turbulent flows. Convergence criteria for initial residual was defined in `fvSolution` file and set equal to 1×10^{-2} for pressure field and 1×10^{-3} for the remaining quantities, the standard values used in OpenFOAM tutorial cases. In what concerns to linear solvers, two different formulations were used: GAMG (*Generalised Geometric-Algebraic Multi-Grid*) for pressure field and PBiCG (*Preconditioned Bi-Conjugate Gradient*) for remaining fields, both with a tolerance between iterations equal to 1×10^{-5}.

4.4 Presentation and discussion of results

In this Section are presented the simulations results and the outcome of a sensitivity analysis concerning convection schemes, grid dimensions, ground roughness and turbulence models.

Results were evaluated in terms of non-dimensional speed-up ratio defined by

$$\Delta \mathrm{S} = \frac{V_h(z) - V_{ref}(z)}{V_{ref}(z)}, \tag{4.2}$$

where $V_h(z)$ and $V_{ref}(z)$ are respectively, the horizontal velocity at the point of study and at RS, and k^*, the non-dimensional TKE defined by

$$k^* = \frac{k}{V_{ref}^2}. \tag{4.3}$$

Data extraction for postprocessing was performed with the developed tool *write_sample* as described in Section 3.4.

Case	Grid	Model	Roughness	Scheme	Iterations	CPU
1	3	Calibrated $k-\varepsilon$	U	upwind	166	324
2	3	Calibrated $k-\varepsilon$	U	linear	284	510
3	3	Calibrated $k-\varepsilon$	U	QUICKV	190	356
4	3	Standard $k-\varepsilon$	U	upwind	185	373
5	3	Standard $k-\varepsilon$	U	linear	279	495
6	3	Standard $k-\varepsilon$	U	QUICKV	192	361
7	3	Atmospheric $k-\varepsilon$	U	upwind	168	352
8	3	Atmospheric $k-\varepsilon$	U	linear	309	596
9	3	Atmospheric $k-\varepsilon$	U	QUICKV	182	352
10	3	Calibrated $k-\varepsilon$	N	upwind	168	338
11	3	Calibrated $k-\varepsilon$	N	linear	285	502
12	3	Calibrated $k-\varepsilon$	N	QUICKV	191	362
13	3	Standard $k-\varepsilon$	N	upwind	187	376
14	3	Standard $k-\varepsilon$	N	linear	281	501
15	3	Standard $k-\varepsilon$	N	QUICKV	194	368
16	3	Atmospheric $k-\varepsilon$	N	upwind	170	340
17	3	Atmospheric $k-\varepsilon$	N	linear	310	576
18	3	Atmospheric $k-\varepsilon$	N	QUICKV	184	361
19	3	Original RNG $k-\varepsilon$	U	QUICKV	175	336
20	3	Modified RNG $k-\varepsilon$	U	QUICKV	175	328
21	3	Atmospheric RNG $k-\varepsilon$	U	QUICKV	175	332
22	3	*Realisable* $k-\varepsilon$	U	QUICKV	492	989
23	3	$k-\omega$	U	QUICKV	191	381
24	3	SST $k-\omega$	U	QUICKV	561	1096
25	1	Atmospheric $k-\varepsilon$	U	upwind	192	205
26	1	Atmospheric $k-\varepsilon$	U	linear	336	331
27	1	Atmospheric $k-\varepsilon$	U	QUICKV	239	238
28	2	Atmospheric $k-\varepsilon$	U	upwind	174	344
29	2	Atmospheric $k-\varepsilon$	U	linear	293	519
30	2	Atmospheric $k-\varepsilon$	U	QUICKV	178	342
31	4	Atmospheric $k-\varepsilon$	U	upwind	167	773
32	4	Atmospheric $k-\varepsilon$	U	linear	255	1115
33	4	Atmospheric $k-\varepsilon$	U	QUICKV	213	962
34	5	Atmospheric $k-\varepsilon$	U	upwind	169	1388
35	5	Atmospheric $k-\varepsilon$	U	linear	255	2112
36	5	Atmospheric $k-\varepsilon$	U	QUICKV	248	2175

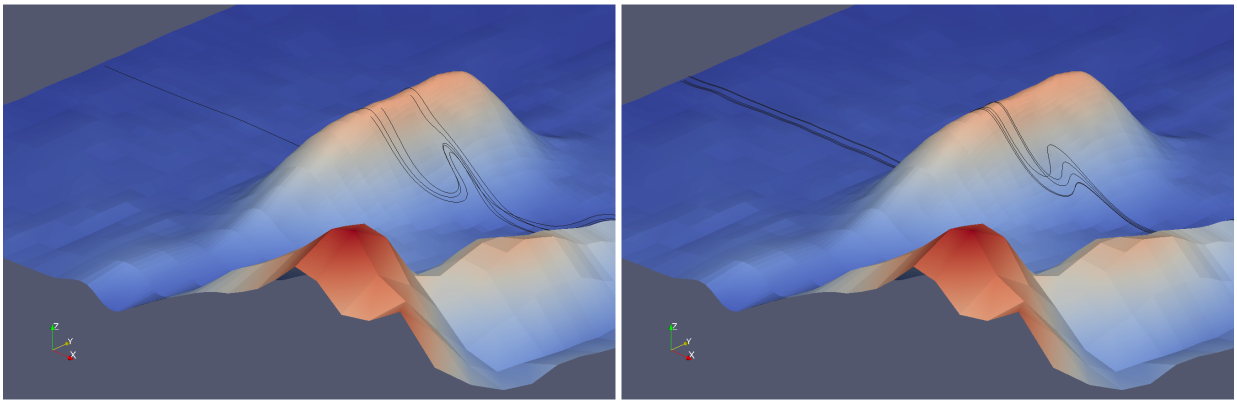
Table 4.4: Performed simulations for the *Askervein hill*. CPU time is given in seconds. U and N stand respectively for Uniform and Non-uniform surface roughness modes. Convection schemes concern exclusively to velocity field U.

4.4.1 Convection schemes

Figure 4.6 shows the results for ΔS along line A−A 10 m AGL in Cases 4 to 9, 25 to 27 and 34 to 36. The purpose of the selected cases are the evaluation of the behaviour of the different convection schemes in two different $k - \varepsilon$ turbulence model constant sets and three different grids dimensions.

Presented results showed a good agreement between experimental and calculated values for all cases in hill windward side and near hill top, with a ΔS error in HT varying from −6.7% with *linear* scheme in Case 35 to −11.4% with *upwind* scheme in Case 4. Best agreement for QUICKV scheme was found in Case 36 with an error equal to −7.3%. The lower computed speed-up in HT when compared to experimental values, observed in all simulations, can be partially explained by the probable too lower height of gathered topographic data in that point, as mentioned in Section 4.1.1.

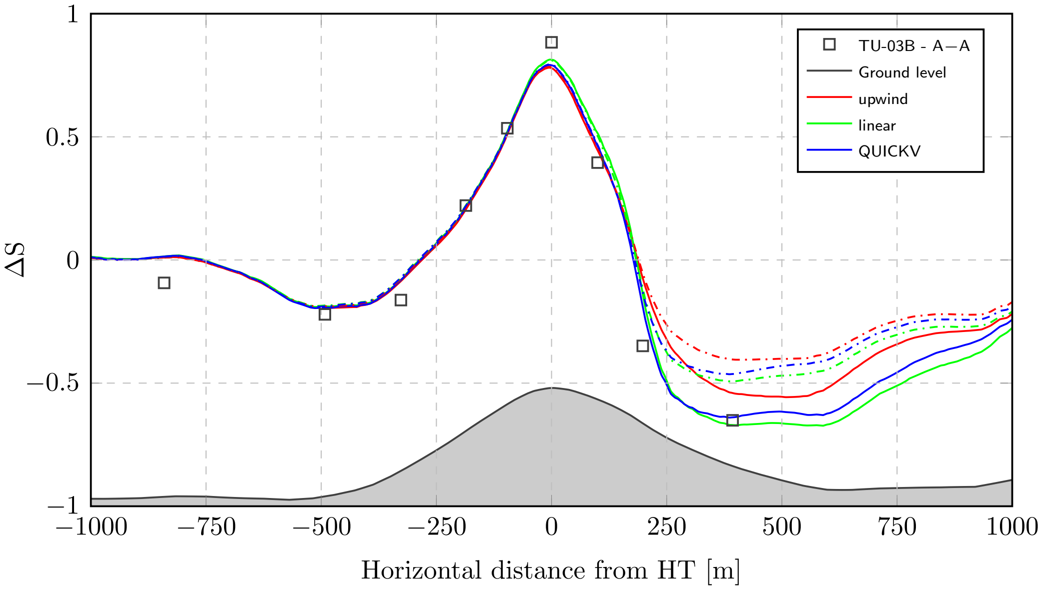
A much higher dependence of the results on case set up variables was found on the hill lee side. In what concerns to interpolation schemes, *linear* and QUICKV showed always a better agreement of ΔS when compared to *upwind*, allowing also to identify a recirculation zone in grids 1, 4 and 5 with Atmospheric $k - \varepsilon$ constants set. This zone, located between HT and CP and approximately 300 m from line B−B towards NE, has been reported in both field campaign [Taylor and Teunissen (1985)] and flow calculations in the reviewed literature [e.g. Kim et al. (2000) and Lopes et al. (2007)]. Its extent was found to be similar for both *linear* and QUICKV schemes and dependent on other variables like the grid dimensions. Figure 4.5 shows a capture of streamlines in the hill lee side near ground level for two grid sizes. A plot of velocity vector glyphs (figures not shown) revealed a three dimensional behaviour of this recirculation zone and a height varying from 1 to 3 m AGL.



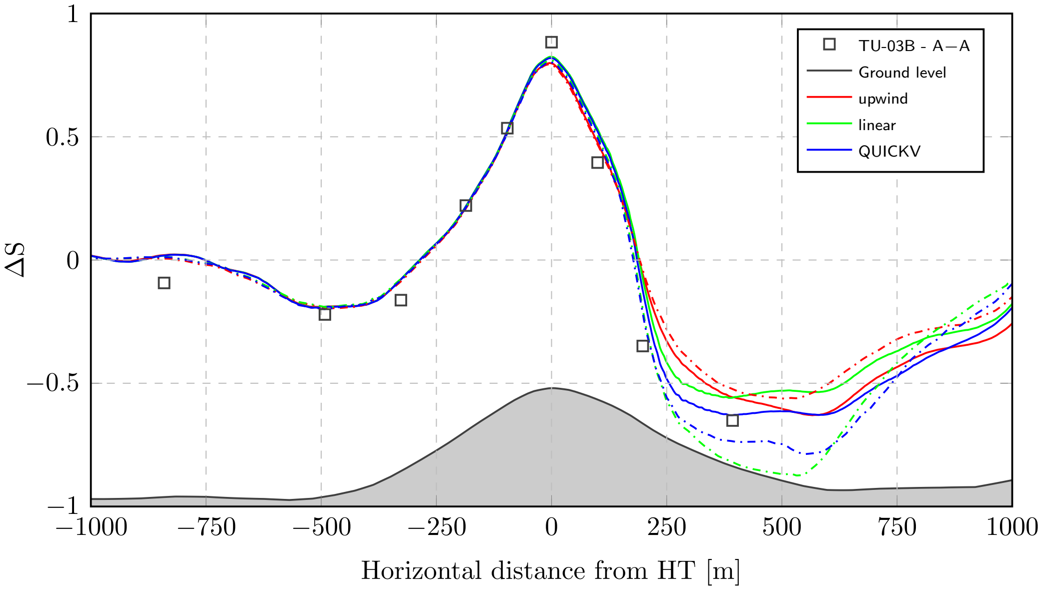
(a) (b)

Figure 4.5: Recirculation zone in *Askervein hill* lee side with Atmospheric $k - \varepsilon$ model and QUICKV scheme. (a) Grid 1: $56 \times 56 \times 30$ (b) Grid 5: $154 \times 154 \times 30$. View from the hill lee side. *Ground* patch and streamlines are magnified 5 times in z direction.

