

UNSTEADY ATMOSPHERIC FLOW SOLUTIONS WITH OPENFOAM
COUPLED WITH THE NUMERICAL WEATHER PREDICTION SOFTWARE,
WRF

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ENGİN LEBLEBİCİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
AEROSPACE ENGINEERING

FEBRUARY 2018

Approval of the thesis:

**UNSTEADY ATMOSPHERIC FLOW SOLUTIONS WITH OPENFOAM
COUPLED WITH THE NUMERICAL WEATHER PREDICTION
SOFTWARE, WRF**

submitted by **ENGİN LEBLEBİCİ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ozan Tekinalp
Head of Department, **Aerospace Engineering**

Prof. Dr. İsmail H. Tuncer
Supervisor, **Aerospace Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Yusuf Özyörük
Aerospace Engineering Department, METU

Prof. Dr. İsmail H. Tuncer
Aerospace Engineering Department, METU

Prof. Dr. Hasan Umur Akay
Department of Mechanical Engineering, Atilim University

Assoc. Prof. Dr. Oğuz Uzol
Aerospace Engineering Department, METU

Assist. Prof. Dr. Mustafa Kaya
Aerospace Engineering Department, YBU

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ENGIN LEBLEBICI

Signature :

ABSTRACT

UNSTEADY ATMOSPHERIC FLOW SOLUTIONS WITH OPENFOAM COUPLED WITH THE NUMERICAL WEATHER PREDICTION SOFTWARE, WRF

Leblebici, Engin

Ph.D., Department of Aerospace Engineering

Supervisor : Prof. Dr. İsmail H. Tuncer

February 2018, 135 pages

Ways to predict wind energy potential and energy generation accurately are sought in many applications such as the wind turbine siting and the short term wind power production estimation. Current wind energy prediction models are based on the statistical analysis of long term observation data and the reconstruction of average wind fields by means of numerical tools. Even if computational fluid dynamics tools based on Navier-Stokes solutions are employed for sectoral wind fields, the reconstruction of wind fields according to prevailing wind speeds and directions are, in general, not accurate for wind field predictions on complex terrains. On the other hand, numerical weather prediction software such as WRF, in which the unsteady atmospheric physics are taken into account, provides time dependent atmospheric flow field solutions, but due to the low resolution and the pressure based vertical coordinate system wind field predictions are inaccurate near ground in complex terrains.

In this work, unsteady atmospheric flows are computed by a coupled solution methodology. Atmospheric flow solutions based on the weather prediction model, WRF, on

a low resolution grid provide the unsteady boundary conditions for *OpenFOAM* solutions on high resolution terrain fitted grids. The usage of WRF predictions as boundary conditions accounts partially for the physical processes such as solar radiation, precipitation, surface heating and the resulting diurnal cycle. *OpenFOAM* solutions with the high resolution topography and computational grid, and a spatially varying roughness model capture complex terrain effects such as flow separation, recirculation, and reattachment zones. The *OpenFOAM* solutions coupled with WRF are performed in parallel. A post-processing tool is also developed to extract the wind field data from the coupled *OpenFOAM* solutions at any given altitude, and to estimate the distributions of wind power density and wind energy generation by time integration along the unsteady flow solution.

The coupled flow solutions are compared with the time series wind speed and direction data obtained from a meteorological measurement tower, Met-mast. It is shown that the coupled flow solutions developed improve the time dependent wind field predictions of WRF by about 12% with respect to the observation data which corresponds to about 40% improvement on wind power production estimations.

Keywords: Computational Fluid Dynamics (CFD) , Wind Energy Prediction, WRF, OpenFOAM

ÖZ

OPENFOAM İLE AKUPLE NUMERİK HAVA TAHMİN MODELİ, WRF KULLANILARAK ZAMANA BAĞLI ATMOSFERİK AKIŞ ÇÖZÜMLERİ

Leblebici, Engin

Doktora, Havacılık ve Uzay Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. İsmail H. Tuncer

Şubat 2018 , 135 sayfa

Rüzgar enerjisi potansiyelini ve güç üretimini doğru bir şekilde tahmin etme yöntemleri, rüzgar türbin yerleşimi ve kısa dönemli rüzgar gücü üretimi tahminleri gibi bir çok uygulamada aranmaktadır. Güncel rüzgar enerjisi tahmin modelleri uzun dönem rüzgar ölçüm verilerinin istatistiksel analizine ve ortalama rüzgar alanlarının sayısal araçlar kullanılarak yeniden yapılandırılması esasına dayanmaktadır. Sektörel rüzgar alanları için Navier-Stokes çözümleri uygulansa bile, rüzgar alanlarının baskın rüzgar hızı ve yönüne göre yeniden yapılandırılması, genel olarak, rüzgar alan tahminleri için hassas değildir. Diğer yandan, daimi olmayan atmosfer fiziğini hesaba katan WRF gibi sayısal hava tahmin yazılımları ile, zamana bağlı atmosferik akış alanı çözümleri elde edilebilse de, düşük çözünürlük ve basınç tabanlı dikey koordinat sisteminden ötürü karmaşık arazilerde çözümler hassas olmamaktadır.

Bu çalışmada, daimi olmayan atmosferik akış çözümleri akuple bir çözüm yöntemi kullanılarak hesaplanmıştır. Sayısal hava tahmin modeli WRF'a dayalı düşük çözünürlükteki atmosferik akış çözümleri, yüksek çözünürlüklü ve topoğrafyaya uyumlu çözüm ağlarındaki *OpenFOAM* çözümlerine daimi olmayan sınır koşulları sağlamakta-

dır. WRF tahminlerinin sınır koşulları olarak kullanımı güneş ışıınımı, yağış, yer ısınması ve ortaya çıkan gece-gündüz döngüsü gibi fiziksel olayların hesaba katılmasını sağlamaktadır. Yüksek çözünürlüklü topoğrafya ve çözüm ağıları ile yere bağlı değişken pürüzlülük modellemesiyle yapılan *OpenFOAM* çözümleri akış ayrılması, sirkülasyon ve yeniden yapışma gibi karmaşık arazi etkilerini yakalamaktadır. WRF ile akuple *OpenFOAM* çözümleri paralel olarak uygulanmıştır. Ayrıca, yerden belirli bir yükseklikteki akış alanlarını, daimi olmayan WRF ile akuple *OpenFOAM* çözümlerinden çıkaran ve zamana bağlı çözümler boyunca entegrasyonu ile rüzgar enerjisi ve rüzgar potansiyeli yoğunluğu dağılımlarının tahmini yapan bir son işlem aracı geliştirilmiştir.

Akuple çözüm yönteminin sonuçları, bir meteorolojik ölçüm direğinden alınan rüzgar hız ve yön zaman serisi verileri ile karşılaştırılmıştır. Geliştirilen akuple çözüm yönteminin, WRF'nin zamana bağlı rüzgar hız tahminlerini, deneysel verilere göre, üretilen rüzgar gücü tahmininde %40 'lük iyileşmeye karşılık gelen, %12 civarında iyileştirdiği gösterilmiştir.

Anahtar Kelimeler: Hesaplamalı Akışkanlar Dinamiği (HAD), Rüzgar Enerji Tahmini, WRF, OpenFOAM

To Boncuk

ACKNOWLEDGEMENTS

To begin as is custom, I would like to express my gratitude to my supervisor Prof. Dr. İsmail H. Tuncer as he had a great contribution to who I am now for better or worse, both personally and academically.

I would like thank Umut Erdal A.K.A Grandfather of Sea for helping me when no one else can and reminding me to respect the Elders.

I wish to express my sincere thanks to my former team-mate Gökhan Ahmet for giving me his interpolation routines and his support to go on with my PhD.

As for my latter team-mate and my (not so) little brother Tansu Sevine, thank you for sticking up with me. Without your company, this dissertation would not be possible.

Also, I wish to express my sincere thanks to my soon expanding family and its cause, my fiancée Derya Kaya for their thrust and understanding throughout this 6 years.

Last but never the least, I am grateful for the times with my cats: Tirt, Uzay, Mars and Boncuk. Without them my thesis might have been finished earlier but I wouldn't have it any other way.

This work is supported by TUBITAK under the Project No: 215M385 : “Microscale Modelling coupled with Mesoscale Weather Prediction Models” and is part of FP7 Project: “New European Wind Atlas (NEWA)”. This support is greatly acknowledged.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGEMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xiv
LIST OF FIGURES	xv
LIST OF ABBREVIATIONS	xx
CHAPTERS	
1 INTRODUCTION	1
1.1 Atmospheric Boundary Layer and Wind	2
1.2 Wind Power Prediction/Estimation Models	7
1.2.1 Linearized Models	10
1.2.2 CFD Based Models	10
1.2.3 Numerical Weather Prediction Software: Weather Research Forecast, WRF	12
1.3 OpenFOAM Open Source Library	14
1.4 Objectives	16
2 METHODOLOGY	19

2.1	Problem Definition and Numerical Model	19
2.2	WRF Solutions	22
2.3	High Resolution Topography Modelling and Grid Generation	23
2.4	Extraction of Boundary Condition Profiles from WRF	25
2.5	OpenFOAM Flow Solver	31
2.5.1	File Structure in an OpenFOAM Solution	32
2.5.2	OpenFOAM Boundary Condition Classes	35
2.5.2.1	mixedFvPatchField Class	36
2.5.2.2	InletOutlet Class	36
2.6	Developement of Time and Spatially Varying BC class . . .	39
2.7	Parallel Implementation of TimeVaryingMixed Bound- ary Condition	43
2.8	Turbulence Modelling	48
2.8.1	(Unsteady) Reynolds Averaged Navier-Stokes Equa- tions (URANS)	49
2.8.1.1	$k - \epsilon$ Model	50
2.8.1.2	Realizable $k - \epsilon$ Model	51
2.8.1.3	$k - \omega$ SST Model	53
2.9	Near Wall Treatment and Roughness	54
2.10	Coriolis Acceleration	57
2.10.1	Implementation of Coriolis Acceleration in Pim- pleFOAM Solver	61
3	RESULTS AND DISCUSSIONS	63
3.1	Validation of Coriolis Acceleration implemented in Pimple- FOAM Solver	64

3.2	Case Study: Wind Farm in Mut Mersin, Turkey	66
3.3	Case Study: Alaiz Wind Field in Navarra, Spain	73
3.3.1	WRF Solutions	74
3.3.2	High Resolution Terrain Modeling for OpenFOAM and Grid Generation	75
3.3.3	Grid Independence Study	76
3.3.4	Comparison of Turbulence Models	80
3.3.5	Unsteady OpenFOAM solutions coupled with WRF	81
3.3.5.1	Comparison of <code>timeVaryingMixed</code> Boundary Conditions with WRF results	83
3.3.5.2	Effects of Complex Terrain	84
3.3.5.3	Velocity Profiles at MP5 Met-Mast and Diurnal Cycle	94
3.3.5.4	Effects of Coriolis Acceleration	97
3.3.5.5	Effects of Surface Roughness	99
3.3.5.6	Comparison with WRF and Observa- tion Data	103
3.3.5.7	Sensitivity Analysis on the WRF res- olution	109
3.3.6	Daily Wind Power Production Estimation	115
4	CONCLUSION	123
	CURRICULUM VITAE	133

LIST OF TABLES

TABLES

Table 2.1	OpenFOAM base unit dimensions	35
Table 3.1	Location and velocity of the air particles at the exit plane	66
Table 3.2	WRF model settings	74
Table 3.3	Effects of the Coriolis acceleration at the MP5 met-mast location . .	99
Table 3.4	Corine roughness classes used	101
Table 3.5	Effects of surface roughness at the MP5 met-mast location	101
Table 3.6	Effects of roughness at the locations with different surface roughness	102
Table 3.7	rRMSE of the wind speed predicted by WRF and the coupled Open- FOAM solutions with respect to the met-mast data	108
Table 3.8	rRMSE of the wind direction predicted by WRF and the coupled OpenFOAM solutions with respect to the met-mast data	109
Table 3.9	rRMSE of wind speed predictions with respect to the met-mast data	114
Table 3.10	RMSE of wind direction predictions with respect to the met-mast data	114

LIST OF FIGURES

FIGURES

Figure 1.1 Atmospheric Boundary Layer representation by night and profiles for flow variables [45, 26]	5
Figure 1.2 Atmospheric Boundary Layer representation by night and profiles for flow variables [45, 26]	6
Figure 1.3 Sectoral solutions used in wind atlas methodology (left) and lin- earised model used in WASP (right)	11
Figure 1.4 WRF η coordinate system	13
Figure 1.5 OpenFOAM C++ library	14
Figure 2.1 Coupling OpenFOAM solution with WRF	20
Figure 2.2 Numerical modelling for WRF coupled OpenFOAM solutions	20
Figure 2.3 Simplified flowchart of WRF coupled OpenFOAM solutions	22
Figure 2.4 Flowchart of WRF solutions	23
Figure 2.5 Flowchart of mesh pre-processing (MeshPrep)	24
Figure 2.6 Flowchart of interface program between WRF and OpenFOAM . . .	25
Figure 2.7 Horizontal (left) and vertical(right) Arakawa C-type staggered grid	26
Figure 2.8 Nested Arakawa C-type staggered grid	27
Figure 2.9 Trilinear interpolation	28
Figure 2.10 WRF and OpenFOAM solution domains and close-up views	28

Figure 2.11 Stretching method	29
Figure 2.12 Example of an extracted BC profile from WRF	29
Figure 2.13 Cause of pseudo-shifting in ground-level	30
Figure 2.14 Before - after modification	30
Figure 2.15 OpenFOAM class structure	31
Figure 2.16 OpenFOAM case structure	33
Figure 2.17 timevaryingmixed BC class algorithm	40
Figure 2.18 Parallel case structure	44
Figure 2.19 Domain decompositions for a case with TimeVaryingMixed BC using the 1st and 2nd approaches	48
Figure 2.20 Components of the Earth's rotational velocity	60
Figure 2.21 PimpleFOAM algorithm and modification on UEqn.H	62
Figure 3.1 Solution domain for the validation of the Coriolis acceleration im- plementation	64
Figure 3.2 Streamlines under the effect of the Coriolis acceleration	64
Figure 3.3 Streamlines based on the OpenFOAM solutions and the Euler in- tegration	65
Figure 3.4 WRF and OpenFOAM solution domains	67
Figure 3.5 Topography of the Mut region and the modelled ground surface . .	67
Figure 3.6 Computational grid for OpenFOAM solutions in the Mut region . .	68
Figure 3.7 OpenFOAM (right) vs WRF (left) solutions (inflow)	68
Figure 3.8 OpenFOAM (right) vs WRF (left) solutions (outflow)	68
Figure 3.9 Unsteady OpenFOAM solutions (6, 11,18 ,24th Hour)	69

Figure 3.10 Unsteady OpenFOAM solutions (top) and WRF solutions (bottom) at 6, 11, 18, 24th hour	69
Figure 3.11 Velocity contours and streamlines on terrain following surfaces 25m and 100m above the ground at a)≈6 b)≈12 c)≈18 d)≈24 hours . . .	70
Figure 3.12 Vertical slice for comparison with/out Coriolis acceleration	71
Figure 3.13 Wind speed distribution with (a) and without (b) Coriolis acceleration and the difference (c)	72
Figure 3.14 Domain decomposition by METIS (left 8, right 10 partitions) . . .	72
Figure 3.15 Efficiency of parallel computations	72
Figure 3.16 Alaiz mountain range	74
Figure 3.17 Nested domains for WRF solutions (4 nests (left), 3 nests (right)) .	75
Figure 3.18 Alaiz topography and modelled ground surface	76
Figure 3.19 Computational grid for OpenFOAM solutions in the Alaiz region .	77
Figure 3.20 Grid resolutions in the Alaiz region	78
Figure 3.21 Effect of the vertical grid resolution on the velocity profile (left) and the zoomed view (right)	79
Figure 3.22 Effect of horizontal resolution on the velocity profiles (left) and zoomed view (right)	79
Figure 3.23 Velocity profiles predicted by the turbulence models	81
Figure 3.24 Domain decomposition for the Alaiz region with 8 and 16 partitions	82
Figure 3.25 Inflow/outflow velocity profiles in OpenFOAM and WRF solutions	84
Figure 3.26 Vertical and horizontal cuts in the OpenFOAM solution domain . .	85
Figure 3.27 Wind speed and turbulent kinetic energy distributions in the vertical cut	86

Figure 3.28 Streamlines colored with velocity magnitude on the vertical cut . .	87
Figure 3.29 Streamlines colored with velocity magnitude on the vertical cut . .	88
Figure 3.30 Streamlines colored with velocity magnitude on the horizontal cut .	89
Figure 3.31 Streamlines colored with velocity magnitude on the horizontal cut .	90
Figure 3.32 Streamlines colored with velocity magnitude on the horizontal cut .	91
Figure 3.33 Streamlines colored with velocity magnitude and the 3D representation of the flow field	92
Figure 3.34 Streamlines colored with velocity magnitude and the 3D representation of the flow field	93
Figure 3.35 Velocity profiles at the met-mast location before(left) and after(right) mid-day	95
Figure 3.36 Velocity profile predicted by OpenFOAM solution at about 5 o'clock (left) and the diurnal cycle description for the night-time (right)	96
Figure 3.37 Velocity profile predicted by the OpenFOAM solution at about 12 o'clock (left) and the diurnal cycle description for the night-time (right) . .	96
Figure 3.38 Wind direction convention for the met-mast data	97
Figure 3.39 Wind speed and direction time-series at the MP5 met-mast location 118 <i>m</i> and 40 <i>m</i> above the ground with and w/o the Coriolis acceleration .	98
Figure 3.40 Corine roughness and roughness distribution in the Alaiz region . .	100
Figure 3.41 Wind speed and direction time-series 118 <i>m</i> and 40 <i>m</i> above the ground at the MP5 mast location with the smooth and the rough terrains .	100
Figure 3.42 Wind speed and direction time-series 118 <i>m</i> and 40 <i>m</i> above the ground at Point A with the smooth and the rough terrains	102
Figure 3.43 Wind speed and direction time-series at 118 <i>m</i> above the ground . .	103
Figure 3.44 Wind speed and direction time-series at 102 <i>m</i> above the ground . .	104

Figure 3.45 Wind speed and direction time-series at 90m above the ground . . .	105
Figure 3.46 Wind speed and direction time-series at 78m above the ground . . .	106
Figure 3.47 Wind speed time-series at 40m above the ground	107
Figure 3.48 Wind speed and direction time-series 118m above the ground at the met-mast location	110
Figure 3.49 Wind speed and direction time-series 102m above the ground at the met-mast location	111
Figure 3.50 Wind speed and direction time-series 90m above the ground at the met-mast location	112
Figure 3.51 Wind speed and direction time-series 78m above the ground at the met-mast location	113
Figure 3.52 Wind speed time-series 40m above the ground at the met-mast location	113
Figure 3.53 Power curve of a generic wind turbine	115
Figure 3.54 Unsteady OpenFOAM solutions coupled with WRF at 40m above the ground	117
Figure 3.55 Wind power density at 40m above the ground	117
Figure 3.56 Wind power density at 90m above the ground	118
Figure 3.57 Wind power density at 118m above the ground	118
Figure 3.58 Daily wind power production estimation at 40m above the ground .	120
Figure 3.59 Wind turbines in the Alaiz region	120
Figure 3.60 Daily wind power production estimation at 90m above the ground	121
Figure 3.61 Daily wind power production estimation at 118m above the ground	121

LIST OF ABBREVIATIONS

ABL	Atmospheric Boundary Layer
CFD	Computational Fluid Dynamics
METIS	Serial Graph Partitioning and Fill-reducing Matrix Ordering
NWP	Numerical Weather Prediction
WAsP	Wind Atlas Analysis and Application Program
WRF	Weather Research Forecast
(U)RANS	(Unsteady) Reynolds-Averaged Navier-Stokes
u, v, w	Velocity in Easting, Northing, vertical direction in WRF
U	Velocity vector in <i>OpenFOAM</i>
p	Static pressure in <i>OpenFOAM</i>
ϕ	Mass flow rate in <i>OpenFOAM</i>
τ_w	Wall shear stress
ρ	Density
μ	Dynamic viscosity
μ_t	Dynamic turbulent viscosity
ν	Kinematic viscosity
ν_t	Turbulent kinematic viscosity
ϵ	Turbulent dissipation
P_k	Turbulent kinetic energy production
k	Turbulent kinetic energy
S	Modulus of the mean rate-of-strain tensor
$\overline{\Omega_{ij}}$	Mean rate-of-rotation tensor in a rotating reference frame
y^+	Non-dimensional distance from the wall
K_s	Roughness height

C_s	Roughness constant
K_s^+	Non-dimensional roughness height
φ	Latitude
R_o	Rossby number

CHAPTER 1

INTRODUCTION

The increase in energy demand all around the world due to industrialization and the pollution that comes with it lead the governments and investors towards the direction of renewable and clean, green energy. As one of the most significant sources of green energy, wind and efficient ways to harness it, are currently among the main research topics[17, 29].

Site-selection for wind farms/turbines and wind resource assessment are of the main branches of wind energy research. The aim of wind resource assessment is simply to predict the power output and best positions for the installation of wind turbines. As the initial investment cost of wind farms is quite high including preliminary surveys, met-mast installations and turbine installations; wind resource assessment is of significant importance [19, 18].

One of main problems that gradually pop-up for the wind assessment studies is the inaccuracy in complex terrains. The past experiences in the operation of commercial wind farms have confirmed that effects from a local complex terrain on the site characteristics of each turbine have a significant influence on the power output and perhaps even on the viability of a wind energy project[11, 7, 12, 38, 40].

Most of the commercial wind assessment tools either use linearised models or computational fluid dynamics (CFD) models, which employ Wind Atlas Methodology [49] based on meteorological mast data. A more detailed approach is to use numerical weather prediction (NWP) models. Further up in the complexity ladder, there are a few NWP coupled CFD or LES models which have the most extensive modelling of physics and are computationally the most expensive ones.

Even if the wind energy is clean and renewable, once wind turbines are installed, there are still some issues to overcome before they can be utilized reliably. The main problems of using wind power in large scales are its high variability, limited predictability, limited dispatchability, and non-storability. A coal power plant, for example, can be turned on or off, and can adjust its output to the demand. Wind power, however, cannot be controlled by power system operators because wind farms cannot increase their power generation upon request when there is not sufficient wind. Wind farms can only reduce the output. Also, wind cannot be stored like coal, natural gas, or atoms for future power generation [56].

Therefore, in order to utilize the generated wind power efficiently, accurate and reliable short-term predictions are needed, even more so for countries with a high wind power potential [32]. While the widely used linear or CFD based statistical wind prediction models are fairly useful for annual wind energy predictions, their accuracy is not adequate for short-term wind predictions as these models work with statistical averages and don't take into account the time dependent variation in the wind speed. Moreover, wind forecasting errors are magnified by the fact that wind power increases with the cube of the wind speed [32].

In order to understand and cope with the problems in the current wind prediction models, understanding the underlying principles behind wind energy is essential. In this section some background information about the wind energy is given along with the objective of this work and the tools to be used.

1.1 Atmospheric Boundary Layer and Wind

The "Atmospheric Boundary Layer", ABL, (also called the Planetary Boundary Layer, PBL) can be defined as the part of the atmosphere which is highly influenced by the Earth's surface and responds to the forcing from the surface within a time scale about an hour [45]. This layer's thickness and structure vary diurnally along with the flow variables within such as velocity, temperature and pressure [26]. The thickness of the ABL ranges from $3km$ down to $100m$ in high latitude (north-south poles) and usually considered to be in the order of $1km$.

There are 3 main forcing mechanisms that effect the characteristics of ABL.

- Pressure gradient force
- Surface frictional force
- Buoyancy force

Forcing due to pressure gradient \vec{F}_p is straight forward;

$$\vec{F}_p = -\frac{1}{\rho}\vec{\nabla}p \quad (1.1)$$

Surface frictional force \vec{F}_f is defined as follows

$$\vec{F}_f = -a\vec{V} \quad (1.2)$$

where a is the friction parameter that depends on the surface roughness. While this equation looks simple, there are 2 important ramifications: The friction force acts in the opposite direction to the wind, thus retards the flow. The parameter a is not easily determined as it depends on the wind speed and the surface roughness.

Atmospheric boundary layers can be divided into 4 sublayers:

- The roughness sub-layer: the thin layer whose height is in the order of roughness elements such as man-made structures, plants, trees etc..
- The surface layer: which is also called constant flux layer, is around 100 m thick. The flow characteristics such as wind velocity, temperature, pressure greatly varies with altitude and the turbulence characteristics are influenced by the topography mostly. Vertical heat and momentum fluxes are approximately constant in this layer.
- The well-mixed layer: Flow characteristics are relatively constant with altitude but Coriolis effects from the Earth's rotation are more effective as the wind velocity is higher than in the surface layer. It is the buffer layer between the capping inversion layer.

- Capping inversion layer: Also called as the inversion layer, this layer acts as a bridge between the well-mixed buffer layer and the free atmosphere in which the geostrophic wind conditions apply.

The buoyancy force is generated due to the density changes in the ABL. For an air particle with a differential volume, the gravitational force acting on it can be given as:

$$\vec{F}_b = (\rho_0 + \Delta\rho) \vec{g} \quad (1.3)$$

where ρ_0 is the density of the air particle, $\Delta\rho$ is the density change and \vec{g} is the gravitational acceleration. Thus, when the density changes within the differential volume, gravitational force acting on the particle becomes different resulting in a net force which is the buoyancy force. It is important to note that once ρ and ρ_0 are equal no buoyancy force is generated. Buoyancy is closely related to atmospheric stability. Stability is defined using the concept of potential temperature, θ , which is defined as the temperature that an unsaturated parcel of dry air would have if brought adiabatically and reversibly from its initial state to the ground pressure, p_0 . Potential temperature can be calculated as:

$$\theta = T \left(\frac{p_0}{p} \right)^{R/c_p} \quad (1.4)$$

where T is the temperature, p_0 is the pressure at the surface, p is the pressure at the level where temperature is measured, R is the universal gas constant and c_p is the specific heat of air. By its definition, potential temperature θ is a quantity that does not change if you move an air parcel up or down adiabatically.

There are 3 types of atmospheric flow characteristics (stability conditions) :

- Stable Atmosphere: This condition is where $\frac{d\theta}{dz} < 0$. For this condition, the heat flux from the surface to the air in the vicinity is negative, ($H < 0$). This means that the air in the vicinity of the ground is being cooled by the ground, increasing the air's density. This results in the air being stagnant in altitude as the buoyancy acts towards the ground on it.

- **Neutral Atmosphere:** This condition is where $\frac{d\theta}{dz} = 0$. For this condition, the heat flux from the surface to the air in the vicinity is zero, ($H=0$). This means that there is no cooling or heating effect from the ground, so the effect of buoyancy is negligible.
- **Unstable Atmosphere:** This condition is where $\frac{d\theta}{dz} > 0$. For this condition to occur, the heat flux from the surface to the air in the vicinity is positive, ($H>0$), meaning that the ground is warming up the air in the vicinity. As the air gets warmer, its density decreases causing an upwards airflow due to buoyancy.

A typical representation of the atmospheric boundary layer for day and night is given in Figure 1.1 and Figure 1.2. As seen in these figures, all flow characteristics change diurnally.

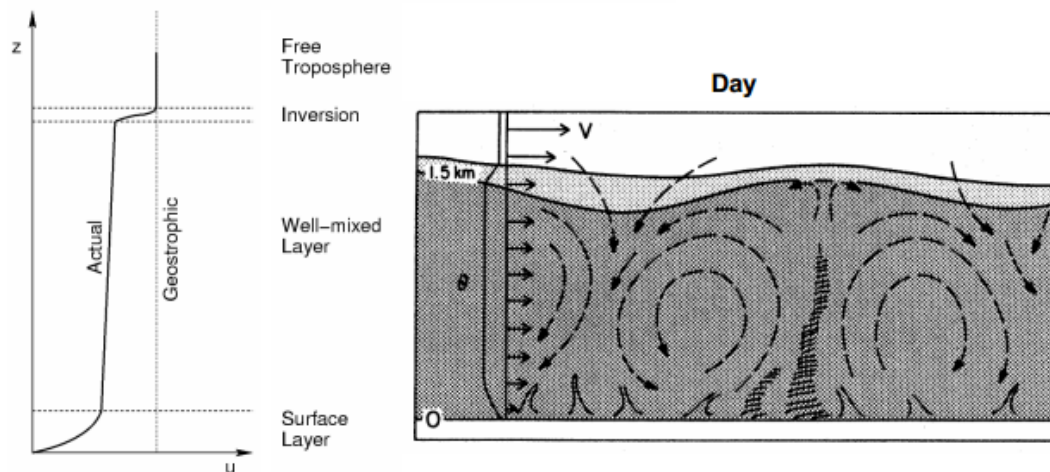


Figure 1.1: Atmospheric Boundary Layer representation by night and profiles for flow variables [45, 26]

As the radiation from the Sun hits earth's surface, it is absorbed by the ground and heats the atmosphere. This results in the heating of the atmosphere, more intense near the ground. As the density of air near the ground decreases, air moves up and a vertical highly turbulent flow is generated by buoyancy as shown in Figure 1.1. Inversely, at night as the air cools down, its density increases near the ground, no significant vertical flow is generated and atmosphere becomes stable as shown in Figure 1.2

In the surface layer, as the vertical heat and momentum fluxes are negligible, wind

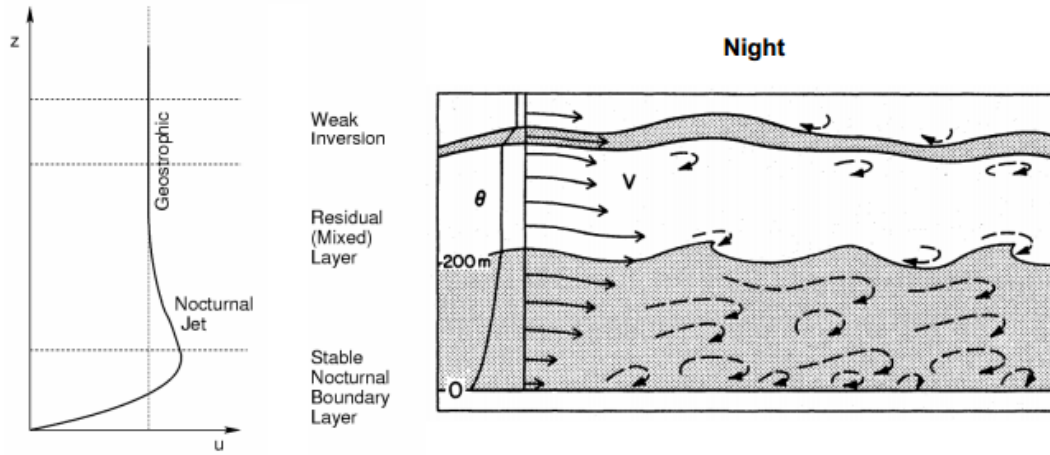


Figure 1.2: Atmospheric Boundary Layer representation by night and profiles for flow variables [45, 26]

profiles can usually be approximated by two methods: logarithmic wind profile and power law.

In logarithmic wind profile:

$$u(z) = \frac{u^*}{\kappa} \ln \left(\frac{z}{z_0} \right) \quad (1.5)$$

where u^* is the friction velocity and κ is the von Karman constant as mentioned earlier. Note that the friction velocity u^* and the shear stress τ are defined as:

$$u^* = \sqrt{\frac{\tau}{\rho}}, \quad \tau = \mu \frac{du}{dz} \quad (1.6)$$

As seen in Equations 1.5 and 1.6 that the approximated wind profile depends on the rate of change of velocity with respect to altitude, $\frac{du}{dz}$. So, turbulence plays an important role in its characteristics.

While the logarithmic wind profile embodies some physics behind [30], for a more straightforward approximation of wind profile, the power law is used:

$$u(z) = u_r \left(\frac{z}{z_r} \right)^\alpha \quad (1.7)$$

where u_r is the reference wind speed at altitude z_r and α is the wind shear exponent which is determined based on experimental data.

In short, the physics of the Atmospheric Boundary Layer suggests 4 important issues to be considered.

- Atmospheric Boundary Layer's thickness can be assumed to be around $1km$. Since there are a great number of disturbances such as roughness elements, topography, heating/cooling from the ground, stability etc, atmospheric flows are essentially turbulent.
- As most of the wind turbines reside within 200 meters from the ground level, it is safe to assume that the part of the atmospheric boundary layer of interest is the surface layer (AKA constant flux layer). Since the flow properties and turbulence characteristics vary abruptly and are mostly influenced by the topography in this layer, high resolution terrain modelling is of utmost importance.
- While the wind profile can be approximated by logarithmic profiles in the surface layer where most of the wind turbines reside in, abrupt changes in flow direction and speed occurs both in time and space because of the disturbances terrain features create in the upcoming flow.
- Characteristics of the ABL changes diurnally due to the heating from the Sun. In order to correctly model this behaviour of the ABL, unsteady models should be used along with unsteady boundary conditions.

1.2 Wind Power Prediction/Estimation Models

Several factors are considered in the traditional wind power prediction and the siting of wind turbines:

- On-site meteorological observation data: Traditional wind power prediction methods all require wind speed observation data which is obtainable from meteorological masts (met-mast). As the even the annual wind is highly variable,

on-site measurements that are taken for less than 5 years may lead to highly inaccurate annual energy predictions.

- Extrapolation of met-mast data to turbine hub height: As the cost of met-masts usually go up exponentially with height, extrapolation of met-mast data to turbine heights are usually done using logarithmic wind profiles or power law which were given in the "Atmospheric Boundary Layer and Wind" section.
- Complex terrain and roughness effects: Complex topography such as hills, mountains, valleys changes the wind flow characteristics drastically. Also, changes in the terrain types (such as woodlands into irrigated lands) and sheltering objects are usually modelled as variable roughness elements. Effects of roughness on the wind flow is significant especially in surface layer of the ABL. [14]
- Power predictions with wind turbine power curves: Wind turbine power gives the estimated power generation according to hub height wind speeds. Accurate power curves for the turbines to be installed are obviously necessary in order to calculate the power output of a wind turbine. But even if the annual averages or wind speed distributions are correctly estimated as the wind varies abruptly with time, accurate short-term predictions are really hard to come by.
- Power losses: There are several factors that hinders the power generated by a turbine such as wake losses from other turbines, high wind speed shutdowns, etc.. that each of which requires attention.

Although there are several factors to take into account for wind power prediction, scope of this study is focused on the accurate determination of time dependent wind fields. Thus, most of the discussions in the following pages will be about that.

Currently, most of the commercial wind assessment tools either use linearised models or CFD based models, that use the Wind Atlas Methodology [49] based on measurements from meteorological masts. The main idea in Wind Atlas Methodology is to obtain 12 or more steady sectoral wind field solutions that sweeps 360° for a region that contains the location of measurements as seen in Figure 1.3 (left) and correlate these sectoral solutions to reconstruct the time averaged flow field estimations

based on the point observation data. Linearised and CFD based models differs in the methodology to obtain the sectoral solutions. The reconstructed wind field is then used for the micro-siting of the wind turbines and the annual energy production estimation of the wind farm. It is important to note that both CFD based and linearised models depend on observation data that is collected for at least a year (usually 3-5 years) to yield reasonable results for wind speed distributions meaning that at least a years worth of time and money is lost. A more detailed approach is to use NWP models which generates time dependent flow fields instead of the statistical flow reconstruction methods. Also, there are a few NWP coupled CFD or LES models which have the most extensive modelling of physics and computationally the most expensive ones.

Beaucage [4], compared a linearised model (*WAsP*), a CFD based model (*MeteodynWT*), a mass consistent model that downscales NWP results (*SiteWind*) and NWP coupled LES model (*ARPS*) at four different sites with different climate conditions. The statistical wind reconstructions based on sectoral wind solutions of *WAsP*, *Meteodyn*, *SiteWind* are compared with averaged results of 72 days samples equally weighted for each month in a 5 year period obtained from time dependent *ARPS* model. He concluded that the *SiteWind* and *ARPS* model produced the lowest root mean square error (RMSE) as measured between the predicted and observed mean wind speeds. The RMSE for the linear Jackson-Hunt type model was 29% greater than the coupled NWP models and for the RANS model 58% greater than the coupled NWP models. The key advantage of the coupled NWP models appears to be their ability to simulate the unsteadiness of the flow as well as phenomena due to atmospheric stability and other thermal effects. It should be noted that using time averaged velocity for error can be misinterpreted as the positive and negative differences may cancel out to give a low mean error. [30]. It is also observed that all four numerical models performed better in flat terrain than in complex terrain.

In the following subsections, 3 types of commonly used models (linearised, CFD based and NWP models, namely) are described in detail.

1.2.1 Linearized Models

Linearised models are based on the early study by Jackson and Hunt [23] and known accordingly as Jackson and Hunt models. Linearised models employ fast, linearised solvers assuming sheltering obstacles, roughness and mountainous topography as small perturbations on a constant background wind field. Using point wind data measurement(s) for long period (1-10 years) generalized regional wind data is derived. Reversing this process, wind climatology of a specific location is obtained. A basic flowchart of the process is given in Figure 1.3(right). The wind atlas data set then can be subsequently applied for estimation of the wind climate and wind power potential, as well as for siting of specific wind turbines[34].

WAsP (Wind Atlas Analysis and Application Program) from Risø is the most popular one among these linearised flow models that use wind atlas methodology [34]. The *WAsP* model, like other linearised models such as *MS3DJH/MsMicro3* [47, 50] and *MSFD* [5], uses linearised equation sets that are developed for neutrally stable, attached flow. While the computational effort is negligible with respect to CFD based or NWP models, linearised models cannot simulate flow features such as separation and recirculation, thus their application is limited to simple geometries. Bowen [10] stated that *WAsP* prediction errors may be significant if the local climate or terrain lie outside its normal operational envelope and specified a terrain slope of 0.3 after which the prediction errors become large due to flow separation. Ayotte's study [3] stated that a slope higher than 0.2 even on relatively smooth surfaces can produce significant errors for mean wind speed estimates using linearised model, *MSFD*. Politis, [37] in his study about turbine wake effects in complex terrains, concluded that the velocity deficit and the wake geometry cannot be predicted by a linear superposition of the terrain and the wake effect as the presence of complex topography affects the wake development in a complicated way.