The poorer agreement of the *upwind* scheme, mainly on the hill lee side, was attributed not only to its smaller order of accuracy, but also to the absence of a downstream numerical component on its formulation. The boundary layer flow in the hill



(a)



(b)

Figure 4.6: Speed-up ratio (ΔS) 10 m AGL along line A—A with three different interpolation schemes. (a) Standard $k-\varepsilon$ model (– · –) and Atmospheric $k-\varepsilon$ model (——) with grid 3 ($76 \times 76 \times 30$) (cases 4 to 9). (b) Grid 1 ($56 \times 56 \times 30$) (– · –) (cases 25 to 27) and grid 5 ($154 \times 154 \times 30$) (——) (cases 34 to 36) with Atmospheric $k-\varepsilon$ model.

windward side, being mathematically classified as a parabolic flow [Ferziger and Perić (2002)], is mainly affected by what happens upstream, and therefore not sensible to the lack of a numerical downstream component in *upwind*, giving rise to similar results for all interpolation schemes. On the other hand, the more elliptic behaviour of the flow in the lee side, with the need of information to travel from downstream to upstream, make

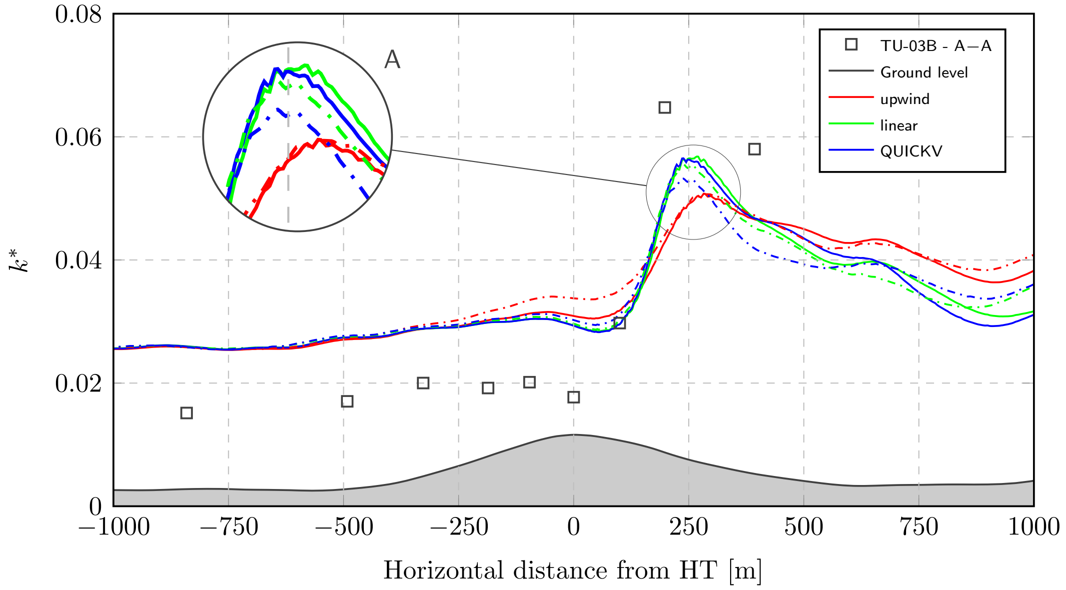


Figure 4.7: Non-dimensional TKE (k^*) 10 m AGL along line A—A with three different convection schemes for velocity field U and Atmospheric $k-\varepsilon$ model. Grid 3 ($76\times76\times30$) (– · –) and grid 5 ($154\times154\times30$) (——).

upwind to give higher and inaccurate flow speeds, avoiding also to capture an inversion in flow direction.

The *upwind* scheme behaviour found in lee side has proven to be similar in both Standard and Atmospheric $k-\varepsilon$ constant sets, but improved by the use of finer grids in lieu of coarser grids, a phenomenon reported in Versteeg and Malalasekera (2007). Grid refinement revealed also an increase of ΔS with *linear* and QUICKV schemes, and a decrease with *upwind* [see Figure 4.6 (b)].

The inability of *upwind* scheme to describe peaks and fast changes in the flow is also shown in Figure 4.7, concerning the calculated non-dimensional TKE 10 m AGL along line A—A for two different grids and Atmospheric $k-\varepsilon$ model. At the lee side of the hill, *upwind* revealed again to be less responsive to changes in flow when compared to other schemes, noticed mostly on the coarser grid. The agreement between calculated and experimental TKE was found to be more dependent on other variables than interpolation schemes and will therefore be addressed in Sections 4.4.3 and 4.4.4.

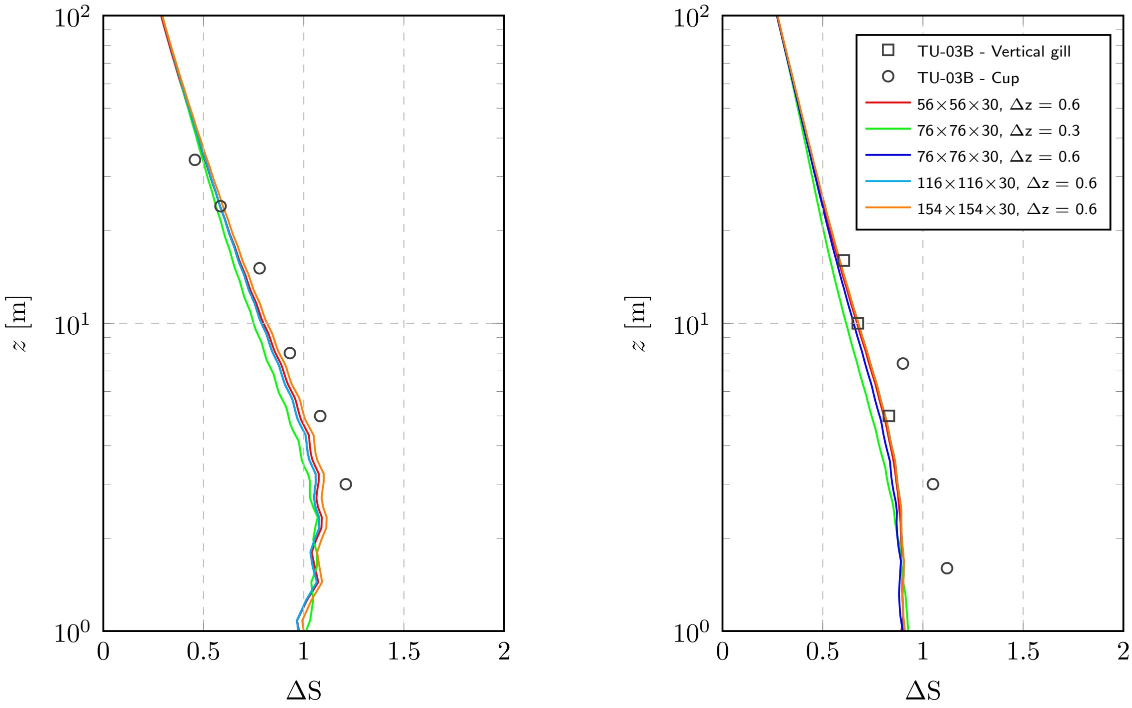
Detail A in Figure 4.7 shows a ripple in k^* plot, observed also in the remaining simulations for both k^* and ΔS between HT and 250 m towards hill downstream 10 m AGL and bellow, and whose origin was not possible to identify.

The higher iterations number of *linear* scheme in all simulations was explained by its oscillatory behaviour [Ferziger and Perić (2002)], causing the need of more iterations for the accomplishment of predefined convergence criteria. To QUICKV was attributed the best balance between accuracy and computational effort.

4.4.2 Grid dimensions

For the evaluation of the importance of grid dimensions, simulations were performed for each of the grids in Table 4.1 with Atmospheric $k-\varepsilon$ model and QUICKV.

Figure 4.8 shows vertical profiles of speed-up ratio in HT and CP', being CP' located approximately 28 m from CP towards North. Calculated results were considered to agree well with cup anemometers in HT and nearly overlap with vertical gill anemometers in CP', with cup anemometers in this point presenting a measured speed approximately 10% higher when compared to vertical gill, as noticed in field campaign report. Horizontal grid dimensions revealed to have a minor importance in these results, with its slight differences vanishing at $z \approx 25$ m AGL, a height already above the *inner layer* where flow starts to loose its dependence on surface characteristics. Reducing Δz lead to an increase of ΔS near the ground surface and a decrease above, making this numerical result globally less interesting, as can be seen in Figure 4.8 for grid 2 ($76 \times 76 \times 30$ with $\Delta z = 0.3$ m). The ripple mentioned in Section 4.4.1 can also be observed in Figure 4.8 (a) for $z \approx 10$ m and bellow.

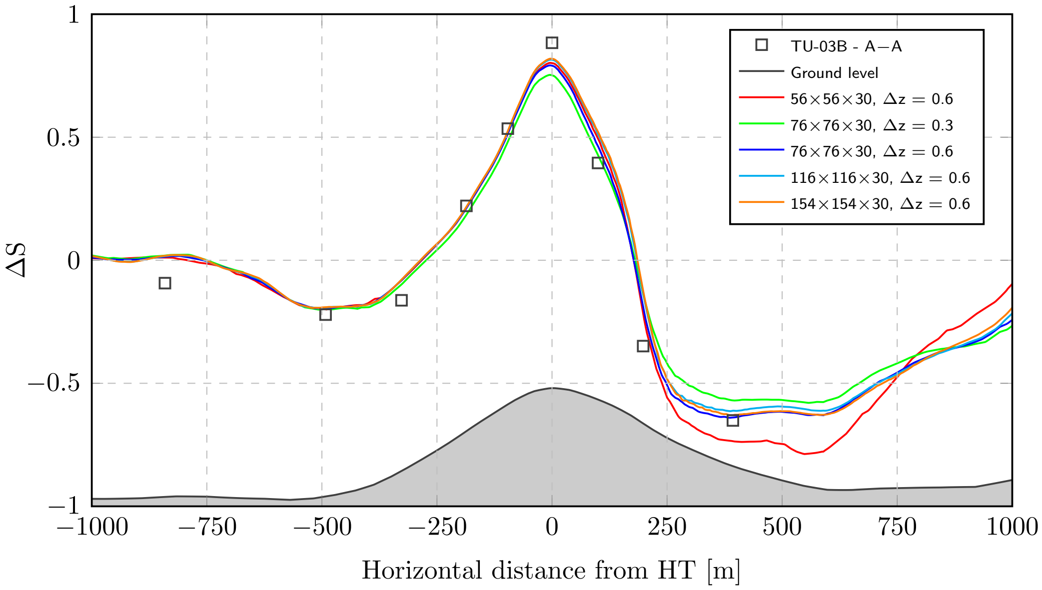


(a)

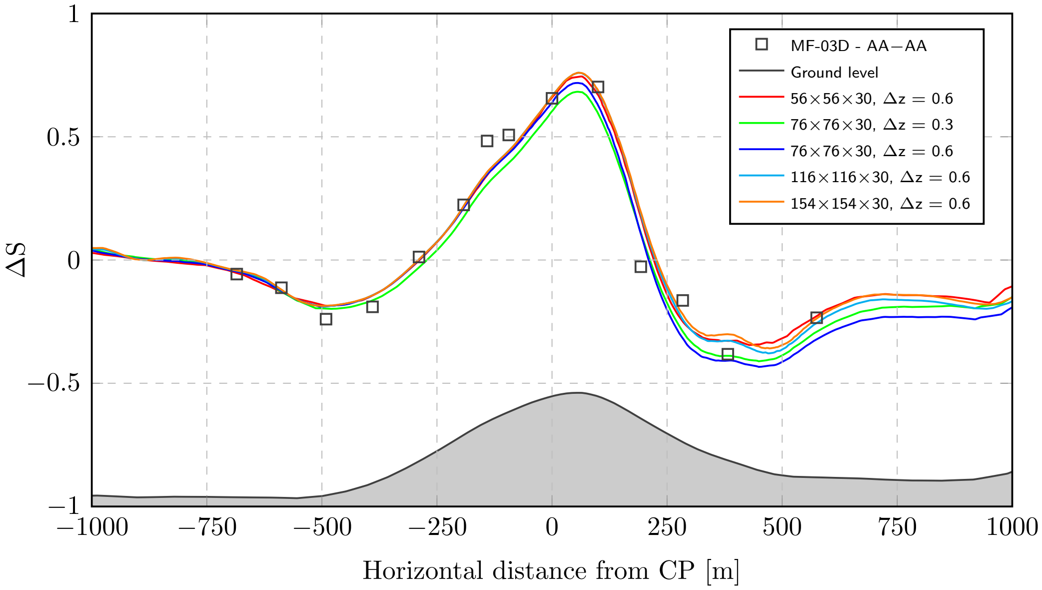
(b)

Figure 4.8: Vertical profiles of speed-up ratio (ΔS) in (a) HT and (b) CP' with five different grid dimensions. Atmospheric $k-\varepsilon$ model and QUICKV scheme.