1.2.2 CFD Based Models

CFD based power prediction models are used in order to overcome the inaccuracies in complex terrains by using Reynolds Averaged Navier Stokes (RANS) solutions for

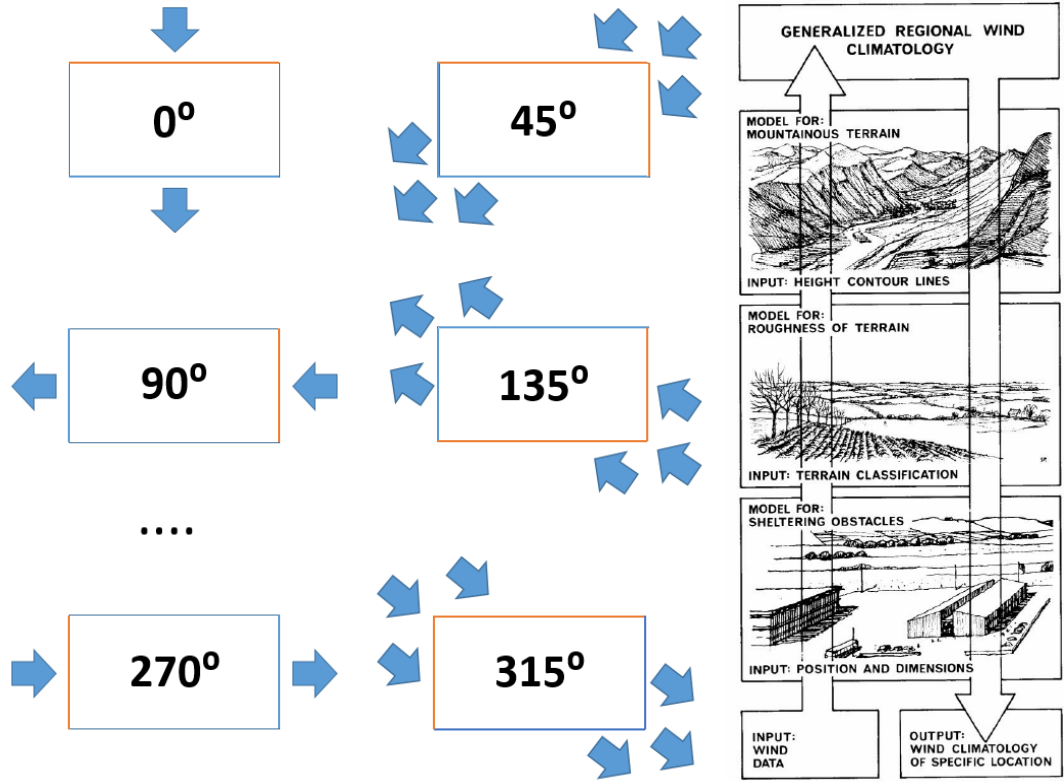


Figure 1.3: Sectoral solutions used in wind atlas methodology (left) and linearised model used in WAsP (right)

steady sectoral flow field solutions to sweep over 360° . These sectoral solutions are obtained by giving steady and spatially uniform logarithmic wind profiles as inflow velocity boundary conditions and giving the opposite sides as outlets as given with red and blue lines respectively in Figure 1.3(left). Usually other sides (if any) are given as symmetry BCs. Sectoral solutions are then interpolated to reconstruct the time averaged flow field estimations based on the point observation data obtainable from met-masts.

Ayotte [3] stated that RANS based *Raptor_{NL}* model significantly improved the results with respect to linear model *MSFD* for slopes upto 0.4 for smooth and 0.3 for rough surfaces but there is a significant deviation from measurement on the leeward side for hills for slopes higher than that. As the flow reconstruction methodology in CFD based models is based on the interpolation of sectoral solutions and obtaining wind distributions for time averaged flow fields, complex terrain effects such as separation are somewhat smoothed out. As an example, the direction of the wakes behind obstructing features such as mountains change with respect to the inflow wind

directions in sectoral solutions. As all sector solutions are used in the interpolations to reconstruct the flow field, effects of each sectors' wakes smooth out the averaged results.

Aside from the inaccuracies in complex terrain, both linearised and CFD based models' outputs are statistical annual averages, meaning that accurate short term predictions are out-of question as there is no time dependent wind flow information and since wind power is proportional to the cube of wind speed. On the other hand, a short term energy production estimation of a wind farm requires a more accurate time dependent wind field estimation, which is mostly based on numerical weather prediction models such as WRF .

1.2.3 Numerical Weather Prediction Software: Weather Research Forecast, WRF

WRF is a fully compressible, Eulerian, η -coordinate based, nest-able, non-hydrostatic, meso-scale numerical weather prediction (NWP) model with a large suite of options for numerical schemes and parametrization of physical processes [41, 52]. Atmospheric physics such as cloud formations, rain, snow etc. can be included using parametrizations defined in the model. Being a non-hydrostatic model, atmospheric stability and buoyancy is accounted for. Meso-scale models are used for weather phenomenon with a length scale of $2 - 2000km$ and has a maximum feasible resolution of $1km$.

WRF solution is based on the η coordinate in the vertical direction. The vertical coordinate, η , is defined as:

$$\eta = \frac{p - p_{ht}}{p_{hs} - p_{ht}} \quad (1.8)$$

where p is the pressure, p_{hs} is the surface pressure and p_{ht} is the pressure at the top of the model. As seen in Figure 1.4, the η coordinate system causes a poor representation of the surface topography and as of that does not include viscous effects, which are dominant in the surface layer of the ABL, properly.

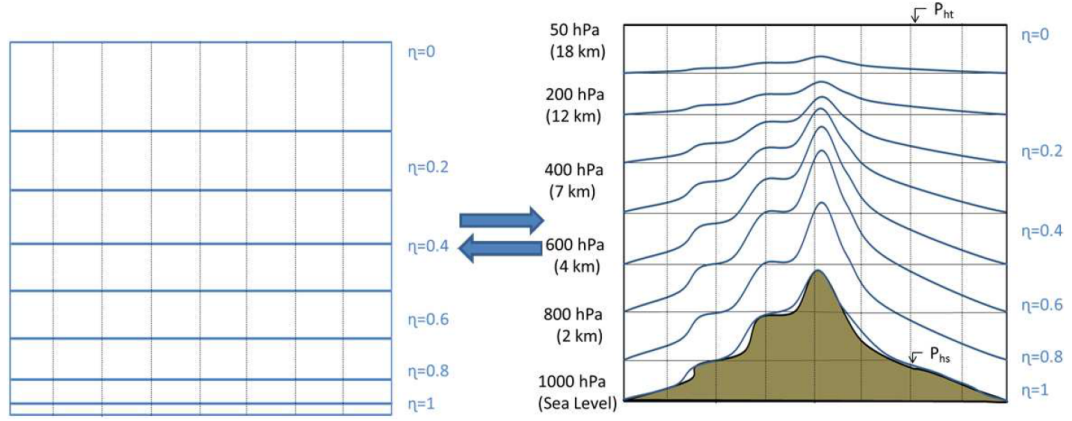


Figure 1.4: WRF η coordinate system

WRF needs basically 2 inputs, which are the terrain features, the initial and boundary conditions for a region of interest. The local terrain data can be obtained from UCAR (University Corporation of Atmospheric Research) server via WRF whereas the time dependent initial and boundary conditions for the WRF solution may be obtained from ECMWF (European Centre of Medium Range Weather Forecast).

WRF can be run in nests which are the smaller domains that derive the initial and boundary conditions from their parent domain. WRF has a nesting ratio of 1:3 meaning that horizontal resolution can be increased 3 times with each smaller nest, but at the expense of storage space and computational resources. It is important to note that the numerical schemes for WRF are explicit thus subject to the CFL stability condition. This makes increasing the horizontal resolution a demanding work as increasing the resolution results in decreasing the time step size in order for the numerical schemes to be stable. As an example increasing the horizontal resolution by nesting (1:3 scale) increases the number of cells by 9 times, decreases the allowable time stepsize to 1/3 resulting in 27 times more storage space and computational time [35].

NWP models like WRF are used in meso-scales ($1 - 2000\text{km}$) and downscaled to turbine hub heights in micro-scale models (1km or less) for a wind power prediction. Downscaling usually involves the usage of logarithmic wind profiles for average wind fields at a given altitude near the ground. Even if the WRF provides the time dependent wind field information, there are several reports of inaccuracy in the near ground levels. Zhang [55] reported that while WRF's solutions are accurate in simple terrains,

its accuracy in complex terrains suffers from both the models inability to reproduce accurate atmospheric conditions in the lower atmosphere and the representative issues due to mismatches between the model and actual terrain and increasing the vertical resolution may not help with latter issue. A study by Jimenez [25] reveals that the wind speed is underestimated at the sites located at mountain tops and hills, and it is overestimated at plains and valleys by WRF . A study conducted by Wang for the low-level winds simulated by WRF shows an overestimation of low-level winds. [51].

1.3 OpenFOAM Open Source Library

OpenFOAM (Open Field Operation and Manipulation) is a free, open source C++ library for Computational Fluid Dynamics which has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics and electro-magnetics [21]. It includes tools for meshing, pre- and post-processing. All the features of the software (solvers, meshing, pre-post processing) are able to run in parallel. The overall structure of *OpenFOAM* is shown in Figure 1.5.

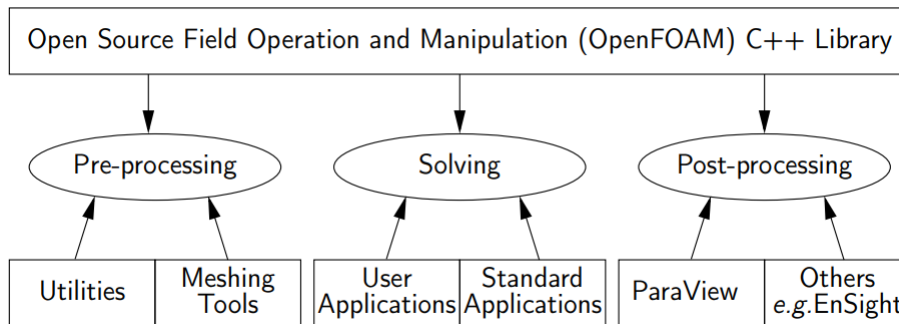


Figure 1.5: OpenFOAM C++ library

As an open source software, *OpenFOAM* is fully customizable. The capabilities of the software can be improved or changed to fit the requirements of the user. Since all the functions (numerical methods, meshing, physical models) are compiled in the form of shared libraries, new executables can easily be linked with these shared libraries.

OpenFOAM uses a syntax representation which is very similar to the mathematical equation to be solved by the help of Object Oriented Programming and Operator Overloading [24]. For example, the conservation of momentum equation:

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p$$

```
solve
(
  fvm::ddt(rho,U)
+  fvm::div(phi,U)
-  fvm::laplacian(mu,U)
==
-  fvc::grad(p)
);
```

can be represented as simple as the code snippet given in the grey-box. With features such as operator overloading that comes with the C++ language, it is relatively easy for users to implement additional terms into the equations to be solved or define their own algorithms.

OpenFOAM is first and foremost a C++ library with well over a million lines of code. It has many contributors and is developing at an exponential rate but as the structure of the code is complex and a proper documentation is lacking, an efficient usage of this tool might become tiresome.

OpenFOAM is becoming widely used for wind research, recently. Most of the studies related to wind energy with *OpenFOAM* are focused on modelling of wind turbines' wakes [43, 44, 28]. There are some studies that take into account the ABL stability effects [16] which is an unsteady phenomena but even then, the common practice is to use RANS or LES with steady and uniform BCs derived from logarithmic wind velocity profiles as the main focus is to understand the effects ABL stability generates on turbine wakes.

OpenFOAM is also used in slightly more uncommon subjects which are simulations of wind fields [46] or wind farms [22] on complex terrains for the estimation of wind power. The boundary conditions are in general given with steady logarithmic velocity profiles.

There exist a few applications that couple NWP models with a Navier-Stokes solver but the coupling is done either by using body force terms in the micro-scale solver and imposing periodic boundary conditions [20, 54] or by using spatially varying steady

velocity profiles derived from a NWP model. The derivation of the steady velocity profiles are based on either time-averaging NWP predictions on the outer boundaries of the CFD domain or on the instantaneous NWP prediction [9, 54]. There are no definitive reports on successful coupling through both unsteady and spatially varying BCs derived from a NWP model.

OpenFOAM is chosen as the solver of choice for this work because of its useful features and being open source for relatively easy access for modifications.

1.4 Objectives

The main objective of the study is to improve the accuracy of the current wind power prediction models by developing a coupled flow solution methodology based on the atmospheric weather forecast software, WRF, and the open source Navier-Stokes solver, *OpenFOAM*.

The current wind power assessment/prediction models can be given within 3 groups with advantages and disadvantages summarized below:

• Linearized Models

Advantages:

- Fastest Models
- Reliable in smooth, simple topography
- Accurate for annual predictions

Disadvantages:

- Assumes attached flow, cannot simulate flow separation, ..
- Not accurate in complex terrain
- No unsteady physics

• CFD based Models

Advantages:

- Still fast, slower than linearized models
- More accurate than linearised models in complex terrain
- Accurate for annual predictions

Disadvantages:

- Flow reconstruction methodology by sector solutions smooths out complex terrain effects to some extent
- No unsteady physics

● **Numerical Weather Prediction Models**

Advantages:

- Unsteady physics are taken into account
- Several models for ABL, stability, weather events etc...

Disadvantages:

- Meso-scale model, horizontal resolution is low.
- Explicit Model, hard to increase resolution, relatively slower than other two
- Because of η coordinate assumption and low resolution, not accurate in the vicinity of complex terrains.

In order to improve the short-comings of current wind power prediction models while keeping the methodology computationally feasible 3 important requirements are identified:

- CFD based solutions on terrain fitted high resolution grids: A high horizontal resolution is needed to resolve complex terrains and Navier-Stokes solutions are proved to be good at simulating complex terrain effects such as flow separation, recirculation etc...
- Coupling CFD solutions with a weather prediction model: Unsteady atmospheric physics can be accounted for via coupling WRF with an unsteady Navier-stokes solver.
- Modelling real-time unsteady atmospheric flows accurately rather than using

statistical data. In order to achieve this requirement, computations should take much less time than the predicted phenomena. A parallel implementation of the methodology is therefore needed.

The methodology developed for unsteady *OpenFOAM* solutions coupled with WRF to satisfy these requirements is given in the following chapter. Two validation case studies, one in Mersin/Mut region in Turkey and another one in Navarre/Pamplona region in Spain are presented in the results. Final remarks and future work related to the methodology are given in the Conclusion chapter.

CHAPTER 2

METHODOLOGY

In this work, a coupled atmospheric flow solution methodology, which uses the predictions of the numerical weather prediction model WRF on a low resolution grid to provide the unsteady boundary conditions for *OpenFOAM* solutions on a high resolution terrain fitted grid, is developed. The unsteady WRF solution is first obtained over the geographical domain of interest. Then, a high resolution grid for the coupled *OpenFOAM* solution is generated using a high resolution terrain model. Once the *OpenFOAM* solution domain is generated, the cell center coordinates of the outer boundary cells are extracted for the interpolation of the flow variables from the WRF solution. The unsteady boundary conditions for the *OpenFOAM* solution are then interpolated tri-linearly in space and linearly in time from the WRF solution. A new BC class named `timeVaryingMixed` is developed and added to the *OpenFOAM* source tree to implement the interpolation of the coupled boundary conditions in a parallel computing environment. Unsteady coupled atmospheric solutions are then performed in parallel using *OpenFOAM*'s incompressible flow solver, `pimpleFOAM` with the modifications for the inclusion of the Coriolis acceleration, and the spatially varying surface roughness data.

2.1 Problem Definition and Numerical Model

The main steps to achieve the objectives of the study are:

- To obtain CFD based atmospheric flow solutions on terrain fitted high resolution grids

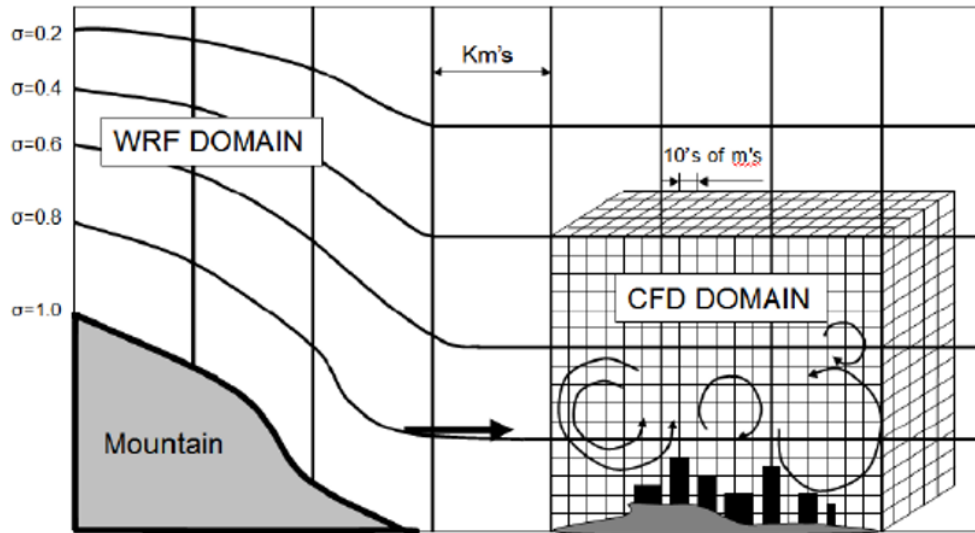


Figure 2.1: Coupling OpenFOAM solution with WRF

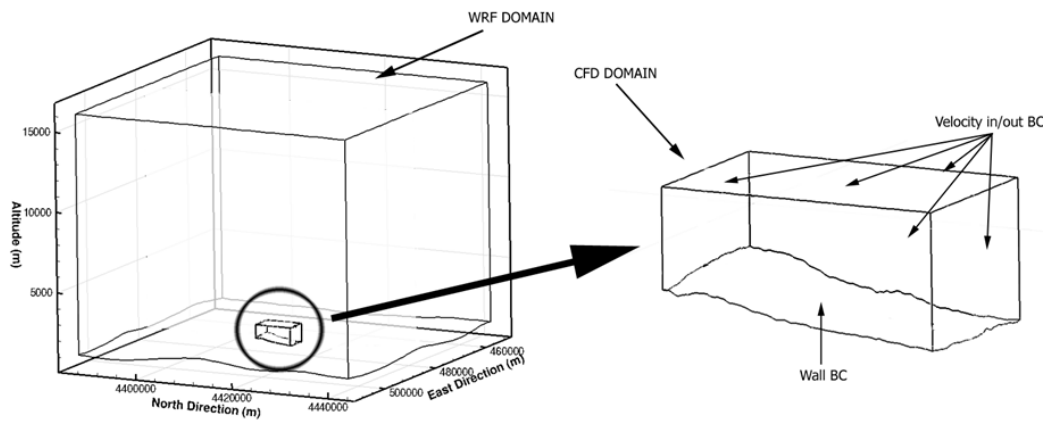


Figure 2.2: Numerical modelling for WRF coupled OpenFOAM solutions

- To couple the unsteady CFD solutions with a weather prediction model

The numerical model for the problem is sketched Figures 2.1 and 2.2; where the coarser domain outside is the low resolution WRF solution domain whereas the finer domain in the middle is the *OpenFOAM* domain.

In the coupled *OpenFOAM* solutions, incompressible fluid assumption is made due to the fact that the wind speed is on the order of 10 m/s. As a result, the conservation of mass and momentum equations are decoupled from the conservation of energy equation. In addition, a neutral atmosphere is assumed and the heating of the Earth's surface and heat transfer effects are neglected.

For the pressure velocity coupling, the `pimpleFOAM` algorithm which is a combination of `SIMPLE` and `PISO` is used. `pimpleFOAM` employs multiple iterative cycles over the same time step and is known to be quite robust for relatively large time steps. The time stepping in `pimpleFOAM` is dynamic and is based on Courant number. `pimpleFOAM` is modified to include the effects of Coriolis acceleration for a solution on a rotating reference frame.

Usage of WRF predictions as boundary conditions for the *OpenFOAM* solution domain accounts for the unsteady variations in the flow field outside *OpenFOAM* domain and provides accurate, more realistic boundary conditions. The spatially and time varying boundary conditions are imposed on the side and the top boundaries of the *OpenFOAM* solution domain. Even though a neutral atmosphere is assumed and heat transfer and buoyancy effects are neglected in *OpenFOAM* solutions, these effects are partially introduced in the solution through the inflow boundary conditions extracted from WRF. In order to prevent the divergence of the *OpenFOAM* solutions due to a possible violation of the conservation of mass, gradients of the flow variables at the outflow boundaries are set to zero. The new boundary condition class developed `timeVaryingMixed` implements the Dirichlet and the Neumann type boundary conditions dynamically depending on the velocity field.

A high resolution topography with spatially varying roughness model is used to account for the surface canopy on the terrain surface. Rough wall functions available within *OpenFOAM* are used along with the spatially varying roughness model for the near-wall treatment.

A parallel implementation of the developed methodology is achieved to reduce the computational time to feasible levels for real-time, short-term wind energy forecasting.

A simplified flowchart for the overall process is given in Figure 2.3. Sections with dashed lines are explained in detail in their respective sections.

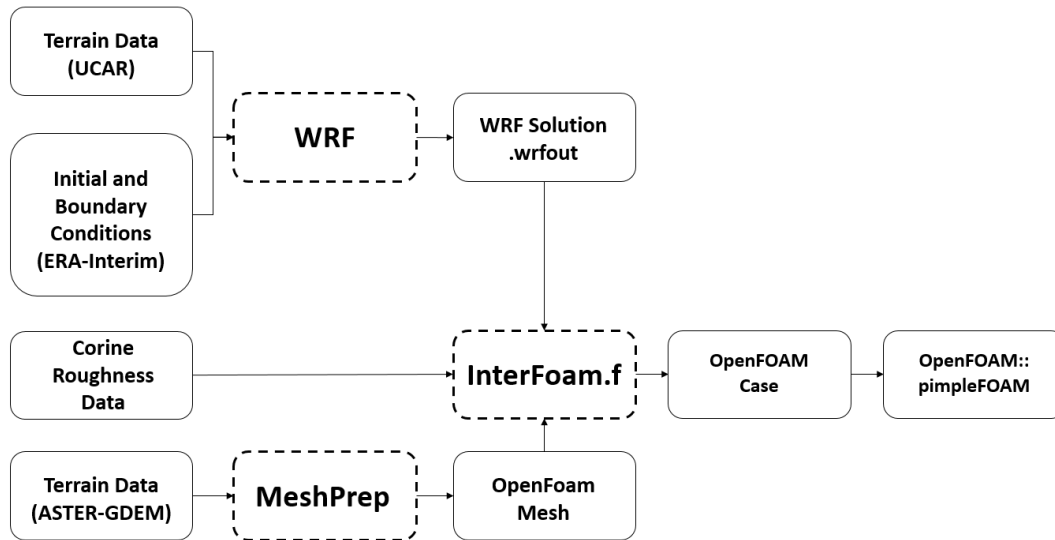


Figure 2.3: Simplified flowchart of WRF coupled OpenFOAM solutions

2.2 WRF Solutions

WRF needs basically 3 inputs which are terrain data, initial and boundary conditions for a region of interest. A detailed flowchart for WRF solutions are given in Figure 2.4.

A static data set covering all the world, which is available in the WRF website [1], is used for the terrain data. Terrain data includes terrain height, land use category, soil categories, ... etc. This data set is provided mostly by UCAR (University Corporation of Atmospheric Research).

Time dependent initial and boundary conditions for the WRF solution are obtainable from ECMWF (European Centre of Medium Range Weather Forecast). ERA-interim reanalysis data from ECMWF is used in this study. This data is provided in .grib format.

Each part of the pre-processing codes (geogrid, ungrib and metgrid) reads parameters from a file, Namelist.input where the bounds of the WRF domain and its nests, *eta* levels for the WRF run are defined along with many physical options.

The purpose of *geogrid* is to define the simulation domains, and interpolate various terrestrial data sets to the model grids. *geogrid* will interpolate soil categories, land

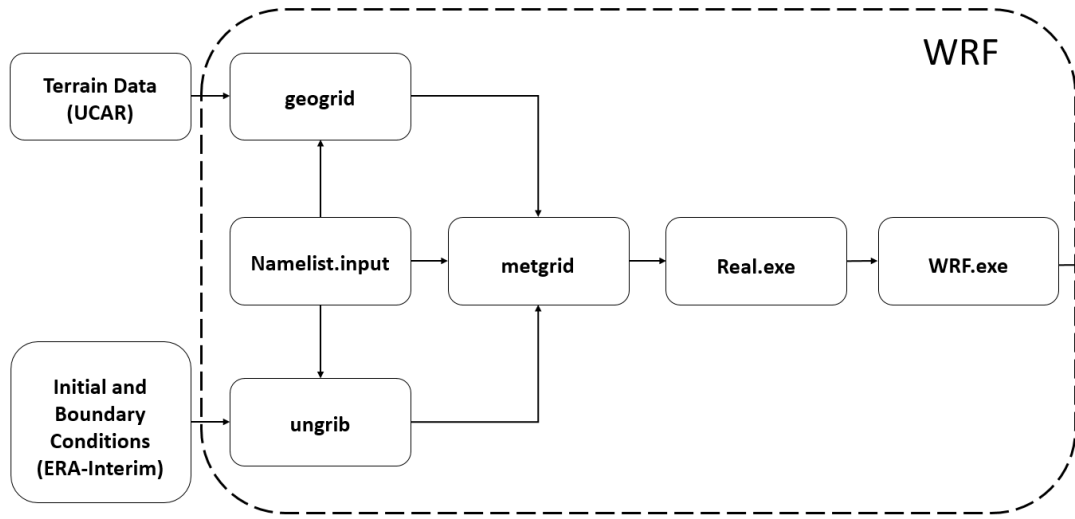


Figure 2.4: Flowchart of WRF solutions

use category, terrain height, annual mean deep soil temperature, monthly vegetation fraction, monthly albedo, maximum snow albedo, and slope category to the model grids.

The *ungrib* program reads GRIB files, degribes the data, and extracts meteorological fields into an intermediate data format. There are 2 separate mandatory grib files which are the surface level data and η level data.

After both processes are done, metgrid program horizontally interpolates the intermediate format meteorological data that are extracted by the ungrib program onto the simulation domains defined by the geogrid program. The interpolated metgrid output can then be ingested by the real.exe program.

real.exe program vertically interpolates meteorological fields to WRF eta levels, finishing the preprocessing.

2.3 High Resolution Topography Modelling and Grid Generation

The computational grid generation for *OpenFOAM* solutions are based on a high resolution terrain topography obtained from ASTER GDEM, which is a product of METI and NASA. The worldwide elevation data has about 1.5 arc-sec spatial resolution

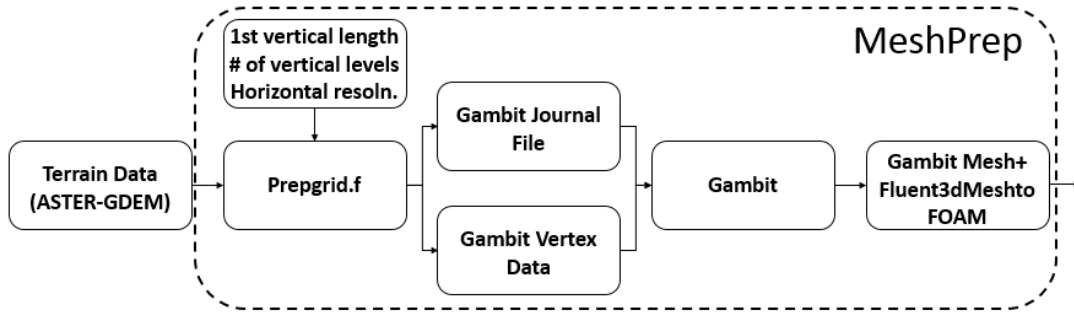


Figure 2.5: Flowchart of mesh pre-processing (MeshPrep)

which is approximately 30 meters. *GAMBIT* software (ANSYS-Fluent) is used to create structured stretching grids based on a *GAMBIT* journal file created automatically by the mesh pre-processing program. A detailed flowchart of the mesh pre-processing procedure is given in Figure 2.5.

ASTER-GDEM data is read by the *Prepgrid.f* along with the 1st vertical length of the structured grids, number of vertical levels and horizontal *OpenFOAM* grid resolution. *Prepgrid.f* converts the terrain height data from ASTER-GDEM into *GAMBIT* Vertex Data format and creating the journal file for *GAMBIT* to generate the structured grids.

The algorithm for the journal file is as follows: Horizontal resolution, first length in vertical direction and number of vertical levels are provided by the journal file. Then, the vertex data file is read and a model for the topography is generated via *GAMBIT*'s Import Vertex Data and Create Surface from Vertex Rows options using information for the nodes at ground level. After that, a flat surface, which is a projection of the topography at the desired altitude is generated. These two surface are then connected via 4 edges to bound the solution domain. First, the upper and lower boundaries of the solution domain are meshed using the given horizontal resolution. After that, the edges connecting these two surface are meshed such that the first length of each edge must be equal to that of others and also the number of points on each edge must be equal. In order to do that, a size function with a fixed starting length is fitted to all 4 connecting edges by the journal file which also restricts the number of points for the edge mesh.

Using this approach, the vertical resolution can be increased as much as desired, independently of the horizontal resolution and without increasing the number of cells exponentially as structured mesh is used.

GAMBIT is then used to generate rapidly stretching grids in the vertical direction, using a journal file automatically generated for given vertical and horizontal resolutions, as explained previously. After the mesh generation is done in *GAMBIT*, resulting grids are converted to *OpenFOAM* format using the intrinsic *OpenFOAM* function `fluent3DMeshToFoam`.

2.4 Extraction of Boundary Condition Profiles from WRF

For the extraction of boundary condition profiles for the high resolution *OpenFOAM* solution domain from the low resolution WRF data, an interface program, *interfoam.f* is written. A detailed flowchart for *interfoam.f* is given in Figure 2.6.

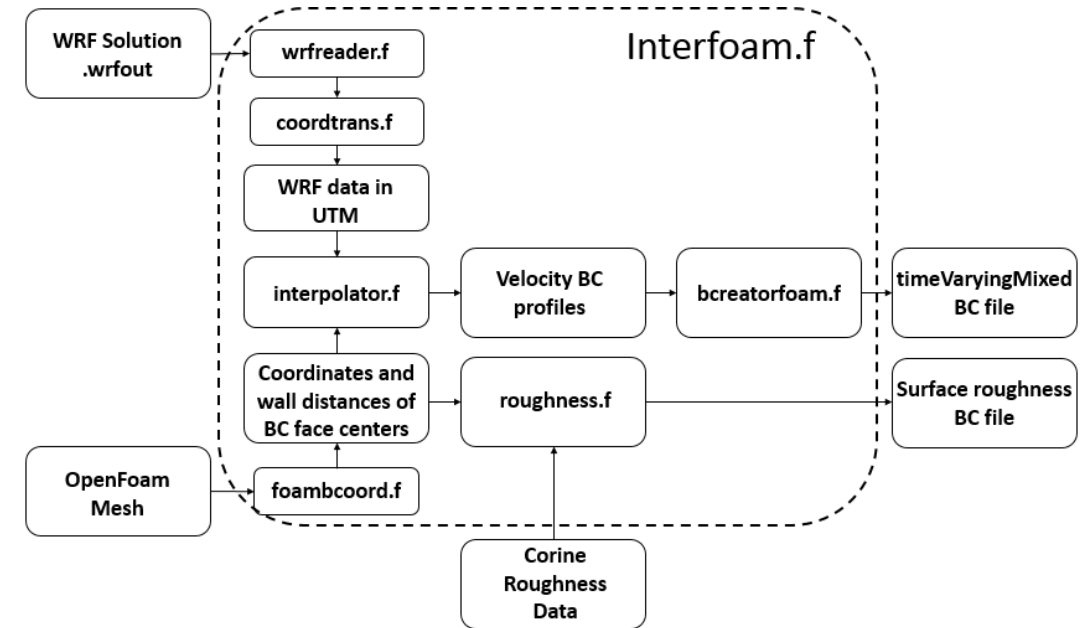


Figure 2.6: Flowchart of interface program between WRF and OpenFOAM

OpenFOAM mesh is read by *foambcoord.f* to extract boundary faces cell center coordinates of *OpenFOAM* domain along with their cell wall distances.

In the meantime output of the WRF solutions is read and manipulated by *wrfreader.f*. WRF uses C-type Arakawa staggered grids in order to increase the resolution and ac-

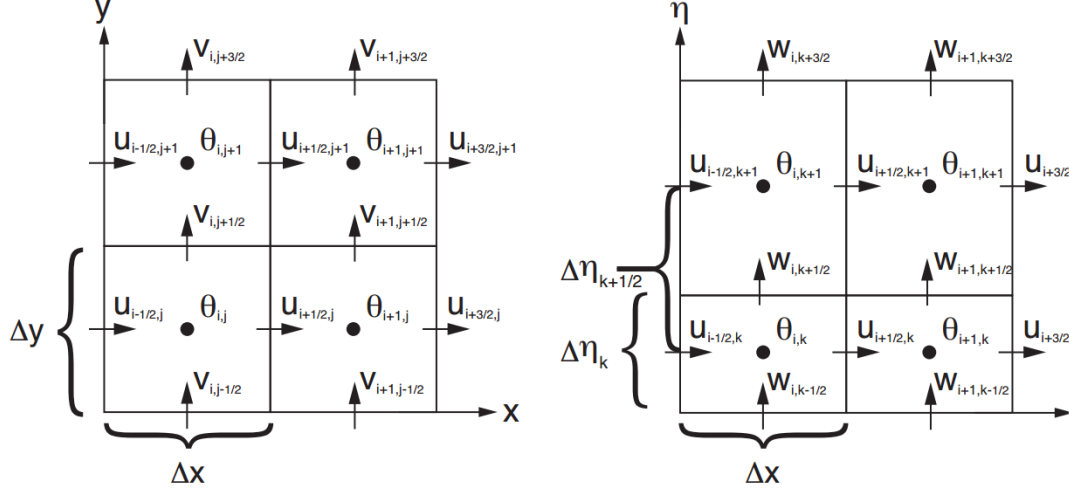


Figure 2.7: Horizontal (left) and vertical(right) Arakawa C-type staggered grid

curacy of solutions [41]. In this type of grids, u components are evaluated at the centers of the left and right grid faces whereas the v components are evaluated at the centers of the northern and southern grid faces. Non-vectorial quantities (θ) such as density, temperature are defined in the cell centers as shown in Figure 2.7. Nests of WRF also use this convention while increasing horizontal resolution as shown in Figure 2.8.

To use WRF data for the extraction of BC profiles, all variables are interpolated to cell centers of WRF cells and a new structured grid convention that takes these WRF cell centers as nodes is created by *wrfreader.f*.

As the WRF uses Latitude, Longitude which are given in degrees and non-dimensional η levels as a coordinate system, these coordinates are converted into Northing, Easting in the Universal Transverse Mercator (UTM) coordinate system and Altitude which are all in meters by *coordtrans.f*. For all transformations WGS84 (World Geodetic System 1984) datum is used.

Once the new grid structure from the WRF data in UTM and *OpenFOAM* domains boundary face centers' coordinates along with their wall distances are extracted, these are passed down to *interpolator.f* to generate BC profiles at each WRF output interval for each boundary face center in the *OpenFOAM* domain.

For the tri-linear interpolation of WRF variables the code developed in Gökhan Ahmet's PhD thesis [2] is used in *interpolator.f*. Mesh distribution along altitudinal

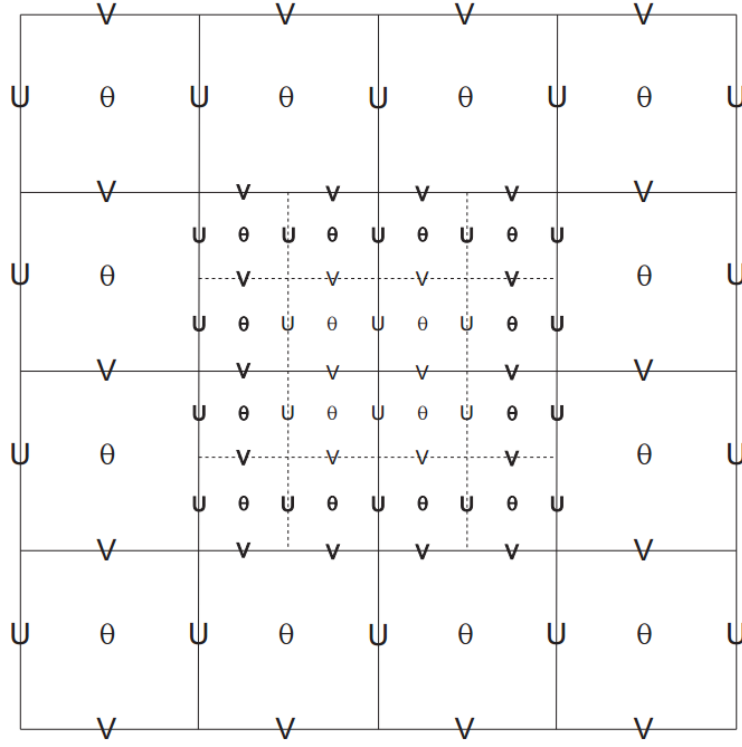


Figure 2.8: Nested Arakawa C-type staggered grid

direction, latitude and longitude values are kept constant. Yet, demanded point in the solution domain is firstly searched in the latitudinal and longitudinal direction by means of i and j value simultaneously to improve calculation time. Then nearest k value is searched starting from the node that is found previously. So, the nearest node in the flow field is defined.

Using the index of the nearest node, the values of latitude longitude and elevation of surrounding nodes are compared and the cell in which the demanded point resides in is found. After finding the 8 corners of the octagonal grid tri-linear interpolation is performed. Calculation can be shown simply in Figure 2.9 and Equation 2.1.

It should be noted that WRF has a horizontal resolution of $1km$ and a vertical resolution of about $30m$ on the ground which stretches rapidly. In addition, as shown in Figure 2.10, the surface boundaries in the WRF and *OpenFOAM* domains differ significantly mainly due to the high resolution topographic data used in the generation of the *OpenFOAM* domain, and due to the η coordinate system employed in WRF .

As the resolutions of the WRF and high resolution topography differs, even if all terrain data in both are correct, ground levels of WRF and *OpenFOAM* grids may not

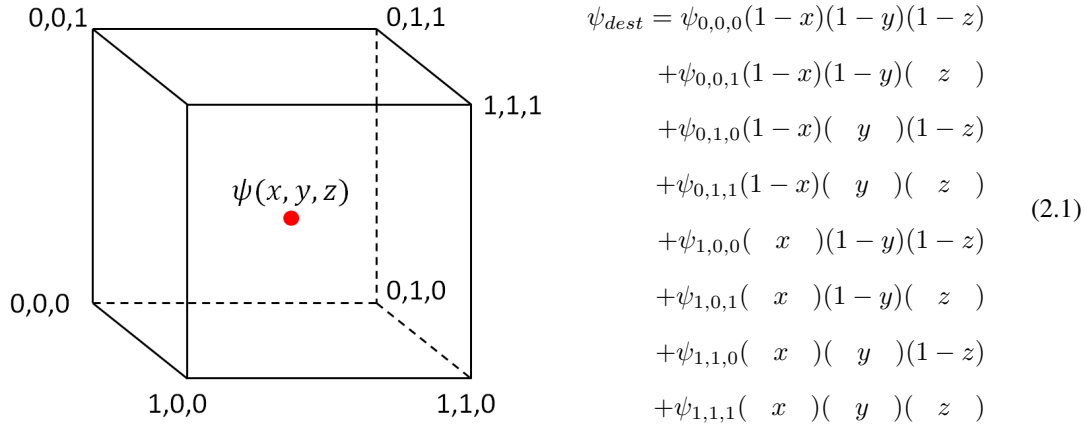


Figure 2.9: Trilinear interpolation

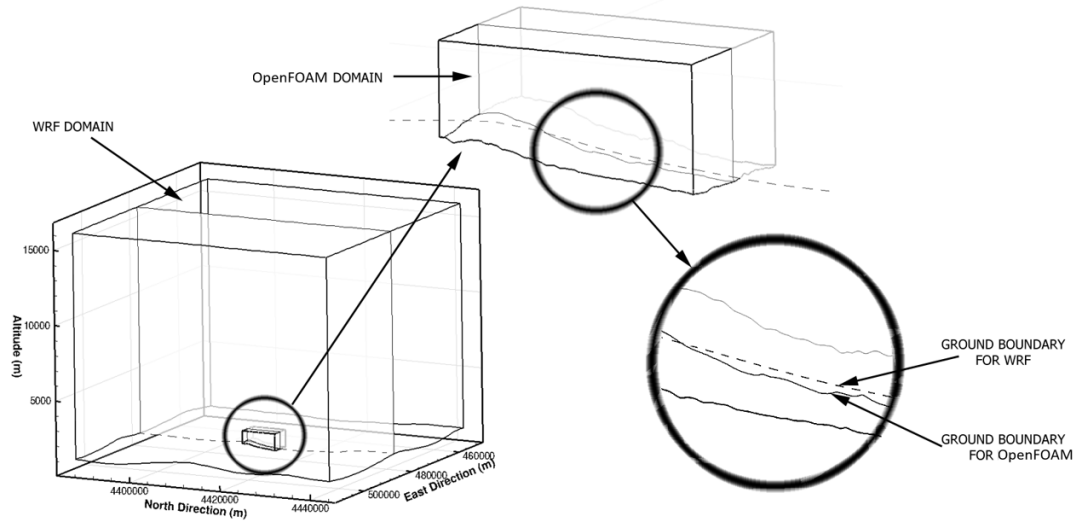


Figure 2.10: WRF and OpenFOAM solution domains and close-up views

match as shown in Figure 2.10. For avoiding undefined regions or pseudo-shiftup of the ground level, stretching methodology developed in in Gökhan Ahmet's PhD thesis [2] is used with several modifications for high slope topography for matching ground levels.

In the stretching approach, finding the extreme differences between WRF and *OpenFOAM* ground levels, a limit altitude is defined. Keeping the velocities identical for the higher altitudes of this limit, for the lower region, velocity profile is stretched to equalize the ground levels of domains and the limit. With this "stretching approach", which is shown in Figure 2.11, upper region of the limit, velocities are directly taken from WRF unchanged. For the lower region, distances between ground levels of each

solver and the limit are calculated. Using these distances, velocities are redistributed proportionally along this line.

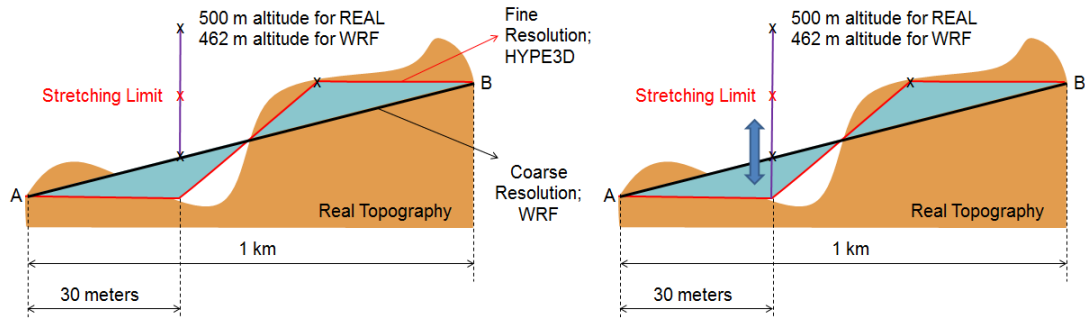


Figure 2.11: Stretching method

One example of the boundary condition profiles extracted from WRF on a high resolution *OpenFOAM* mesh with this interpolation method is given in Figure 2.12

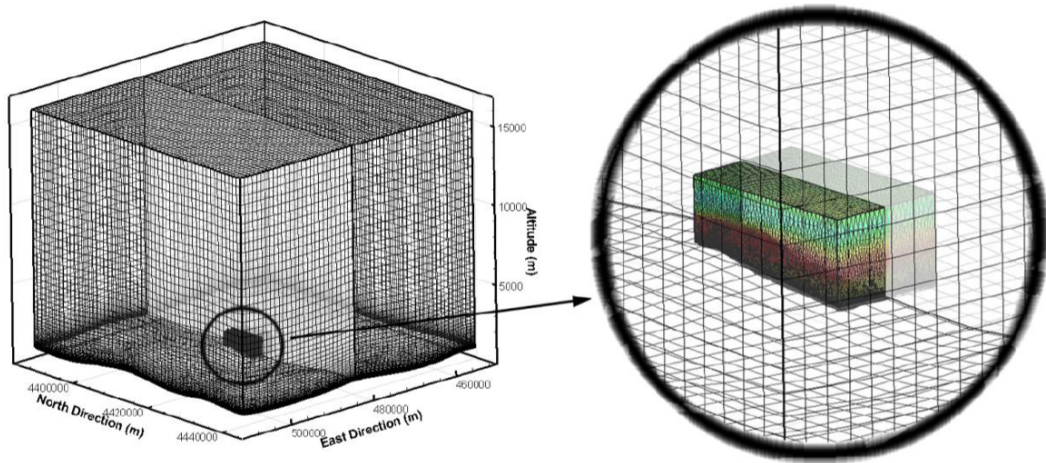


Figure 2.12: Example of an extracted BC profile from WRF

While the method performs adequately in simple geometries, there are some discrepancies in steep slopes because of the methodology to find which cell a point resides in. The methodology to find which cell a demanded point is in:

- Find the nearest node i and j
- Find the nearest node in k
- Check the elevations of $(i,j,k-1)$ $(i,j,k+1)$ nodes
- Decide which WRF cell the point resides in

To understand the pseudo-shifting in ground-level due to this algorithm, suppose that the demanded point is a little over the top surface of a skewed cell as shown Figure 2.13. Nearest nodes in i and j indexes are found as the grey dots in WRF domain (left). Checking the elevations of $(i,j,k-1)$ $(i,j,k+1)$ nodes, the algorithm decides that the point is between (k) th and $(k-1)$ th index, while in reality it is between (k) th and $(k+1)$ th index resulting in the shift in in boundary condition profiles as shown in Figure 2.14 (left). In order to remedy this, demanded point in x and y coordinates is projected onto each k th level and the its altitude at each k th level are calculated using tri-linear interpolation of k th level nodes' altitudes given with grey dots in Figure 2.13(right). Decision of the cell in which the demanded point resides, is done using the altitude of these projections. Resulting boundary condition profile after the modification is shown in 2.14 (right).

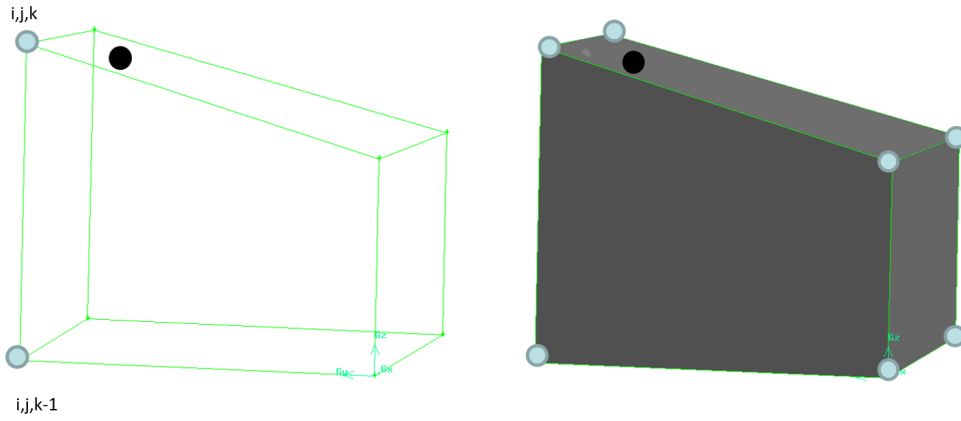


Figure 2.13: Cause of pseudo-shifting in ground-level

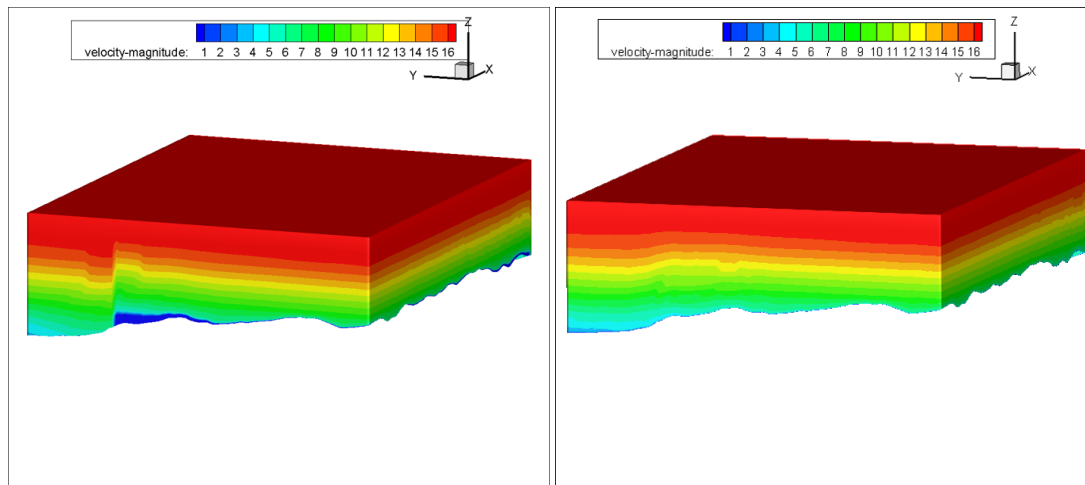


Figure 2.14: Before - after modification

2.5 OpenFOAM Flow Solver

OpenFOAM library is composed of a large number of classes. Classes in *OpenFOAM* as well as in C++, are defined using constructors and member functions belonging to the class and should contain at least 2 files. The *.c* extension file is used to define the class, whereas the *.h* extension one is usually used to define the member functions and objects used in the class and the objects or the member functions' s accessibility. A schematic of an *OpenFOAM* class is given in Figure 2.15

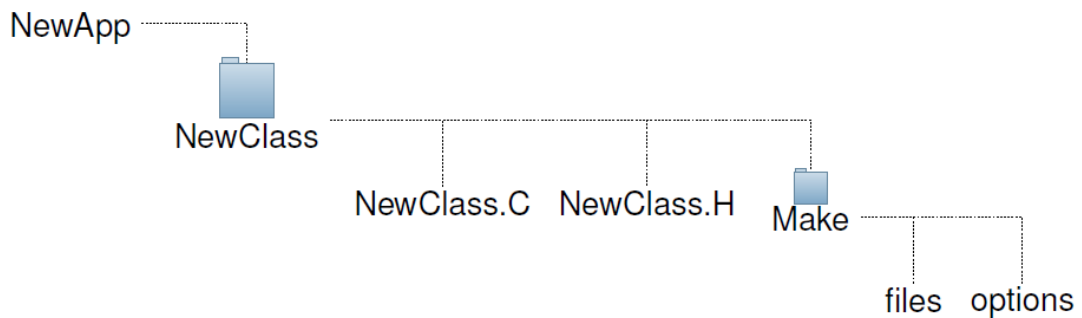


Figure 2.15: OpenFOAM class structure

Usage of a class in another class is provided by using `#include otherclass.h` line. For example, a custom boundary condition class can be defined using main BC classes such as `fixedValue` (class where variables are defined on the face centers), `FixedGradient` (class where gradient of variables are defined on the face centers) or `mixed` (where both classes are used simultaneously). For using a modified version of an existing class, dependencies of the existing class must be removed first using the *OpenFOAM* function `wclean`.

`Make/option` file, defines the names and paths to the previously existent header files `header.h` that belongs to classes other than the developed class. In `Make/option` file, paths that contain header files are defined as `-I <path>` whereas for libraries `-L <libraryPath>` command is used. In `Make/file` file, the developed class's name and its path is defined in a similar fashion.

User defined libraries are compiled into their separate shared library using `wmake libso` command which is *OpenFOAM*'s application based on `make` for libraries. A shared library generated this way is easily accessible by adding

`#include sharedlib.so` line.

More detailed information about *OpenFOAM* classes will be given in the section where implementation of the newly developed boundary condition class `timeVaryingMixed` is explained.

2.5.1 File Structure in an OpenFOAM Solution

For a flow simulation, *OpenFOAM* needs 3 separate folders which are `system/`, `constant/` and `0/` namely. The file structure of an *OpenFOAM* case file is given in Figure 2.16. The `system/` folder and files underneath holds the information about the numerical method and input/output settings for the case. In the `constant` folder properties related to the solution domain and the fluid(s) to be worked with are given. In `0/` folder, boundary conditions for the problem are given separately for each flow variable. Main case folders and files underneath for an *OpenFOAM* simulation are given in a more detailed sense below:

- **System/:** contains `controlDict`, `fvSchemes` and `fvSolution` files

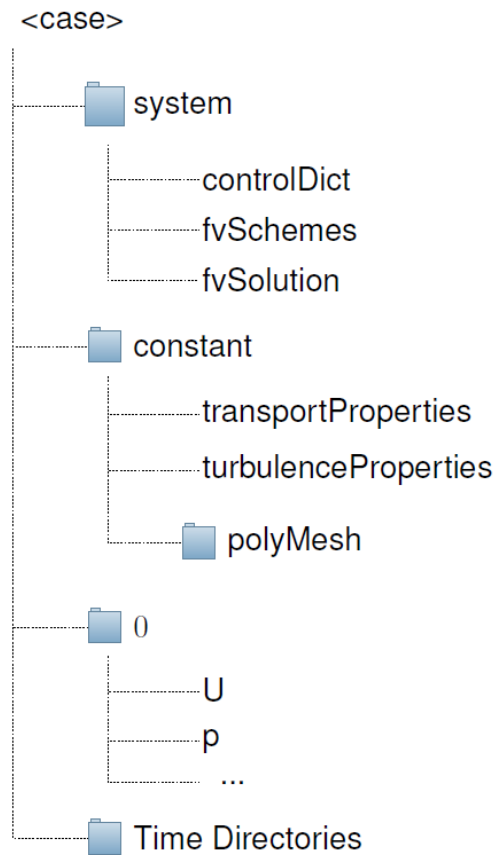


Figure 2.16: OpenFOAM case structure

- `controlDict`: In this file, input/output and time step parameters such as which solver to use, time step period, output period per time step, start-finish times of the simulation, etc .. are defined. User generated shared libraries to be used in the simulation are specified here.
- `fvSchemes`: In this file, numerical schemes to be used in the solver are given explicitly such as using which numerical schemes to calculate laplacian of U , divergence of p , etc ..
- `fvSolution`: Solver parameters such as convergence criteria, under-relaxation factors for variables, etc .. are defined in this file.

- **Constant/**: This folder consist of `transportProperties`, `turbulenceProperties` files and `polyMesh` folder:

- `transportProperties`: In this file, the physical properties of the fluid to be worked with such as kinematic viscosity, heat capacity are given.
- `turbulenceProperties`: This file specifies the turbulence model to be used and defines the coefficients for the turbulence model.
- `polyMesh/` folder: Data related to the solution domain and its mesh are stored in separate files such as *points*, *edges*, *faces*, *cells* and *boundaries* in this folder.

- **0/:** In this folder, boundary and initial conditions for each flow variable are defined in separate files such as ($p, U, vs.$). As an example:

```
FoamFile
{
    version 2.0;
    format ascii;
    class volVectorField;
    object U;
}
dimensions [0 1 -1 0 0 0 0];
internalField uniform (-2.78225 -6.6085 -0.391);
boundaryField
{
    inlet
    {
        type inletOutlet;
        inletValue nonuniform List<vector>
            14794
        (
            ( 0.0000000E+00 0.0000000E+00 0.0000000E+00 )
            ( 0.0000000E+00 0.0000000E+00 0.0000000E+00 )
            ...  })
    }
```

For this boundary condition file, `volVectorField` entry specifies that the variable is a volumetric vector quantity whereas `object U` entry defines which flow variable is considered. `dimensions` and `InternalField` objects that are defined in `volVectorField` class specifies the dimensions for the variable (m/s for this case) and gives the initial conditions for the volumetric vector quantity U , respectively. Dimensions of variables which are determined based on Table 2.1 are given with `dimension` entry.

`BoundaryField` entry is used to define the boundary conditions for U variable. In this example, `inletOutlet` boundary condition class is used for a patch named "inlet" which was defined on `constant/polyMesh` folder. A keyword entry belonging to `inletOutlet` class, namely `inletValue`, is of `nonuniform List <vector>` template class meaning that each face

Table 2.1: OpenFOAM base unit dimensions

No	Property	SI Unit
1	Mass	kilogram (kg)
2	Length	metre(m)
3	Time	second(s)
4	Temperature	Kelvin (K)
5	Quantity	mole (mol)
6	Current	ampere (A)
7	Luminous intensity	candela (cd)

in "inlet" patch have a different value and these values are all vectors.

`nonuniform List <vector>` class takes the number of entries in the first line (for this case it is the number of faces that belongs to the "inlet" patch 14794) and the vector value of each face in the following lines. As seen in the above example, *OpenFOAM*'s data entries are based on classes and keywords belonging to those classes. This kind of data entries that consist of keywords are called *dictionary*. Detailed information about the `inletOutlet` class is given in the Method section where implementation of the newly developed boundary condition class `timeVaryingMixed` is explained.

- **Time Directories:** These directories share the same format and files as the `0/` directory but instead of boundary and initial conditions, flow solutions at different times are stored in directories named according to the simulation time.

2.5.2 OpenFOAM Boundary Condition Classes

In *OpenFOAM* boundary conditions are defined by C++ classes. To understand the structure of *OpenFOAM* boundary condition classes, a BC class given previously, `inletOutlet`, some of whose properties the developed boundary condition `timeVaryingMixed` inherits, will be investigated.

`inletOutlet` BC class inherits and is closely related to `mixedFvPatchFields` which is one of the main BC classes as mentioned earlier.

2.5.2.1 mixedFvPatchField Class

`mixedFvPatchField` class constructs the flow variable as a linear combination of the flow variable defined at the boundary surface, and the value and gradient of the flow variable given at the cell center with Equation 2.2.

$$x_p = wx_p + (1 - w) \left(x_c + \frac{\nabla \cdot x}{\Delta} \right) \quad (2.2)$$

where x_p is the value at a patch (basically boundary cells face center), x_c is the value at cell center and w is the value fraction. Right hand side term $(x_c + \frac{\nabla \cdot x}{\Delta})$ is the boundary face center value calculated using the cell center value and the gradient of the flow variable. Thus, when $w = 1$ the boundary condition only uses the given value at the face center and disregards the derivative, whereas for $w = 0$ the value at the face center is calculated using only the derivative of the flow variable at cell center and its cell center value, but this time disregarding the given value at the face center.

2.5.2.2 InletOutlet Class

As mentioned earlier, *OpenFOAM* BC classes consist of a `.c` extension file where the main code of the class along with the member operators /functions are defined, and a `.h` extension header file where the accessibility of the class for / to other existing classes are given and the member functions and operators used in `.c` extension file are declared. For simplicity's sake, only the `.c` extension file will be investigated.

`.c` extension file can be examined in three separate parts; constructors, member functions and member operators namely.

- **Constructors:** Constructors, as the name suggests, create the objects to be used in their corresponding classes. While there are several constructors in BC classes, one of the constructors for the `inletOutlet` class is given for simplicity's sake.

```

template <class Type>
Foam::inletOutletFvPatchField<Type>::
inletOutletFvPatchField
(
    const fvPatch& p,
    const DimensionedField<Type, volMesh>& iF,
    const dictionary& dict
)
:
    mixedFvPatchField<Type>(p, iF),
    phiName_(dict.lookupOrDefault<word>("phi", "phi"))
{
    this -> refValue() = Field<Type>("inletValue", dict, p.size());
    this -> refGrad() = pTraits<Type>::zero;
    this -> valueFraction() = 0.0;
}

```

First, `inletOutlet` BC class is defined as a template class with three objects p which is of `fvPatch`, iF of `DimensionedField<Type, volMesh>` and `dict` which is of dictionary classes defined in *OpenFOAM*.

`inletOutlet` inherits the `mixedFvPatchField` BC class which is one the main BC classes. In the following lines, `phiName` object defined in `(mixedFvPatchField<Type>(p, iF))` is assigned to `phi` variable in the solution. `phi` variable is nothing but the mass flow rate entering or exiting the solution domain. Afterwards `refValue()`, `refGrad()` and `valueFraction()` which are member functions belonging to the `mixedFvPatchField` class; are initialized.

`this -> refValue() = Field<Type>("inletValue", dict, p.size());` line assigns the data defined using the `inletValue` keyword from the previously defined dictionary `dict` to the `refValue()` member function for the `mixedFvPatchField` class. Dictionary `dict` is nothing other than the *U* file given in the 0/ folder. The most important thing to note about constructors is that they are executed only once at the beginning of a simulation in *OpenFOAM*.

- **Member functions:** Operations performed in a BC class are done using member functions. The member functions have direct access to all the data members and member functions of the class. The most important member function for BC classes in *OpenFOAM* is the `updateCoeffs()`. This member function applies the boundary conditions for the face centers in a patch and is called at the beginning of each iteration. In this example, `updateCoeffs()` member function (belonging to `inletOutlet` BC class) reads the `phi` variable that `phiName_` variable points as mentioned in the constructors subsection. Read data is now a part of the `inletOutlet` class under the name `phip`. After that this data is passed to `valueFraction()` function which was a part of `mixedFvPatchField` class:

```
// * * * * * Member Functions * * * * *
template<class Type>
void Foam::inletOutletFvPatchField<Type>::updateCoeffs()
{
    if (this->updated())
    {
        return;
    }
    const Field<scalar>& phip = this-> patch().template
        lookupPatchField<surfaceScalarField, scalar>
        (
            phiName_
        );
    this->valueFraction() = 1.0 - pos(phip);
    mixedFvPatchField<Type>::updateCoeffs();
}
```

`(this->valueFraction() = 1.0 - pos(phip);)` `Pos()` function outputs 1 if the function variable is positive and 0 otherwise. `valueFraction` function will be used in `updateCoeffs` member function belonging to `mixedFvPatchField` class for the example.

- **Member operators:** Member operators which are used by member functions are explained in this section. When the member operator of `inletOutlet` class is examined `this->valueFraction()*this->refValue() + (1 - this->valueFraction())*ptf` line is similar to Equation 2.2.

As explained in `MixedFvPatchField` class `valueFraction()` function makes the boundary condition acts as a Neumann or Dirichlet type boundary condition. As previously mentioned, `(this -> valueFraction() = 1.0 - pos(php);)` line makes the `valueFraction()` function 1 if the `phi` variable is negative and 0 if `phi` is positive. `Phi` variable is defined as the mass flow rate in the member function part. So, when there is inflow from the boundary, `Phi` variable becomes negative in turn making the `valuefraction()` function 1. This results in `inletOutlet BC` class acting as Dirichlet BC when the flow is entering the solution domain and Neumann BC otherwise.

```
// * * * * * Member Operators * * * * * //
template<class Type>
void Foam::inletOutletFvPatchField<Type>::operator=
(
    const fvPatchField<Type>& ptf
)
{
    fvPatchField<Type>::operator=
    (
        this->valueFraction()*this->refValue()
        + (1 - this->valueFraction())*ptf
    );
}
```

2.6 Developement of Time and Spatially Varying BC class

In this section , the main algorithm of the developed time and spatially varying BC class that uses WRF data ,`timeVaryingMixed`, is explained. Algorithm for the developed `timeVaryingMixed BC` class is given in Figure 2.17. Light blue boxes in the algorithm stands for preprocessing and data passed down to `timeVaryingMixed`

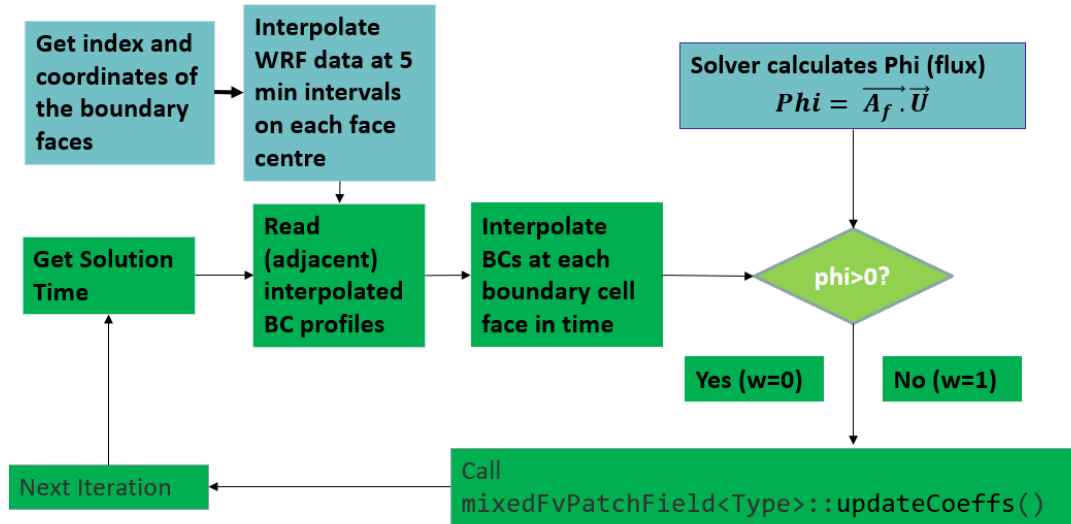


Figure 2.17: timevaryingmixed BC class algorithm

class in which the processes inside is shown with green boxes.

For the preprocessing part, low resolution unsteady WRF solutions are obtained in 5 minute intervals. After that, latitude longitude and η coordinates are converted into easting northing and altitude in meters. Once the conversion is done, ground levels or the WRF and *OpenFOAM* solution domain are matched and WRF variables are interpolated onto face centers of the high resolution *OpenFOAM* domain's boundary cells, tri-linearly in space as given in the Extraction of Boundary Condition Profiles from WRF sub-section.

As mentioned in the `inletOutlet` class section, constructors are called only once at the beginning of the simulations whereas member functions which the boundary conditions are called at each iteration. Thus the main part of the algorithm must be implemented into the member functions.

`this -> refValue() = Field<Type>("inletValue", dict, p.size());` line reads the data defined by the keyword `inletValue`, from the dictionary `dict`, according to the number of face centers given by the `p.size()` function; as mentioned in the part constructors of the `inletOutlet` class is explained.

In the developed boundary condition class, `timeVaryingMixed`, main idea is define each BC profile obtained from WRF with 5 minute intervals at *OpenFOAM* do-

main's boundary cells face centers, with a different keyword entry. Afterwards, according to the simulation time, adjacent BC profiles can be read and interpolated in time. For that purpose, *bcreatorfoam.f* in *interfoam.f* main program, reads the boundary condition profiles from WRF extracted by the *interpolator.f* to write the dictionary entry used for `timeVaryingMixed` BC class. The dictionary, consists of different keyword entries for each BC profile extracted from WRF , whose structure is given below:

```
FoamFile
{
    version 2.0;
    format ascii;
    class volVectorField;
    object U;
}
dimensions [0 1 -1 0 0 0 0];
internalField uniform (-2.78225 -6.6085 -0.391);
boundaryField
{
    inlet
    {
        type timeVaryingMixed;
        inletValue1 nonuniform List<vector>
            14794
        (
            ( 0.0000000E+00 0.0000000E+00 0.0000000E+00 )
            ( 0.0000000E+00 0.0000000E+00 0.0000000E+00 )
            ...)
        inletValue2 nonuniform List<vector>
            14794
        (
            ( 0.0000000E+00 0.0000000E+00 0.0000000E+00 )
            ( 0.0000000E+00 0.0000000E+00 0.0000000E+00 )
            ...)
        ...
    }
}
```

As an example `inletValue1` keyword belongs the BC profile at `time=0` seconds whereas `inletValue2` keywords holds the data of the BC profile for `time=300` seconds. For reading and interpolating 5 minute interval BC profiles in time, `updateCoeffs()` function for `timeVaryingMixed` which inherits the `inletOutlet BC` class is defined as below:

```
template<class Type>
void Foam::timevaryingmixedFvPatchField<Type>::
updateCoeffs()
...
    int step,ustep;
    int zaman;
    std::string prev,next;
    zaman=this->db().time().value();
    step=(zaman/300)+1;
    ustep=step+1;
    char intStr[4];

    sprintf(intStr,"% d", step);
    prev="inletValue" + string(intStr);

    sprintf(intStr,"%d", ustep);
    next="inletValue" + string(intStr);

    refValueLow_ = Field<Type>(prev, dict, p());
    refValueHigh_ = Field<Type>(next, dict, p());

    this->refValue() = refValueLow_+((refValueHigh_-
    refValueLow_)/300)*(zaman%300);

    mixedFvPatchField<Type>::updateCoeffs();
```

In the following expression `zaman = this->db().time().value()` the simulation time is assigned to `zaman` variable, which in turn decides which adje-

cent BC profiles will be read according to keywords assigned to the BC profiles. For the interpolation of BC profiles in time, 2 member functions `refValueLow_` and `refValueHigh_` are defined. `refValueLow_` holds the values that belongs to the previous adjacent BC profile whereas `refValueHigh_` holds the next one.

Time dependent interpolation is done by the line `this->refValue() = refValueLow_ + ((refValueHigh_ - refValueLow_) / 300) * (zaman%300);`. Afterwards `updateCoeffs()` function belonging to the `mixedFvPatchField` class is called to apply the boundary condition. `w` parameter in `updateCoeffs()` of the `mixedFvPatchField` class, which was explained previously in `mixedFvPatchField` section, is updated dynamically during the simulation depending on the value of mass flux, ϕ . Thus, when $w = 1$ at the inflow boundaries, the flow variable extracted from the WRF solution is applied as the boundary value whereas when $w = 0$ at the outflow boundaries the value at the face center is evaluated based on the cell gradient of the flow variable.

2.7 Parallel Implementation of TimeVaryingMixed Boundary Condition

Parallel processing in *OpenFOAM* is based on domain decomposition. This domain decomposition and reconstruction is done using intrinsic `decomposePar` and `reconstructPar` utilities, respectively. `decomposePar` utility decomposes the case into partitions using either of 4 methods:

- Simple: uses geometric decomposition based on coefficients by direction. E.g 2 pieces in x-direction 1 piece in y direction.
- Hierarchical: same as geometric decomposition except the order in which the directional splitting is done is specified in `decomposeParDict`. E.g first in x-direction with 2 parts, then in y-direction with 3 parts
- Scotch: minimizes number of processor boundaries based on the weighting for each partition
- Manual: Allocation of each cell is specified manually in `decomposeParDict`

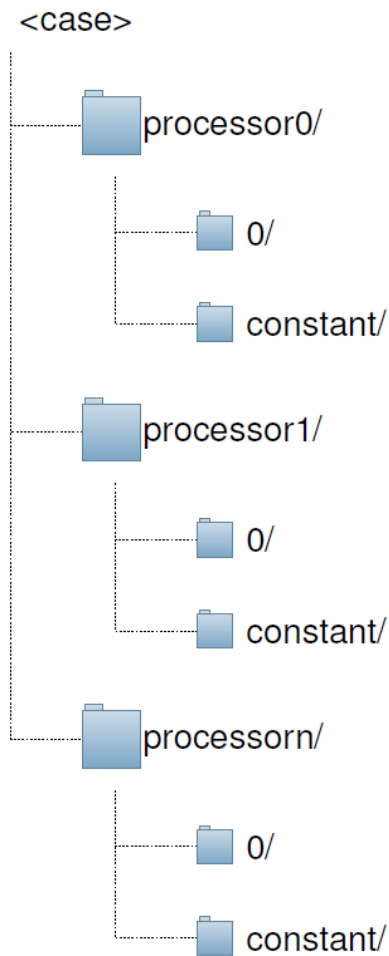


Figure 2.18: Parallel case structure

Once `decomposePar` is run with its dictionary `decomposeParDict` in which the required entries are given, it decomposes the case into specified number of partitions as seen in Figure 2.18.

In the partition case files, the boundary and initial conditions with necessary entries are given in `0/` folder whereas the grid points, faces and cell for a partition are in `constant/` folder. Interface boundaries are detected and named according to between which partitions the interfaces lies. An example of a boundary condition file in `processor1/0/` folder is given below:

```

FoamFile
{
    version 2.0;
    format ascii;
    class volVectorField;
    object U;
}
dimensions [0 1 -1 0 0 0 0];
internalField uniform (-2.78225 -6.6085 -0.391);
boundaryField
{
    inlet
    {
        inletValue1 nonuniform 0();
        type timevaryingmixed;
        value nonuniform 0();
    }
    yer
    {
        type fixedValue;
        value uniform (0 0 0);
    }
    procBoundary1to0
    {
        type processor;
        value uniform (-3.9662 -14.0808 0.0935);
    }
    procBoundary1to2
    {
        type processor;
        value uniform (-3.9662 -14.0808 0.0935);
    }
    ...
}

```

In the decomposed BC file, boundary type information is given first for all boundaries for the undecomposed case, and interfacing boundaries for each partition are defined as `type processor` with an unused entry `value`, afterwards.

The developed BC class (`timeVaryingMixed`) cannot be run in parallel using `decomposePar` automatically because of two reasons:

- In the developed BC class `TimeVaryingMixed`, the required entries consist of the interpolated WRF data for each cell at 5 minute intervals which are given in their respective keywords such as `inletValue1` for the 00:00 data set. `decomposePar` does not support more than 1 data entries of type `nonuniform List<vector>` to evaluate the BCs as seen above given in line `inletValue1 nonuniform 0();`. Only `inletValue1` keyword is read by `decomposePar` according to the simulation time which is 00:00 at initialization. The problem arises when the `updateCoeffs` function of the `TimeVaryingMixed` BC class tries to read adjacent WRF derived BC profiles to interpolate the value of BC in time (E.g `inletValue1`, `inletValue2`) for each cell.
- Interpolation of BC profiles from the 5 minute WRF data is done for the extracted face centers of the `TimeVaryingMixed` boundary as given in the Development of `TimeVaryingMixed` Boundary Condition Class. The second and most important reason is that the indexes and the ordering of cell faces belonging to `TimeVaryingMixed` boundary changes when decomposed.

In order to remedy these problems, all cells that has a face belonging to the `TimeVaryingMixed` BC is grouped into only one partition. The first approach to do that is done by:

- Decompose the domain with any method (E.g Simple)
- Write out the ownership of the cells to processors
- Find the cells that are adjacent to the `timeVaryingMixed` boundary
- Read the ownership data and change the processor numbers that belongs to cells adjacent to the `timeVaryingMixed` boundary
- Decompose the domain with manual method using the modified ownership data

While the previous method enables parallel processing, it does not guarantee the number of processor faces are minimized, resulting in sub-optimal parallel performance. So a second approach is devised using METIS (Serial Graph Partitioning and Fill-reducing Matrix Ordering) to minimize the number of interfaces given that the all cells that has a face belonging to the `TimeVaryingMixed` BC is grouped into only one of the partition.

METIS takes the neighbouring information of each cell in the form of graph file. Data Structure of a graph file is given below:

```
1st weight, 1st neigh.  of 1st cell, 2nd neigh.  of 1st cell, ...
2nd weight, 1st neigh.  of 2nd cell, 2nd neigh.  of 2nd cell, ...
...
nth cell's weight ....
```

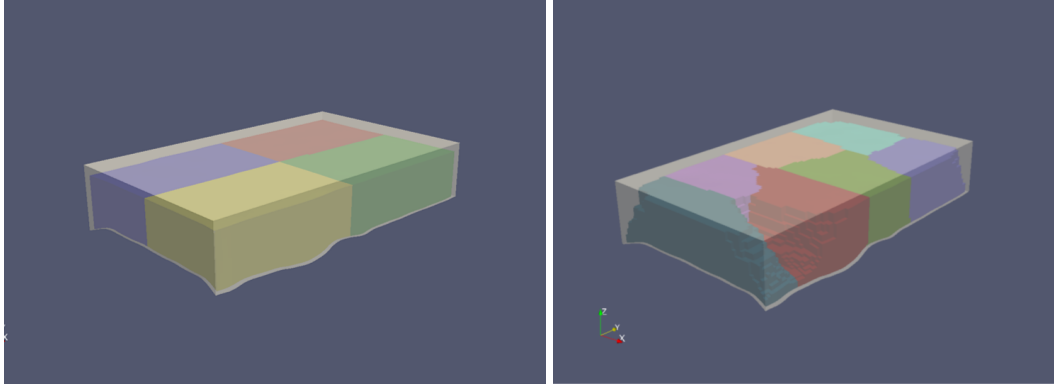
where weight is the measure of processor power needed for a cell and taken as number of faces for the cell as they are proportional.