The plot of calculated ΔS along lines A–A and AA–AA 10 m AGL for cases 9, 27, 30, 33 and 36 (grids 1 to 5) is shown in Figure 4.9. Error between calculated and experimental ΔS in HT [Figure 4.9 (a)] varied from -7.3% in grid 5 ($154 \times 154 \times 30$) to -15.0% in grid 2 ($76 \times 76 \times 30$ with $\Delta z = 0.3$ m). Greatest error in HT for grids with



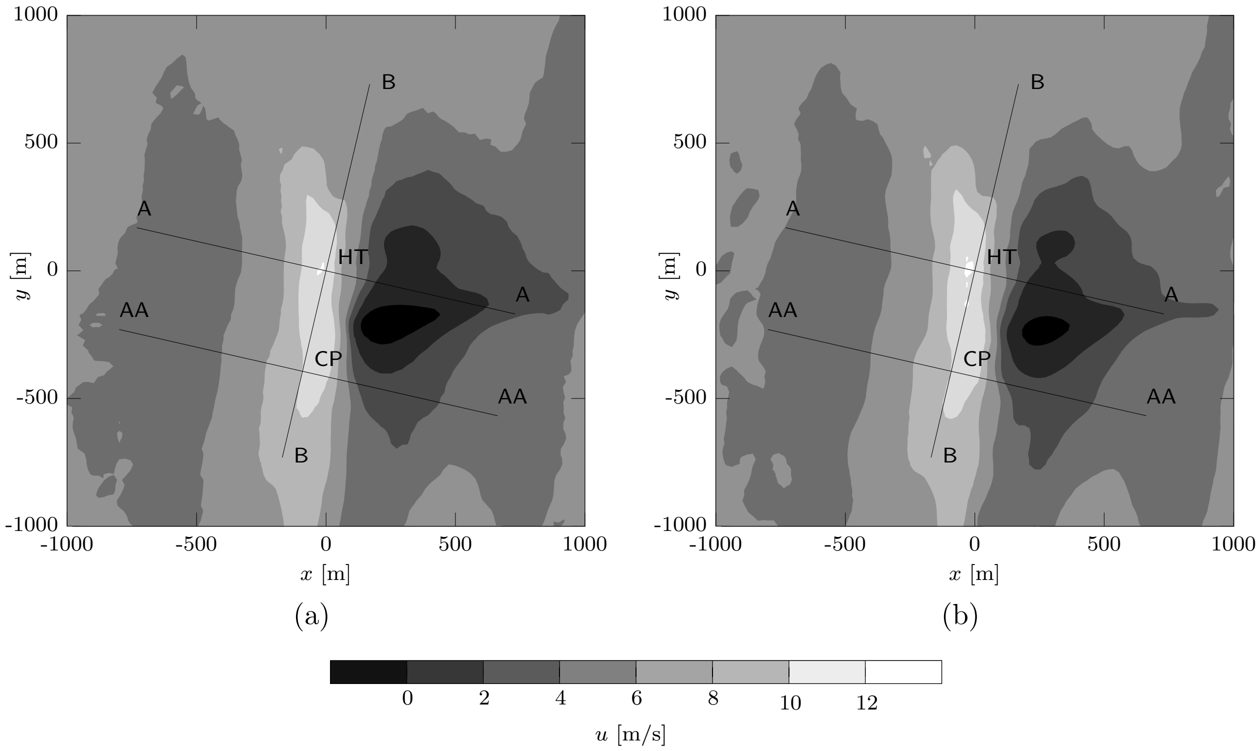
(a)



(b)

Figure 4.9: Speed-up ratio (ΔS) 10 m AGL along lines (a) A−A and (b) AA−AA with five different grid dimensions. Atmospheric $k - \varepsilon$ model and QUICKV scheme.

$\Delta z = 0.6$ m was found on grid 3 ($76 \times 76 \times 30$) and equal to −10.6%. In CP [Figure 4.9 (b)], calculated speed-up values closely matched experimental data, being the largest error equal to −8.0% and also found in grid 2. Lack of agreement between experimental and calculated values near 125 m and 250 m, respectively, upstream and downstream CP along line AA−AA, were attributed to a too low resolution of elevation model in gathered topographic data [Figure 4.9 (b)].



(a)

(b)

Figure 4.10: Recirculation zone in *Askervein hill* lee side with Atmospheric $k-\varepsilon$ model and QUICKV scheme. Contour fill plots of streamwise velocity 2 m AGL. (a) Grid 1 ($56 \times 56 \times 30$). (b) Grid 4 ($116 \times 116 \times 30$).

Although calculated results were considered to agree well with experimental values, was not possible to identify a trend in horizontal grid refinement. In what concerns to vertical discretisation, grid refinement revealed to induce an increase in speed-up downstream hill crest and a decrease in the upstream, partially agreeing with Castro et al. (2003) who reported this same behaviour along line A–A downstream hill top.

Deviation of ΔS plot along line A–A in grid 1 ($56 \times 56 \times 30$), starting at 250 m from hill downstream, was explained by the larger extent of the recirculation zone achieved with this grid and to its higher proximity to line A–A when compared to line AA–AA, as shown in Figure 4.10. Extension of this zone in grid 5 was found to be similar to the one achieve with grid 4 and its plot omitted for a sake of simplicity. Absence of flow inversion in grids 2 and 3 was explained as a consequence of the weakness and intermittence of this recirculation zone [Taylor et al. (1987)], making it impossible to capture over the whole extent of the steady state formulations.

4.4.3 Ground roughness

For the evaluation of the importance of aerodynamic roughness length, simulations for the two roughness modes described in Section 4.2.3 with $k-\varepsilon$ model, grid 3 ($76 \times 76 \times 30$) and QUICKV scheme were performed. Standard and Calibrated $k-\varepsilon$ constants sets presented similar sensitivity to aerodynamic roughness length when compared to the Atmospheric one, and its results were therefore omitted in this work.

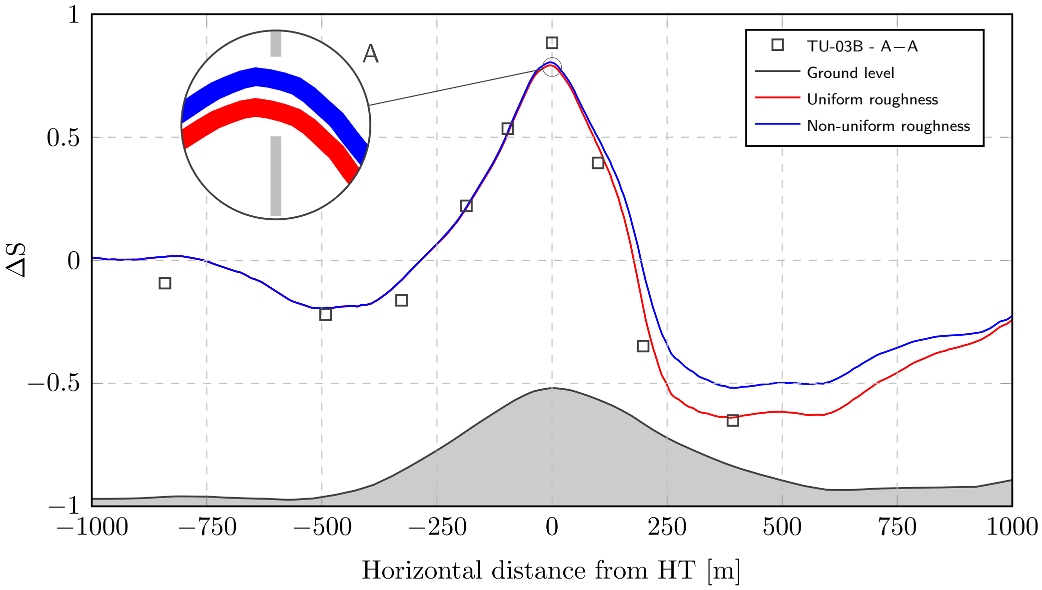


Figure 4.11: Speed-up ratio (ΔS) 10 m AGL along line A—A for Uniform and Non-uniform roughness modes. Atmospheric $k-\varepsilon$ model, QUICKV scheme and grid 3 ($76 \times 76 \times 30$).

Figure 4.11 presents the calculated speed-up ratio 10 m AGL along line A—A for the two roughness modes. For the Non-uniform roughness mode, with $z_0 = 0.01$ m in HT in opposition to $z_0 = 0.03$ m for the Uniform roughness mode, speed-up in HT was found to be equal to 0.80, a value slightly higher then 0.79 as found for Uniform roughness mode, agreeing with the theoretical background in Section 2.1.4, in which the higher the aerodynamic roughness length, the lower the wind speed for the same observed height. In the hill windward, the low dependence of the flow in the diffusive term [Ferziger and Perić (2002)] explained the overlapping of results of both roughness modes. For the lee side, although results were considered to behave as expected taking into account the theoretical prescriptions, the higher and inaccurate calculated ΔS when compared to experimental data lead to conclude that the considered Non-uniform mode may describe better the surface roughness in the hill windward side and hill crest, and worst in the lee side. This conclusion was considered to be reinforced by the prevailing wind direction of 210° that can erode more the canopy at higher heights in the windward and hill crest and much less on the hill lee side.

The vertical profiles of speed-up ratio and non-dimensional TKE in HT for each of the mentioned roughness modes are presented in Figure 4.12. Main conclusion is the accomplishment of the expected behaviour of OpenFOAM ν_t wall function for different aerodynamic roughness lengths. This behaviour agrees respectively, with Eq. 3.1 in which the higher the z_0 the lower the ΔS, and with Eq. 2.5, where there is a direct proportionality between turbulent viscosity and TKE.

The error between calculated and experimental values in HT was reduced for the Non-uniform roughness mode when compared to Uniform roughness mode, noticed more in the measuring points closer to ground surface. ΔS error at $z = 3$ m was reduced from −12.7%