In this graph file, defining all the cells that has a `timeVaryingMixed` boundary face as a single cell and giving the modified neighbouring data, restricts the decomposition to keep all cells in a single partition while METIS optimizes the number of interface boundaries. The steps to extract and utilize this decomposition is given below:

- Find all neighboring cells for the `timeVaryingMixed` boundary
- Renumber the indexes of cells such that all `timeVaryingMixed` boundary cells are indexed as the 1st cell
- Find the neighbouring cells for the newly generated 1st cell
- Writeout neighbouring info for remaining cells unchanged
- Use METIS and reconvert the ownership info into original cell numbering

- Use `decomposePar` manually with the converted ownership data

An example for domain decompositions using the first and second approach is given in Figure 2.19. Each color in Figure 2.19 represents a different partition whereas the transparent light grey section is the partition where all the cells belonging to the `timeVaryingMixed` are grouped together.



1st Approach (5 parts)

2nd Approach (8 parts)

Figure 2.19: Domain decompositions for a case with `TimeVaryingMixed` BC using the 1st and 2nd approaches

While the second approach is obviously better the first in terms of parallel performance, it still has a bottleneck in parallel performance. Since all the cells that has a face belonging to `TimeVaryingMixed` are grouped together via defining them as a single cell, the minimum number of cells that partition can have, is fixed. Thus, number of partitions that can be used for the parallel processing efficiently is limited.

2.8 Turbulence Modelling

Turbulence is a state of fluid motion with chaotic changes due to random and chaotic three-dimensional vorticity patterns. Turbulent flow is characterised with increased energy dissipation, mixing, heat transfer and drag. Recent understanding of turbulence suggest that turbulence is caused by eddies which are coherent vorticity patterns whose motions are correlated with each other. While turbulence is present everywhere, its existence is amplified with increasing Reynolds number.

There are several methods to simulate the turbulence and 4 important approaches are summarized below:

- (U)RANS equations: (Unsteady) Reynolds Time Averaged Navier-Stokes equations are solved while modelling all scales of turbulence according to Boussinesq hypothesis.
- Large Eddy Simulations (LES): Based on Smagorinsky [42], in LES, larger scale eddies are computed while the smaller scales are modelled. LES method requires a fair amount of computational effort due to meshes should have the resolution to resolving large eddies.
- Detached Eddy Simulations (DES): Only regions where flow is separated, are modelled by LES while (U)RANS equations are employed everywhere else. It is a middle ground between (U)RANS and LES methods in terms of computational power.
- Direct Numerical Simulations (DNS): All scales of turbulence are resolved. As a result DNS method suffers from immense computational effort requirements due to meshes should be smaller than Kolmogorov length scale. There are only few solutions of DNS with supercomputers for only simple cases (ex. channel flow with Reynolds number up to 590) as the method is not feasible for engineering problems. [36]

Only feasible option for the problem presented in this work is (U)RANS models, as the time is of essence for short term wind power predictions.

2.8.1 (Unsteady) Reynolds Averaged Navier-Stokes Equations (URANS)

URANS equations are derived from 2 hypothesis.

Reynolds averaging: Reynolds [39] suggested that turbulent velocity can be decomposed into an time-averaged mean part u and a fluctuating part u' . When implemented into Navier-Stokes equations and time-averaged for mean flow variables,

RANS equations are obtained as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \quad (2.3)$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_l}{\partial x_l} \right) \right] + \frac{\partial}{\partial x_j}(-\overline{\rho u'_i u'_j}) \quad (2.4)$$

If the flow velocity is small, air can be assumed incompressible and above equations reduce to;

$$\frac{\partial u_j}{\partial x_j} = 0 \quad (2.5)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{\partial \overline{u'_i u'_j}}{\partial x_j}. \quad (2.6)$$

Boussinesq Hypothesis: Boussinesq suggested that the Reynolds stresses were related directly to the main stress which is based on the idea that viscous and Reynolds stresses have similar effects on the mean flow and can be written for incompressible flows [8]:

$$-\overline{\rho u'_i u'_j} = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.7)$$

Where μ_t is the eddy viscosity which is additional viscosity due to turbulence. If the turbulent viscosity can be obtained from other time averaged flow variables, then the turbulence closure can be achieved.

2.8.1.1 $k - \epsilon$ Model

$k - \epsilon$ model is a two equation model which solves the transport equations for two new variables: turbulent kinetic energy k and turbulent dissipation per unit mass ϵ

[31]. Most basic form (where buoyancy and compressibility effects are neglected) for transport equations for k and ϵ are:

$$\frac{\partial k}{\partial t} + \frac{\partial}{\partial x_i}(ku_i) = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + P_k - \epsilon \quad (2.8)$$

$$\frac{\partial \epsilon}{\partial t} + \frac{\partial}{\partial x_i}(\epsilon u_i) = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] + C_{1\epsilon} \frac{\epsilon}{k} P_k - C_{2\epsilon} \frac{\epsilon^2}{k} \quad (2.9)$$

where P_k represents the generation of turbulence kinetic energy due to the mean velocity gradients and calculated as:

$$P_k = -\overline{u'_i u'_j} \frac{\partial u_j}{\partial x_i} \quad (2.10)$$

Turbulent kinematic viscosity ν_t is calculated as:

$$\nu_t = C_\mu \frac{k^2}{\epsilon} \quad (2.11)$$

and model constants are:

$$C_{1\epsilon} = 1.44, \quad C_{2\epsilon} = 1.92, \quad C_\mu = 0.09, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.3 \quad (2.12)$$

2.8.1.2 Realizable $k - \epsilon$ Model

Realizable $k - \epsilon$ model is a modification of the standard $k - \epsilon$ to account for deficiencies in high shear flows and inaccuracy for the spreading rate of jets. Most basic form for transport equations for k and ϵ are:

$$\frac{\partial k}{\partial t} + \frac{\partial}{\partial x_j}(ku_j) = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + P_k - \epsilon \quad (2.13)$$

$$\frac{\partial \epsilon}{\partial t} + \frac{\partial}{\partial x_j}(\epsilon u_j) = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] + C_1 S \epsilon - C_2 \frac{\epsilon^2}{k + \sqrt{\nu \epsilon}} \quad (2.14)$$

where,

$$C_1 = \max \left[0.43, \frac{\eta}{\eta + 5} \right], \quad \eta = S \frac{k}{\epsilon}, \quad S = \sqrt{2 S_{ij} S_{ij}} \quad (2.15)$$

Turbulent kinematic viscosity ν_t is calculated in the same manner as in $k - \epsilon$ model:

$$\nu_t = C_\mu \frac{k^2}{\epsilon} \quad (2.16)$$

But this time instead of being constant, C_μ is calculated as:

$$C_\mu = \frac{1}{A_0 + A_s \frac{k U^*}{\epsilon}} \quad (2.17)$$

$$U^* \equiv \sqrt{S_{ij} S_{ij} + \tilde{\Omega}_{ij} \tilde{\Omega}_{ij}}, \quad \tilde{\Omega}_{ij} = \Omega_{ij} - 2 \epsilon_{ijk} \omega_k, \quad \Omega_{ij} = \overline{\Omega_{ij}} - \epsilon_{ijk} \omega_k \quad (2.18)$$

where $\overline{\Omega_{ij}}$ is the mean rate of rotation tensor viewed in a rotating reference frame with the angular velocity ω_k

A_0 and A_s are given by:

$$A_0 = 4.04, \quad A_s = \sqrt{6} \cos \phi \quad (2.19)$$

which are calculated according to:

$$\phi = \frac{1}{3} \cos^{-1}(\sqrt{6}W), \quad W = \frac{S_{ij} S_{jk} S_{ki}}{\tilde{S}^3}, \quad \tilde{S} = \sqrt{S_{ij} S_{ij}}, \quad S_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \quad (2.20)$$

Model constants are given by:

$$C_{1\epsilon} = 1.44, \quad C_2 = 1.9, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.2 \quad (2.21)$$

Although this model increases the computational power, its performance in boundary layers with adverse pressure gradient is better with respect to the standard $k-\epsilon$ model.

2.8.1.3 $k-\omega$ SST Model

The SST $k-\omega$ turbulence model [33] is a two equation eddy viscosity model which is a modification of the Wilcox's $k-\omega$ model [53] that solves transport equations for turbulent kinetic energy k and specific dissipation rate ω . The model acts as a $k-\omega$ model for the inner parts of the boundary layer and switches its behaviour to $k-\epsilon$ in the free stream, using the advantages of both models. Transport equations for the turbulent kinetic energy k and specific dissipation rate ω are given as follows:

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = P_k - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_k \nu_T) \frac{\partial k}{\partial x_j} \right] \quad (2.22)$$

$$\frac{\partial \omega}{\partial t} + U_j \frac{\partial \omega}{\partial x_j} = \alpha S^2 - \beta \omega^2 + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_\omega \nu_T) \frac{\partial \omega}{\partial x_j} \right] + 2(1-F_1) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \quad (2.23)$$

where the production of turbulent kinetic energy P_k is:

$$P_k = \min \left(\tau_{ij} \frac{\partial U_i}{\partial x_j}, 10 \beta^* k \omega \right) \quad (2.24)$$

Turbulent viscosity is calculated as follows:

$$\nu_T = \frac{a_1 k}{\max(a_1 \omega, S F_2)} \quad (2.25)$$

Auxiliary relations for the turbulence closure are given as:

$$F_1 = \tanh \left\{ \left\{ \min \left[\max \left(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{y^2 \omega} \right), \frac{4 \sigma_{\omega 2} k}{C D_{k\omega} y^2} \right] \right\}^4 \right\} \quad (2.26)$$

$$F_2 = \tanh \left[\left[\max \left(\frac{2 \sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{y^2 \omega} \right) \right]^2 \right] \quad (2.27)$$

$$CD_{k\omega} = \max \left(2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-10} \right) \quad (2.28)$$

$$\alpha_1 = \frac{5}{9}, \alpha_2 = 0.44, \beta_1 = \frac{3}{40}, \beta_2 = 0.0828, \beta^* = \frac{9}{100} \quad (2.29)$$

$$\sigma_{k1} = 0.85, \sigma_{k2} = 1, \sigma_{\omega 1} = 0.5, \sigma_{\omega 2} = 0.856 \quad (2.30)$$

2.9 Near Wall Treatment and Roughness

It is known that the presence of land covers such as forests or vegetation or sheltering objects such as buildings effect the wind flow significantly [14]. In atmospheric flow solutions surface roughness models are used to model these effects. These models are typically implemented in turbulent boundary layers by means of rough wall functions.

Turbulent boundary layers can be divided into three sub-layers depending on the value of the non-dimensional distance from a wall, y^+ . y^+ is defined as:

$$y^+ = \frac{yu_\tau}{\nu} \quad (2.31)$$

At the first level near the bounding surface ($y^+ < 5$), there is the viscous sub-layer where the viscosity effects are dominant. In the outer part ($30 < y^+ < 200$), there exists a fully turbulent zone where the flow is characterised by the turbulence. Between them ($5 < y^+ < 30$), there exists a buffer zone in which both viscosity and turbulence are both influential. The non-dimensional velocity profiles in these regions can be given as:

$$u^+ = y^+ \quad \text{viscous sublayer} \quad (2.32)$$

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) \quad \text{buffer layer} \quad (2.33)$$

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) \quad \text{fully turbulent zone} \quad (2.34)$$

where u^+ is

$$u^+ = u/u_\tau \quad (2.35)$$

and u_τ is the shear velocity that can be calculated by:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad (2.36)$$

The usual way for handling the high gradients that exists near wall boundaries is to mesh such that the value of the first vertical level resides within $y^+ < 1$. But using this approach in an atmospheric flow solution where the solution domain is covers kilometre squares requires too much computational power. The remedy is to use wall functions with a decent vertical resolution mesh that doesn't need to cover the viscous sub-layer. The wall functions forces the non-dimensional velocity profiles within the first vertical levels to follow a certain 'law of the wall'. For a smooth wall, a typical wall function forces the following non-dimensional velocity profile:

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) , \quad \kappa = 0.4178 , \quad E = 9 \quad (2.37)$$

The surface roughness is typically defined with two variables: the roughness height (K_s) which is usually given as sand grain roughness and non-dimensional roughness constant (C_s) that varies from 0 to 1. (K_s) can be obtained from a region interest using the global Corine Roughness Data Set. On a rough wall, the velocity is slowed in presence of roughness elements. Cebeci and Bradshaw [15] suggested a wall function that is based on empirical formulae.

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) - \Delta B , \quad \kappa = 0.4178 , \quad E = 9 \quad (2.38)$$

where K_s^+ is the non-dimensional roughness height and

$$K_s^+ = \frac{K_s u_\tau}{\nu} = \frac{c_\mu^{1/4} k^{1/2}}{\nu} \quad (2.39)$$

ΔB is the retardation factor due to roughness that depends on the 3 sub-regimes K_s^+ is in:

- Hydrodynamically smooth ($K_s^+ < 2.25$), $\Delta B = 0$
- Transition
 $(2.25 < K_s^+ < 90)$, $\Delta B = \frac{1}{K} \ln \left[\frac{K_s^+ - 2.25}{87.75} + C_s K_s^+ \right] \sin[0.4258(\ln K_s^+ - 0.811)]$
- Fully Rough ($K_s^+ > 90$), $\Delta B = \frac{1}{\kappa} \ln(1 + C_s K_s^+)$

In *OpenFOAM*, like most of the turbulence closures, shear velocity is calculated using the turbulent kinetic energy in the following way:

$$u_\tau = \frac{c_\mu^{1/4} k^{1/2}}{\nu} \quad (2.40)$$

The effects of surface roughness is implemented in *OpenFOAM* via the intrinsic the wall function `nutkRoughWallFunction` for k - ϵ turbulence closure by changing the E parameter when $K_s^+ < 2.25$ and using the smooth wall function given in Equation 2.37, instead of applying a retardation factor ΔB as in Equation 2.38.

$$E_{dash} = E / fnRough(K_s^+, C_s) \quad (2.41)$$

$fnRough(K_s^+, C_s)$ function changes values according to the K_s^+ value as

- ($K_s^+ < 90$), $\left[\frac{K_s^+ - 2.25}{87.75} + C_s K_s^+ \right] \sin[0.4258(\ln(K_s^+) - 0.811)]$
- ($K_s^+ > 90$), $(1 + C_s K_s^+)$

The turbulent viscosity is then evaluated to satisfy the wall function as in Equation 2.38 as follows:

$$\nu_t = \nu \left(\frac{y^+ \kappa}{\ln(E_{dash} y^+)} - 1 \right) \quad (2.42)$$

In this study, the roughness height, (K_s), for a region of interest is obtained from the global Corine Roughness Data Set. The roughness constant C_s , is set to an average value of 0.5 which corresponds to sand grain roughness. This implementation of roughness is reported to cause unrealistic results if the first vertical level of the grid has a height greater than the roughness height. [6]. So, care must be taken to avoid when meshing the solution domain.

2.10 Coriolis Acceleration

Coriolis acceleration is the acceleration that acts on a particle that is moving in a rotating reference frame, perpendicular to its direction of motion. For atmospheric flow simulations, Coriolis acceleration results from the rotation of Earth and Newton's Law of motion being only valid in the inertial reference frame. A generic vector $A(x, y, z, t)$ in Cartesian coordinates is expressed as:

$$\vec{A} = A_x \vec{i} + A_y \vec{j} + A_z \vec{k} \quad (2.43)$$

in an inertial reference frame and,

$$\vec{A} = \hat{A}_x \hat{\vec{i}} + \hat{A}_y \hat{\vec{j}} + \hat{A}_z \hat{\vec{k}} \quad (2.44)$$

in a rotating frame of reference. Taking the time derivative of \vec{A} in the inertial frame:

$$\frac{d\vec{A}}{dt} = \left[A_x \frac{d\vec{i}}{dt} + \vec{i} \frac{dA_x}{dt} \right] + \left[A_y \frac{d\vec{j}}{dt} + \vec{j} \frac{dA_y}{dt} \right] + \left[A_z \frac{d\vec{k}}{dt} + \vec{k} \frac{dA_z}{dt} \right] \quad (2.45)$$

since the unit vectors of the inertial frame $\vec{i}, \vec{j}, \vec{k}$ does not change in time:

$$\frac{d\vec{i}}{dt} = \frac{d\vec{j}}{dt} = \frac{d\vec{k}}{dt} = 0 \quad (2.46)$$

then Equation 2.45 reduces to:

$$\frac{d\vec{A}}{dt} = \frac{dA_x}{dt}\vec{i} + \frac{dA_y}{dt}\vec{j} + \frac{dA_z}{dt}\vec{k} \quad (2.47)$$

Derivative of \vec{A} in the rotating reference frame:

$$\frac{d\vec{A}}{dt} = \left[\hat{A}_x \frac{d\vec{i}}{dt} + \vec{i} \frac{d\hat{A}_x}{dt} \right] + \left[\hat{A}_y \frac{d\vec{j}}{dt} + \vec{j} \frac{d\hat{A}_y}{dt} \right] + \left[\hat{A}_z \frac{d\vec{k}}{dt} + \vec{k} \frac{d\hat{A}_z}{dt} \right] \quad (2.48)$$

As the derivative of a constant vector doesn't change in an inertial or rotating frame, Equation 2.47 and Equation 2.48 should yield the same result. Thus,

$$\frac{d\vec{A}}{dt} = \frac{dA_x}{dt}\vec{i} + \frac{dA_y}{dt}\vec{j} + \frac{dA_z}{dt}\vec{k} = \left[\hat{A}_x \frac{d\vec{i}}{dt} + \vec{i} \frac{d\hat{A}_x}{dt} \right] + \left[\hat{A}_y \frac{d\vec{j}}{dt} + \vec{j} \frac{d\hat{A}_y}{dt} \right] + \left[\hat{A}_z \frac{d\vec{k}}{dt} + \vec{k} \frac{d\hat{A}_z}{dt} \right] \quad (2.49)$$

rearranging Equation 2.49 :

$$\left[\frac{dA_x}{dt}\vec{i} + \frac{dA_y}{dt}\vec{j} + \frac{dA_z}{dt}\vec{k} \right] = \left[\frac{d\hat{A}_x}{dt}\vec{i} + \frac{d\hat{A}_y}{dt}\vec{j} + \frac{d\hat{A}_z}{dt}\vec{k} \right] + \left[\hat{A}_x \frac{d\vec{i}}{dt} + \hat{A}_y \frac{d\vec{j}}{dt} + \hat{A}_z \frac{d\vec{k}}{dt} \right] \quad (2.50)$$

Left hand side of the Equation 2.50 is the time derivative of \vec{A} in an inertial frame, whereas the first term in the right hand side is the time derivative of \vec{A} in a rotating frame and the second term equals the effects of rotation.

Since the velocity is defined as:

$$\vec{V} = \frac{d\vec{r}}{dt} \quad (2.51)$$

The derivative of the the unit vectors in rotating frame can be given as:

$$\frac{d\hat{i}}{dt} = \vec{\Omega} \times \hat{i}, \quad \frac{d\hat{j}}{dt} = \vec{\Omega} \times \hat{j}, \quad \frac{d\hat{k}}{dt} = \vec{\Omega} \times \hat{k} \quad (2.52)$$

Thus, Equation 2.47 becomes:

$$\left[\frac{dA_x}{dt} \hat{i} + \frac{dA_y}{dt} \hat{j} + \frac{dA_z}{dt} \hat{k} \right] = \left[\frac{d\hat{A}_x}{dt} \hat{i} + \frac{d\hat{A}_y}{dt} \hat{j} + \frac{d\hat{A}_z}{dt} \hat{k} \right] + \left[\hat{A}_x (\vec{\Omega} \times \hat{i}) + \hat{A}_y (\vec{\Omega} \times \hat{j}) + \hat{A}_z (\vec{\Omega} \times \hat{k}) \right] \quad (2.53)$$

Since Newton's Law of Motion is only valid on inertial reference frame, the derivative of \vec{V} is needed in the inertial frame. So, replacing the \vec{A} with Velocity \vec{V} in Equation 2.53 yields:

$$\left(\frac{d\vec{V}_{inertial}}{dt} \right)_{inertial} = \frac{d\vec{V}_{inertial}}{dt} + \vec{\Omega} \times \vec{V}_{inertial} \quad (2.54)$$

since the inertial velocity is velocity in the rotating reference frame plus rotational velocity:

$$\left(\frac{d\vec{V}}{dt} \right)_{inertial} = \frac{d}{dt} (\vec{V} + \vec{\Omega} \times \vec{r}) + \vec{\Omega} \times (\vec{V} + \vec{\Omega} \times \vec{r}) \quad (2.55)$$

Rearranging the terms:

$$\left(\frac{d\vec{V}}{dt} \right)_{inertial} = \frac{d\vec{V}}{dt} + 2\vec{\Omega} \times \vec{V} + \vec{\Omega}^2 \vec{r} \quad (2.56)$$

Left hand side of Equation 2.56 is the acceleration following the motion in an inertial system. As for the right hand side; first term is the rate of change of relative velocity following the relative motion in a rotating reference frame, the second term is the Coriolis acceleration and the third term is the centrifugal acceleration. Lastly,

applying Newton's Law of Motion yields:

$$\left(\frac{d\vec{V}}{dt} \right)_{inertial} = \frac{\sum \vec{F}}{m} , \quad \frac{d\vec{V}}{dt} + 2\vec{\Omega} \times \vec{V} + \vec{\Omega}^2 \vec{R} = \frac{\sum \vec{F}}{m} \quad (2.57)$$

It is important to note that the \vec{V} in the Equation 2.57 are defined in rotating frame (with respect to Earth).

As mentioned earlier, for the case of atmospheric flows, Coriolis acceleration results from the rotation of Earth and Newton's Law of motion being only valid in the inertial reference frame.

A representation of Earth and its rotational velocity components are given in Figure 2.20. For a given location the vertical component of the Earth's rotational velocity Ω_v , is $\vec{\Omega} \sin \varphi$, whereas the horizontal component, Ω_h is $\vec{\Omega} \cos \varphi$.

As derived in Equation 2.57, Coriolis acceleration is the term $2\vec{\Omega} \times \vec{V}$. In a more explicit sense for atmospheric flows:

$$2\vec{\Omega} \times \vec{V} = 2 \begin{vmatrix} i & j & k \\ 0 & \Omega \cos \varphi & \Omega \sin \varphi \\ u & v & w \end{vmatrix} \quad (2.58)$$

So the magnitude and direction of the Coriolis acceleration depends on the Latitude. As an example, Coriolis acceleration deflects the flow direction to the right of the direction of the flow in Northern Hemisphere, whereas for the Southern Hemisphere the deflecting due to the Coriolis acceleration is to the left. Effects of Coriolis acceleration is proportional to the Rossby number which is defined as:

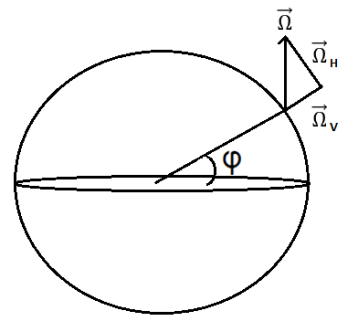


Figure 2.20: Components of the Earth's rotational velocity

$$R_o = \frac{V}{L2\Omega \sin \varphi} \quad (2.59)$$

where V is the velocity, L is the characteristic length, Ω is the rotational speed and

φ is the angle between the axis of rotation and the direction of the flow. Also, for flow solutions over a topography, φ is the Latitude and Ω is the rotational speed of the Earth which is $\Omega = (2\pi/23h\ 56m\ 4secs) = 7.2921 \times 10^{-5}sec^{-1}$. It is important to note that the effects of Coriolis acceleration can be neglected when Rossby number is much greater than 1, $R_o \gg 1$.

2.10.1 Implementation of Coriolis Acceleration in PimpleFOAM Solver

The Coriolis acceleration is successfully implemented into pimpleFOAM solver by utilizing useful features *OpenFOAM* such as object oriented programming and operator overloading. The modifications to the pimpleFOAM source code are given in Figure 2.21.

In the first column, a short representation of pimpleFOAM algorithm is shown. Firstly, in the PIMPLE pressure corrector loop, equations defined in *Ueqn.H* file is solved which is nothing but conservation of momentum equation as seen in the second column where parts of the *Ueqn.H* is shown. After that, pressure velocity coupling is done using the equation *Peqn.H*. As shown in the previous section, Coriolis acceleration can be used as a simple source term which was $2\vec{\Omega} \times \vec{V}$ where Ω 's vertical and horizontal components only depends on the Latitude. So, simply defining Ω as a vector variable in the *Ueqn.H* and adding $2\vec{\Omega} \times \vec{V}$ source term to the left hand side to the *Ueqn*, effects of the Coriolis acceleration is implemented into pimpleFOAM.

```

...
while (runTime.run())
{
    #include "readTimeControls.H"
    #include "CourantNo.H"
    #include "setDeltaT.H"
    runTime++;

    //PIMPLE corrector loop
    while (pimple.loop())
    {
        #include "UEqn.H"
    //Pressure corrector loop
        while (pimple.correct())
        {
            #include "pEqn.H"
        }
        if (pimple.turbCorr())
        {
            turbulence->correct();
        }
    }
}
....

```

```

...
dimensionedVector Omega
(
    "Omega",
    dimensionSet(0,0,-1,0,0),
    vector(0,5.848E-5,4.356E-5)
);
tmp<fvVectorMatrix> UEqn
(
    fvm::ddt(U)
    + fvm::div(phi, U)
    + turbulence->divDevReff(U)
    + 2*Omega ^ U
    ==
    fvOptions(U)
);
...

```

Figure 2.21: PimpleFOAM algorithm and modification on UEqn.H

CHAPTER 3

RESULTS AND DISCUSSIONS

In this study, a coupled flow solution methodology with an atmospheric weather forecast software, WRF , and *OpenFOAM* , an open source Navier-Stokes solver, is developed. In this section, a validation study for the implementation of the Coriolis acceleration in the *pimpleFOAM* solver is first performed. Further validation studies and analysis of wind fields are presented for the Mut region in Mersin/Turkey and the Alaiz region in Navarra/Spain.

The validation of the new boundary condition class developed, *timeVaryingMixed*, and its parallel performance are carried out in the Mut region. The analysis of wind fields and the assessment of the coupled flow solutions are performed in the Alaiz region, where meteorological mast data are available at 5 different altitudes. Grid independence studies for vertical and horizontal grid resolutions, and the comparison of the flow fields predicted by various turbulence models are first performed. Unsteady variation of wind velocities at the met-mast location are compared with the met-mast data at 5 different altitudes. The predictions of atmospheric flow features such as flow separation, reattachment, diurnal cycle and capping inversion, are assessed. Next, a sensitivity analysis for the effect of WRF grid resolution on the unsteady *OpenFOAM* solutions is performed. Finally, distributions of wind potential density at any altitude and the corresponding production of wind energy are obtained by integrating the proper combinations of the flow variables in time.

3.1 Validation of Coriolis Acceleration implemented in PimpleFOAM Solver

The Coriolis acceleration implemented into the `pimpleFOAM` solver is validated on a 2D flow field which is similar to what is suggested by Tapia[46]. The flow in an open channel is solved by the modified `pimpleFOAM` solver and the deflection in the flow due to the Coriolis acceleration is compared with a simple 1D Euler integration.

The solution domain which is given in Figure 3.1 is of size $20km \times 2km$ in x and y directions with a resolution of $100m \times 100m$. The leftmost inflow boundary condition is given as `FixedValue` $10m/s$ for velocity and `zeroGradient` for pressure. Other boundaries are given as `zeroGradient` for velocity and `FixedValue` $0 Pa$ for pressure boundary condition. The steady state solutions obtained are given in Figure 3.2.

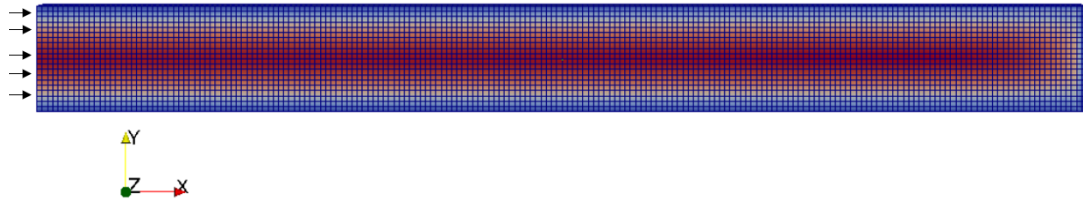


Figure 3.1: Solution domain for the validation of the Coriolis acceleration implementation

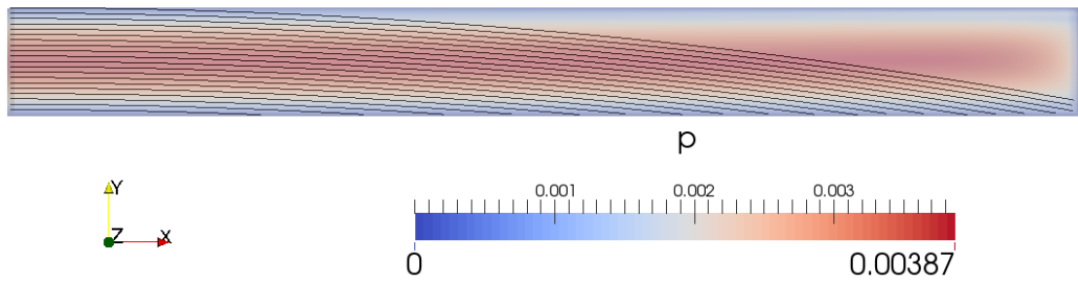


Figure 3.2: Streamlines under the effect of the Coriolis acceleration

For the flow field shown in Figure 3.1, the only forcing on the fluid particles is due to the Coriolis acceleration, other than the negligible pressure gradient and gravity. For this 2D case, forcing due to Coriolis acceleration which acts opposite to Coriolis

acceleration is:

$$-2\vec{\Omega} \times \vec{V} = -2 \begin{vmatrix} i & j & k \\ 0 & \Omega \cos \varphi & \Omega \sin \varphi \\ u & v & 0 \end{vmatrix} = 2v\Omega \sin \varphi \vec{i} - 2u\Omega \sin \varphi \vec{j} = a_x \vec{i} + a_y \vec{j} \quad (3.1)$$

The motion of an air particle on $x - y$ plane that is under the effect of Coriolis acceleration may also be obtained by integrating the equations of motion in x and y directions as given in Equation 3.2 and 3.3:

$$a_x^{n-1} = -2U^{n-1}\Omega \sin \varphi, \quad U^n = U^{n-1} + a_x^{n-1}\Delta t, \quad x^n = x^{n-1} + U^{n-1}\Delta t \quad (3.2)$$

$$a_y^{n-1} = -2V^{n-1}\Omega \sin \varphi, \quad V^n = V^{n-1} + a_y^{n-1}\Delta t, \quad y^n = y^{n-1} + V^{n-1}\Delta t \quad (3.3)$$

A streamline obtained from the steady state *OpenFOAM* solution that has the same starting point as the streamline obtained from the Euler integration are compared in Figure 3.3. The position and the flow velocities at the exit of the solution domain are compared in Table 3.1. As both Figure 2.20 and Table 3.1 suggest, the Coriolis acceleration is successfully implemented into the *OpenFOAM* solver, *pimpleFOAM*.

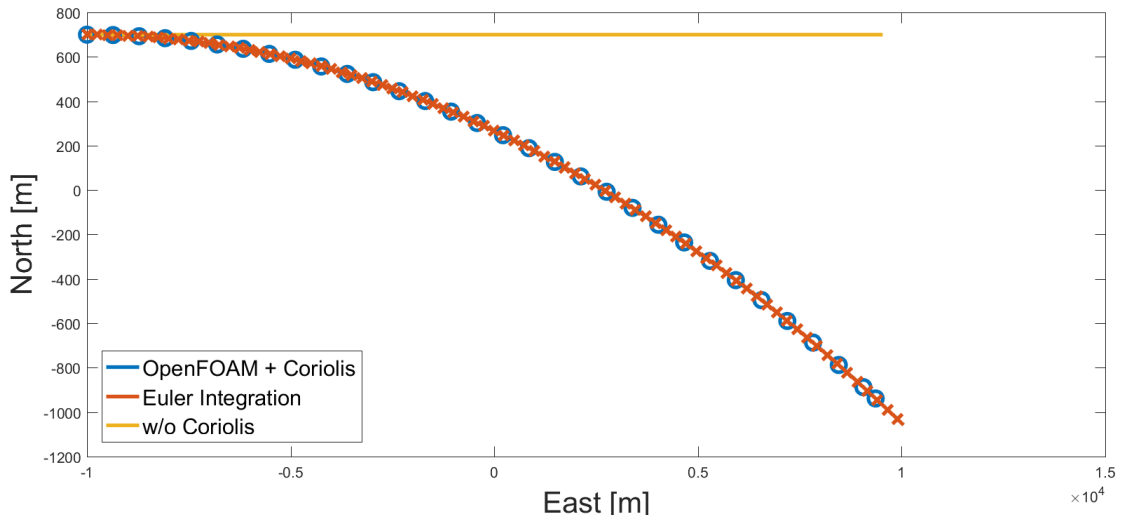


Figure 3.3: Streamlines based on the *OpenFOAM* solutions and the Euler integration

Table 3.1: Location and velocity of the air particles at the exit plane

	Integration Time	X (m)	Y (m)	U (m/s)	V (m/s)
OpenFOAM	1385 sec	3816.592	-130.52	9.928	-1.1993
Euler Int.	1385 sec	3816.736	-130.725	9.928	-1.1989

3.2 Case Study: Wind Farm in Mut Mersin, Turkey

The preliminary case study is done in Mut district in Mersin providence. Mut lies on the foothills of the mountain range Taurus and on the waterfront of Göksu river. A medium scale wind farm consisting of 16 wind turbines with a total of $50Mw$ installed power is located in the district. While being near a mountain range, the topography of the region is relatively smooth and simple. The prevailing wind direction in the district is from North to South.

Nested WRF solutions are first obtained for a 1 month period, within a parent domain of $3km$ horizontal resolution and a nested domain of $1km$ resolution. The computations are performed for the date 04.04.2010 starting from 00:00 GMT 0 to 12:00. The parent and the nested solution domains are of 100×79 (horizontal) $\times 50$ (vertical) size, and 88×67 (horizontal) $\times 50$ (vertical) size, respectively. WRF's solution domain and its nests are given in Figure 3.4.

The unsteady boundary conditions needed for the *OpenFOAM* solution at its domain boundaries, which fall into the innermost nest of WRF domain as seen in Figure3.4, are extracted from the WRF solution at 5-minute time intervals. These velocity profiles are used by the developed BC class *timeVaryingMixed* for imposing boundary conditions for the *OpenFOAM* domain.

The computational grid for *OpenFOAM* solutions are based on a high resolution terrain topography obtained from ASTER GDEM, which is a product of METI (The Ministry of Economy, Trade, and Industry of Japan) and NASA. The worldwide elevation data has about 1.5 sec^{-1} spatial resolution which is approximately 38 m . The region of interest is $1.8km \times 3km$ in East-West and North-South directions. The topography of the region and the modelled surface is given in Figure 3.5.

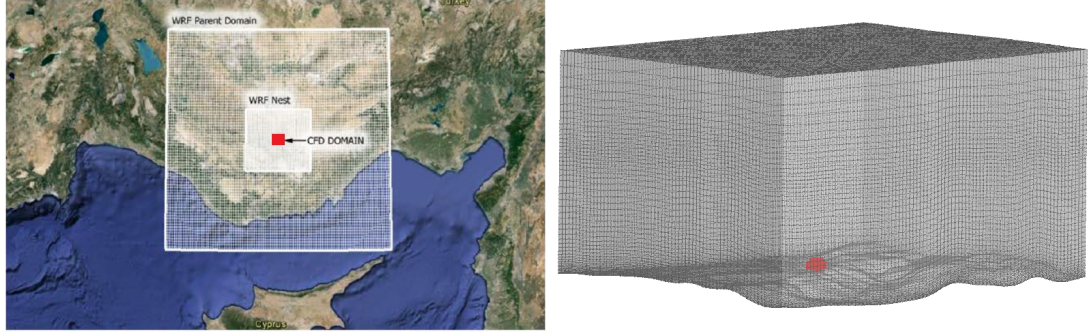


Figure 3.4: WRF and OpenFOAM solution domains

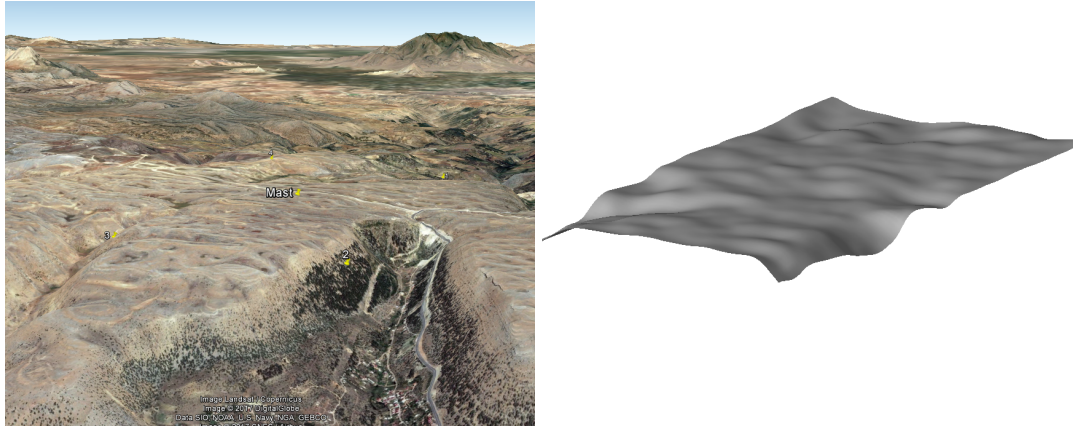


Figure 3.5: Topography of the Mut region and the modelled ground surface

The terrain fitted computational grids have a uniform horizontal resolution of $30m$ and a vertical resolution of $1m$ on the terrain surface. The $61 \times 100 \times 27$ grid stretches rapidly in the vertical direction in order to capture the turbulent atmospheric boundary layers efficiently as shown in Figure 3.6. The turbulent boundary layers dominate the flow characteristics at the surface layer of the atmosphere where majority of the wind turbines reside.

In Figure 3.7 and Figure 3.8 the velocity boundary conditions imposed on the *OpenFOAM* solutions are compared against the WRF solution interpolated on the *OpenFOAM* grid at an arbitrary time step along the 24 hour solution. As expected, the agreement is stronger at the inflow boundaries where the Dirichlet type boundary conditions are applied through the new boundary condition class developed. It should be noted that the zero-gradient boundary condition is applied at the outflow boundaries. The velocity vectors at the outflow boundaries of both the *OpenFOAM* solution domain and the inner nested domain of WRF solution are compared at various times along the unsteady solution in Figures 3.9 and 3.10, respectively. It is observed that the velocity

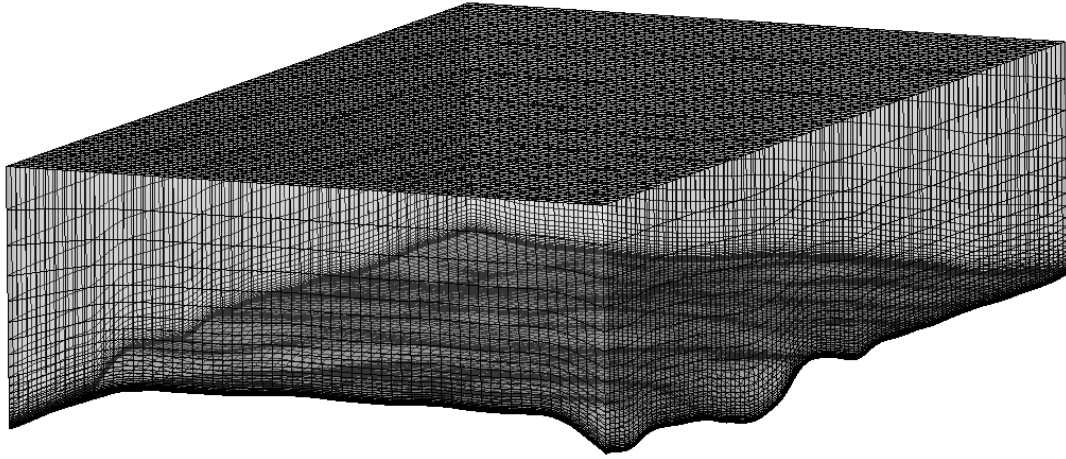


Figure 3.6: Computational grid for OpenFOAM solutions in the Mut region

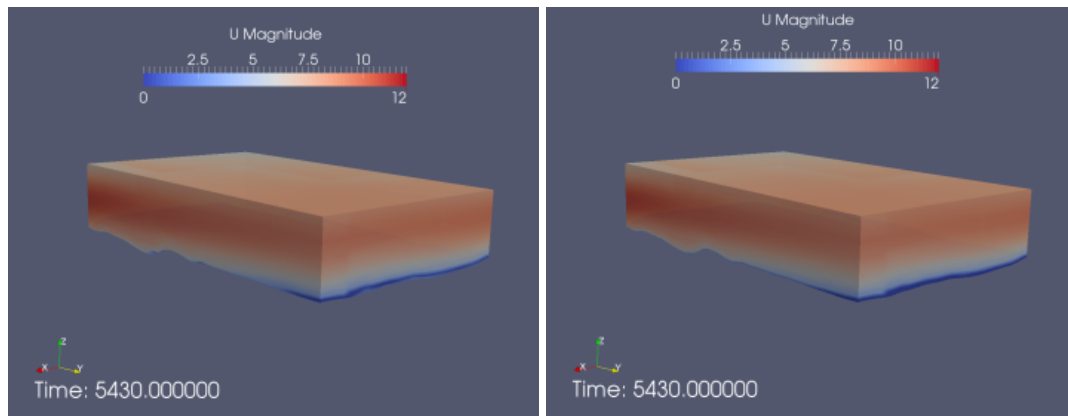


Figure 3.7: OpenFOAM (right) vs WRF (left) solutions (inflow)

directions are in a general agreement, yet the high vertical resolution in the *OpenFOAM* solution resolves the atmospheric boundary layer profile better.

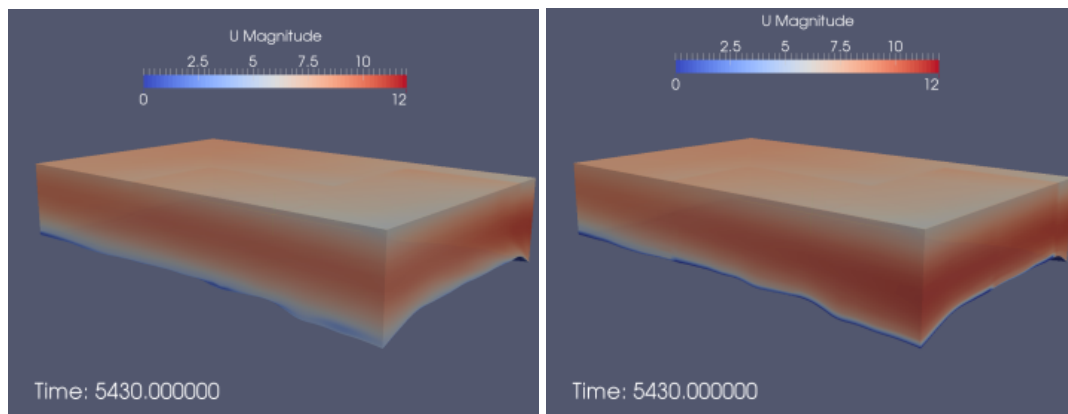


Figure 3.8: OpenFOAM (right) vs WRF (left) solutions (outflow)

Figure 3.11 shows the streamlines and the distribution of velocity magnitudes on two

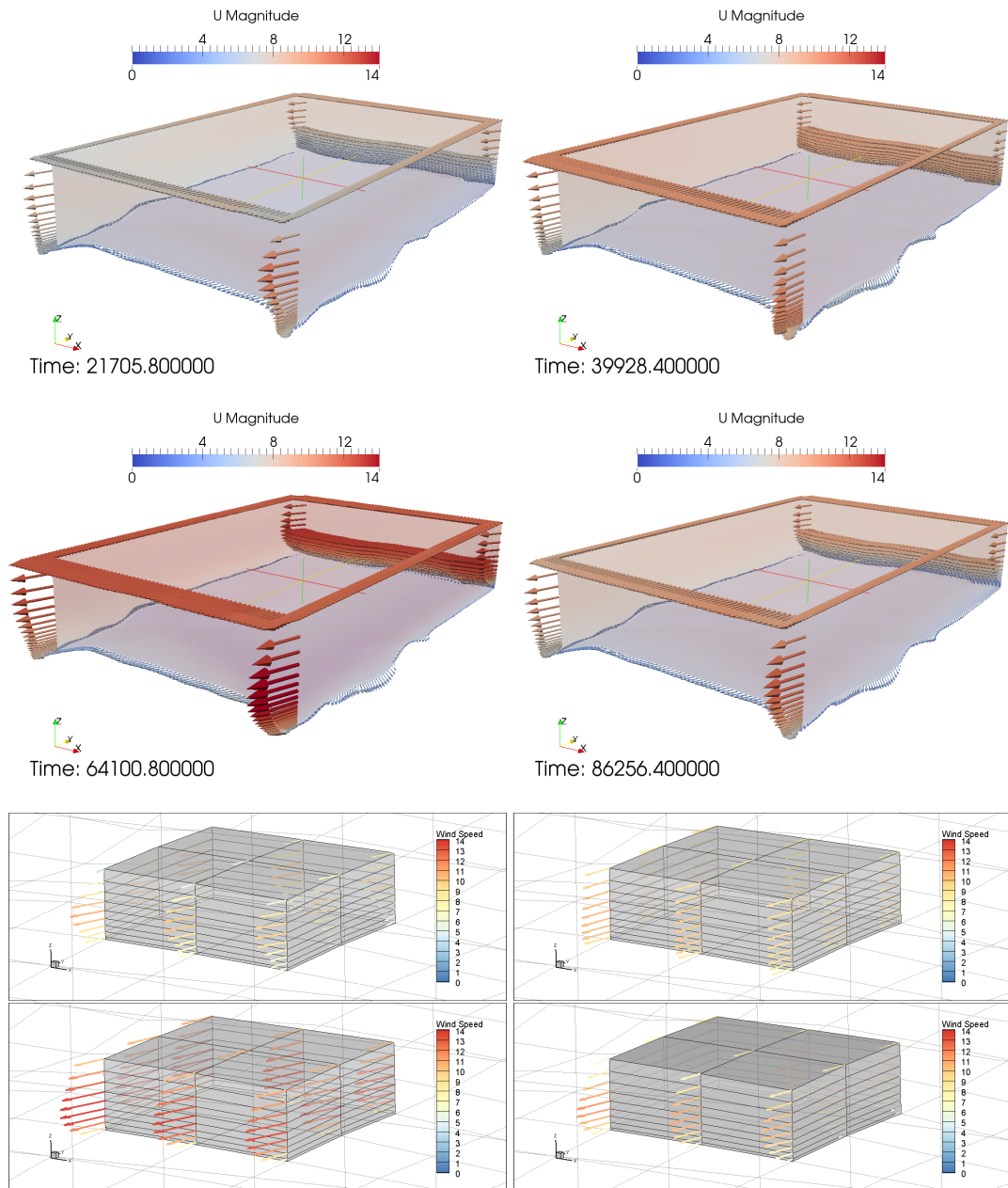


Figure 3.10: Unsteady OpenFOAM solutions (top) and WRF solutions (bottom) at 6, 11, 18, 24th hour

terrain following surfaces 25m and 100m above the ground at various instances. Having known the unsteady variation of the wind velocity in the vertical direction makes it possible to predict the power production of a wind farm located in the solution domain. The short term power production of an individual wind turbine can be evaluated based on the variation of the wind velocity at the hub height or in the turbine blade plane.

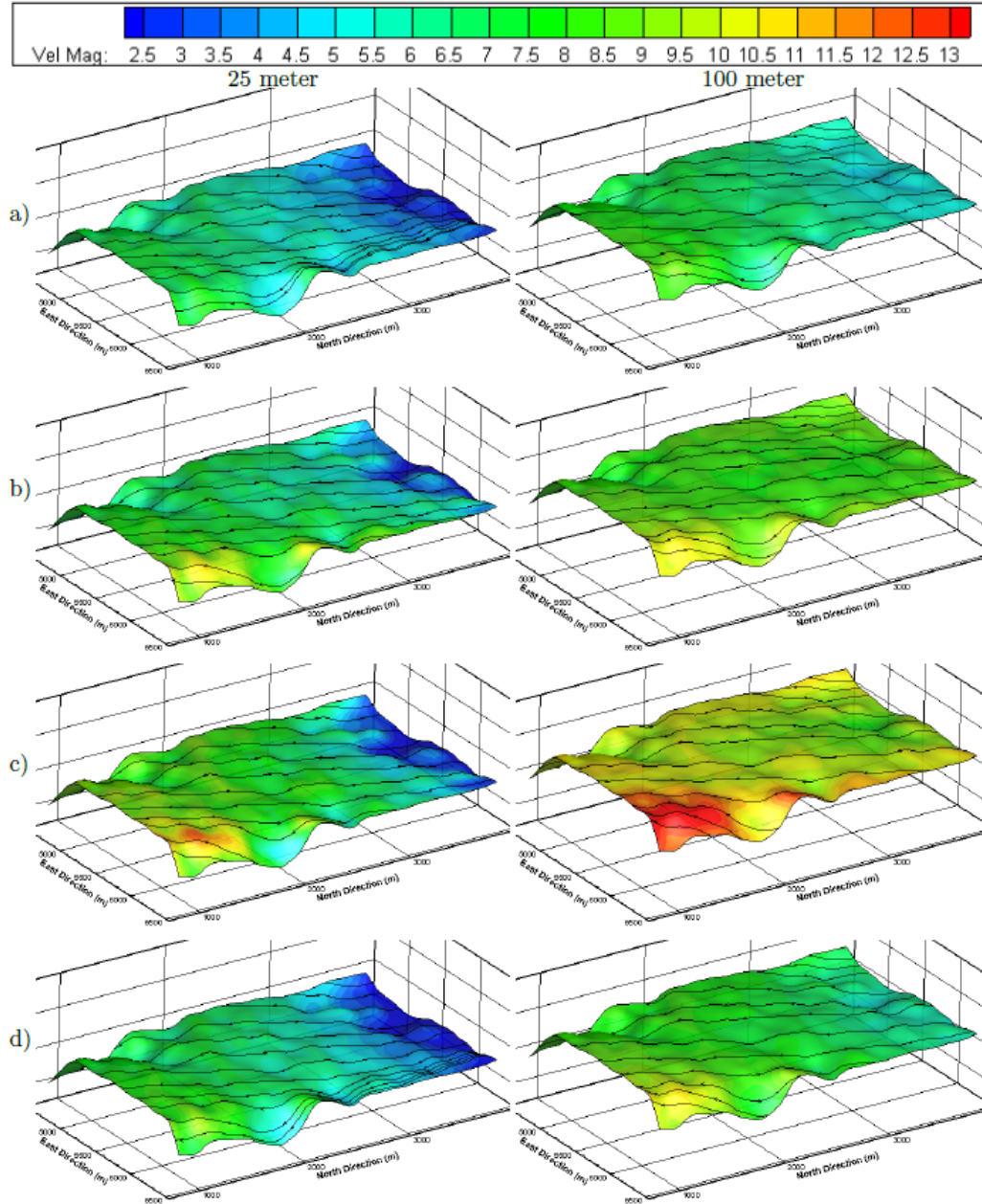


Figure 3.11: Velocity contours and streamlines on terrain following surfaces 25m and 100m above the ground at a) ≈ 6 b) ≈ 12 c) ≈ 18 d) ≈ 24 hours

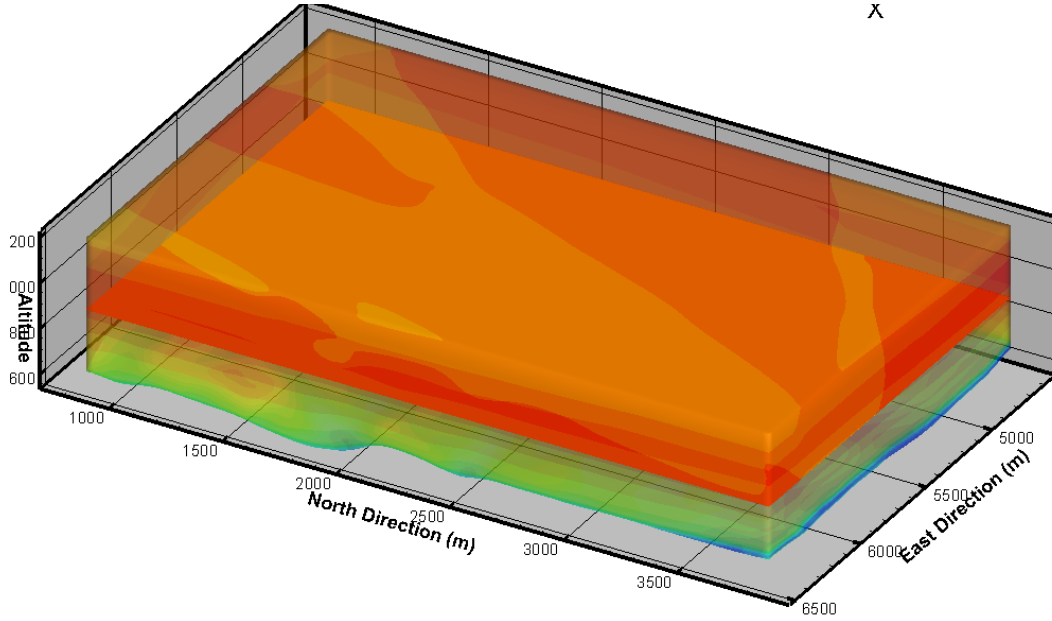
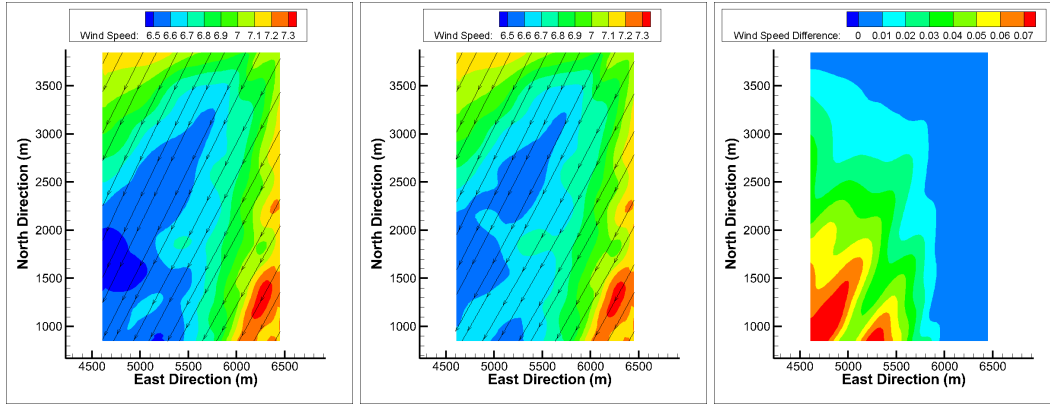


Figure 3.12: Vertical slice for comparison with/out Coriolis acceleration

Effects of the Coriolis acceleration are investigated on a z cut at $800m$ altitude shown in Figure 3.12. Wind speed contours and streamlines at the cut which are obtained with and without the Coriolis implementation into `pimpleFOAM` are given along with wind speed differences in Figure 3.13. As seen in Figure 3.13, the influence of the Coriolis acceleration on wind speed is less than 1% and increases as the wind passes through the computational domain. Upon closer inspection it can be seen that the Coriolis acceleration deflects the flow direction to the right slightly as expected. A more through analysis about the influence of the Coriolis acceleration is given in the Alaiz case study.

The serial computations for a 24-hour simulation take about 60 CPU hours on a computer with a 2.3 GHz Intel processor. In parallel computations, the domain decomposition of the computational grid is given in Figure 3.14, where the the outer boundary cells are collected in a single partition. In the figure different colors denote a different partition, whereas the outer transparent grey domain includes the boundary cells where `timeVaryingMixed` type boundary conditions are imposed.



(a) (b) (c)
Figure 3.13: Wind speed distribution with (a) and without (b) Coriolis acceleration and the difference (c)

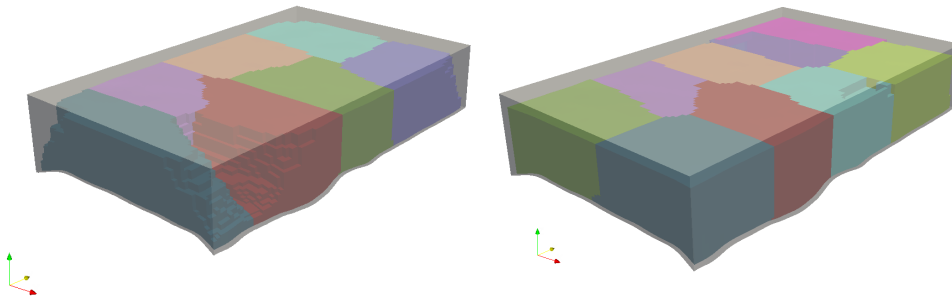


Figure 3.14: Domain decomposition by METIS (left 8, right 10 partitions)

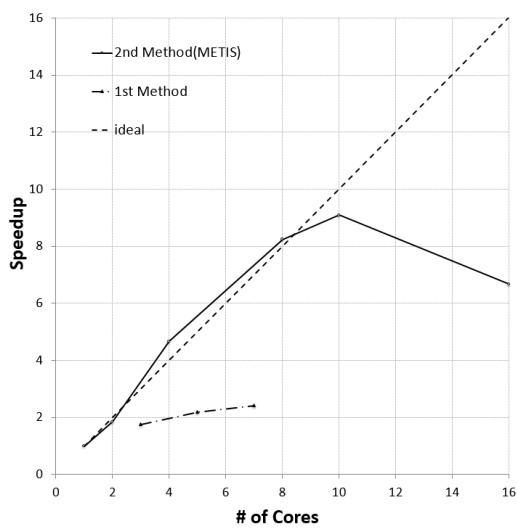


Figure 3.15: Efficiency of parallel computations

The *OpenFOAM* solutions coupled with WRF for a 24 hour period are obtained using 3,5,7 partitions with the first method in which the grid is first partitioned by *OpenFOAM*, then the outer boundary cells are collected in an additional partition. In the second method the boundary cells are first collected into a single giant cell and all the cells are partitioned by METIS into 2,4,8,10 and 16 partitions.

The parallel efficiencies of these methods are assessed and the resulting speed-up curves are given in Figure 3.15. As expected the parallel performance of the first method is poorer in comparison to the latter due to the sub-optimal number of partition faces. In the case of METIS based partitioning, a super-linear speed up is observed with low number of partitions. However, with 16 partitions, the number of cells in the partitions become unbalanced and the fixed number of boundary cells in a single partition not only limits the parallel efficiency of the computations but also increases the computational time needed. If the grid used in the Mut case study is considered, it is of $61 \times 100 \times 27$ size with a total number of 164700 cells and 14794 outer boundary cells. The number of outer boundary cells limits the number of partitions to be used to $164700/14794 \approx 11$ in order to have a near uniform cell distribution among all the partitions. Therefore, decomposing the cells in the domain into more partitions than 11 provides no improvement in parallel computations and reduces the parallel efficiency. In the current study, the METIS based partitioning is used and the optimum number of partitions is determined based on the number of outer boundary cells.

3.3 Case Study: Alaiz Wind Field in Navarra, Spain

Alaiz Mountain range is located in Navarre providence, Spain, towards $15km$ south-east of Pamplona. A wide valley lies to the north at around $700m$ lower altitude, and there are several wind farms to the south on the complex terrain as seen in Figure 3.16. The prevailing wind directions are from North and from South. In this region there are 5 met-masts that extend up to $118m$ above the ground, and are located upstream of several wind turbines.

One of the important features of the region is that the topography is complex with slopes upto 2, which far exceeds the operation envelopes of linearised and CFD based prediction models (max. ≈ 0.4) as mentioned in the Introduction chapter. The presence of a mountain and a complex terrain cause flow features such as flow separation and reattachment. Another feature is that the roughness of the region can be characterised by two levels that sharply differ. The southern and the western parts of the region are mostly covered with $10 - 15m$ tall beech trees. The northern and eastern

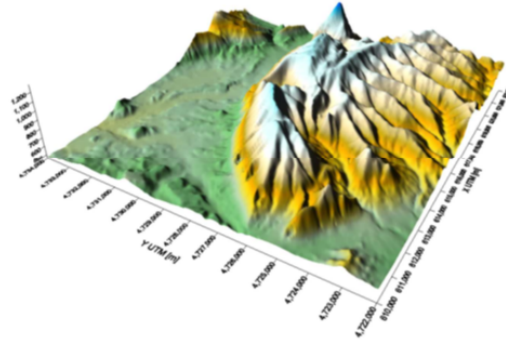


Figure 3.16: Alaiz mountain range

Table 3.2: WRF model settings

Micro-physics	Single Moment 5-class scheme
Long/Short wave radiation	rrtmg Scheme
Surface Layer	Monin-Obukhov Similarity Scheme
Land-Surface	Noah Land Surface Model
Boundary Layer	YSU Scheme

parts of the region are covered with low bushes and shrubs.

The observation data from one of the five meteorological masts, MP5, are obtained from CENER (National Renewable Energy Centre) in Spain. The data contain 10 years of wind speed and direction time series with 10 minute intervals at 5 different altitudes (40m, 78m, 90m, 102m, 118m above the ground).

3.3.1 WRF Solutions

The WRF solutions are obtained for the date 01.01.2015 starting from 00:00 GMT 0 to 24:00. The local terrain data for the WRF solution is downloaded automatically from UCAR (University Corporation of Atmospheric Research) server via WRF . The initial and boundary conditions are taken from the ERA-Interim reanalysis data set with $0.25^\circ \times 0.25^\circ$ resolution which corresponds to $\approx 25km$ for the region. The WRF model settings used are summarized in Table 3.2.

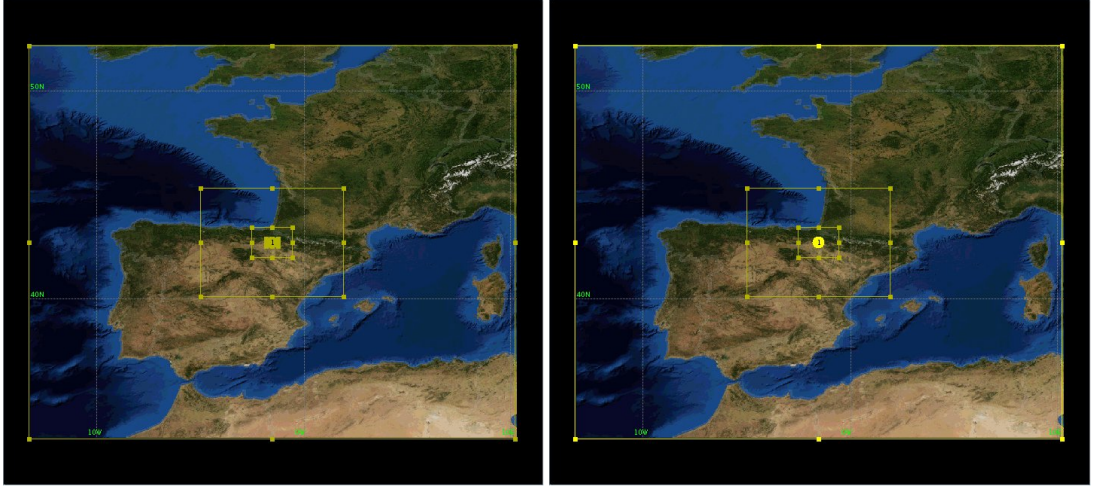


Figure 3.17: Nested domains for WRF solutions (4 nests (left), 3 nests (right))

The WRF solution is performed using two different nest configurations with different resolutions (Figure 3.17) for sensitivity analysis. In the first configuration shown on the left in Figure 3.17, there are 3 nests that are of 88×67 , 76×55 , 61×43 horizontal size from larger to smaller respectively whereas the parent domain is of 100×81 size. All the domains share the same number of η levels which is set to 61. The time step for all the domains is taken as 90 seconds. The surface resolutions of the nests are $24.3km$, $8.1km$, $2.7km$ and $0.9km$ respectively starting from the parent domain to the innermost nest. In the second configuration shown on the right in Figure 3.17 only two nests are employed, The surface resolution of the innermost nest is now $2.7km$.

3.3.2 High Resolution Terrain Modeling for OpenFOAM and Grid Generation

The computational grid for the *OpenFOAM* solutions are based on a high resolution terrain topography obtained from ASTER GDEM as in the Mut case study with a horizontal resolution of $38m$. The region of interest which is $6.5km$ in East-West direction and $5.2km$ in the north-south direction is chosen such that the mountain range and the complex topography surrounding the met-mast is contained within. The topography and the modelled surface of the Alaiz region are given in Figure 3.18.

The top of the solution domain is given as the projection of the topography as a flat surface at $2200m$ altitude and the first vertical length is given as $2m$ as the length of

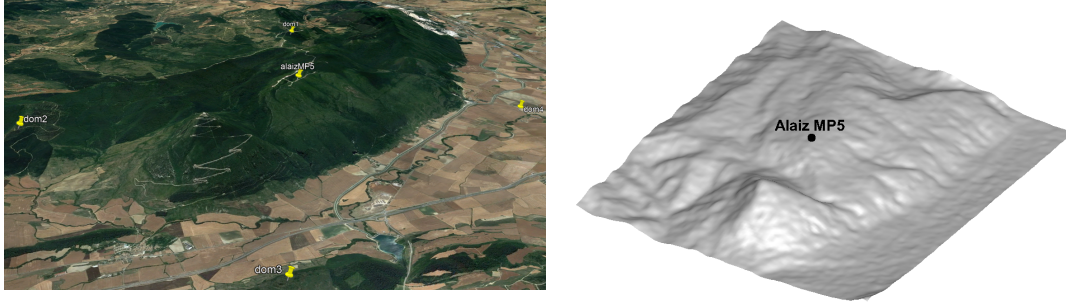


Figure 3.18: Alaiz topography and modelled ground surface

the first grid level should not exceed the roughness height as mentioned in the Near Wall Treatment and Roughness sub-section. Using the high resolution topography, *GAMBIT* software is used to generate rapidly stretching grids in the vertical direction, using a journal file automatically generated for given vertical and horizontal resolutions as given in the Methodology chapter. Grids for 4 different vertical and 3 horizontal resolutions are generated for a grid independence study which are given later in this section. A sample grid with $38m$ horizontal resolution and 25 vertical levels are given in Figure 3.19.

3.3.3 Grid Independence Study

Preliminary grid independence studies are conducted on the Alaiz region with a spatially varying BC profile from WRF for the date 01.01.2015 at 00:00 with different horizontal resolutions and number of vertical levels. The topographical model of the region is the same as the one used in the previous section with $38m$ horizontal resolution ASTER-GDEM data set. The velocity profiles obtained at the MP5 met-mast location are then compared with each other.

Firstly, rapidly stretching structured grids with various number of vertical levels are generated as seen in Figure 3.20(left), while keeping horizontal resolution and the first vertical distance constant at $38m$ and $2m$ respectively.

The resultant velocity profiles from structured grids with 13, 25, 38 and 50 vertical levels are compared at the MP5 met-mast location in Figure 3.21. Although no significant changes are observed in the vicinity of the topography which is at approxi-

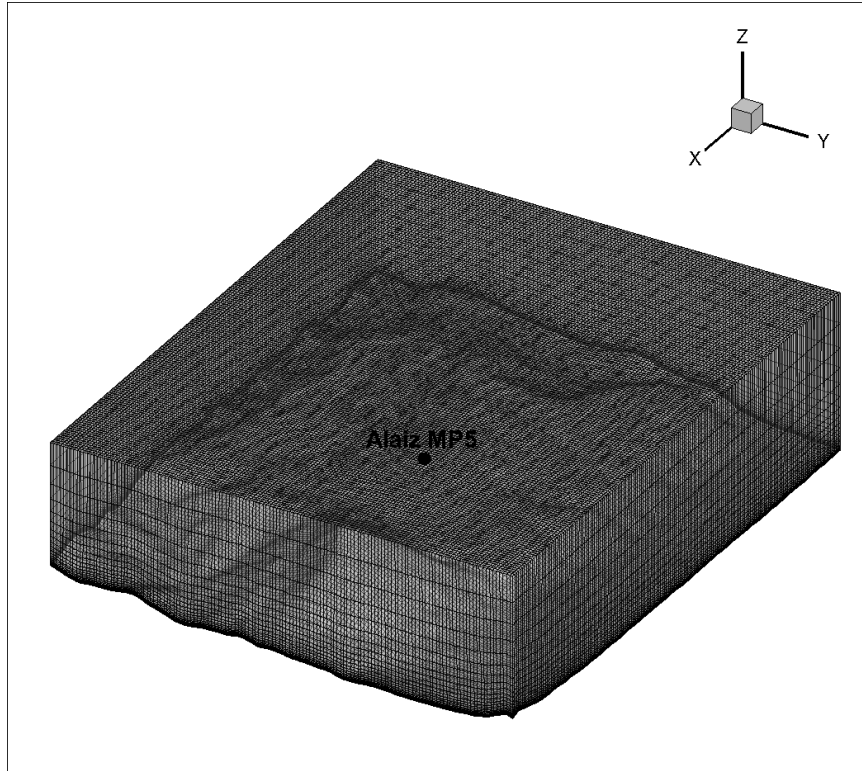
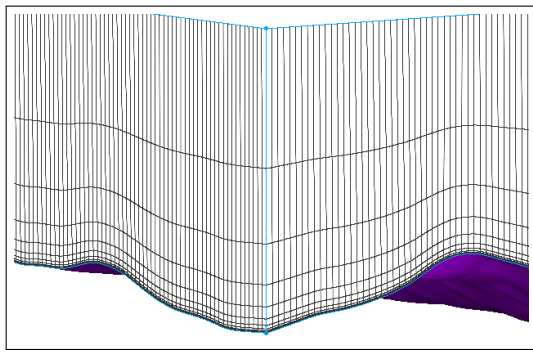


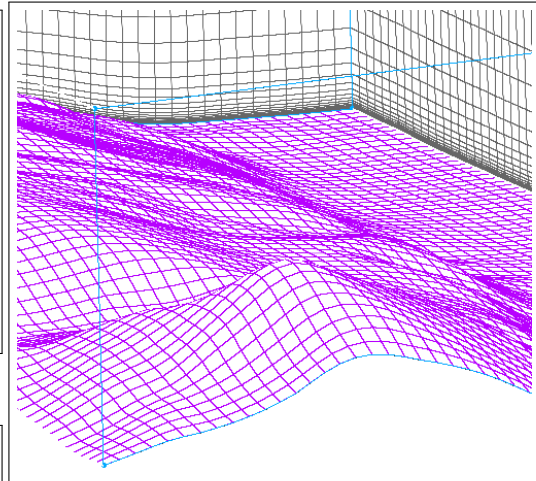
Figure 3.19: Computational grid for OpenFOAM solutions in the Alaiz region

mately $1116m$ altitude, it can be seen that the difference between the 13 vertical level case and the others increases with the altitude. The difference between the velocity magnitude with 50 vertical level grids is less than 5% for that of 13 vertical levels whereas for the 25 vertical levels case, it is less than 2%. This result is to be expected as the first vertical length of all 4 cases are the same, $2m$. At the top of the domain, the case with 13 vertical levels differs from the others, which is attributed to the low vertical resolution at the top.

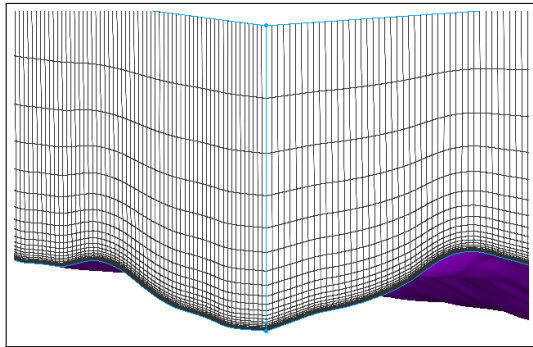
Next, structured grids with 25 vertical levels are generated with $30m$, $38m$ and $60m$ horizontal resolutions as seen in Figure 3.20(right). The velocity profiles are compared at the met-mast location in Figure 3.22. As seen, the velocity profiles except the one with $60m$ horizontal resolution are in reasonable agreement. The difference in velocity profiles between the $30m$ and $38m$ horizontal resolution is less than 1%, but with the $60m$ horizontal resolution case there exists significant differences in near the ground level as seen in Figure 3.22 (right). In the light of these results, 25 vertical levels and $38m$ horizontal resolution are chosen to be used in the remaining *OpenFOAM* solutions.