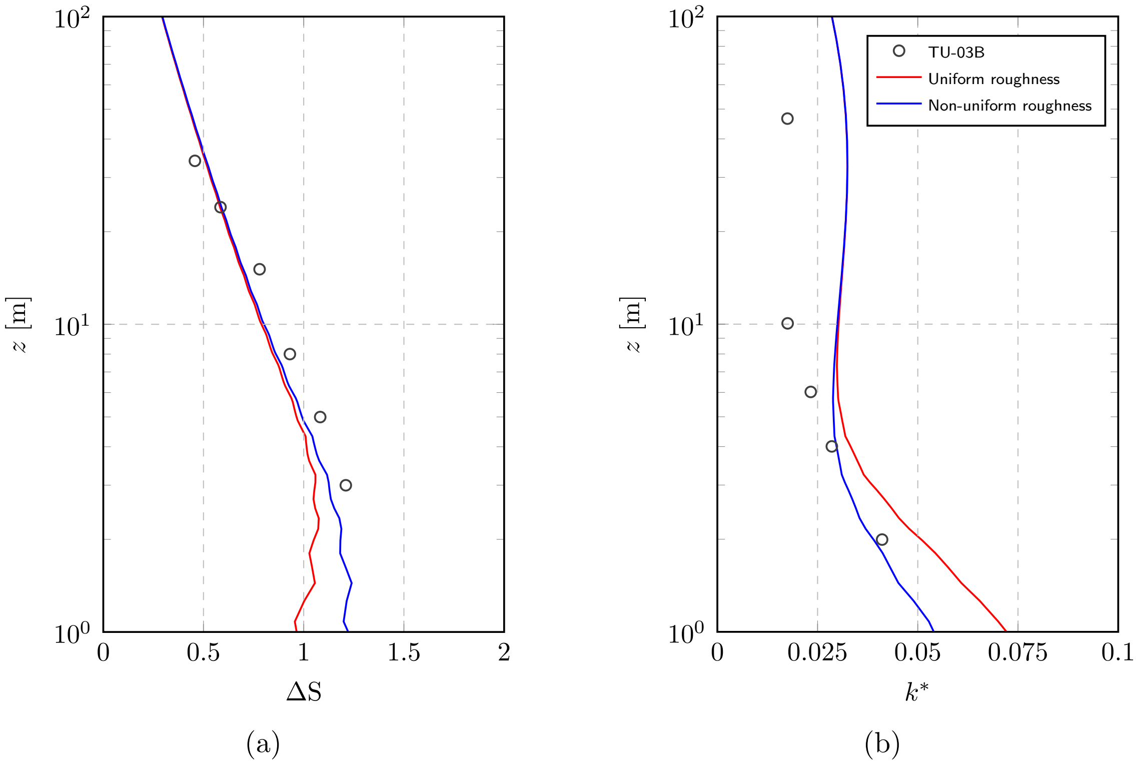


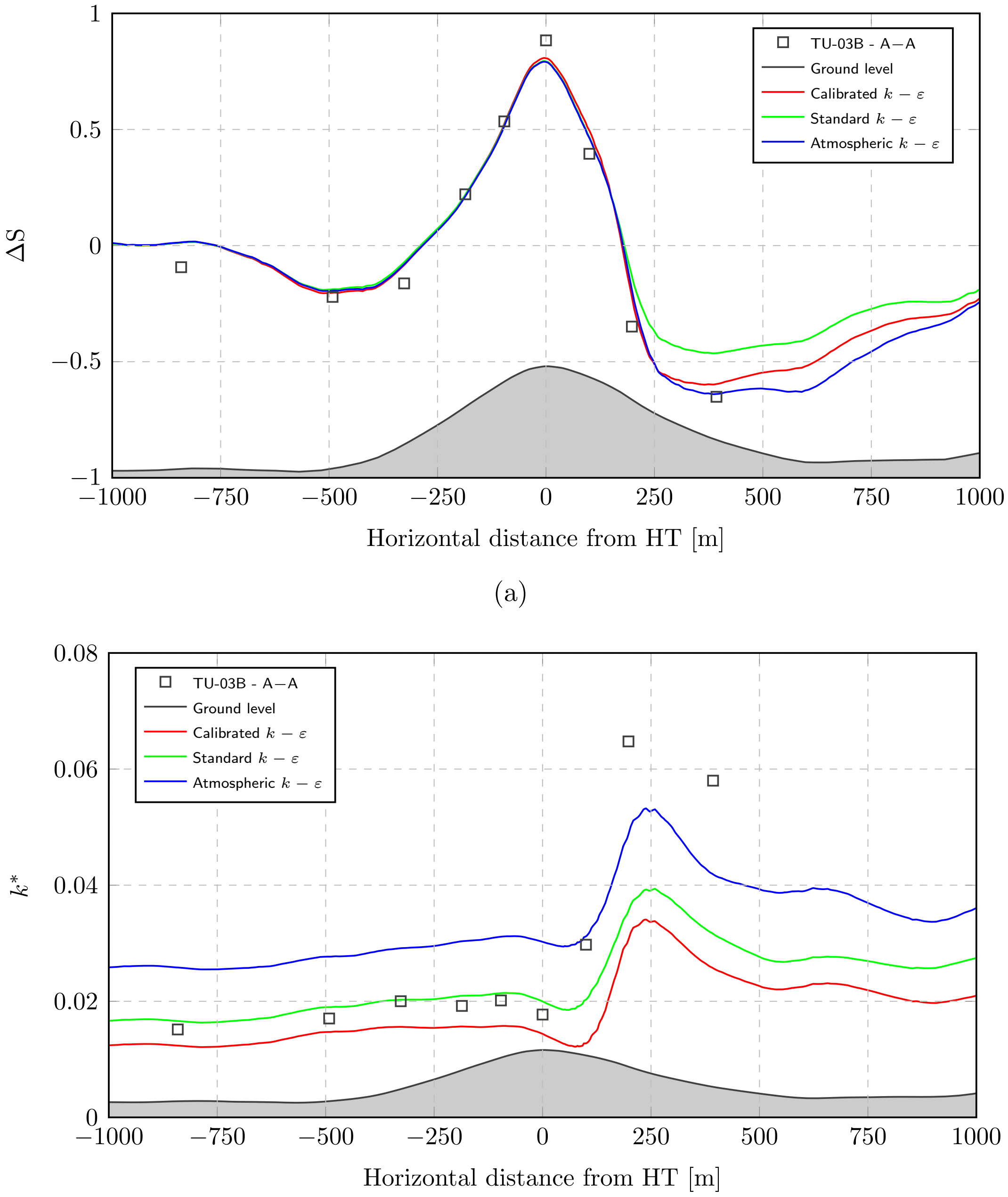
Figure 4.12: Vertical profiles in HT of (a) speed-up ratio (ΔS) and (b) non-dimensional TKE (k^*) for Uniform and Non-uniform roughness modes. Atmospheric $k - \varepsilon$ model, QUICKV scheme and grid 3 ($76 \times 76 \times 30$).

to -6.9% and k^* error at $z = 2$ m reduced from $+27\%$ to -5.2%. Influence of roughness modes in vertical profiles vanished at $z \approx$ 10 m AGL, a height near the *inner layer* limit, as noticed also by Castro et al. (2003) and interpreted here as the calculated height of influence of ground roughness in flow vertical profiles.

4.4.4 Turbulence models

In this Section are presented the results for each of the turbulence models and formulations described in Section 4.2.4, all with QUICKV scheme, grid 3 ($76 \times 76 \times 30$) and Uniform roughness mode. A particular attention was given to the different $k - \varepsilon$ and *RNG* $k - \varepsilon$ constants sets, the two models more explored in the reviewed literature for the simulation of atmospheric flows over complex terrain.

Results presented in first place concern the evaluation of the Standard, Atmospheric and Calibrated constant sets of the $k - \varepsilon$ model. The calculated speed-up ratio and non-dimensional TKE for each of these tunings are plotted in Figure 4.13. In the same manner as the previously presented simulations, influence of the set up variables in ΔS results was noticed mostly on the lee side [see Figure 4.13 (a)]. In this side of the hill, the lower and more accurate ΔS plot was found with the Atmospheric tuning with $C_\mu = 0.033$. The lack of a trend relating C_μ and ΔS in the lee side was interpreted as a possible consequence of the previously identified recirculation zone to be in verge of its formation for these specific

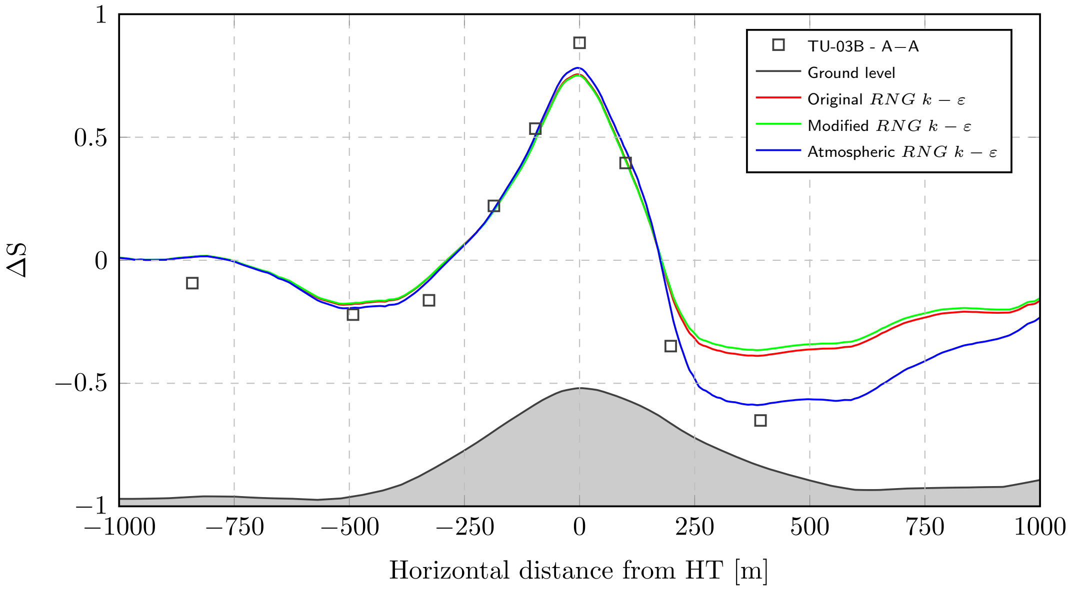


(a)

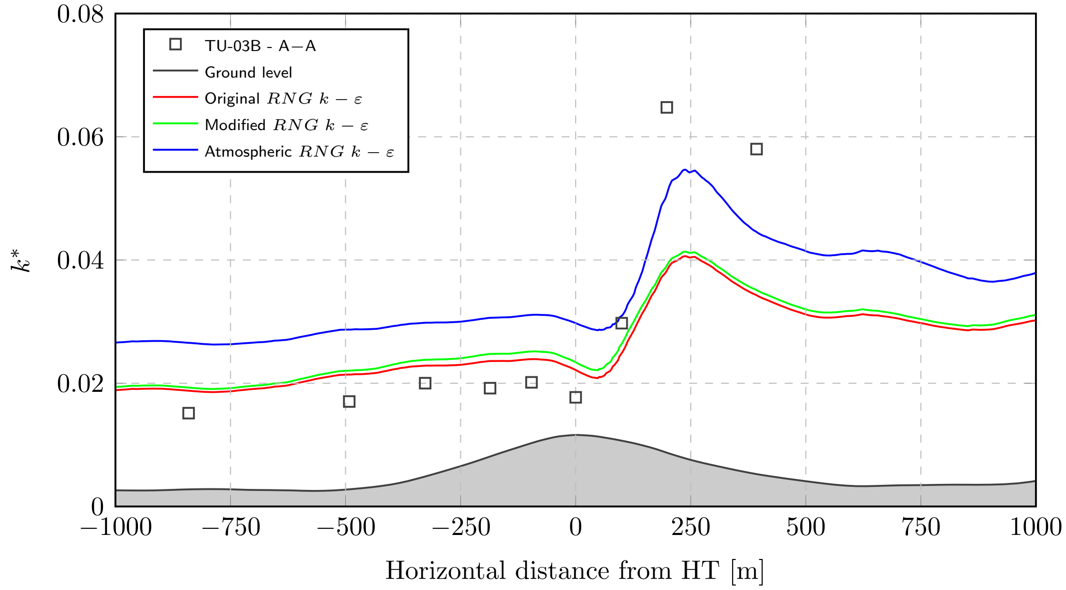
(b)

Figure 4.13: (a) Speed-up ratio (ΔS) and (b) non-dimensional TKE (k^*) 10 m AGL along line A—A with three different $k - \varepsilon$ model constants sets. QUICKV scheme and grid 3 ($76 \times 76 \times 30$).

steady state formulation, influencing mean velocity field 10 m AGL. In HT, the increase of C_μ revealed an increase of ΔS agreement with experimental data, with its error varying from -10.5% for $C_\mu = 0.033$ to -8.7% for $C_\mu = 0.116$. Behaviour of calculated non-dimensional TKE for each of the constants sets agrees with Eq. 3.4, in which the higher the C_μ the lower the k value, with Standard and Calibrated formulations presenting a good agreement with experimental data in the windward side and a poorer agreement in the lee



(a)



(b)

Figure 4.14: (a) Speed-up ratio (ΔS) and (b) non-dimensional TKE (k^*) 10 m AGL along line A—A with three different $RNG\ k-\varepsilon$ model constant sets. QUICKV scheme and grid 3 ($76 \times 76 \times 30$).

side when compared to Atmospheric constant tuning [see Figure 4.13 (b)]. Magnitude of discrepancy between calculated and measured TKE values in the hill lee side was considered to be in line with reviewed bibliographic sources and explained by Kim and Patel (2000) as a possible consequence of the unsteady flow separation, whose influence on the turbulent field is not correctly modelled by the RaNS equations.