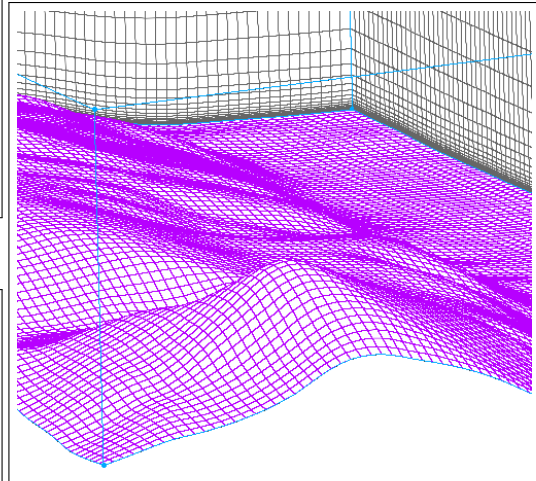
13 vertical levels



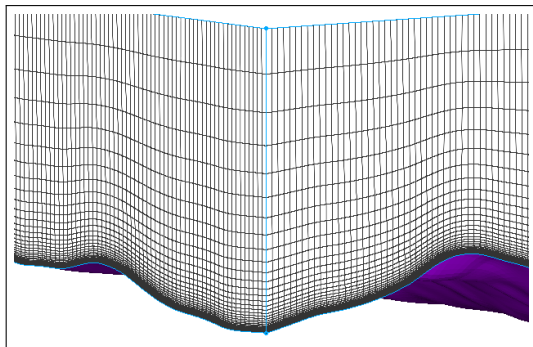
60 m horizontal resolution



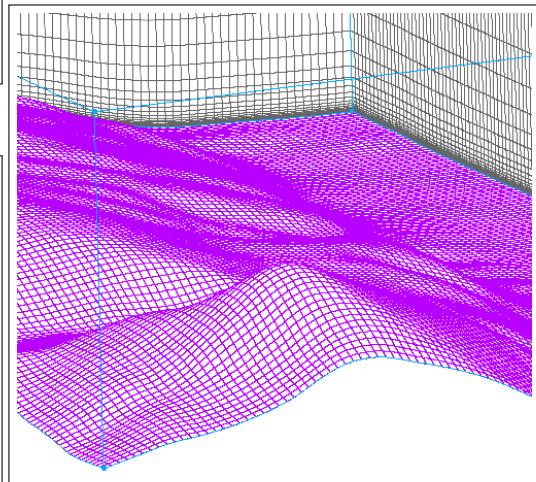
25 vertical levels



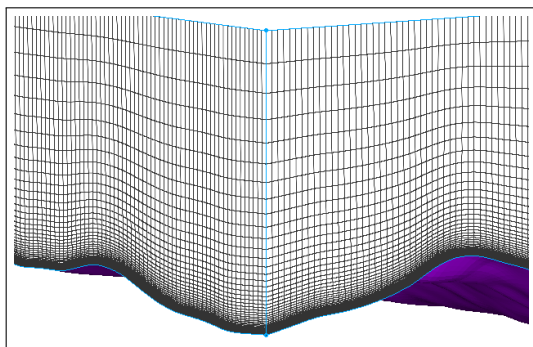
38 m horizontal resolution



38 vertical levels



30 m horizontal resolution



50 vertical levels

Figure 3.20: Grid resolutions in the Alaiz region

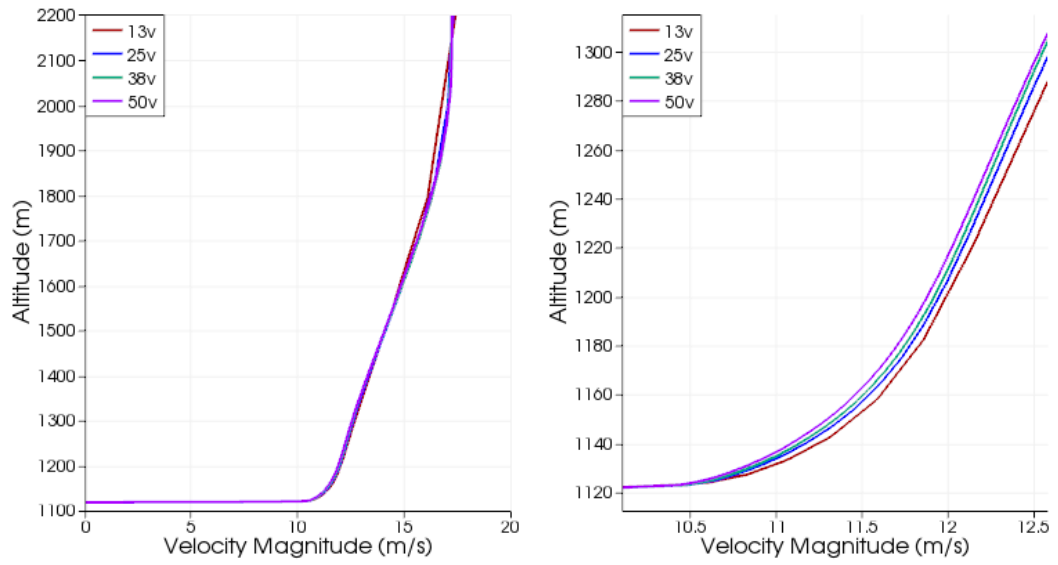


Figure 3.21: Effect of the vertical grid resolution on the velocity profile (left) and the zoomed view (right)

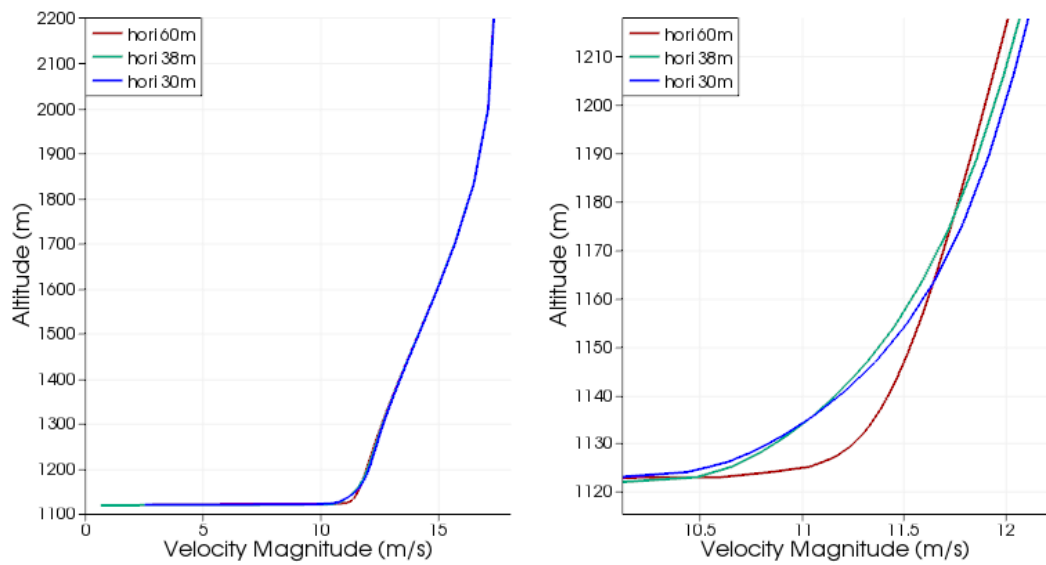


Figure 3.22: Effect of horizontal resolution on the velocity profiles (left) and zoomed view (right)

3.3.4 Comparison of Turbulence Models

Velocity profiles with different turbulence models are compared on the optimum grid configuration which has 25 vertical levels, 38m horizontal resolution, and 2m vertical resolution on the ground.

$k - \epsilon$, $k - \omega$ SST and realizable $k - \epsilon$ turbulence models with a steady spatially varying velocity boundary conditions from WRF for the date 01.01.2015 at 00:00 are used with the same constant free stream turbulence. Since $k - \epsilon$ and realizable $k - \epsilon$ models use k and ϵ values as the boundary conditions, the constant values for k and ϵ are determined with the conventions for external flows suggested in the literature:

$$k = \frac{3}{2} (U I)^2 \quad (3.4)$$

$$\epsilon = C_\mu \frac{k^{\frac{3}{2}}}{l} \quad (3.5)$$

where I is the turbulence intensity and taken as 10 % according to the studies conducted in Alaiz [13], U is the mean velocity taken as the mean inflow velocity, C_μ is the turbulence model constant taken as 0.09. l is the turbulent length scale and it is approximated in external flows by:

$$l = 0.07\delta_{ABL} \quad (3.6)$$

where δ_{ABL} is the length of the atmospheric boundary layer and taken as 1km.

For $k - \omega$ SST model ω is given as:

$$\omega = \frac{\sqrt{k}}{l} \quad (3.7)$$

where k is turbulent kinetic energy and l is the turbulent length scale.

As seen in Figure 3.23 the velocity profiles for 3 different turbulence models are

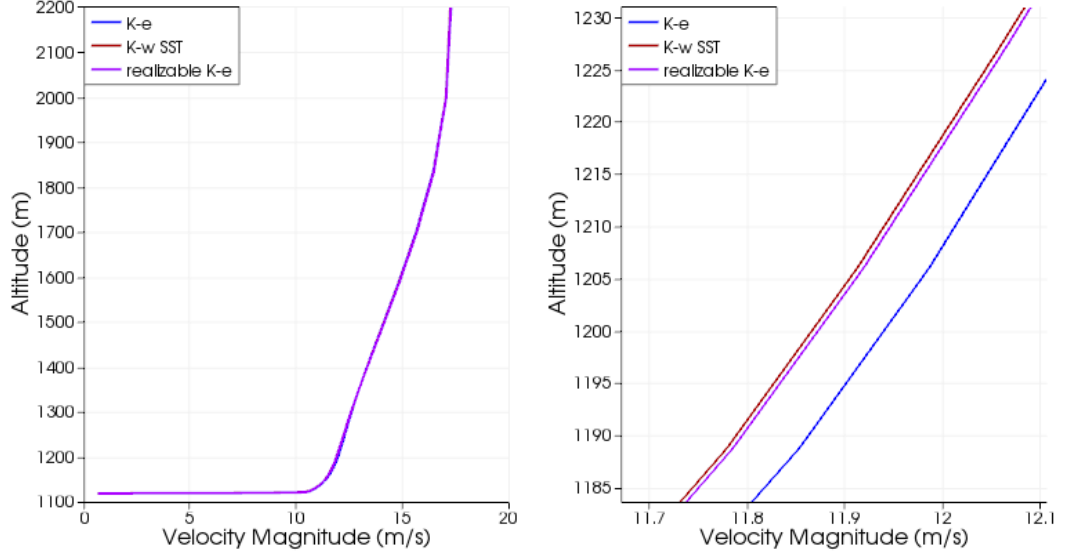


Figure 3.23: Velocity profiles predicted by the turbulence models

in good agreement. The difference between standard $k - \epsilon$ model with respect to $k - \omega SST$ and realizable $k - \epsilon$ is observed to be less than 1% at all altitudes. Also, it is important to note that the maximum difference between the realizable $k - \epsilon$ and $k - \omega SST$ model is less than 0.1%. As $k - \omega SST$ and realizable $k - \epsilon$ turbulence closures take more time then the standard $k - \epsilon$ model and the $k - \epsilon$ model is reported to have more reliable results in atmospheric flow simulations [27, 48], the standard $k - \epsilon$ model is used for the remaining unsteady *OpenFOAM* solutions.

3.3.5 Unsteady *OpenFOAM* solutions coupled with WRF

OpenFOAM based unsteady atmospheric flow solutions coupled with WRF predictions are performed by using the methodology developed for the date 01.01.2015 starting from 00:00 to 24:00. Horizontal dimensions for the region of interest is $6.5km$ in East-West direction and $5.2km$ in North-South direction.

Rapidly stretching structured grids with $38m$ horizontal resolution and 25 vertical levels with the first vertical length of $2m$ is used as decided from the mesh independence studies. The $168 \times 137 \times 25$ structured grid generated with the topographical model for the region has 575400 cells. As the k^+ value is around 800, standard wall

functions for the $k - \epsilon$ turbulence closure is used for near wall treatment.

Unsteady boundary condition profiles obtained from $0.9km$ horizontal resolution WRF predictions are utilized using the developed BC class `timeVaryingMixed`. Boundary condition type for each at each 38266 boundary faces are changed between Neumann and Dirichlet dynamically based on the WRF data at the simulation time. `timeVaryingMixed` imposes time interpolated velocities extracted from the WRF solution for the inflow, whereas the velocity gradients are set to zero at the outflow boundaries.

`pimpleFOAM` solver with implemented Coriolis acceleration is used for the unsteady coupled solutions. `pimpleFOAM` uses multiple cycling over the same time step using the last iteration final value as initial guess for the next iteration (outer correction loops) and under-relaxation of the variables between consequent outer iterations. The main advantage of the solver is that the time step can be adjusted according to the maximum Courant number, which in turn reduces instabilities. Convergent results with a maximum of 40 Courant number with 30 corrector steps is achieved for the Alaiz case study.

Domain decompositions for this case is done with METIS and examples with 8 and 16 partitions are given in Figure 3.24. Each color in Figure 3.24 represents a different partition whereas the transparent light grey section is for the main partition where all the cells belonging to the `timeVaryingMixed` are grouped together.

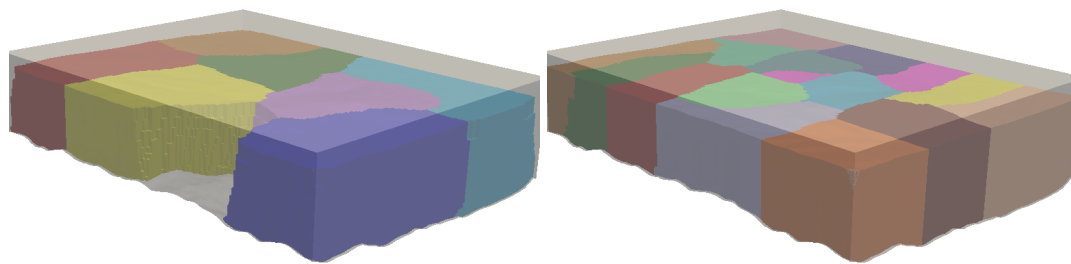


Figure 3.24: Domain decomposition for the Alaiz region with 8 and 16 partitions

Parallel implementation for the developed BC class is done for 16 processors as the number of faces belonging to `timeVaryingMixed` limits parallel performance as mentioned in the parallel implementation section in Method chapter. Using the par-

allel implementation methodology developed, accurate daily runs can be completed within 4 hours using 16 2.3 Ghz cores.

3.3.5.1 Comparison of `timeVaryingMixed` Boundary Conditions with WRF results

Inflow and outflow profiles for the *OpenFOAM* domain in Figure 3.25 (left) are compared with WRF solution on the $8 \times 6 \times 13$ subset of WRF grids given in Figure 3.25 (right) that barely encloses the high resolution *OpenFOAM* domain. As both domains exists in different physical spaces, Latitude and Longitude coordinates of the WRF solution are just scaled to match the size of the *OpenFOAM* domain whereas η coordinates are transformed to altitude based on the pressure from WRF solution. So, changes in the height of WRF sub-domain can be observed upon closer inspection due to changes in pressure over time. x and y axis points towards East and North direction for both *OpenFOAM* and WRF solutions.

As seen in Figure 3.25, the topographical model for the WRF is far smoother and simpler than the complex terrain model used in *OpenFOAM*. This demonstrates the need for matching the ground levels of both models to avoid undefined regions on outer boundaries, before the extraction of BC profiles from the WRF solution. It is also evident from Figure 3.25 that the wind speed, direction and profile change drastically in both time and space. It can be said that assuming spatially constant and steady logarithmic profiles is a huge oversimplification for the short-term wind dynamics. Velocity magnitude contours of both WRF and *OpenFOAM* solution are in good agreement as well as velocity profiles at inflow. There are some differences for the outflow profiles as expected, because of the `timeVaryingMixed` BC switching to zero-gradient for velocity in case of outflow.

Surprising thing to note is that even if the topographical model for the WRF domain is not complex, there are deflections on flow direction to the East on the North-East corner of the domain, especially evident in 15th and 24th hour. These deflections are also seen in the *OpenFOAM* solutions North-East corner inflow profiles. Also, high shear velocity profiles at the South-East corner of the domain are maintained in the *OpenFOAM* domain indicating that the stretching method is utilized successfully.

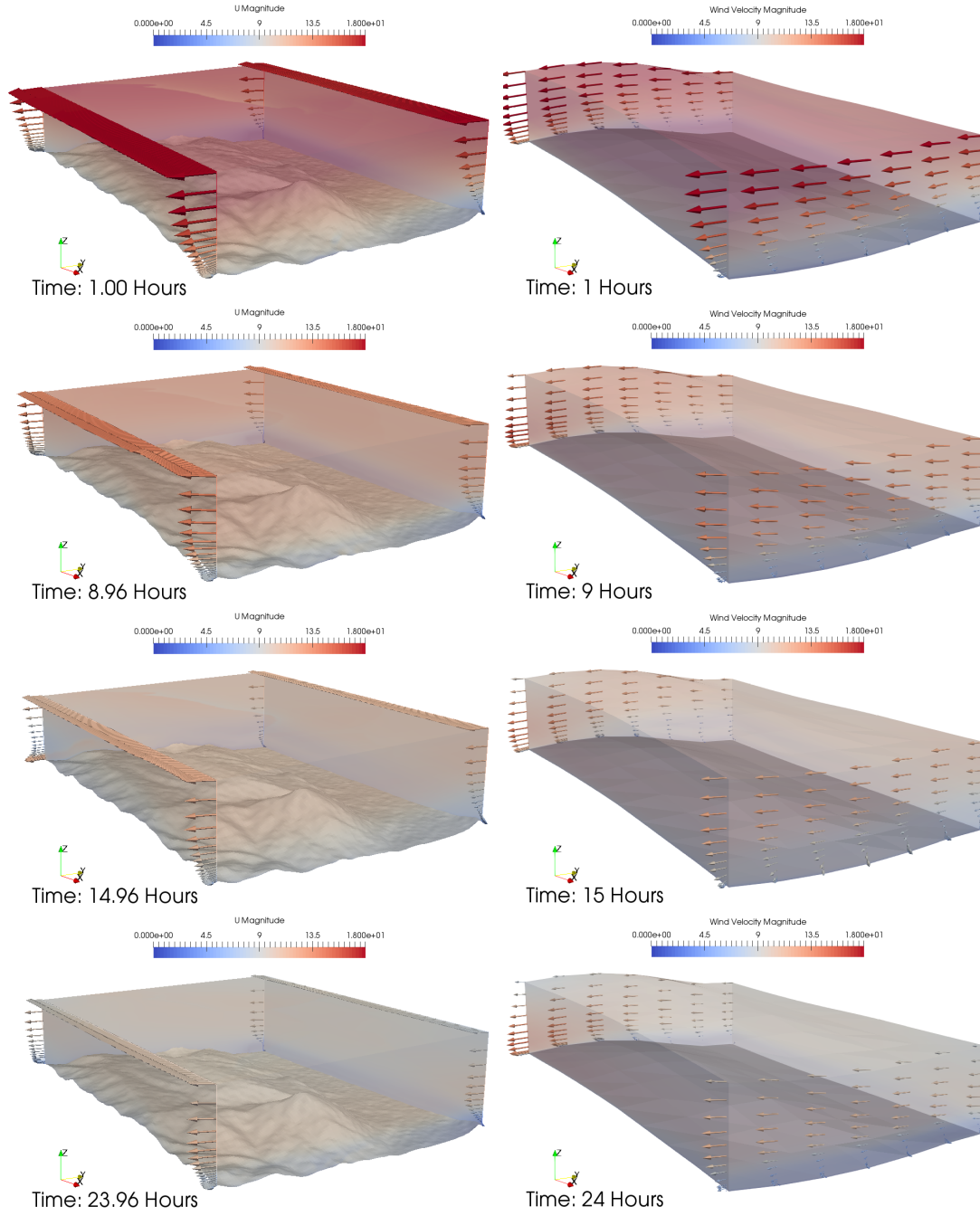


Figure 3.25: Inflow/outflow velocity profiles in OpenFOAM and WRF solutions

3.3.5.2 Effects of Complex Terrain

In order to examine the effects of complex topography, the unsteady flow field data are extracted from two cuts which are at $x = 620350m$ easting which coincide with a mountain top in Alaiz, and $z = 1040m$ altitude as shown in Figure 3.26.

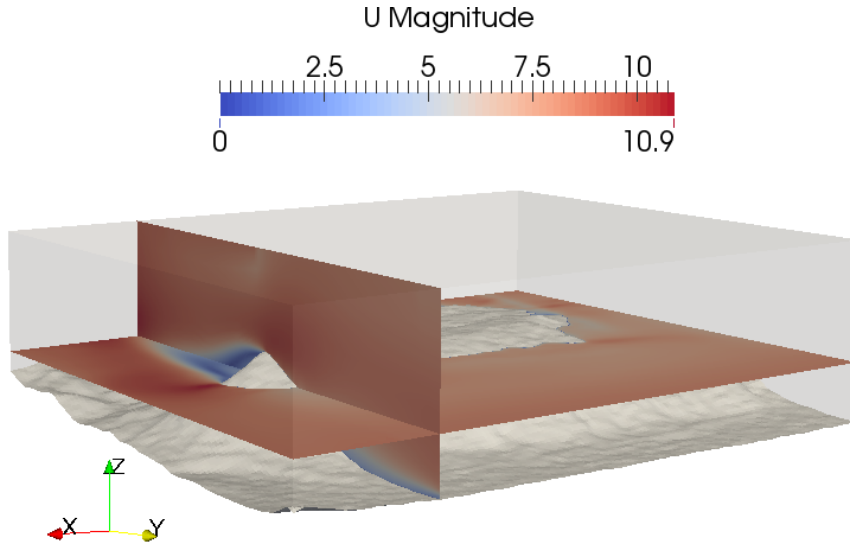


Figure 3.26: Vertical and horizontal cuts in the OpenFOAM solution domain

In Figure 3.27, contours of velocity magnitude and turbulent kinetic energy, k , are given side by side for 6 different simulation time covering the whole day on the extracted x cuts. It is clear from Figure 3.27 that the velocity magnitude increases as the altitude increases at all times which is as expected. The wake behind the mountain is clearly seen from the velocity magnitude contours and the distance covered by the wake increases with the incoming wind velocity. The main factor responsible for the generation of turbulent kinetic energy is seen to be the influence of the mountain which makes sense. It can also be deduced from the 2.9 Hour and 12.33 Hour contours that turbulent kinetic energy increases with increasing wind velocity which is consistent with the theory. It can be said that physics of the problem is still captured to an extent, even if the turbulent kinetic energy, k , and turbulent dissipation rate, ϵ , are given constant according to the conventions in literature as mentioned in the Turbulence Modelling sub-section.

Next, streamlines colored with velocity magnitude on the x cuts are examined in Figure 3.28 and Figure 3.29 at the same 6 times covering the whole day. As seen in Figure 3.28 and Figure 3.29, there is a recirculation zone in the wake of the mountain at around 3rd hour of the simulation, dying out at around the 9th hour and reappearing at around 17.5th hour. Taking a closer look at the 12.33rd hour, a small recirculation zone near the top of the mountain can be observed although the rest of the flowfield seems attached.

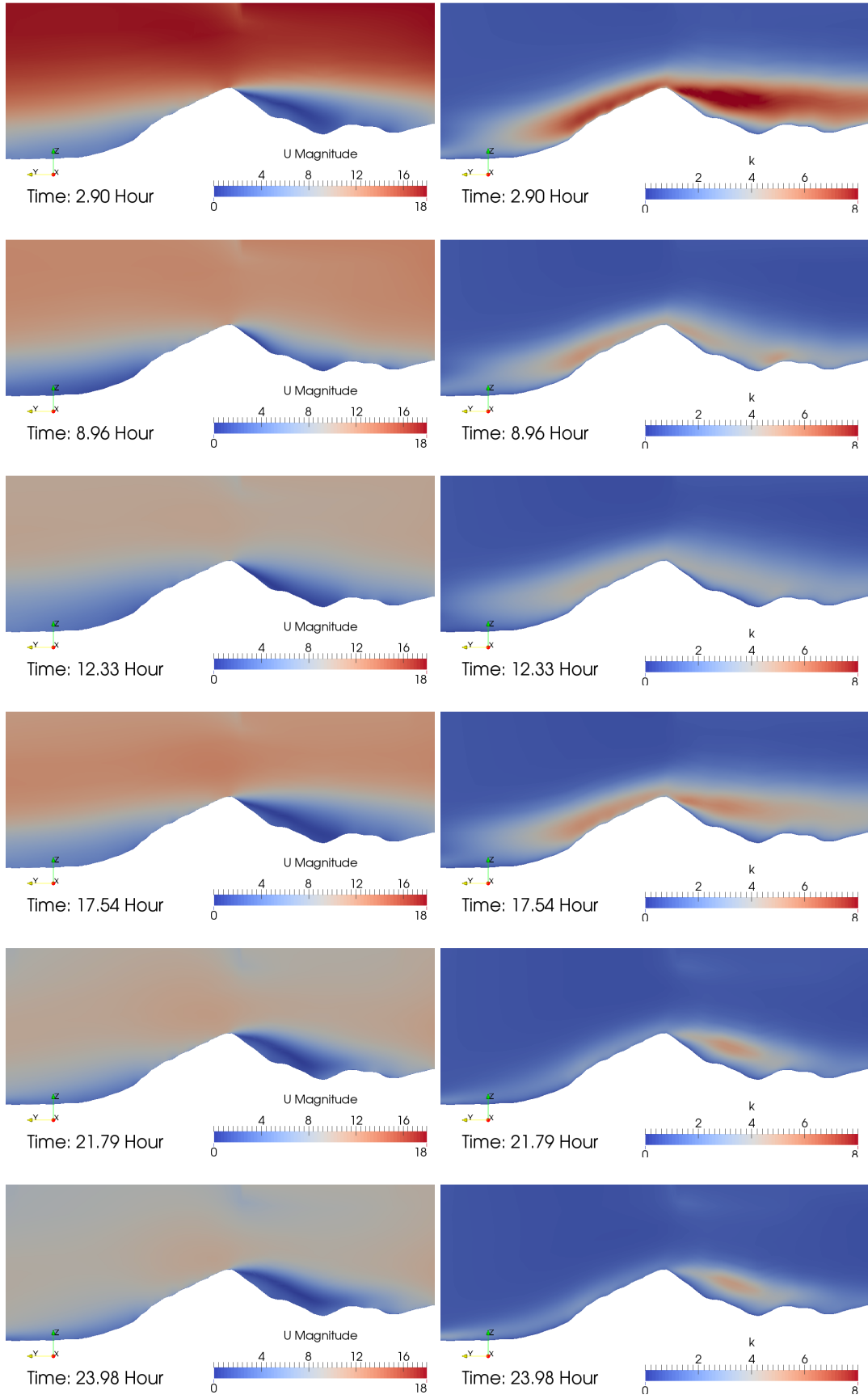


Figure 3.27: Wind speed and turbulent kinetic energy distributions in the vertical cut

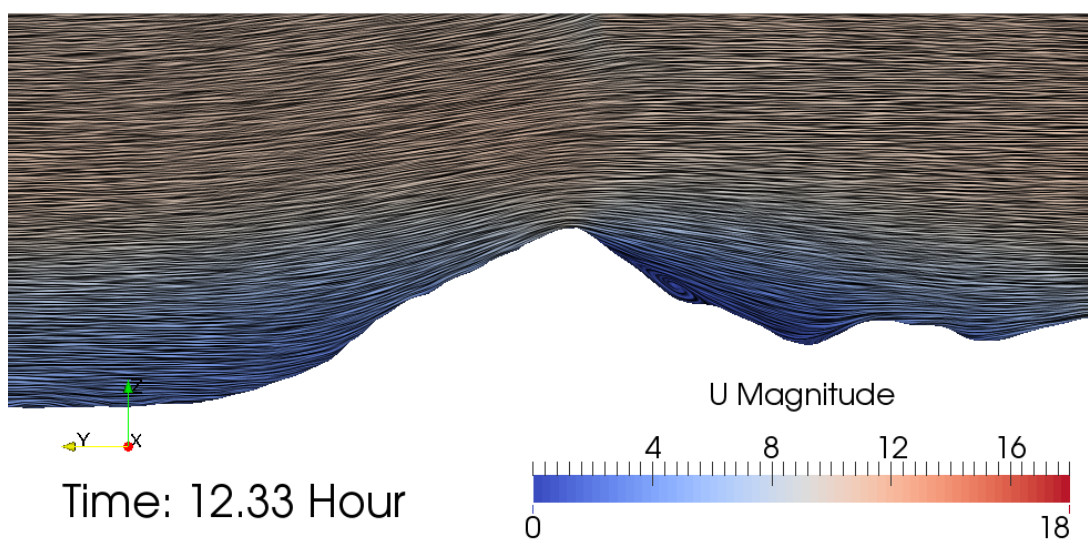
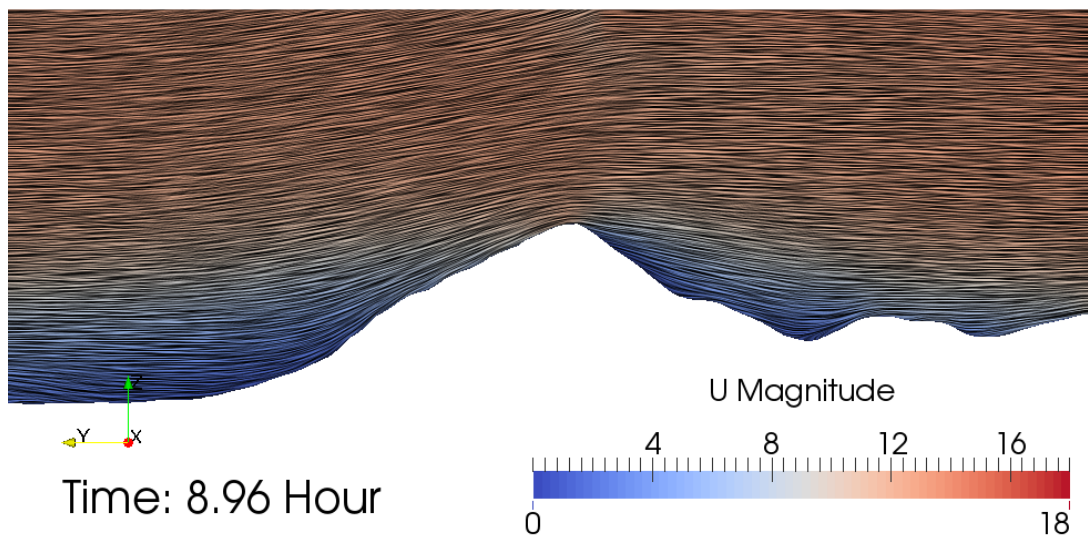
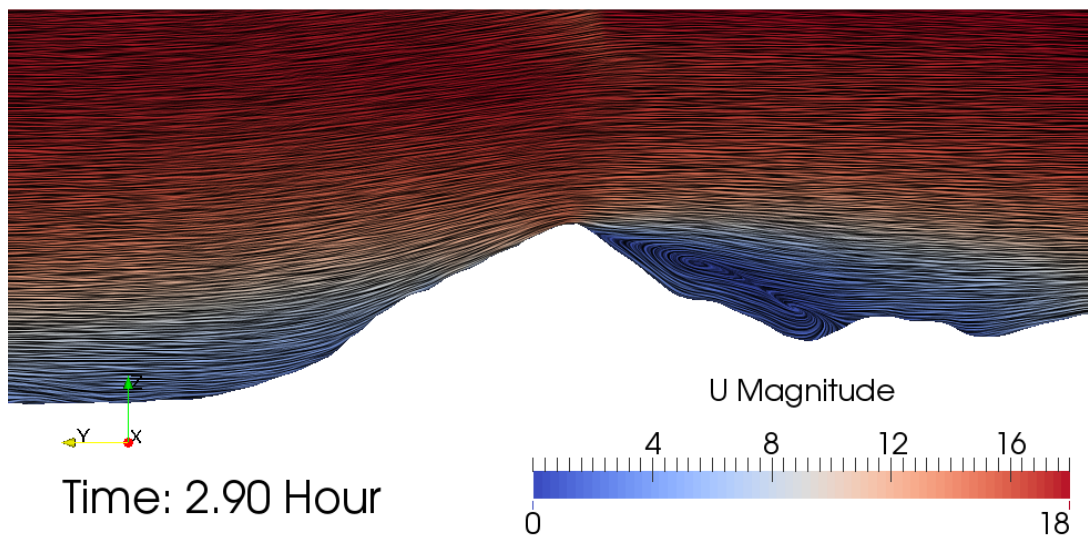


Figure 3.28: Streamlines colored with velocity magnitude on the vertical cut

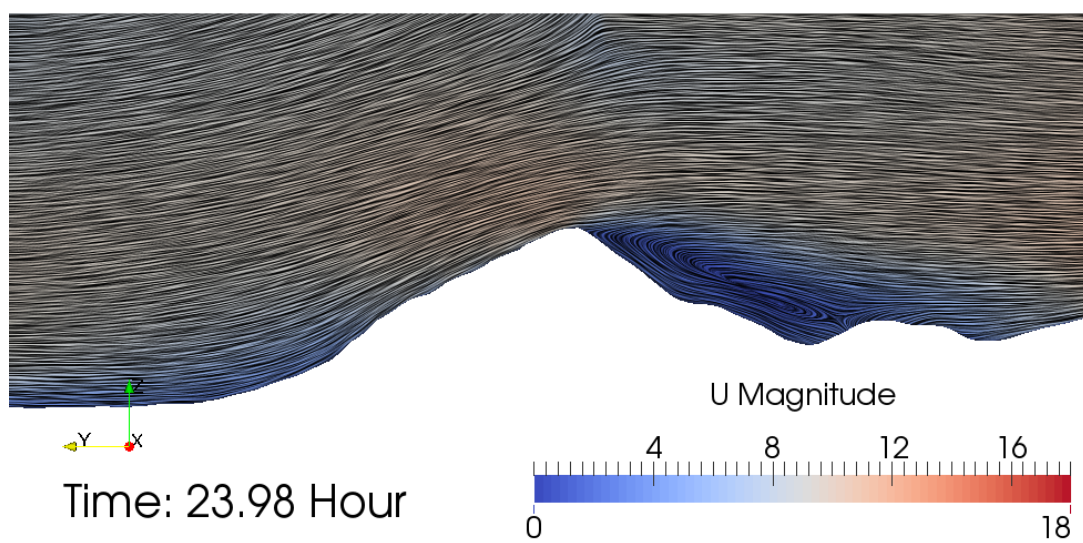
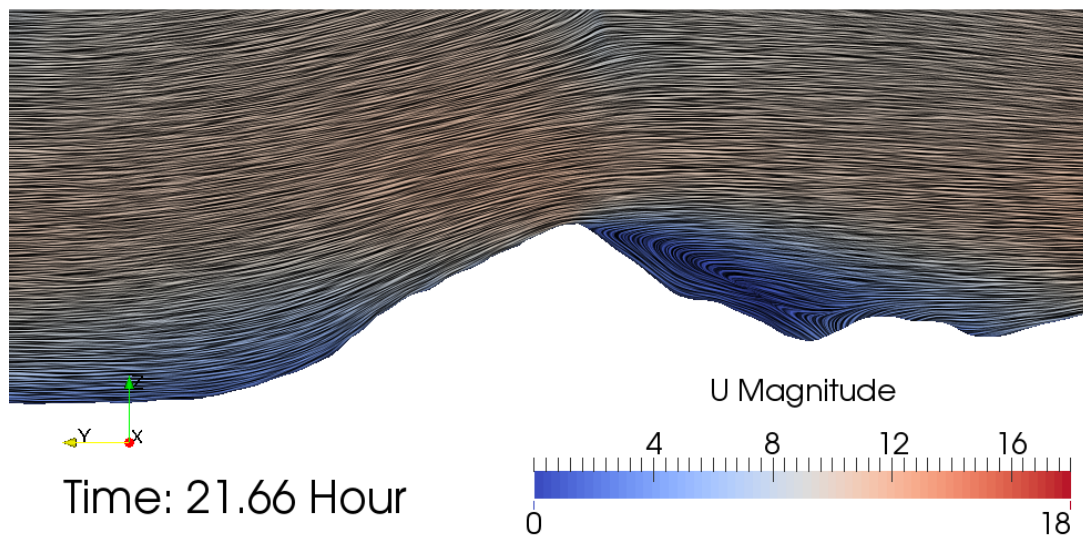
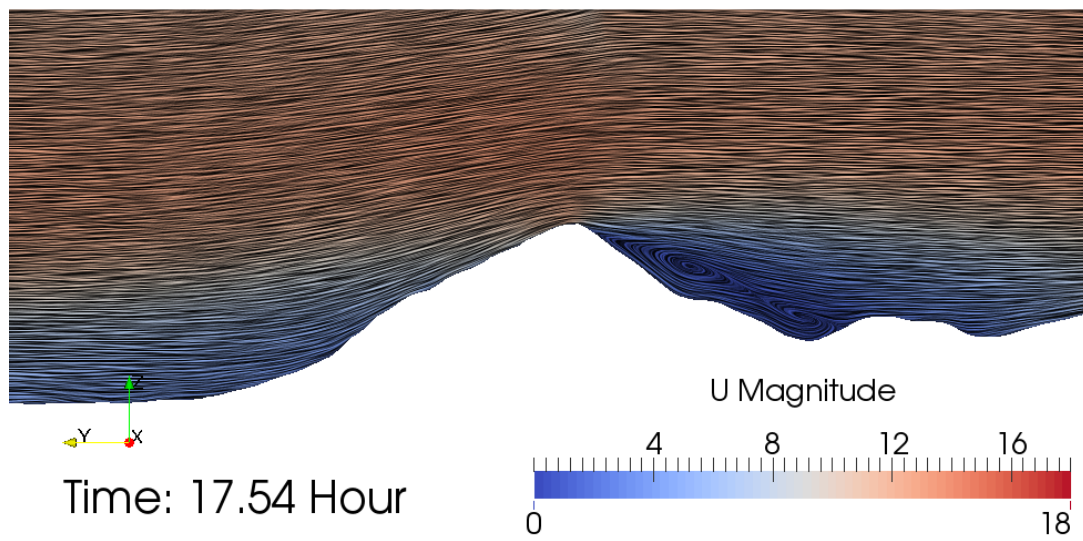