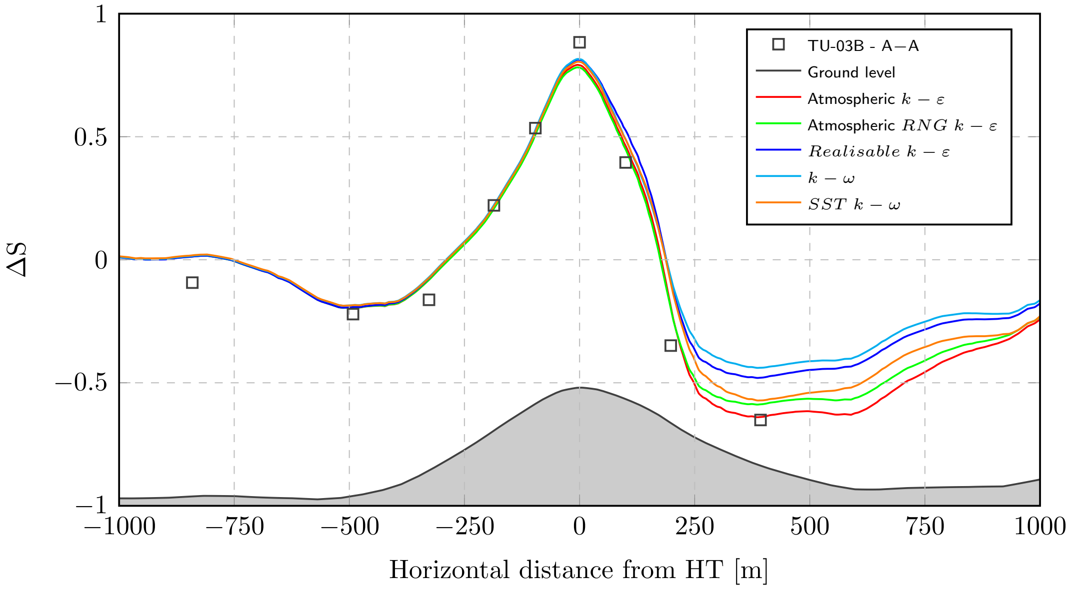
Calculated ΔS and k^* for each of the $RNG\ k-\varepsilon$ model constant sets are presented in Figure 4.14. Once more, influence of model parameters in calculated ΔS is mainly noticed in the hill lee side, with the Atmospheric tuning showing the best agreement between calculated and experimental data. The similar results between Original and Modified constant tunings, with both plots nearly overlapping along line A−A, were attributed to the topography gentle slope [see Speziale and Thangam (1992)]. Oppositely to the $k-\varepsilon$ model, the formulation with the lower C_μ yielded the higher speed-up in HT, with an error equal to −11.7% for the Atmospheric formulation, a value slightly higher than the one found for Atmospheric $k-\varepsilon$. In the overall, $RNG\ k-\varepsilon$ model showed worst results when compared to $k-\varepsilon$, missing this way an agreement with the conclusions of Kasmi and Masson (2010) for the use of $RNG\ k-\varepsilon$ on the simulation of atmospheric flows over complex terrain. In what concerns to k^*, Atmospheric set revealed again an increase in TKE along all line A−A and the worst agreement between calculated and experimental data in the hill windward. Similarly to $k-\varepsilon$ model, non-dimensional TKE was considered to be grossly predicted in the lee side of the hill.

Figure 4.15 shows the comparison of the simulated turbulence models results for calculated ΔS and k^* along line A−A. For the $k-\varepsilon$ and the $RNG\ k-\varepsilon$ models, only the results for the Atmospheric constants sets are presented in the following. The relative error between calculated and experimental data is presented in Table 4.5 for HT and nearly 400 m from HT towards downstream, the furthest mast from hill top.

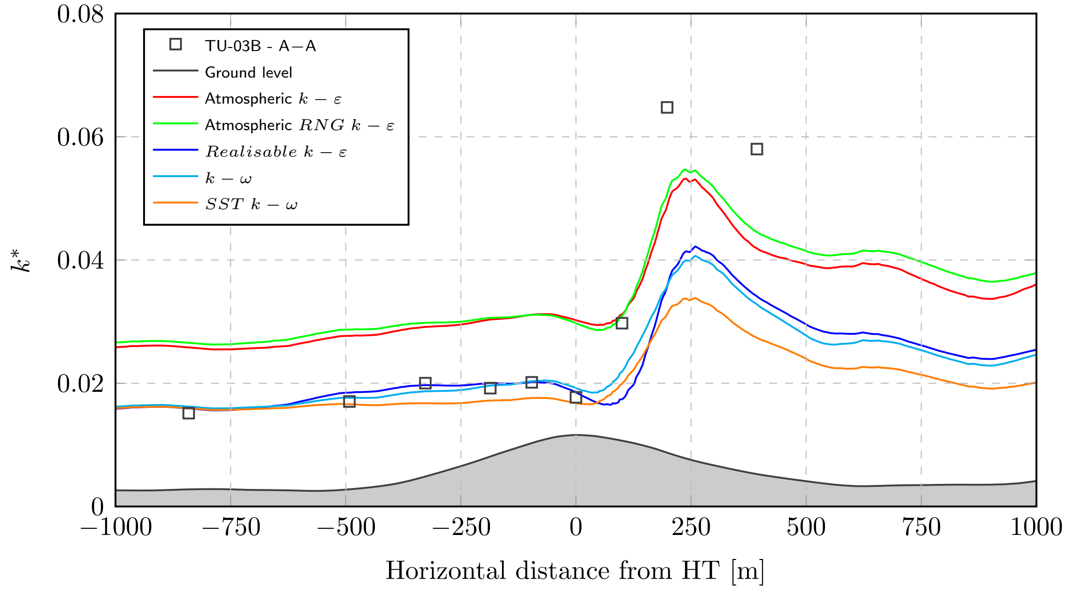
Model	**ΔS** error **HT**	$\boldsymbol{k^*}$ error **HT**	**ΔS** error **HT** +400	$\boldsymbol{k^*}$ error **HT** +400
Atmospheric $k-\varepsilon$	−10.5%	+70.1%	+1.8%	−27.9%
Atmospheric $RNG\ k-\varepsilon$	−11.7%	+68.1%	+9.7%	−23.2%
Realisable $k-\varepsilon$	−8.1%	+4.5%	+26.4%	−41.2%
$k-\omega$	−7.8%	+8.5%	+32.6%	−43.4%
$SST\ k-\omega$	−9.0%	−5.5%	+12.2%	−52.6%

Table 4.5: Relative error between calculated and experimental data along line A−A for five different RaNS models. QUICKV scheme and grid 3 (76 × 76 × 30).

For speed-up, the lowest error in HT was found for $k-\omega$, indicating its better agreement for the simulation of boundary layer flows with favourable pressure gradients when compared to $k-\varepsilon$ models, as mentioned by Wilcox (1994). At the hill lee side, the lowest error was achieved with Atmospheric $k-\varepsilon$, a formulation already discussed in this Section. With grid 3, a recirculation zone was identified only in the simulation using $SST\ k-\omega$ model, confirming this way the superior capabilities of this model to predict zones of flow separation, as reported by Menter (1992). With CFX commercial software and $SST\ k-\omega$, Memon and Kondreddi (2005) have also identified this recirculation zone in the *Askervein* lee side. Location of this zone was determined recurring to paraFoam, but its extent was not compared to the ones obtained with $k-\varepsilon$ model with grids 1, 4 and 5. Speed-up prediction in the lee side of the hill has been improved when using



(a)

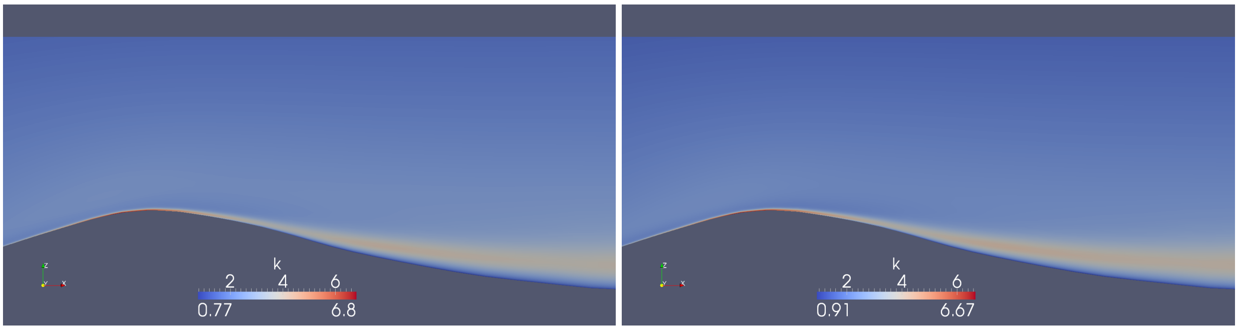


(b)

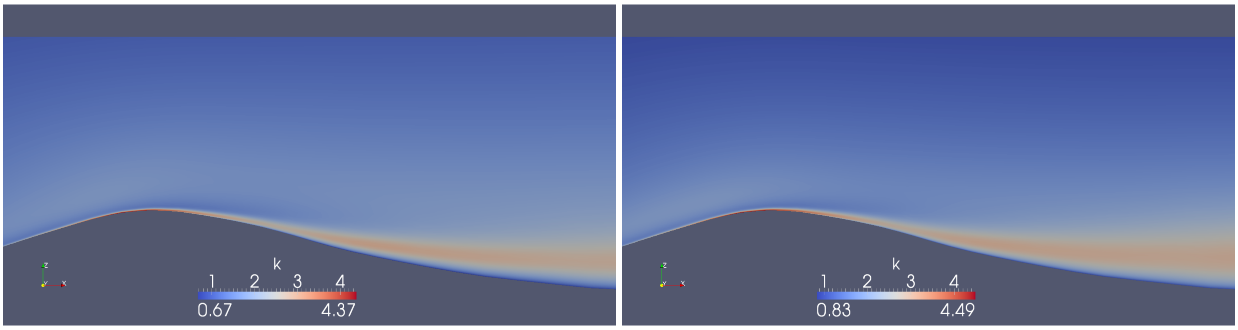
Figure 4.15: (a) Speed-up ratio (ΔS) and (b) non-dimensional TKE (k^*) 10 m AGL along line A—A with five different RaNS models. QUICKV scheme and grid 3 ($76 \times 76 \times 30$).

$SST\ k-\omega$ in lieu of $k-\omega$ model.

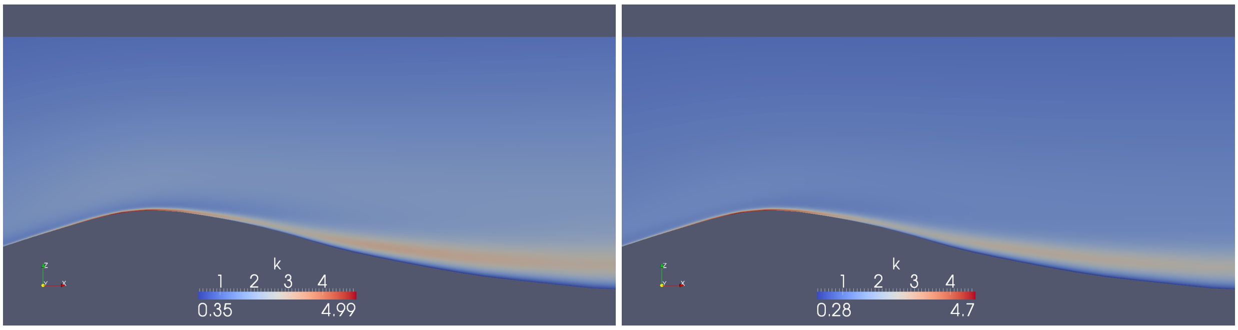
In what concerns to non-dimensional TKE, *Realisable* $k-\varepsilon$ showed the best agreement in HT, nearly matching experimental data in the hill windward. An additional test was performed with this model using a C_μ first guess equal to 0.033. In this test, results for ΔS overlapped the plot in Figure 4.15 and registered number of iterations and CPU time was approximately half of the one presented in Table 4.4 with a C_μ first guess



(a) Atmospheric $k - \varepsilon$ (b) Atmospheric $RNG\ k - \varepsilon$



(c) Standard $k - \varepsilon$ (d) Original $RNG\ k - \varepsilon$



(e) $k - \omega$ (f) $SST\ k - \omega$

Figure 4.16: TKE along a vertical plane matching line A−A for six different RaNS models. QUICKV scheme and grid 3 (76 × 76 × 30).

equal to 0.090. Non-dimensional TKE plots with $k - \omega$ and $SST\ k - \omega$ models were similar to the one with Standard $k - \varepsilon$, which was interpreted as a direct consequence of using identical values for both C_μ and β^* constants for each of these formulations. Effects of RaNS models calibration constants can also be observed in Figure 4.16, where the models with the higher C_μ (or β^*) values presented always lower maximum values of TKE. The difficulty to define consistent and stable free stream boundary conditions for $k - \omega$ based models [reported e.g. in Pope (2000) and Versteeg and Malalasekera (2007)] was considered to be apparently surpassed by combining the k and ε boundary conditions of Castro et al. (2010) in the form $\omega = \varepsilon/(\beta^* k)$ as described in Section 3.3.2.