Figure 3.29: Streamlines colored with velocity magnitude on the vertical cut

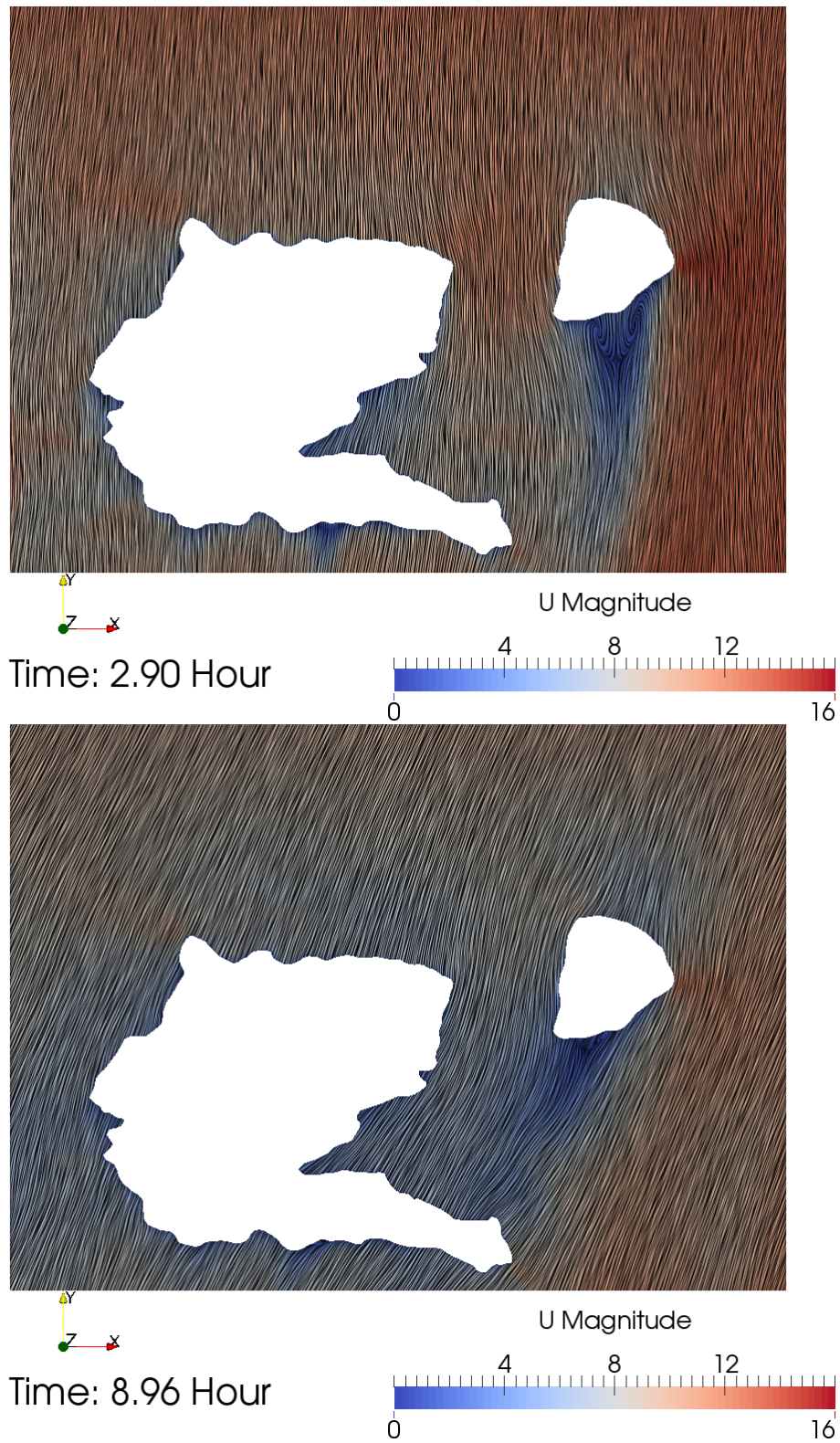


Figure 3.30: Streamlines colored with velocity magnitude on the horizontal cut

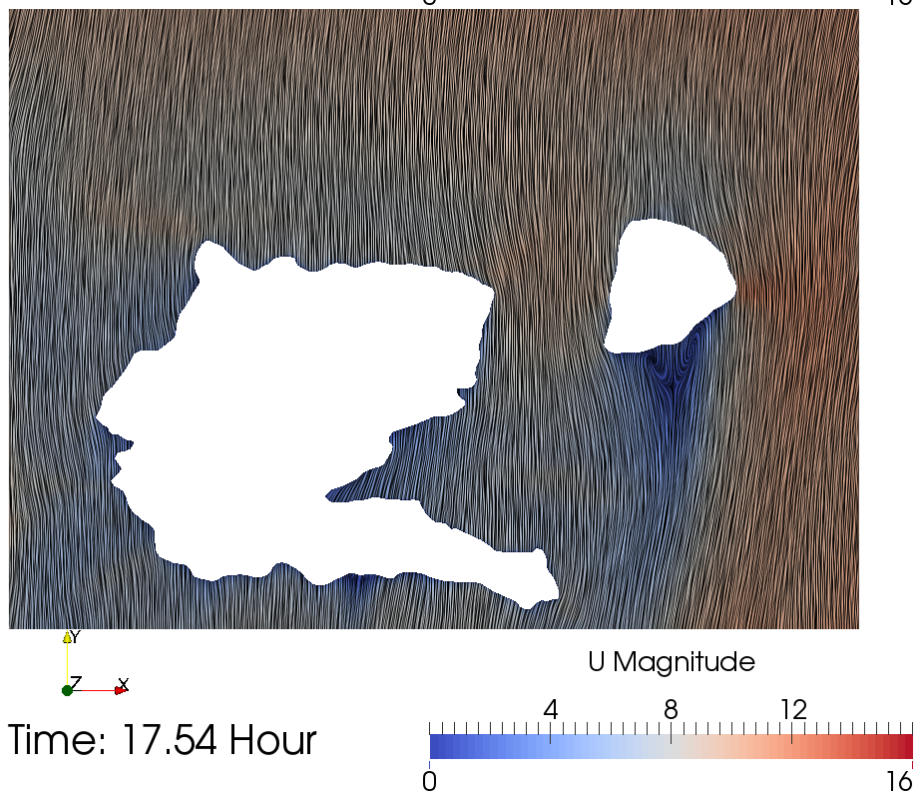
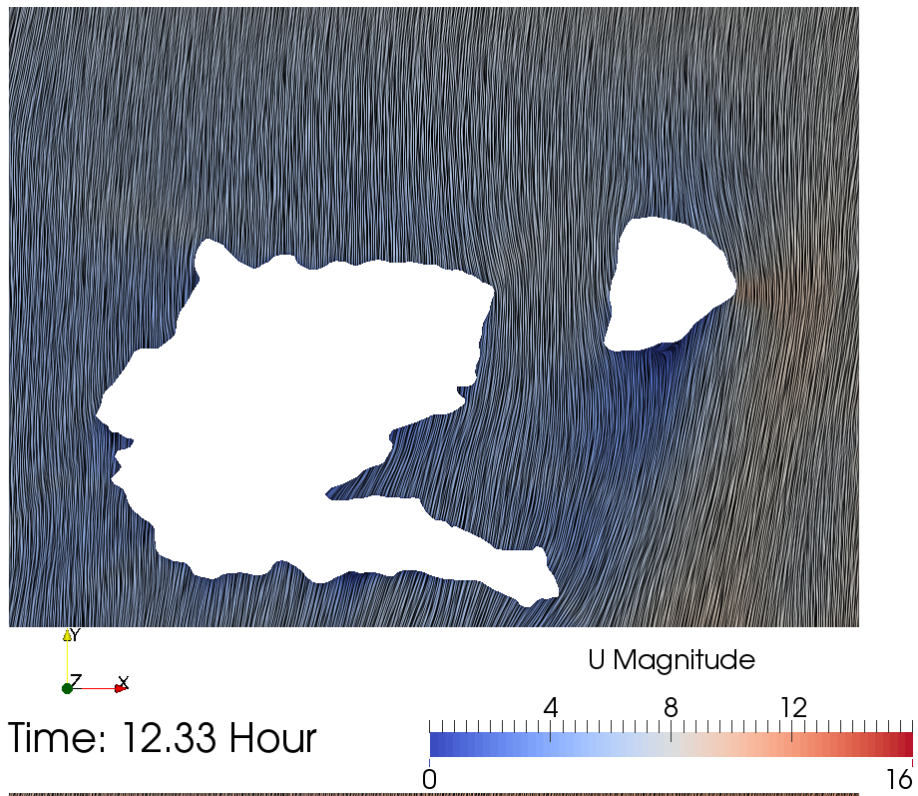


Figure 3.31: Streamlines colored with velocity magnitude on the horizontal cut

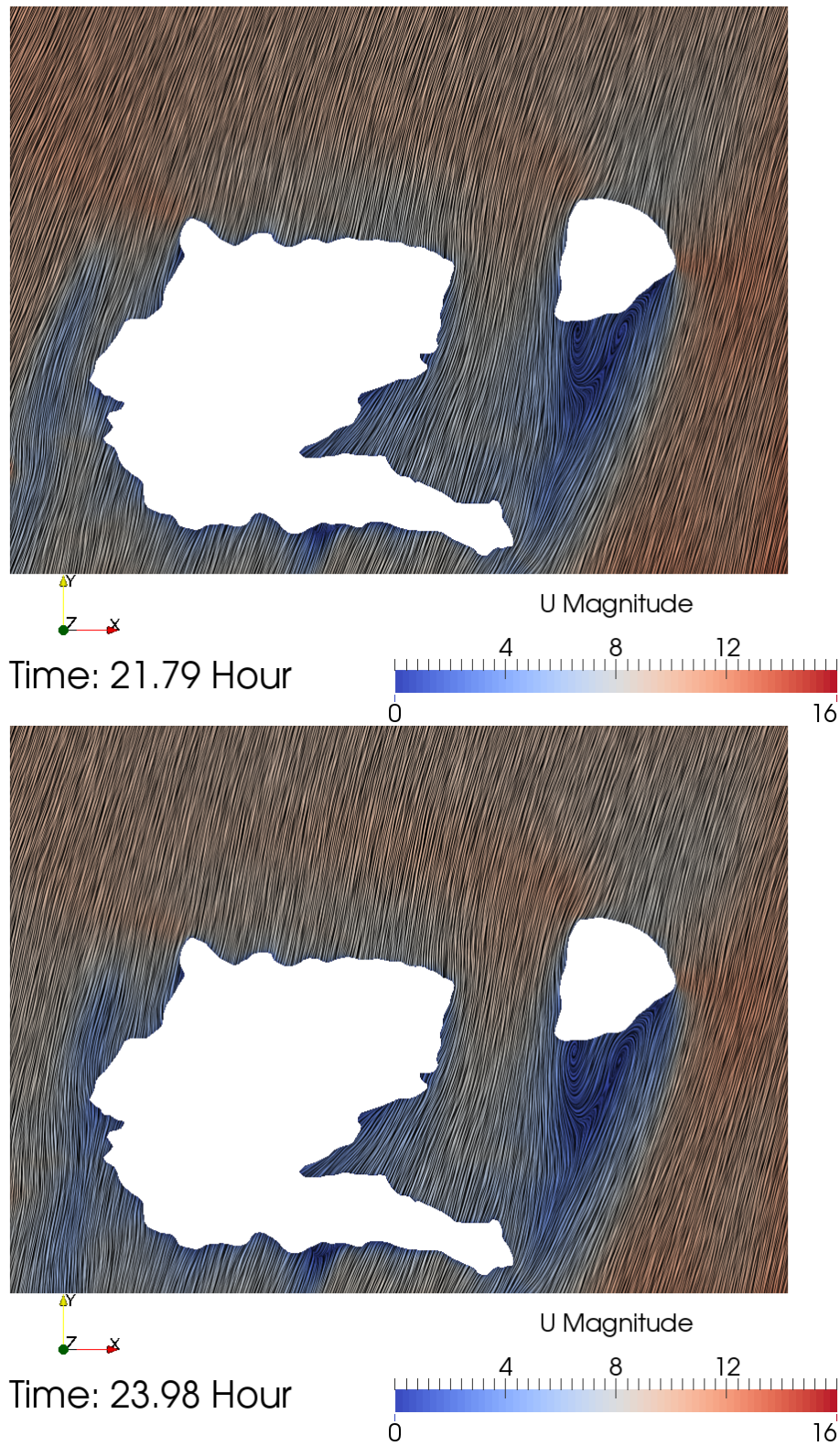


Figure 3.32: Streamlines colored with velocity magnitude on the horizontal cut

The streamlines colored with velocity magnitude on the horizontal cut plane at 1040m

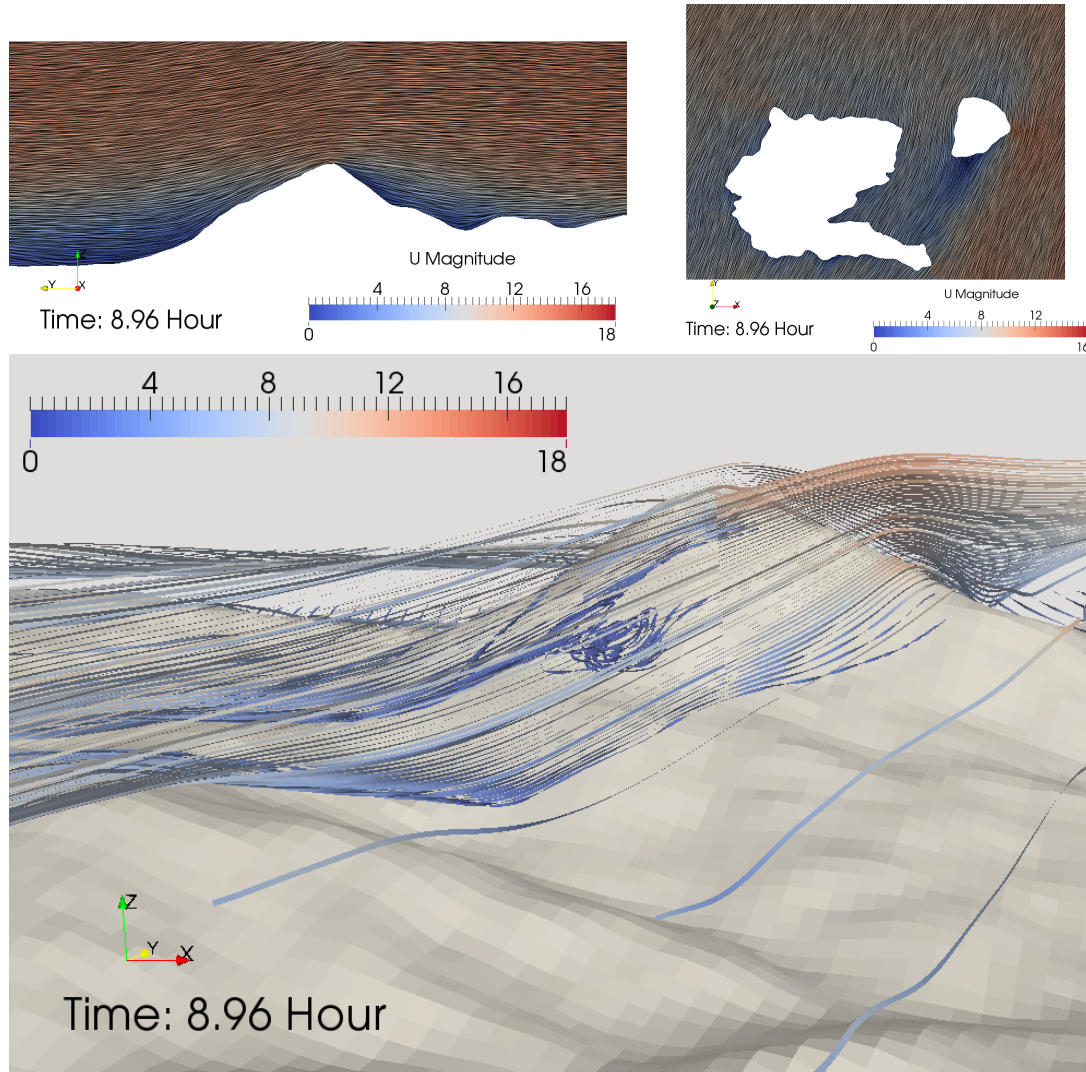


Figure 3.33: Streamlines colored with velocity magnitude and the 3D representation of the flow field

altitude are examined in Figure 3.30 and Figure 3.31. Horizontal direction change of the wake behind the mountain in time is easily seen from Figure 3.30 and Figure 3.31. At 2.9th hour of the simulation the wake behind the mountain is to the south with two distinct circulation bubbles, when at around 9th hour the separation bubbles disperses and the wake is drastically shifted to the south-west, only to shift to the south again at around 12.33rd hour. A similar cycle can be seen at around 17.5th hour again with two distinct circulation bubbles in the southern direction. At about 22th hour the bubbles disperse and the wake turns into the south west direction.

While 2D streamlines at the x and z cuts explains the complex flowfield to an extent, velocity magnitude colored streamlines from these cuts are given side by side along

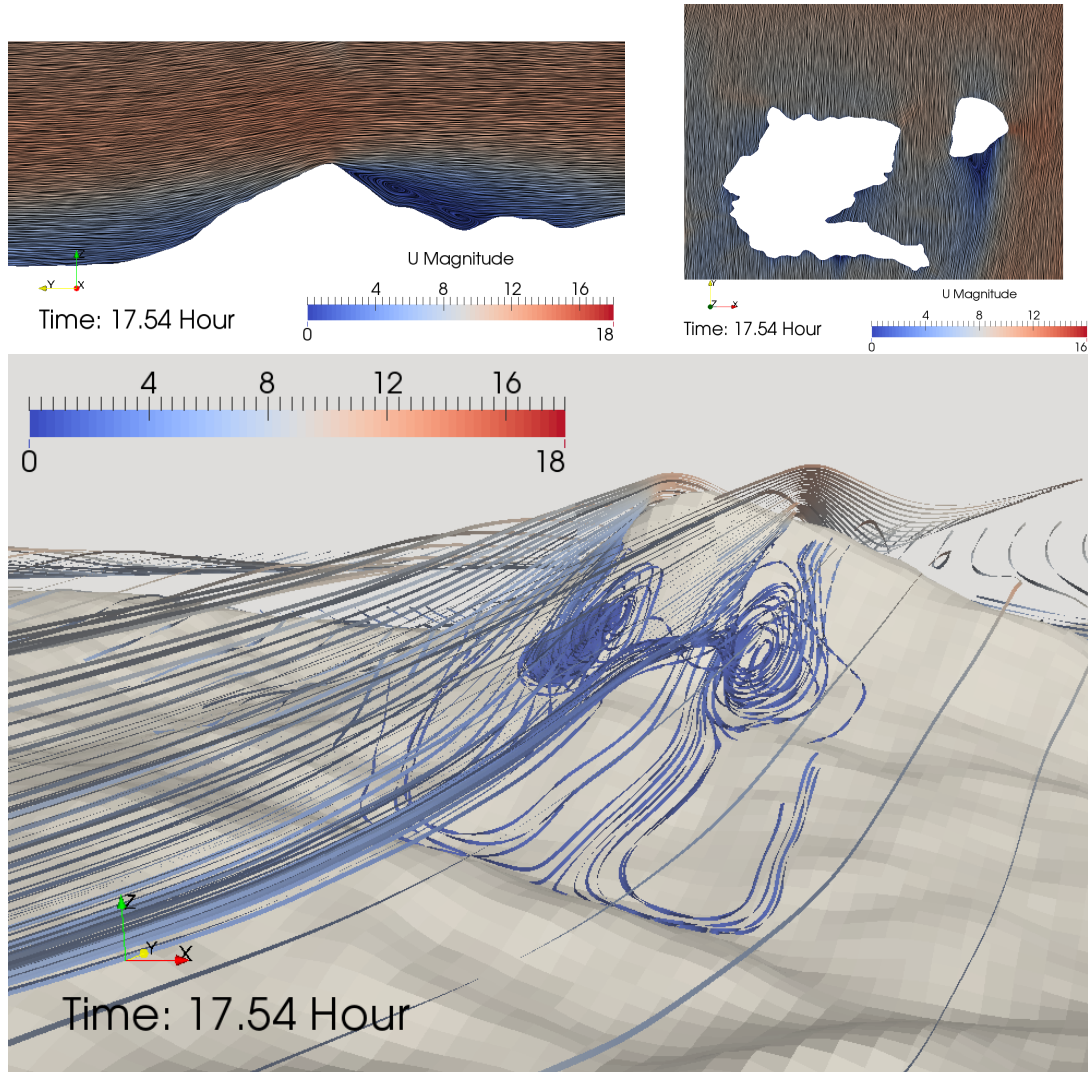


Figure 3.34: Streamlines colored with velocity magnitude and the 3D representation of the flow field

with a 3D representation of flowfield near the mountain's wake at 9th and 17.5th hour, in Figure 3.33 and 3.34 for a better visualization.

In Figure 3.33, a relatively smooth wind field mostly without flow separation is observed on the x cut whereas for the y cut there are a two small vortices just behind the mountain as the wind is coming from North-East direction. The flow pattern behind the mountain can be observed more throughoutly in the 3D representation of the flow field below. It is seen that the flow is mostly attached except for the two small vortices just behind the mountain top at the 9th hour.

Conversely at 17.5th hour in Figure 3.34, even if the wind velocity is similar in mag-

nitude as in the 9th hour, as the flow direction is from the North, parts of the complex topography encountered by the flow is blunter resulting separation zones and reattachments aftermath. Two distinct recirculation bubbles near the mountain top and farther in the wake are observed in the x cut while for the z cut, two vortices created by the flow separation behind the mountain became more distinct as expected. The complex vorticity pattern that is seen vertical in x cut and horizontal in z cut behind the mountain top can be seen in the below figure which has the same streamline seeds as given for the 9th hour. It can be said that time dependent, complex terrain effects can be captured and accounted for using the methodology developed which is not possible with the current wind energy prediction methodologies as mentioned in the Introduction section.

3.3.5.3 Velocity Profiles at MP5 Met-Mast and Diurnal Cycle

Velocity profiles at the MP5 met-mast location are extracted and shown at 3 different times till midday and 3 after midday in Figure 3.35. The same argument about the inlet and outlet profiles is more evident in Figure 3.35, where the magnitude, direction of the wind velocity and shape of the velocity profiles changes drastically with time.

It is important to note stability effects such as diurnal cycle described in the Introduction chapter might be captured with unsteady boundary conditions from WRF even if the governing equation set doesn't contain density variations. The development of the velocity profiles can be given as an example as seen in Figures 3.36 and 3.37. The first velocity profile on the left is taken at around 5 o'clock and the second one is taken at 12 o'clock. As seen from the Figure 3.36, velocity profile at night time is similar to what is described in Stull(2012) for stable atmospheric boundary layer at night [45]. It is worth mentioning that although the velocity profile for the daytime has some similarities with what is shown in Stull 2012 [45] for unstable daytime velocity profile as given in Figure 3.37, its agreement is less when compared to the night time case for stable atmospheric boundary layer. This results is not surprising since the stability results from the heating from the ground and buoyancy effects associated with that, which are not modelled in the methodology developed.

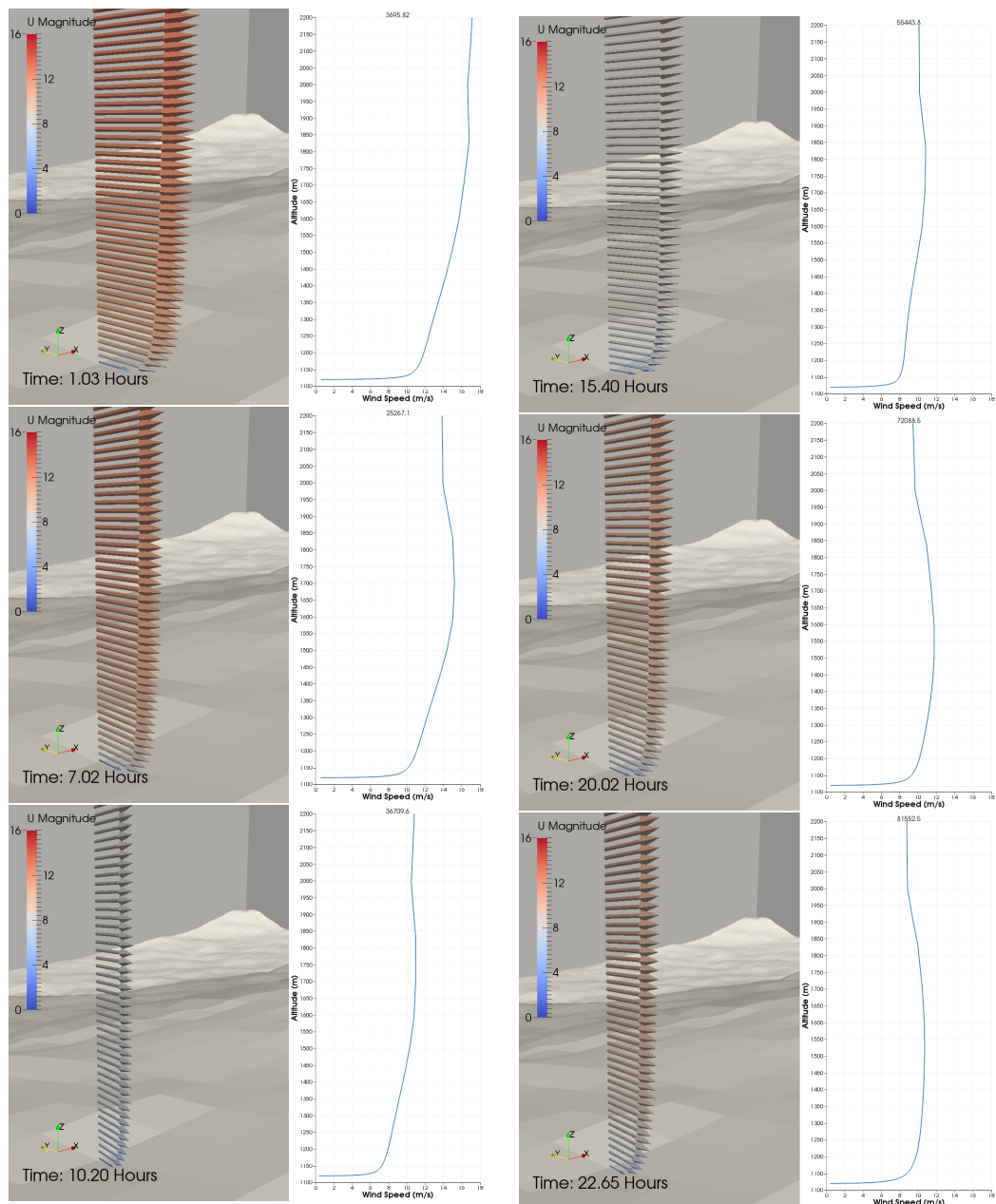


Figure 3.35: Velocity profiles at the met-mast location before(left) and after(right) mid-day

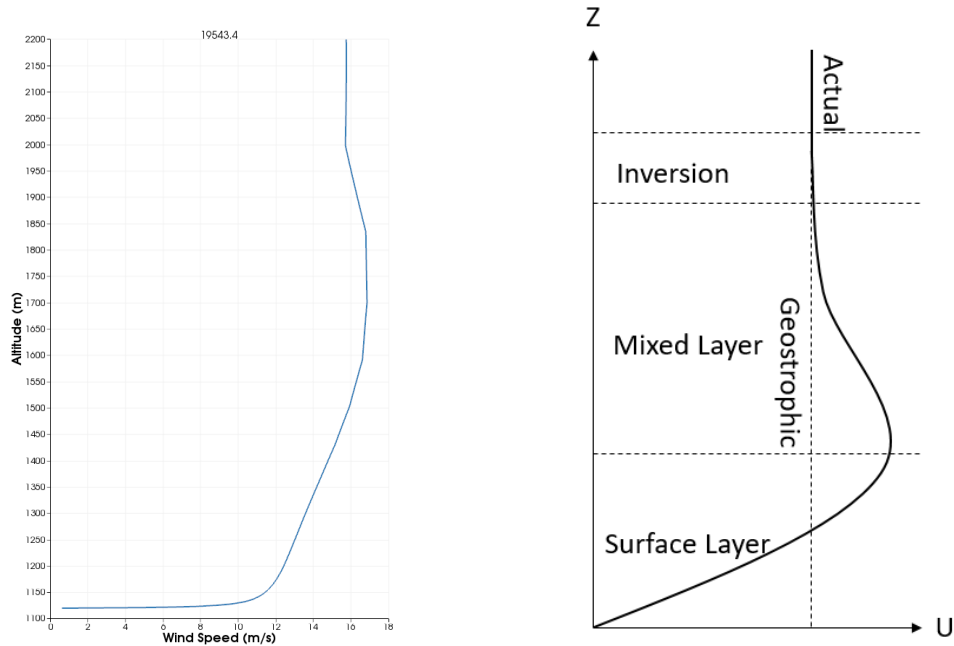


Figure 3.36: Velocity profile predicted by OpenFOAM solution at about 5 o'clock (left) and the diurnal cycle description for the night-time (right)

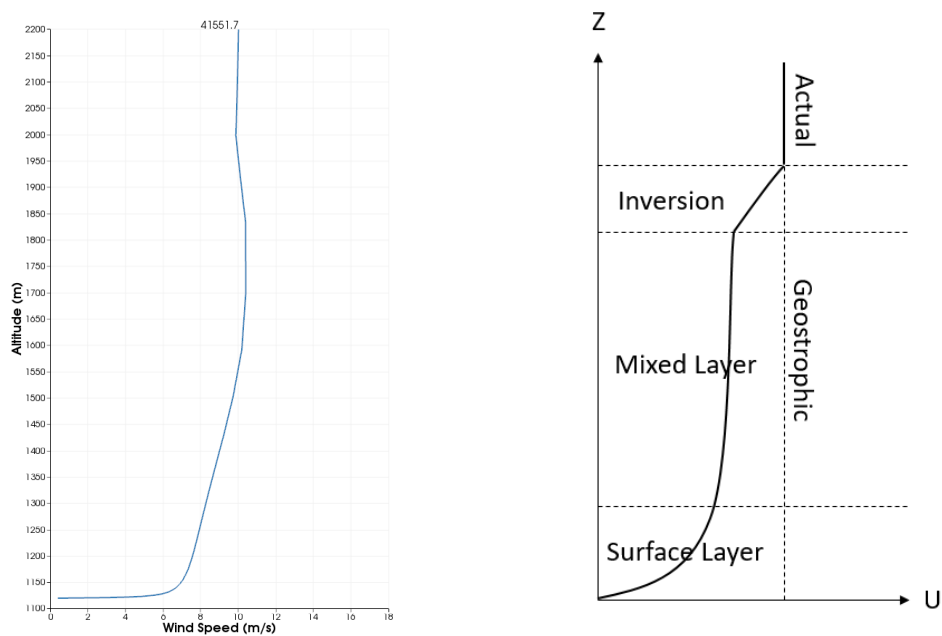


Figure 3.37: Velocity profile predicted by the OpenFOAM solution at about 12 o'clock (left) and the diurnal cycle description for the night-time (right)

3.3.5.4 Effects of Coriolis Acceleration

As mentioned earlier in the Methodology chapter, the Coriolis effects depend on the non-dimensional parameter called the Rossby number which is:

$$R_o = \frac{V}{L2\Omega\sin\varphi} \quad (3.8)$$

where $\Omega = (2\pi/23h\ 56m\ 4secs) = 7.2921 \times 10^{-5}sec^{-1}$ is the rotational speed of the Earth and average wind velocity V is taken as $9m/s$ which is more or less in agreement with both studies conducted by CENER and the average of the time dependent flow velocity from the WRF solution. L is taken as the maximum distance ($5.2km$) for the *OpenFOAM* domain in the North-South direction as the prevailing wind direction is from North to South.

So, the Rossby number for the Alaiz Case can be calculated as

$$R_o = \frac{9m/s}{5200m \cdot 7.29 \times 10^{-5}sec^{-1} \sin 42.69} = 17.50 \quad (3.9)$$

which doesn't seem to be negligible or too significant. In order to investigate its influence, unsteady *OpenFOAM* solutions coupled with WRF are performed with and without the Coriolis acceleration term implemented into *pimpleFOAM*. All the results are obtained with spatially varying roughness models for the region of interest, which is given in the next subsection.

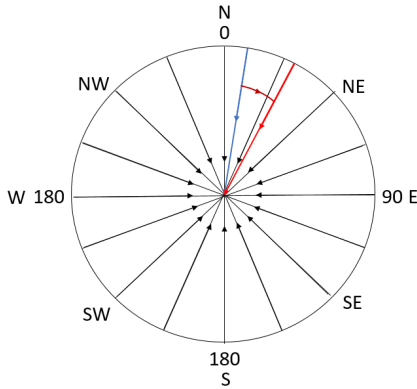


Figure 3.38: Wind direction convention for the met-mast data

Wind direction convention for the met-mast data is given in Figure 3.38. Blue vector in wind direction convention represents a generic wind direction whereas dark red is the deflection due to Coriolis force in Northern Hemisphere and light red vector is the resulting wind direction. Time series for wind speed/direction at the MP5 met-

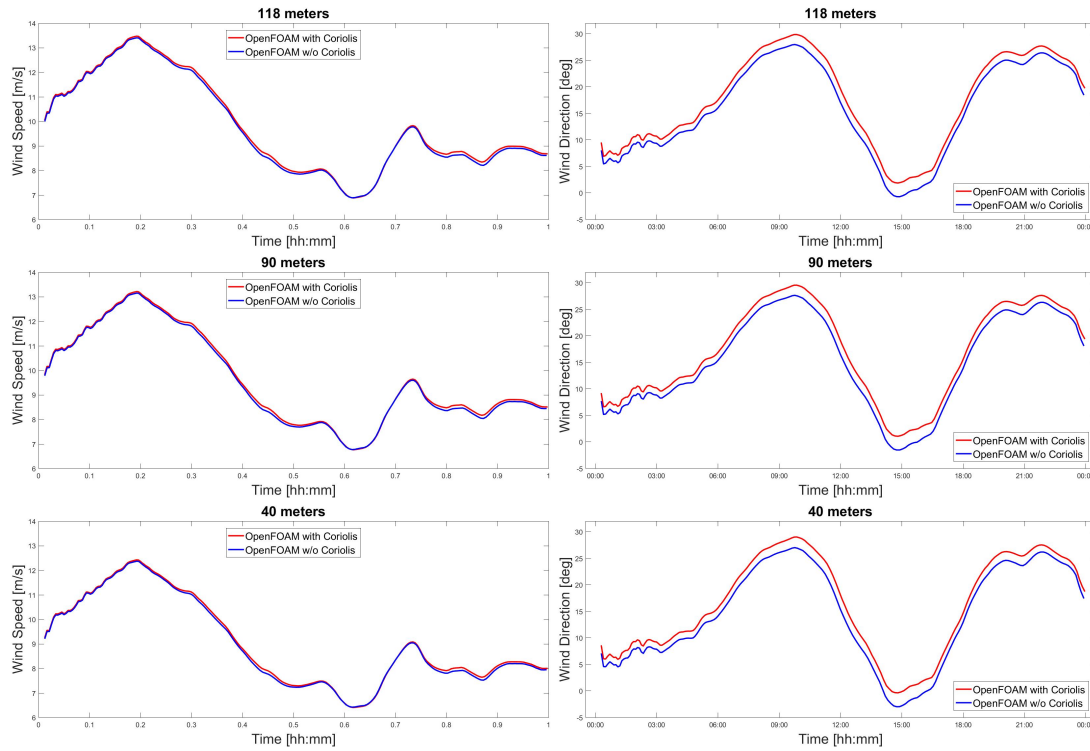


Figure 3.39: Wind speed and direction time-series at the MP5 met-mast location 118m and 40m above the ground with and w/o the Coriolis acceleration

mast 118m, 90m and 40m above the ground are given in Figure 3.39. Likewise, the red and blue lines represents the *OpenFOAM* solutions with and without the Coriolis acceleration, respectively.

It is evident that the Coriolis acceleration doesn't change the wind speed significantly, as expected. On the contrary, there is a positive shift around 2° with slight increments at the 15th hour in the wind direction, as expected. The reason for the slight increments is that the flow direction at the 15th hour is close to 0° , magnifying the effect of Coriolis acceleration on direction.

In order to quantify the effects of the Coriolis acceleration, RMS of the wind direction differences between the *OpenFOAM* solutions with and without the Coriolis acceleration are calculated using Equation 3.10 at 40m, 78m, 90m, 102m and 118m above ground at the MP5 met-mast location. As for the percent change of velocity magnitude due to the Coriolis acceleration, the relative RMS of the speed difference is

calculated as in Equation 3.11.

$$RMS_{Direction} = \sum_{i=1}^n \sqrt{\frac{1}{n} \left[Dir_i^{wCoriolis} - Dir_i^{w/oCoriolis} \right]^2} \quad (3.10)$$

$$rRMSE_{speed} = \sum_{i=1}^n \sqrt{\frac{1}{n} \left[\frac{V_i^{wCoriolis} - V_i^{w/oCoriolis}}{V_i^{wCoriolis}} \right]^2} \quad (3.11)$$

Table 3.3: Effects of the Coriolis acceleration at the MP5 met-mast location

	40 m	78 m	90 m	102 m	118 m
RMS of Direction (°)	1.849	1.811	1.801	1.792	1.781
rRMS of Speed(%)	0.87	0.887	0.892	0.897	0.902

It can be seen from Table 3.3 that the average deflection due to Coriolis acceleration while slightly decreasing, doesn't vary significantly with altitude. Average deflection at these heights is found to be about 1.8°. As for velocity magnitude, the difference is minuscule around 0.9 % but increases slightly but not significantly with increasing altitude which is not surprising as the velocity increases also with altitude.

3.3.5.5 Effects of Surface Roughness

For the investigation of surface roughness effects, *OpenFOAM* solutions coupled with WRF are performed with and without the implementation of spatially varying roughness model for the topography. All results are with the Coriolis implementation in the *pimpleFOAM* solver.

The roughness height profiles for the topography of Alaiz region are extracted via the interface program *Interfoam.f* that reads the Corine roughness data set color codes and interpolates the corresponding roughness height values into the high resolution terrain's cell centres. The roughness length values for each color used Corine roughness map of Alaiz and their types of terrain are given in Table 3.4. Corine roughness map and the resulting spatially varying roughness map for *OpenFOAM* domain's topography are given in Figure 3.40. As mentioned in Methodology chapter,

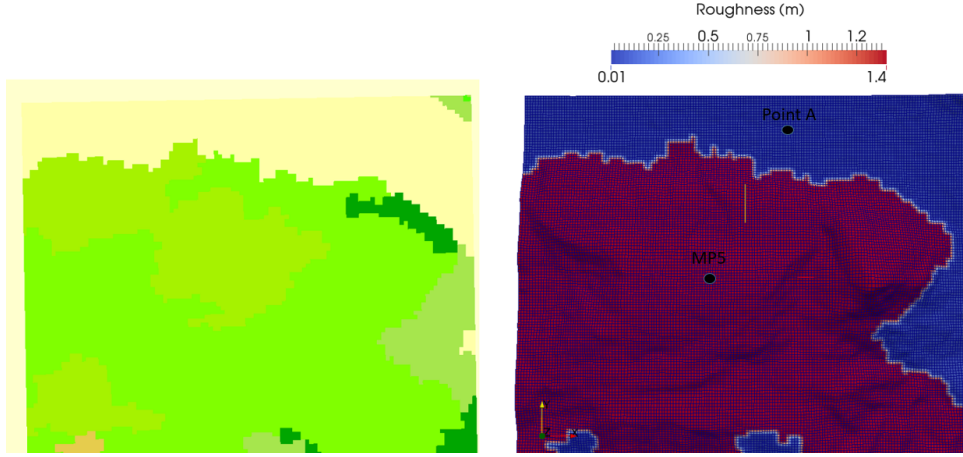


Figure 3.40: Corine roughness and roughness distribution in the Alaiz region

`nutkRoughWallFunction` is used with $k-\epsilon$ model to account for spatially varying roughness. 2 different points, one of which is the location of the MP5 met-mast in the region and the other one being Point A in Figure 3.40 with lower surface roughness is selected for the investigation of time series data from the *OpenFOAM* solutions.