Chapter 5

Conclusions and future work

5.1 Conclusions

In this work tools and procedures were developed for the simulation of neutrally stratified atmospheric flows over complex terrain, using OpenFOAM. These procedures were thereafter applied in the simulation of a real case.

Developed procedure for mesh generation presented in Section 3.3.1 had as outcome a fully structured mesh, where the user has the ability to define the number of control volumes and its dimensions. This procedure revealed to be a systematic method, surpassing the reported set up difficulties and the high computational effort associated to the use of *snappyHexMesh*, the OpenFOAM resource explored in the literature for the generation of a complex terrain numerical mesh [see Tapia (2009) and Martinez (2011)]. For guidance, generation of grid 3 ($76 \times 76 \times 30$) for *Askervein hill* terrain data as described in Section 4.2.1 took less than 20 seconds of CPU time with 1 core in opposition to the use of 20 cores (although CPU time has not been reported) carried by the latter author when running *snappyHexMesh*.

Presented procedure in Section 3.3.2 allowed for the automation of boundary conditions generation, having as input experimental site data supplied by the user. Developed tool revealed to be able to fully generate OpenFOAM boundary conditions files with required data concerning the computational domain inlet vertical profiles, set up of wall functions conditions and definition of first guesses for the iterative resolution process. This tool, although limited to the simulation of neutrally stratified atmospheric flows, can be easily changed to accomplish its applicability to other stratification states or shapes of inlet vertical profiles, achieving this way an higher versatility when compared to OpenFOAM `atmBoundaryLayerInlet` boundary functions.

Mapping of ground surface aerodynamic roughness length was considered to have been successfully accomplished with the developed procedure presented in Section 3.3.3, allowing to the user the definition of idealized roughness conditions, like an uniform aerodynamic roughness length over all terrain surface or making its value variable as a function of topography characteristic values, and the using of real site data. Developed techniques for the visual inspection of generated roughness maps can be also a good base for the development of procedures for the visualization in paraFoam of other important parameters, like the height ASL of ground surface as also carried in this work, or other user defined scalar or vectorial quantities in the postprocessing stage.

The results of the simulations of a neutrally stratified atmospheric flow over the *Askervein hill* presented in Chapter 4 indicated that OpenFOAM can be a promising tool for the simulation of atmospheric flows over complex terrain. In addition to the comparison between calculated and experimental values, was also evaluated the OpenFOAM sensibility to different convection schemes, mesh dimensions, terrain roughness and turbulence models.

In what concerns to convection schemes, and despite some instability issues identified in the *linear* and QUICK schemes, *linear* showed the best agreement between calculated and experimental data, with an error in speed up ratio equal to -6.7% in HT and nearly matching the experimental value 400 m towards hill downstream along line A$-$A. Although QUICKV error in HT was slightly higher (-7.3%), to this scheme was attributed the best balance between accuracy and computational effort.

Performed sensitivity analysis to mesh dimensions revealed no direct relationship between horizontal grid refinement and accuracy in the calculated speed up ratio, the same conclusion presented by other authors [e.g. Kim et al. (2000)]. Vertical grid refinement yielded worst results in both HT, CP' and lee side along line A$-$A. With grids 1, 4 and 5 and QUICKV scheme, a recirculation zone was identified in the lee side of the hill, noticed in both field campaign report and flow calculations in the reviewed literature, and whose extent was larger on the coarser grid.

OpenFOAM wall function to account for the effects of ground surface roughness in the flow was considered to have been successfully tested, with an increase in speed up ratio and a decrease in non-dimensional TKE for a lower value of aerodynamic roughness length, both in HT, where the height of influence of z_0 in the simulated flow was approximately equal to the limit of the *inner layer*. Whereas in HT the decrease of z_0 improved the agreement between calculated and experimental data, in the lee side, with an increase of z_0 in the flow direction, questions arose if the modelled roughness length was the one which could better describe the real site roughness in this side of the hill during measuring campaign.

In the same manner as other case set up variables, influence of turbulence models on gathered results was noticed mostly on the hill lee side. In this side of the hill, the best agreement of calculated speed-up with experimental data was found with Atmospheric $k-\varepsilon$ and the lack of a trend in the use of different constants sets for this model was attributed to a possible influence of the intermittent behaviour of the recirculation zone in mean velocity field 10 m AGL. Models $k-\omega$ and $SST\ k-\omega$ proved, respectively, its better adequacy to favourable pressure gradients and adverse pressure gradients with flow separation, and unlike the aforementioned by other authors [e.g. Kasmi and Masson (2010)], no benefit was found in the use of $RNG\ k-\varepsilon$ in lieu of $k-\varepsilon$. TKE was considered to be well predicted in the hill windward and hill top for models with

$C_\mu = \beta^* = 0.090$, and grossly predicted in the lee side, being the latter explained as a limitation of the RaNS models.

The intermittent behaviour and the high dependence of the recirculation zone on the numerical parameters of the simulation [Castro et al. (2003)] lead to conclude that more work should be done to evaluate the quality of the gathered results in this side of the hill.

5.2 Future work

In the following, suggestions for future work are presented, always from the point of view of the use of OpenFOAM for wind resource assessment over complex terrain and excluding the possibility of code modifications.

1. Evaluate how different first guesses calculated values, whether for internal field or wall functions, can reduce the solution iterations number and/or CPU time.

2. Perform simulations in topographies with an higher terrain complexity, carefully evaluating the quality of the solution in zones of adverse pressure gradients and flow separation, whether recurring to time dependent simulations or to the analysis of mean field fluctuations between iterations.

3. Assess the validity of OpenFOAM wall function `nutkAtmRoughWallFunction` in terms of canopy height and its morphology. Additionally, a vertical shift in *ground* patch z coordinates, a strategy present e.g. in WAsP, can also be explored to model the zero plane displacement height mentioned in Section 2.1.4.

4. Explore other OpenFOAM solvers in order to include heat transfer effects, following e.g. the Boussinesq approximation [White (1999)] and allowing this way for the simulation of non-neutrally stratified flows as intended in an early stage of this work. Developed procedure for the generation of boundary conditions shall also be adapted in compliance.

5. Develop a mesoscale data coupling procedure able to use meteorological historical data as boundary conditions instead of idealized boundary conditions, recurring e.g. to the heat transfer capabilities developed in the last topic and to OpenFOAM `timeVaryingMappedFixedValue` boundary function.

6. Explore the OpenFOAM capabilities to perform large-eddy simulations and evaluate its adequacy in the simulation of atmospheric flows over complex terrain.

Bibliography

P. S. Arya. *Introduction to Micrometeorology.* Academic Press, 2001.

A. C. M. Beljaars, J. L. Walmsley, and P. A. Taylor. A mixed spectral finite-difference model for neutrally stratified boundary-layer flow over roughness changes and topography. *Boundary-Layer Meteorology*, 38:273–303, 1987.

F. A. Castro. *Numerical Methods for the Simulation of Atmospheric Flows over Complex Terrain (in Portuguese).* PhD thesis, Faculty of Engineering of Porto, 1997.

F. A. Castro, J. M. L. M. Palma, and A. S. Lopes. Simulation of the Askervein Flow. Part 1: Reynolds Averaged Navier–Stokes Equations ($k-\varepsilon$ Turbulence Model). *Boundary-Layer Meteorology*, pages 501–530, 2003.

F. A. Castro, C. S. Santos, and J. C. Costa. Development of a meso-microscale coupling procedure for site assessment in complex terrain. *Proceedings EWEC 2010, Warsaw*, 2010.

B. J. Choudhury and J. L. Monteith. A four-layer model for the heat budget of homogeneous land surfaces. *Quarterly Journal of the Royal Meteorological Society*, 114 (480):373–398, 1988.

J.-F. Corbett, S. Ott, and L. Landberg. A mixed spectral-integration model for neutral mean wind flow over hills. *Boundary-Layer Meteorology*, 128:229–254, 2008.

W. P. Elliot. The growth of the atmospheric internal boundary layer. *Transactions, American Geophysical Union*, 39:1048–1054, 1958.

J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics.* Springer, 3rd edition, 2002.

B. Gardiner. Airflow Over Forests and Forest Gaps. Technical report, MBWEA Tree Workshop Forestry Commission, March 2004.

J. R. Garratt. *The Atmospheric Boundary Layer.* Cambridge University Press, 1992.

R. Gasch and J. Twele. *Wind Power Plants: Fundamentals, Design, Construction and Operation.* Springer, 2002.

gfortran. `http://gcc.gnu.org/fortran`. [Online; acessed 31-March-2013].

GNU Octave. `http://www.gnu.org/software/octave`. [Online; accessed 01-April-2013].

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

D. M. Hargreaves and N. G. Wright. On the use of the k–ε model in commercial CFD software to model the neutral atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, 95:355–369, 2007.

P. S. Jackson and J. C. R. Hunt. Turbulent wind over a low hill. *Quarterly Journal of the Royal Meteorological Society*, 101:929–955, 1975.

M. Z. Jacobson. *Fundamentals of Atmospheric Modeling.* Cambridge University Press, 2005.

L. H. Kantha and C. A. Clayson. *Numerical Models of Oceans and Oceanic Processes.* Academic Press, 2000.

A. Kasmi and C Masson. Turbulence modeling of atmospheric boundary layer flow over complex terrain: a comparison of models at wind tunnel and full scale. *Wind Energy*, 2010.

H. G. Kim and V. C. Patel. Test of turbulence models for wind flow over terrain with separation and recirculation. *Boundary-Layer Meteorology*, pages 5–21, 2000.

H. G. Kim, V. C. Patel, and C. M. Lee. Numerical simulation of wind flow over hilly terrain. *Journal of Wind Engineering and Industrial Aerodynamics*, pages 5–21, 2000.

B. E. Launder and B. I. Sharma. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in Heat and Mass Transfer*, 1: 131–138, 1974.

B. E. Launder and D. B. Spalding. The numerical computation of turbulent flow. *Computer Methods in Applied Mechanics and Engineering*, 3:269–289, 1974.

A. Silva Lopes, J. M. L. M. Palma, and F. A. Castro. Simulation of the Askervein flow. Part 2: Large-eddy simulations. *Boundary-Layer Meteorology*, 125:85–108, 2007.

B. Martinez. Wind resource in complex terrain with openfoam. Master's thesis, Technical University of Denmark, 2011.

P. J. Mason and J. C. King. Measurements and predictions of flow and turbulence over an isolated hill of moderate slope. *Quarterly Journal of the Royal Meteorological Society*, 111:617–640, 1985.

N. Memon and V. R. Kondreddi. Wind resource assessment in complex terrain using cfd. Master's thesis, Technical University of Denmark, 2005.

F. Menter, J. C. Ferreira, and T. Esch. The SST turbulence model with improved wall treatment for heat transfer predictions in gas turbines. *The International Gas Turbine Congress*, pages 1–7, 2003a.

F. R. Menter. Improved two-equation $k - \omega$ turbulence models for aerodynamic flows. *NASA Ames Research Center Technical Memorandum 103975*, 1992.

F. R. Menter and T. Esch. Elements of industrial heat transfer predictions. *16th Brazilian Congress of Mechanical Engineering (COBEM)*, 2001.

F. R. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the SST turbulence model. *Turbulence, heat and mass transfer*, 2003b.

MS-Micro/3. `http://www.zephyrnorth.com`. [Online; acessed 01-March-2013].

T. R. Oke. *Boundary Layer Climates.* Methuen: London and New York, 1987.

OpenFOAM. `http://www.openfoam.org`. [Online; accessed 03-March-2013].

OpenFOAM Programmer's Guide. OpenFOAM Programmer's Guide v.2.1.1, 2012.

OpenFOAM User Guide. OpenFOAM User Guide v.2.1.1, 2012.

L. M. S. Paiva, G. C. R. Bodstein, and W. F. Menezes. Numerical simulation of atmospheric boundary layer flow over isolated and vegetated hills using RAMS. *Journal of Wind Engineering and Industrial Aerodynamics*, 97:439–454, 2009.

J. M. L. M. Palma, F. A. Castro, L. F. Ribeiro, A. H. Rodrigues, and A. P. Pinto. Linear and nonlinear models in wind resource assessment and wind turbine micro-siting in complex terrain. *Journal of Wind Engineering and Industrial Aerodynamics*, 96:2308–2326, 2008.

S. B. Pope. *Turbulent Flows.* Cambridge University Press, 1st edition, 2000.

G. D. Raithby, G. D. Stubley, and P. A. Taylor. The Askervein hill project: A finite control volume prediction on three-dimensional flows over the hill. *Boundary-Layer Meteorology*, 39:107–132, 1987.

M. R. Raupach. Simplified expressions for vegetation roughness length and zeroplane displacement as functions of canopy height and area index. *Boundary-Layer Meteorology*, 71:211–216, 1994.

M. R. Raupach and A. S. Thom. Turbulence in and above plant canopies. *Annual Review of Fluid Mechanics*, 13:97–129, 1981.

P. J. Richards and R. P. Hoxey. Appropriate boundary conditions for computational wind engineering models using the k-ε turbulence model. *Journal of Wind Engineering and Industrial Aerodynamics*, 46,47:145–153, 1993.

M. L. Salby. *Fundamentals of Atmospheric Physics.* Academic Press, 1996.

J. R. Salmon, H. W. Teunissen, and R. E. Mickle. The Kettles hill project: field observations, wind tunnel simulations and numerical model preductions for flow over a low hill. *Boundary-Layer Meteorology*, 43:309–343, 1988.

C. S. Santos. Renewable Energies 1. Lecture Notes (in Portuguese). ISEP, 2012.

R. H. Shaw and A. R. Pereira. Aerodynamic roughness of a plant canopy: a numerical experiment. *Agricultural Meteorology*, 26:51–65, 1982.

T. H. Shih, W. W. Liou, A. Shabbir, Z. Yang, and J. Zhu. A new $k - \varepsilon$ eddy viscosity model for high reynolds number turbulent flows - model development and validation. *Computers & Fluids*, (August), 1995.

T.-H. Shih, J. Zhu, and J. L. Lumley. A new reynolds stress algebraic equation model. *Computer Methods in Applied Mechanics and Engineering*, 125:287 – 302, 1995b.

C. G. Speziale and S. Thangam. Analysis of a RNG based turbulence model for separated flows. *International Journal of Engineering Science*, (92), 1992.

SRTM. http://srtm.usgs.gov. [Online; accessed 13-May-2013].

R. B. Stull. *An Introduction to Boundary Layer Meteorology*. Kliwer Academic Publishers, 1988.

SU^2. http://su2.stanford.edu. [Online; accessed 03-March-2013].

J. Sumner, C. Masson, Y. Odemarky, and M. Cehlinz. OpenFOAM simulations of atmospheric flow over complex terrain. *5th OpenFOAM Workshop*, 2010.

X. P. Tapia. Modelling of wind flow over complex terrain using openfoam. Master's thesis, University of Gävle, 2009.

P. A. Taylor. Some numerical studies of surface boundary-layer flow over gentle topography. *Boundary-Layer Meteorology*, 11:439–465, 1977.

P. A. Taylor and R. J. Lee. Simple guidelines for estimating wind speed variations due to small scale topographic features. *Climatological Bulletin*, 18:3–23, 1987.

P. A. Taylor and H. W. Teunissen. The Askervein Hill Project: Report on the Sept./Oct. 1983, Main Field Experiment, Research Report MSRB-84-6. Technical report, Meteorological Services Research Branch Atmospheric Environment Service 4905 Dufferin Street, Downsview, Ontario, Canada M3H 5T4, December 1985.

P. A. Taylor and H. W. Teunissen. The Askervein hill project: overview and background data. *Boundary-Layer Meteorology*, 39:15–39, 1987.

P. A. Taylor, P. J. Mason, and E. F. Bradley. Boundary-layer flow over low hills. *Boundary-Layer Meteorology*, 1987.

Terrain Tool. http://www.ubss.org.uk/terraintool/terraintool.php. [Online; accessed 19-August-2013].

H. W. Teunissen, M. E. Shokr, A. J. Bowen, C. J. Wood, and D. W. R. Green. The Askervein hill project: Wind tunnel simulations at three length scales. *Boundary-Layer Meteorology*, 40:1–29, 1987.

H. W. Tieleman. Roughness estimation for wind-load simulation experiments. *Journal of Wind Engineering and Industrial Aerodynamics*, 91:1163–1173, 2003.

H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Prentice Hall, 2nd edition, 2007.

J. L. Walmsley and P. A. Taylor. Boundary-layer flow over topography: impacts of the Askervein study. *Boundary-Layer Meteorology*, (1987), 1996.

J. L. Walmsley, J. R. Salmon, and P. A. Taylor. On the application of a model of boundary-layer flow over low hills to real terrain. *Boundary-Layer Meteorology*, 23:17–46, 1982.

WAsP. `http://www.wasp.dk`. [Online; accessed 01-March-2013].

F. M. White. *Fluid Mechanics*. Mc Graw-Hill Book Co., 4 edition, 1999.

J. Wieringa. Representativeness of wind observations at airports. *Bulletin of the American Meteorological Society*, 1980.

D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Inc., 1994.

D.C Wilcox. Re-assessment of the scale-determining equation for advanced turbulence models. *AIAA Journal*, 26:1299–1310, 1988.

Windie. `http://www.megajoule.pt`. [Online; accessed 01-March-2013].

WindSim. `http://www.windsim.com`. [Online; accessed 01-March-2013].

WRF-ARW. `http://www.wrf-model.org`. [Online; accessed 14-March-2013].

J. Wyngaard. *Turbulence in the Atmosphere*. Cambridge University Press, 2010.

V. Yakhot and S. A. Orszag. Renormalization group analysis of turbulence. I - Basic theory. *Journal of Scientific Computing*, 1:3–51, 1986.

V. Yakhot, S. A. Orszag, and S. Thangam. Development of turbulence models for shear flows by a double expansion technique. *Physics of Fluids A: Fluid Dynamics*, 4:1510–1520, 1992.

Appendix A

OpenFOAM case folders structure and its content

In this Appendix is presented the folders structure and its content on the integration of the tools and procedures developed during this work for the simulation of atmospheric flows over complex terrain, using OpenFOAM.

Figure A.1 shows a tree view of a typical case, like e.g., the simulations conducted in Section 4. Its content is hereafter described.

- `<case>/0`

  Besides native OpenFOAM files defining model boundary conditions, this folder stores `z` and `z0` files used to force paraFoam to load two extra scalar fields, allowing respectively, for visual inspection of (i) height ASL of *ground* patch and (ii) modelled ground roughness boundary conditions. `k`, `epsilon` and `omega` files are generated by *write_bCs* presented in Section 3.3.2 and `nut`, `z` and `z0` generated by *write_z0* presented in Section 3.3.3.

  - `<case>/0/dat`

    `.dat` files in this folder store the data called by homonym files in folder `<case>/0` via an `#include` instruction in the respective patch definition. `epsilon.dat`, `k.dat`, `omega.dat` and `u.dat` files are generated by *write_bCs* presented in Section 3.3.2 and `z0.dat` and `z.dat` files generated by *write_z0* presented in Section 3.3.3.

- `<case>/constant`

  Its a native OpenFOAM folder and content. `blockMeshDict` file is generated by *write_blockMeshDict* and `RASProperties` file generated by *write_turbulenceProperties*, both tools presented respectively in Sections 3.3.1 and 3.3.4.

```
<case>
|-- 0
|   |-- dat
|   |   |-- epsilon.dat
|   |   |-- k.dat
|   |   |-- omega.dat
|   |   |-- u.dat
|   |   |-- z0.dat
|   |   `-- z.dat
|   |-- epsilon
|   |-- k
|   |-- nut
|   |-- omega
|   |-- p
|   |-- U
|   |-- z
|   `-- z0
|-- constant
|   |-- polyMesh
|   |   `-- blockMeshDict
|   |-- RASProperties
|   `-- transportProperties
|-- _postproc
|   |-- dat
|   |   |-- agl.dat
|   |   |-- aglsurf.dat
|   |   |-- gl.dat
|   |   `-- vprfl1.dat
|   |-- _postprocDict
|   |-- src
|   |   `-- write_sample.src
|   |       `-- write_sample.f90
|   |-- vtk
|   |   |-- aglsurf.vtk
|   |   |-- agl.vtk
|   |   `-- vprfl1.vtk
|   `-- write_sample
|-- _preproc
|   |-- plot
|   |-- _preprocDict
|   |-- src
|   |   |-- write_bCs.src
|   |   |   `-- write_bCs.f90
|   |   |-- write_blockMeshDict.src
|   |   |   `-- write_blockMeshDict.f90
|   |   |-- write_turbulenceProperties.src
|   |   |   `-- write_turbulenceProperties.f90
|   |   `-- write_z0.src
|   |       `-- write_z0.f90
|   |-- utils
|   |   |-- mesh_calc.m
|   |   `-- terraintoolcsv2raw.m
|   |-- vtk
|   |   `-- topo_nrot.vtk
|   |-- write_bCs
|   |-- write_blockMeshDict
|   |-- write_turbulenceProperties
|   `-- write_z0
`-- system
    |-- controlDict
    |-- fvSchemes
    |-- fvSolution
    `-- sampleDict
```

Figure A.1: Example of an OpenFOAM case folders structure and its contents on the integration of the developed tools and procedures.

- `<case>/_postproc`

  Its a folder created for the integration of the postprocessing tools and procedures presented in Section 3.4. Besides sub folders following presented, stores the executable `write_sample` and `_postProcDict`, an ASCII file for the configuration of the postprocessing parameters and whose content is presented in Appendix B.

  - `<case>/_postproc/dat`

    `agl.dat`, `aglsurf.dat` and `vprfl1.dat` files in this folder store the data concerning each of the coordinates sets presented in Section 3.4, which are called by OpenFOAM *sample* utility. `gl.dat` stores the ground level coordinates that match AGL line coordinates in file `agl.dat`. The `.dat` files in this folder are generated by *write_sample* presented in Section 3.4.

  - `<case>/_postproc/src`

    Stores the FORTRAN source code of *write_sample*. In the same folder as the source code is stored a `Makefile` for code compilation via a terminal *make* instruction.

  - `<case>/_postproc/vtk`

    `.vtk` files in this folder store the coordinates sets in the homonym `.dat` files in folder `<case>/_postproc/dat` and allow for a visual inspection of generated postprocessing coordinates in paraFoam. All `.vtk` files in this folder are generated by *write_sample*.

- `<case>/_preproc`

  Its a folder created for the integration of the preprocessing tools and procedures presented in Section 3.3. Besides sub folders following presented, stores the executables `write_bCs`, `write_blockMeshDict`, `write_turbulenceProperties` and `write_z0`, and `_preProcDict`, an ASCII file for the configuration of the preprocessing parameters and whose content is presented in Appendix B.

  - `<case>/_preproc/plot`

    Stores raw `.dat` files generated by *write_bCs* allowing for the plot of boundary conditions profiles at *inlet* patch.

- `<case>/_preproc/src`

  Stores the FORTRAN source code of the developed preprocessing tools. In the same folder as each of the source codes, is stored a `Makefile` for code compilation via a terminal *make* instruction.

- `<case>/_preproc/utils`

  In this folder are stored two GNU Octave scripts: (i) `utils.m` for the calculation of mesh parameters, like expansion factor R, and (ii) `terraintoolcsv2raw.m`, a tool for the conversion of topographic xyz data in CSV (*Comma Separated Values*) format to a raw ASCII file.

- `<case>/_preproc/vtk`

  Stores the VTK file with topographic data to be read by *write_blockMeshDict*. In the present work, this file was generated by *gsurf*, as described in Section 3.3.1.

- `<case>/system`

  Its a native OpenFOAM folder and whose content has already been briefly addressed in Section 2.3. Configuration of `sampleDict` file in order to allow for data extraction as described in Section 3.4 is presented in Appendix B.

Appendix B

Instructions for preprocessing, processing and data extraction

In this Appendix are presented the instructions for preprocessing, processing and data extraction on the integration of the tools and procedures developed during this work for the simulation of atmospheric flows over complex terrain, using OpenFOAM.

In what concerns to topographic data, elevation model of the *Askervein* hill for the simulations presented in Chapter 4 was obtained from SRTM (*Shuttle Radar Topography Mission*) recurring to Terrain Tool, an application distributed under the GNU General Public License. Gathered data in the CSV format was converted to a raw ASCII file with script `terraintoolcsv2raw.m` presented in Appendix A and then manipulated with *gsurf* in order grade nodal spacing and align computational domain with a predefined direction. Its output is `topo_nrot.vtk`, a VTK file in the ASCII format. Being *gsurf* part of Windie, a copyrighted code, its specific set up procedures were here omitted.

B.1 Preprocessing

Configuration of the preprocessing tools and procedures presented in Section 3 is performed in `_preprocDict`, an ASCII file whose content is presented in Figure B.1. Values of the variables in that figure concern the set up of Case 18 in Table 4.4. Developed preprocessing procedure is hereafter described.

B.1.1 Mesh generation

1. Configuration of `_preprocDict`

   Following variables have to be defined by the user in file `_preprocDict`:

   (a) `sky`, the height ASL of computational domain *top* patch.

   (b) `nz`, number of cells (control volumes) in z direction.

   (c) `R`, the OpenFOAM cell expansion factor R, which can be calculated recurring to `mesh_calc.m` utility mentioned in Appendix A giving as input (i) the height

```
######################################################
# CONFIGURATION OF PREPROCESSING TOOLS FOR OpenFOAM #
######################################################

# write_blockMeshDict #
sky          = 700.       # Height ASL of computational domain top patch [m]
nz           = 030        # Number of cells in z direction
R            = 194.76     # R expansion factor

# write_bCs #
zrs          = 10.        # Reference height [m]
ustar        = 0.607      # Friction velocity [m/s]
z0i          = 0.03       # Reference aerodynamic roughness length [m]
delta        = 1000.      # Height of boundary layer [m]
kv           = 0.41       # vonKarman constant
cmu          = 0.033      # kEpsilon model coefficient (betaStar kOmega)

# write_z0 #
z0u          = 0.03       # Uniform aerodynamic roughness length [m]
opt          = 3          # 1 - Uniform roughness mode
                          # 2 - Roughness map in rough_nrot.vtk
                          # 3 - Roughness defined by an equation in write_z0.f90

# write_turbulenceProperties #
model        = 13         # kEpsilon: 11-Calibrated, 12-Standard, 13-Atmospheric
                          # RNGkEpsilon: 21-Original, 22-Modified, 23-Atmospheric
                          # RealizableKE: 3
                          # kOmega: 4
                          # kOmegaSST: 5
```

Figure B.1: Content of file `_preprocDict` for the configuration of preprocessing tools and procedures. Values of variables concern the set up of Case 18 in Table 4.4.

of computational domain in z direction, (ii) the number of intended cells and (iii) the height of the control volume adjacent to *ground* patch.

2. Generation of `blockMeshDict`

   Running the executable `write_blockMeshDict`, OpenFOAM `blockMeshDict` file is created based on `topo_nrot.vtk` *xyz* data and moved to its typical folder.

3. Generation of numerical mesh

   Running the OpenFOAM *blockMesh* utility in case root path, numerical mesh is created.

B.1.2 Boundary conditions

1. Configuration of `_preprocDict`

   Following variables have to be defined by the user in file `_preprocDict`:

   (a) `zrs`, the reference height AGL z in Eq. 3.5 for the calculation of ε internal field first guess.

   (b) `ustar`, the friction velocity u_* for the calculation of boundary conditions and internal field first guesses as described in Section 3.3.2.

   (c) `z0i`, the aerodynamic roughness length z_0 in Eqs. 3.1 and 3.5.

   (d) `delta`, the boundary layer height δ in Eqs 3.1, 3.2 and 3.3.

   (e) `kv`, the vonKarman constant κ.

   (f) `cmu`, the $k - \varepsilon$ model coefficient C_μ in Eqs. 3.2, 3.3 and 3.4. It defines also the wall functions C_μ or β^* value irrespectively of the typical C_μ value of used turbulence model formulation, except for Calibrated $k - \varepsilon$, where $C_\mu =$ `cmu`.

2. Generation of boundary conditions

   Running the executable `write_bCs` are created the `epsilon`, `k` and `omega` files, and the homonym `.dat` files plus `u.dat`. Generated files are moved to destination folders presented in Figure A.1.

Note: Being `p` and `U` files equal for all cases, its content is considered as static and are therefore not generated by any of the developed tools.

B.1.3 Roughness modelling

1. Configuration of `_preprocDict`

   Following variables have to be defined by the user in file `_preprocDict`:

   (a) `z0u`, the aerodynamic roughness length z_0 to be considered in case of choosing the Uniform roughness mode.

   (b) `opt`, the intended roughness mapping mode as described in Section 3.3.3. `rough_nrot.vtk` is generated by a modified version of *gsurf* that reads a roughness map in the form xyz_0 stored in a raw ASCII file and manipulates it in the same form as original *gsurf* does for topographic data. For the equation based roughness mapping mode, equation has to be defined in `write_z0.f90` source code file and recompiled via a terminal *make* instruction.

2. Generation of roughness maps

   Running the executable `write_z0` are created the `nut`, `z.dat` and `z0.dat` files and moved to the destination folders presented in Figure A.1. All roughness information to be read by OpenFOAM ν_t wall function is written in `nut` file proceeded by a `z0 nonuniform List<scalar>` instruction. `z` and `z0` files allow for a visual inspection in paraFoam of scalars (i) z, the average height ASL of each of the four points that define a face, and (ii) z_0, the average aerodynamic roughness length of each of the four points that define a face, both for *ground* patch. The values of each of these scalars are stored in the homonym `.dat` files in folder `<case>/0/dat` and called via an `#include` instruction defined in the scalar *ground* patch boundary conditions. Figures B.2 and B.3 show an excerpt of roughness definition in both `nut` and `z0` files.

```
    ground
    {
        type            nutkAtmRoughWallFunction;
        Cmu             0.033;
        kappa           0.41;
        E               9.8;
        z0              nonuniform List<scalar>
        5776
(
3.000000E-02
3.000000E-02
3.000000E-02
(...)
);
        value                uniform 0;
    }
```

Figure B.2: Excerpt of file `nut` for the configuration of turbulent viscosity ν_t boundary conditions in *ground* patch. `(...)` represents the omission of the remaining values of z_0.

```
    ground
    {
        type            fixedValue;
        value           nonuniform
        (#include "dat/z0.dat");
    }
```

Figure B.3: Excerpt of file `z0` for visual inspection of modelled aerodynamic roughness length z_0.

Note: Being `z` and `z0` files equal for all cases, its content is considered as static and are therefore not generated by any of the developed tools.

B.1.4 Turbulence properties

1. Configuration of `_preprocDict`

   Following variables have to be defined by the user in file `_preprocDict`:

   (a) `model`, a one or two digit variable concerning the intended turbulence model and its specific constants sets.

2. Generation of `RASProperties` file

   Running the executable `write_turbulenceProperties` is created the file `RASProperties` and moved to its destination folder presented in Figure A.1. Constants sets of each of the models is defined as presented in Table 3.4. For the Calibrated $k - \varepsilon$ model, C_μ is taken equal to the value of `cmu` in `_preprocDict` and σ_ε calculated according to Eq. 3.9.

B.2 Processing

For the simulation of the neutrally stratified ABL flows in this work was used the OpenFOAM solver *simpleFoam*, which according to the OpenFOAM User Guide (2012) is indicated for steady-state, incompressible and turbulent flows.

In the case of a simulation without domain decomposition and CPU paralellization, the instruction `simpleFoam > log & tail -f log` in a Linux terminal allows at the same time for solver run, screen output of solver log and to save that log in a `log` file. In order to terminate *simpleFoam* before convergence of a solution, care must be taken to stop both *tail* and *simpleFoam* processes.

B.3 Data extraction for postprocessing

Configuration of the postprocessing tools and procedures presented in Section 3.4 is performed in `_postprocDict`, an ASCII file whose content is presented in Figure B.4. Values of the variables in that figure concern, e.g., the set up of data extraction (i) along a line 10 m AGL rotated 223° from north as in Figure 4.6, (ii) along a surface of 1000 × 1000 m^2 (similar to the one presented in Figure 4.10, but positioned 10 m AGL) and (iii) along a vertical profile positioned in the centre of computational domain and starting in ground level towards 300 m AGL, as partially plotted in Figure 4.8. Developed procedure for data extraction is hereafter described.

1. Configuration of `_postprocDict`

   Following variables have to be defined by the user in file `_postprocDict`:

   (a) `wd`, the prevailing wind direction that was on the base of computational domain alignment upon *gsurf* execution.

   (b) `ang`, the angle of a line of coordinates with the same height AGL, measured from north, clockwise.

```
#######################################################
# CONFIGURATION OF POSTPROCESSING TOOLS FOR OpenFOAM #
#######################################################

# write_sample #
wd             = 210.      # Wind direction [°]
ang            = 223.      # Angle to inspect (from North - before gsurf) [°]
zagl           = 10.       # Height above ground level of line agl [m]
npaglvec       = 1000      # Size of vector that defines line agl
agloffsetx     = 0.        # Offset in x direction of line agl [m]
agloffsety     = 0.        # Offset in y direction of line agl [m]
npvprfl        = 1000      # Size of vectors that define vertical profiles vprfl#
zmast          = 300.      # Height of vertical profiles above ground level [m]
npaglsurf      = 100       # Size of matrix aglsurf (npaglsurf x npaglsurf)
bound          = 0         # Bounding of aglsurf: 0 - Manual; 1 - Auto
xmin           = -1000.    # x and y minimum values of aglsurf [m]
xmax           = +1000.    # x and y maximum values of aglsurf [m]
nmasts         = 2         # Quantity of vertical profiles (masts)
mastcoords:
0.        0.
-5.        10.
```

Figure B.4: Content of file `_postprocDict` for the configuration of postprocessing tools and procedures.

(c) `zagl`, the height AGL of (i) the line of coordinates rotated `ang` degrees and (ii) a surface defined by a matrix of coordinates.

(d) `npaglvec`, the size of the vector that defines the line of coordinates at a height `zagl` rotated `ang` degrees.

(e) `agloffsetx`, the offset in x direction of the generated line of coordinates.

(f) `agloffsety`, the offset in y direction of the generated line of coordinates.

(g) `npvprfl`, the size of the vectors that define each of the vertical profiles.

(h) `zmast`, the height AGL matching the top of the vertical profiles.

(i) `npaglsurf`, the size of the matrix that defines a surface of coordinates AGL. It will be generated a square matrix of `npaglsurf` rows per `npaglsurf` columns.

(j) `bound`, the option for bounding the upper and lower limits of the surface of coordinates AGL. If `bound` $= 0$, the x and y lower and upper limits will be respectively, `xmin` and `xmax`. If `bound` $= 1$, the limits of this surface will match the limits of computational domain.

(k) `nmasts`, the number of the intended vertical profiles. For each vertical profile must be defined a pair of coordinates (`mastcoords`) concerning the profile xy position in computational domain.

    (l) `mastcoords`, the xy coordinates in computational domain of each of the masts position.

2. Generation of coordinates sets

   Running the executable `write_sample` are created (i) the `.dat` files in folder `<case>/_postproc/dat` that store the generated coordinates sets and (ii) the `.vtk` files in folder `<case>/_postproc/vtk` that allow for a visual inspection of the generated coordinates as presented in Figure 3.9. For each vertical profile quantified by the variable `nmasts` is created a `vprfl#.dat` file and a `vprfl#.vtk` file, being `#` $= \{1, 2, 3, ...,$`nmasts`$\}$.

3. Configuration of `sampleDict`

   `sampleDict` native OpenFOAM file must be set up in order to extract the values of the required fields for each of the generated coordinates sets, like in Figure B.5, an excerpt of `sampleDict` configuration in order to extract velocity and TKE fields. For each vertical profile quantified by the variable `nmasts` must be defined a coordinates set similar to the one named as `vprfl1`, adjusting the `.dat` file name suffix from 1 to `nmasts`.

4. Data extraction

   Running the OpenFOAM *sample* utility in the form `sample -latestTime`, the required fields for the each of the coordinate sets are extracted from the converged solution and saved in a raw ASCII file in folder `<case>/sets`.

```
fields
(
    k
    U
);
// Fields to sample.

sets
(
    agl
    {
        type        cloud;
        axis        xyz;
        points      (#include "../_postproc/dat/agl.dat");
    }

    aglsurf
    {
        type        cloud;
        axis        xyz;
        points      (#include "../_postproc/dat/aglsurf.dat");
    }

    vprfl1
    {
        type        cloud;
        axis        xyz;
        points      (#include "../_postproc/dat/vprfl1.dat");
    }

    vprfl2
    {
        type        cloud;
        axis        xyz;
        points      (#include "../_postproc/dat/vprfl2.dat");
    }
);
// Coordinate sets to sample.
```

Figure B.5: Excerpt of file `sampleDict` in order to extract velocity and TKE fields for three generated coordinates sets.