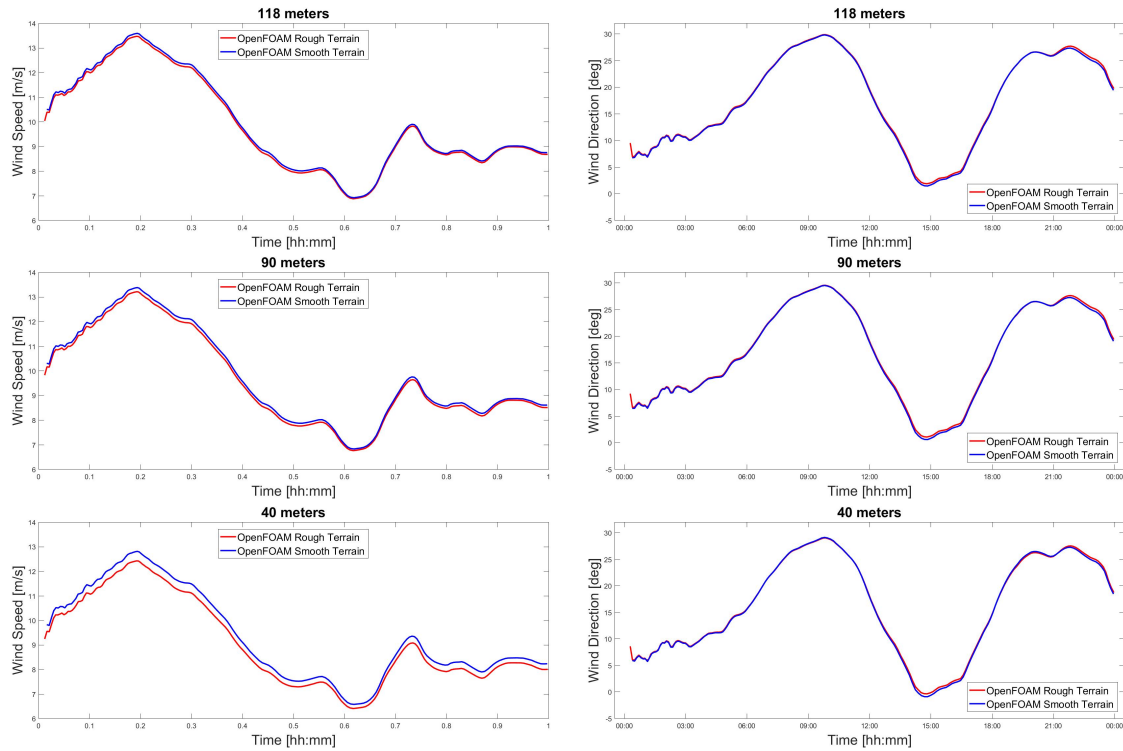


Figure 3.41: Wind speed and direction time-series 118m and 40m above the ground at the MP5 mast location with the smooth and the rough terrains

In Figure 3.41, the time series of wind speed and direction from the *OpenFOAM* so-

Table 3.4: Corine roughness classes used

Class	Color(RGB)	Roughness (m)	Type
2	(255, 0, 0)	0.6000	Discontinuous urban fabric
3	(204, 77, 242)	0.7000	Industrial or commercial units
7	(166, 0, 204)	0.1000	Mineral extraction sites
12	(255, 255, 168)	0.0300	Non-irrigated arable land
18	(230, 230, 77)	0.0300	Pastures
20	(255, 230, 77)	0.0100	Complex cultivation patterns
21	(230, 204, 77)	0.0100	Land principally occupied by agriculture
23	(128, 255, 0)	1.4000	Broad, leaved forest
24	(0, 166, 0)	1.4000	Coniferous forest
25	(77, 255, 0)	1.4000	Mixed forest
26	(204, 242, 77)	0.0500	Natural grasslands
28	(166, 230, 77)	0.0500	Sclerophyllous vegetation
29	(166, 242, 0)	1.4000	Transitional woodland, shrub

Table 3.5: Effects of surface roughness at the MP5 met-mast location

	40 m	78 m	90 m	102 m	118 m
rRMS Speed(%)	4.298	2.113	1.772	1.513	1.246
RMS Direction (°)	0.238	0.249	0.2476	0.244	0.237

lution 118m, 90m and 40m above the ground at the MP5 met-mast location which has a roughness height of 1.4m, with and without the surface roughness model are given with blue and red colors, respectively. It is clear that the presence of surface roughness retards the flow and slows down the wind speed significantly at 40m, yet its influence on wind speed fades away at 118m as the distance above the ground decreases, as expected. The wind direction seems unaffected at both distances above the ground as seen in Figure 3.41.

As in the effects of the Coriolis acceleration section, the RMS of wind direction and the rRMS of the wind speed differences are evaluated as in Equations 3.10, 3.11 , but this time with and without the surface roughness model for the terrain at the MP5 met-mast location at different heights to yield Table 3.5. It is seen from the Table 3.5,

Table 3.6: Effects of roughness at the locations with different surface roughness

	40 m	78 m	90 m	102 m	118 m
rRMS Speed at MP5 (%)	4.298	2.113	1.772	1.513	1.246
rRMS Speed at Point A (%)	3.632	2.183	1.883	1.624	1.347

the direction difference due to the roughness increases slightly with altitude but not significant ($\approx 0.24^\circ$) with respect to the deflection due to the Coriolis acceleration ($\approx 1.8^\circ$). The surface roughness primarily effects the wind speed especially near the ground as expected.

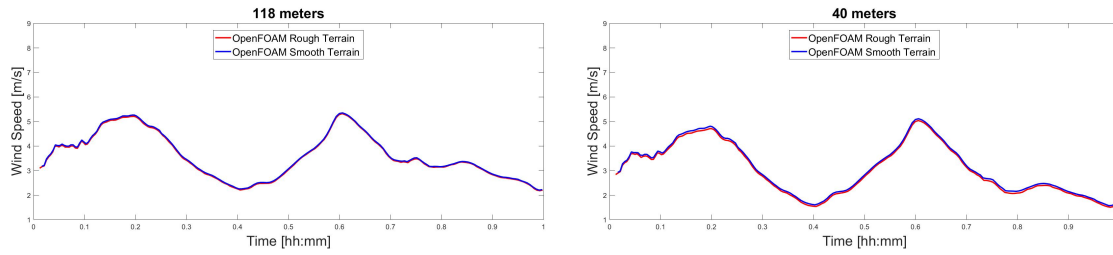


Figure 3.42: Wind speed and direction time-series 118m and 40m above the ground at Point A with the smooth and the rough terrains

For the investigation of different surface roughness in the spatially varying surface roughness model, wind speed time series are extracted are given in Figure 3.42 at Point A with a roughness height of $0.01m$. The trends for wind speed at Point A are similar as for the MP5 met-mast location case, while wind speed being much lower than the MP5 met-mast location. The effect of roughness 40m above the ground at Point A seems less significant than at the MP5 met-mast location, but then again the wind speed is also significantly lower at Point A. So, rRMS values of the wind speed differences due to roughness at the MP5 mast location and Point A are calculated in a similar way with Equation 3.11 and compared in Table 3.6.

$$rRMSE_{speed} = \sum_{i=1}^n \sqrt{\frac{1}{n} \left[\frac{V_i^{Smooth} - V_i^{Rough}}{V_i^{Rough}} \right]^2} \quad (3.12)$$

While the wind speed change is lower at Point A with respect to speed at the MP5 met-mast, percentages of differences due to surface roughness is similar at all distances, except at 40m above the ground which is surprising considering the roughness

height values at these points are vastly different. $40m$ above ground, rRMS of wind speed difference is around 4.3% for the MP5 met-mast location and 3.6% at Point A. The reduction in wind speed difference is thought to be because of the roughness difference. The difference sharply decreases at $78m$ to approximately 2.1% for both locations. Starting from $78m$ and going up, the percentage difference of wind speed is slightly more at Point A then at the MP5 met-mast, which is thought to be due to the wind speed being significantly lower at Point A.

3.3.5.6 Comparison with WRF and Observation Data

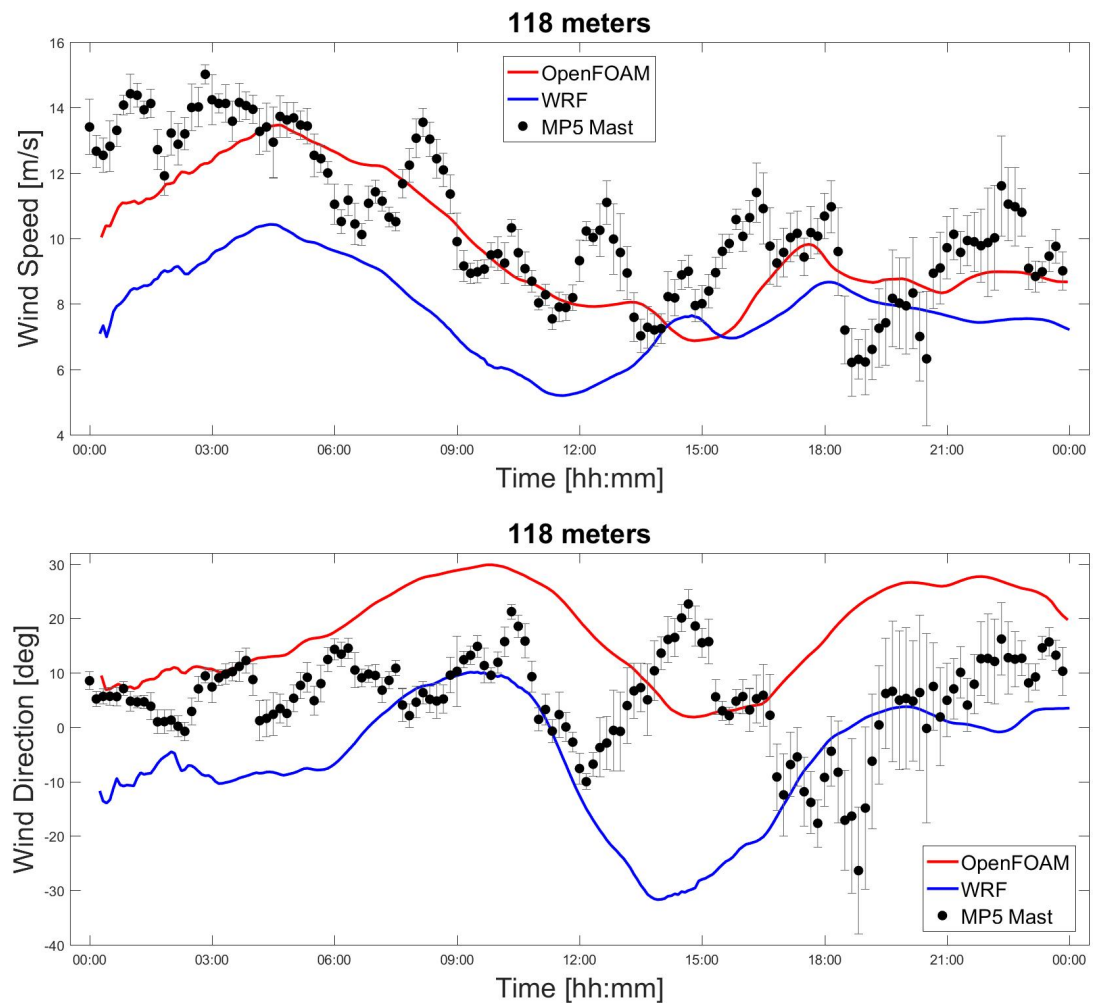


Figure 3.43: Wind speed and direction time-series at $118m$ above the ground

Time series of wind speed and direction from the MP5 met-mast are compared with the predictions of the *OpenFOAM* solution coupled with WRF and WRF alone at $118m$,

102m, 90m and 78m above the ground in Figures 3.43, 3.44, 3.45 and 3.46 respectively. Since the 40m data from the met-mast do not contain the wind direction, only the wind speed is compared at 40m in Figure 3.47. The red lines represent the *OpenFOAM* solutions coupled with WRF with spatially varying surface roughness model and Coriolis implementation into *OpenFOAM*. The WRF results which are taken at altitudes based on the low resolution WRF models terrain level with no stretching/shrinking are shown with blue lines. Observation data from the MP5 met-mast are given with black dots with error bars which have extensions upto one standard deviation provided in the time series data set.

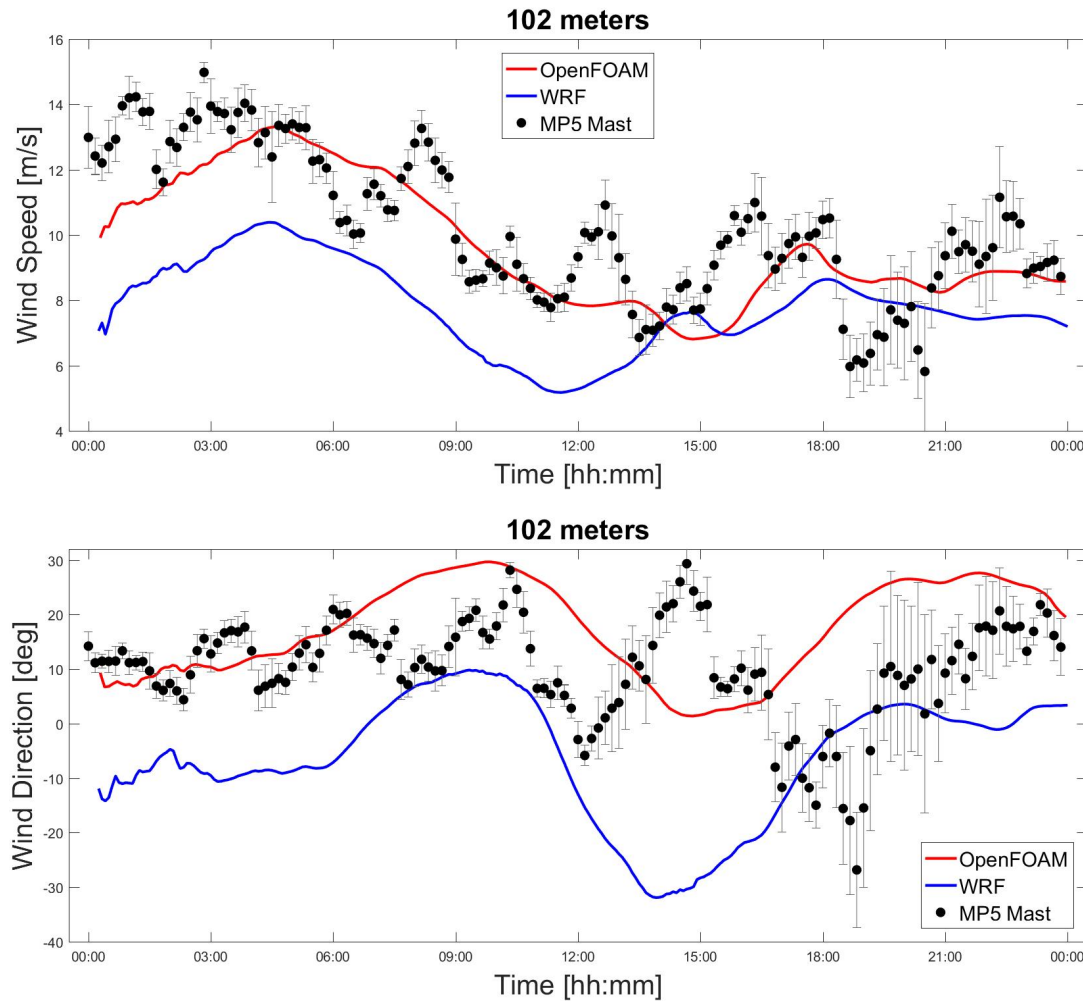


Figure 3.44: Wind speed and direction time-series at 102m above the ground

Looking at the met-mast data on Figures 3.43, 3.44, 3.45, 3.46 and 3.47; there is a general increase in the wind speed as the distance from the ground increases, as expected. Wind speed and direction trends are more or less similar as the distance

from the ground increases. When the variation of velocity magnitude for the met-mast data is investigated; highly oscillatory behaviour is seen between 7th and 18th hour of the simulation. Also there are sharp changes in direction between 10th-12th, 12th-15th and 15th-19th hours, forming an M shape at all distances from the ground. The reason for this phenomenon is thought to be the atmospheric stability as the sun rising and setting for the date 01.01.2015 should be around 7th and 18th hour. As the radiation from the Sun hits earth's surface, it is absorbed by the ground. This results in the heating of the atmosphere, more intense near the ground. As the density of air near the ground decreases, air moves up and a vertical highly turbulent flow is generated by buoyancy. Inversely, at night as the air with more density is near the ground level, no significant vertical flow is generated and atmosphere becomes stable. Surprisingly, the standard deviation of both velocity magnitude and direction are observed between 16th and 24th hour when the atmosphere is stable.

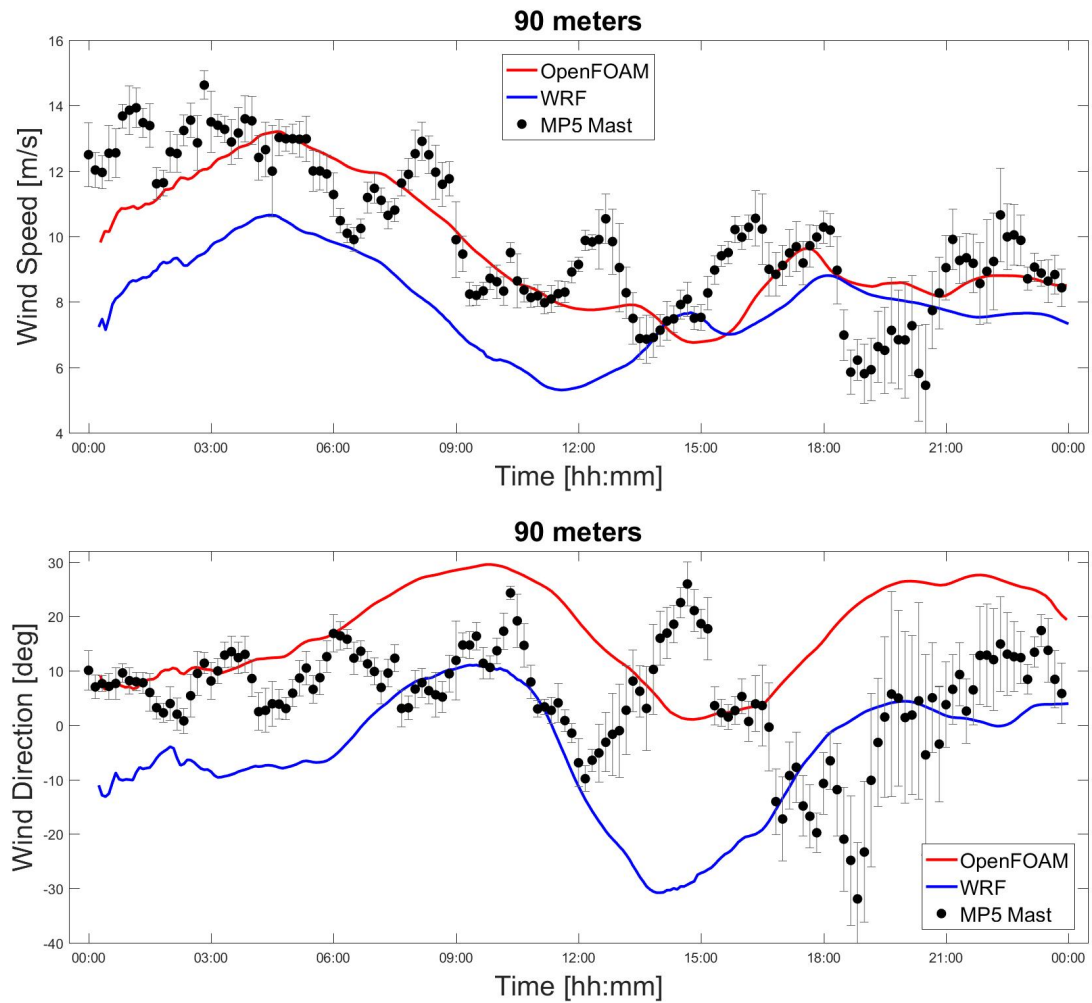


Figure 3.45: Wind speed and direction time-series at 90m above the ground

When WRF predictions are considered, it can be seen that while the variation of wind speed follows a similar trend with respect to the observation data, the WRF results mostly underestimates the wind velocity at all distances above the ground which is consistent with the studies by Jimenez [25], as the low resolution terrain model in WRF skips high slope features such as in the case for Alaiz. The least accurate results with respect to the MP5 met-mast data are observed at 40m. This results are not surprising as the viscous effects are not fully taken into account because of the η coordinate assumption in WRF and also the low resolution of WRF (0.9km). The least accurate results are observed between the start and 14th hours at all distances above the ground. As far as wind direction is concerned, significant error between 12th and 18th hours is observed. Also, the wind direction trends in these hours are opposite of what is predicted by WRF .

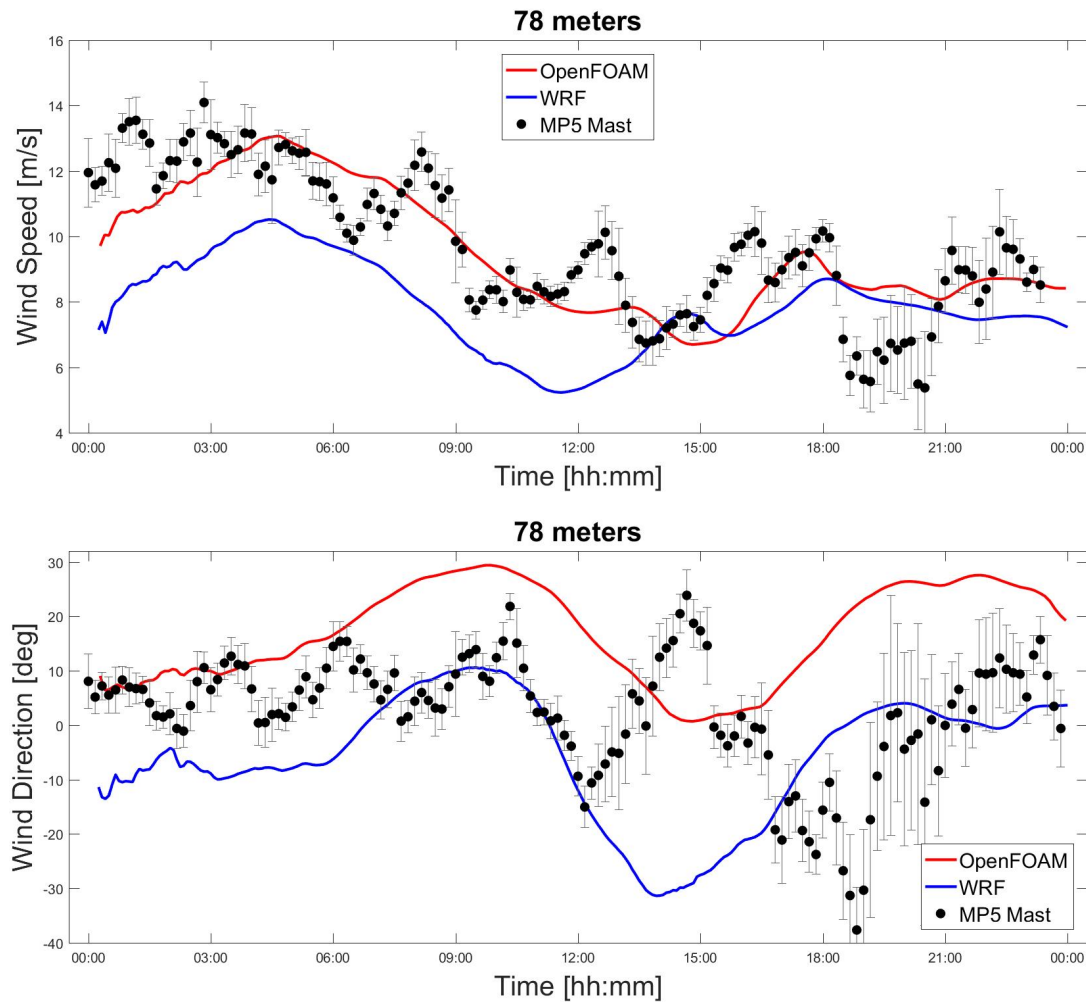


Figure 3.46: Wind speed and direction time-series at 78m above the ground

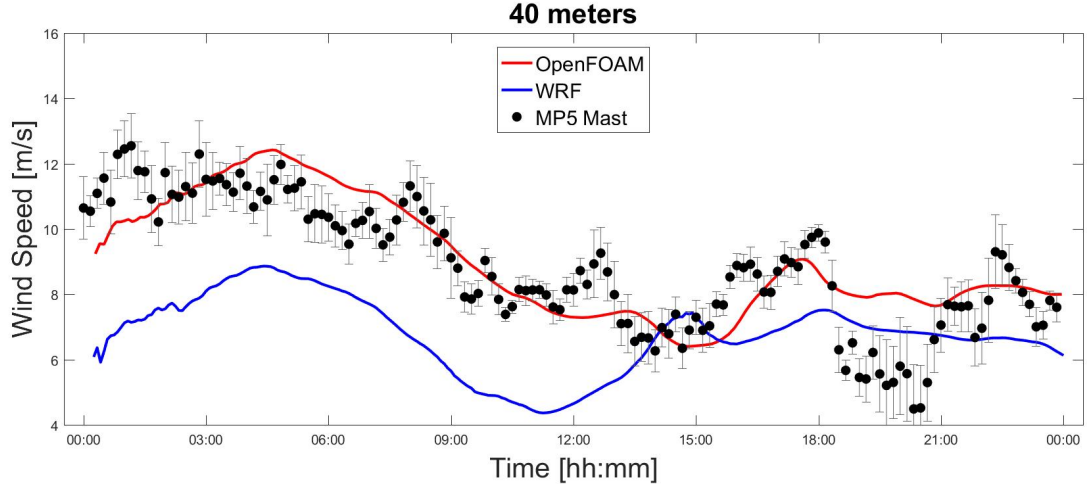


Figure 3.47: Wind speed time-series at 40m above the ground

When the *OpenFOAM* predictions are examined, it can be seen that using *OpenFOAM* with the boundary conditions from WRF greatly improves the velocity magnitude predictions at all distances above the ground and nearly at all times. It is observed that most accurate results for both direction and velocity magnitude with respect to the experimental data are observed during the night-time. The difference between the *OpenFOAM* solutions and the met-mast data increases when a highly oscillatory behaviour in the velocity magnitude from the met-mast data is seen (7th-18th hour). This drop in accuracy in day-time is currently expected as atmospheric stability effects due to heating and buoyancy is neglected in the *OpenFOAM* solutions. Wind direction changes between the 12th and 18th hours are also not captured by the coupled solutions.

In order to quantify the improvement with respect to the WRF predictions, the relative root mean Square errors (rRMSE) of the wind speed from the *OpenFOAM* solutions are obtained using the met-mast data at 10 minute intervals using Equation 3.13.

$$rRMSE_{speed} = \sum_{i=1}^n \sqrt{\frac{1}{n} \left[\frac{V_i^{Mast} - V_i^{OpenFOAM}}{V_i^{Mast}} \right]^2} \quad (3.13)$$

There are 2 important points to note in the evaluation of Equation 3.13. The first one is that any missing data from the met-mast is disregarded. The second one is that the *OpenFOAM* solution is interpolated in time since the time step in the *OpenFOAM*

solution changes with a fixed Courant number employed in the solution.

Table 3.7: rRMSE of the wind speed predicted by WRF and the coupled OpenFOAM solutions with respect to the met-mast data

	40 m	78 m	90 m	102 m	118 m
WRF (%)	29.238	25.378	25.843	27.707	28.337
OpenFOAM (%)	17.827	16.12	16.036	15.608	15.265
Improvement (%)	11.411	9.258	9.807	12.099	13.072

It is shown in Table 3.7, that the methodology developed improved the wind speed results with respect to the met-mast data at all distances above the ground, quantitatively. It is seen that the time dependent rRMS error of the *OpenFOAM* solutions with respect to the mast data decreases with increasing altitude and is around 16%, while for WRF results there is a non-linear trend with an average value of 27%. Surprisingly, accuracy of the WRF results decreases as the altitude increases after 78m above the ground. Improvement over the WRF results also increases after 78m which is primarily due to the same reason. 11.4% improvement is achieved at 40m above the ground using 0.9km resolution WRF data (with 29.38% rRMS error at the same height) as boundary conditions. Average improvement is found to be 11.13% which roughly translates into 37% improvement in estimated energy generation, since the energy is proportional to the cube of wind speed.

Since relative error of wind speed might be misleading as the flow direction becomes 0° at instances, RMS errors for the wind direction are calculated using Equation 3.14 using the same methodology for the wind speed and are given in Table 3.8.

$$RMSE_{Direction} = \sum_{i=1}^n \sqrt{\frac{1}{n} \left[D_i^{Mast} - D_i^{OpenFOAM} \right]^2} \quad (3.14)$$

It is seen from Table 3.8 that the wind direction from WRF are slightly improved with the coupled *OpenFOAM* solutions. Wind direction RMS errors of coupled *OpenFOAM* solution decreases with increasing altitude upto 118m above the ground then which it increases slightly. There is a sharp increase in improvement over the WRF results at 102m above the ground. Average improvement is 1.87°. The best WRF result for direction is obtained at 78m above the ground which is surprising as the resolution

Table 3.8: rRMSE of the wind direction predicted by WRF and the coupled OpenFOAM solutions with respect to the met-mast data

RMSE of Wind Direction (°)	78 m	90 m	102 m	118 m
WRF (°)	17.882	18.619	22.176	18.216
OpenFOAM (°)	20.193	17.634	14.652	16.915
Improvement (°)	-2.331	0.985	7.524	1.302

of the topographical model for WRF solutions is low. Furthermore, the WRF direction results at 78m are more accurate than that of coupled *OpenFOAM* solutions, even if the wind speed results shows a completely opposite situation.

3.3.5.7 Sensitivity Analysis on the WRF resolution

In order to assess the influence of the WRF resolution on the coupled methodology, coupled *OpenFOAM* solutions are performed using 2.7km and 0.9km resolution WRF solutions, and the predictions are compared at the MP5 met-mast location. The time series for the wind speed and the direction at the MP5 met-mast location for the date 01.01.2015 from 00:00 to 24:00 at 118m, 102m, 90m, 78m and 40m above the ground are used for the comparison, and are given in Figures 3.48, 3.49, 3.50, 3.51 and 3.52, respectively.

In Figures 3.48, 3.49, 3.50, 3.51, 3.47; the solid blue lines represent the WRF predictions with 0.9km resolution whereas the dashed blue lines represents the 2.7km WRF predictions. The solid red lines denote the *OpenFOAM* solutions coupled with the WRF solution on the 0.9km resolution grid, while the dashed red lines denote the *OpenFOAM* solutions coupled with the WRF solution on the 2.7km resolution grid.

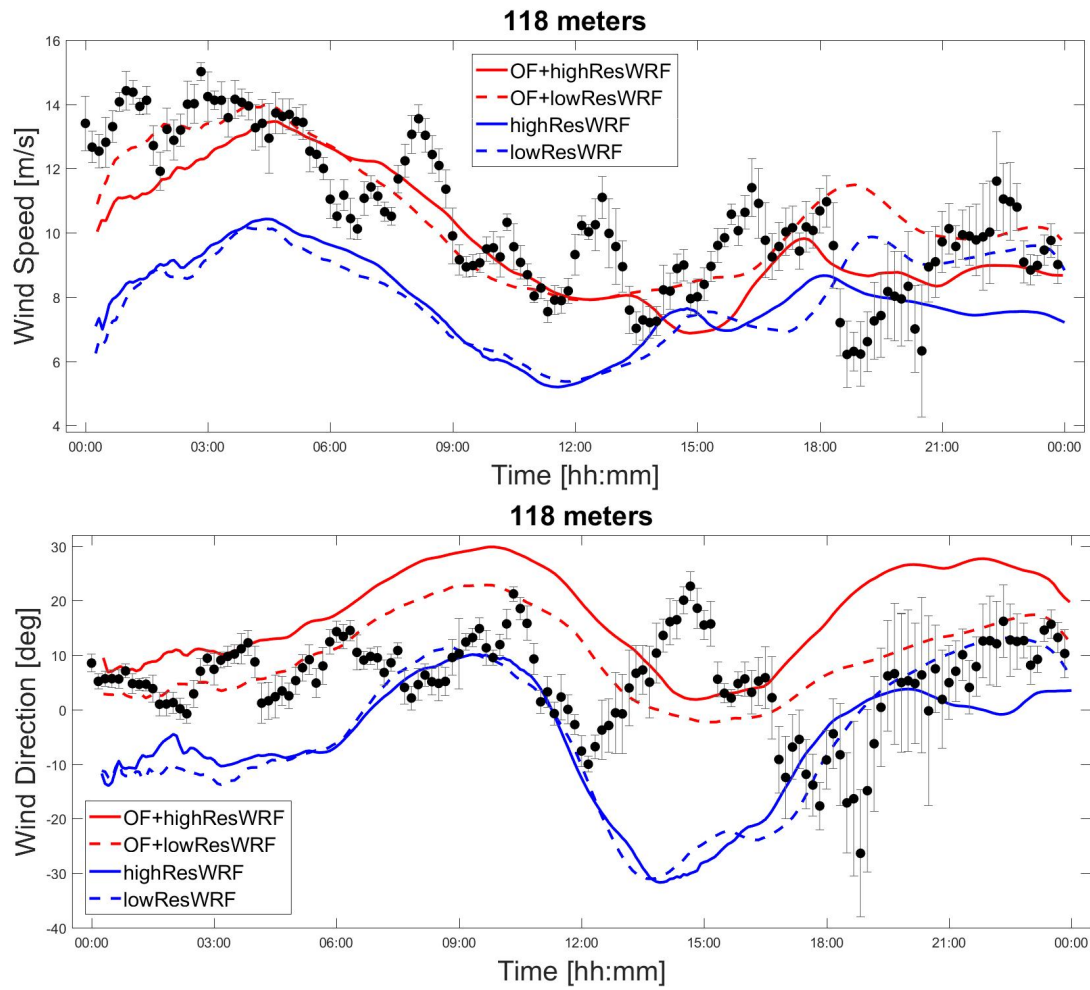


Figure 3.48: Wind speed and direction time-series 118m above the ground at the met-mast location

At the first glance, it is seen that WRF predictions improve significantly in both wind speed and direction as the grid resolution increases from $2.7km$ to $0.9km$. This is expected as complex terrain is modelled more accurately with $0.9km$ resolution even if the same data set is used for the topographical model in the WRF solutions.

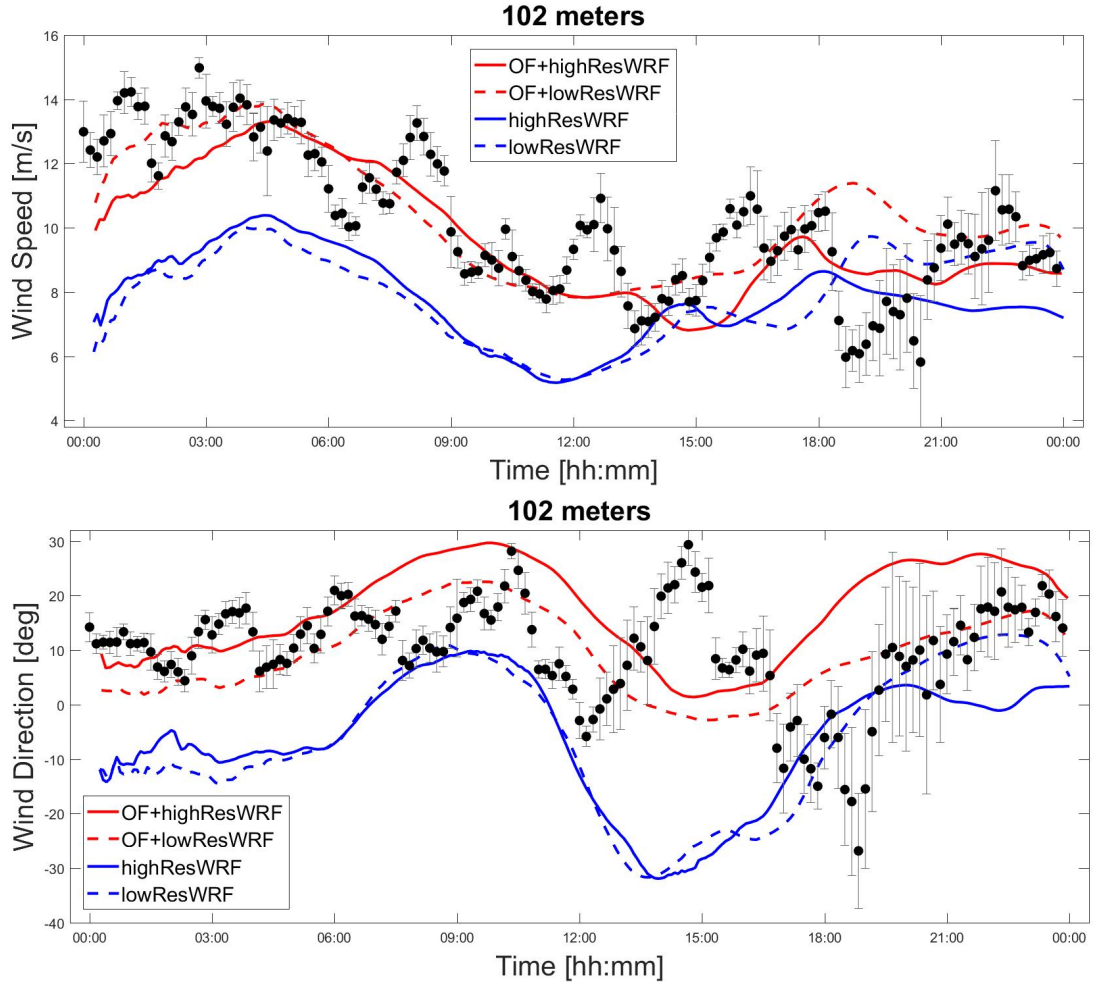


Figure 3.49: Wind speed and direction time-series 102m above the ground at the met-mast location

The same arguments about the time dependent *OpenFOAM* solutions coupled with WRF given in the previous sections also holds for the sensitivity analysis results given in Figures 3.48, 3.49, 3.50, 3.51 and 3.52. For wind speed, it is seen that taking the boundary conditions from either of 0.9km and 2.7km horizontal resolution WRF results, improved the corresponding coupled *OpenFOAM* solutions with respect to the observation data from the MP5 met-mast. The improvement for wind speed over corresponding pure WRF results seems to be higher, when high resolution WRF data with 0.9km horizontal resolution is used as boundary condition instead of the 2.7km horizontal resolution WRF results. For wind direction, usage of high horizontal resolution WRF instead of low resolution one, doesn't seem to improve the results which is unexpected.

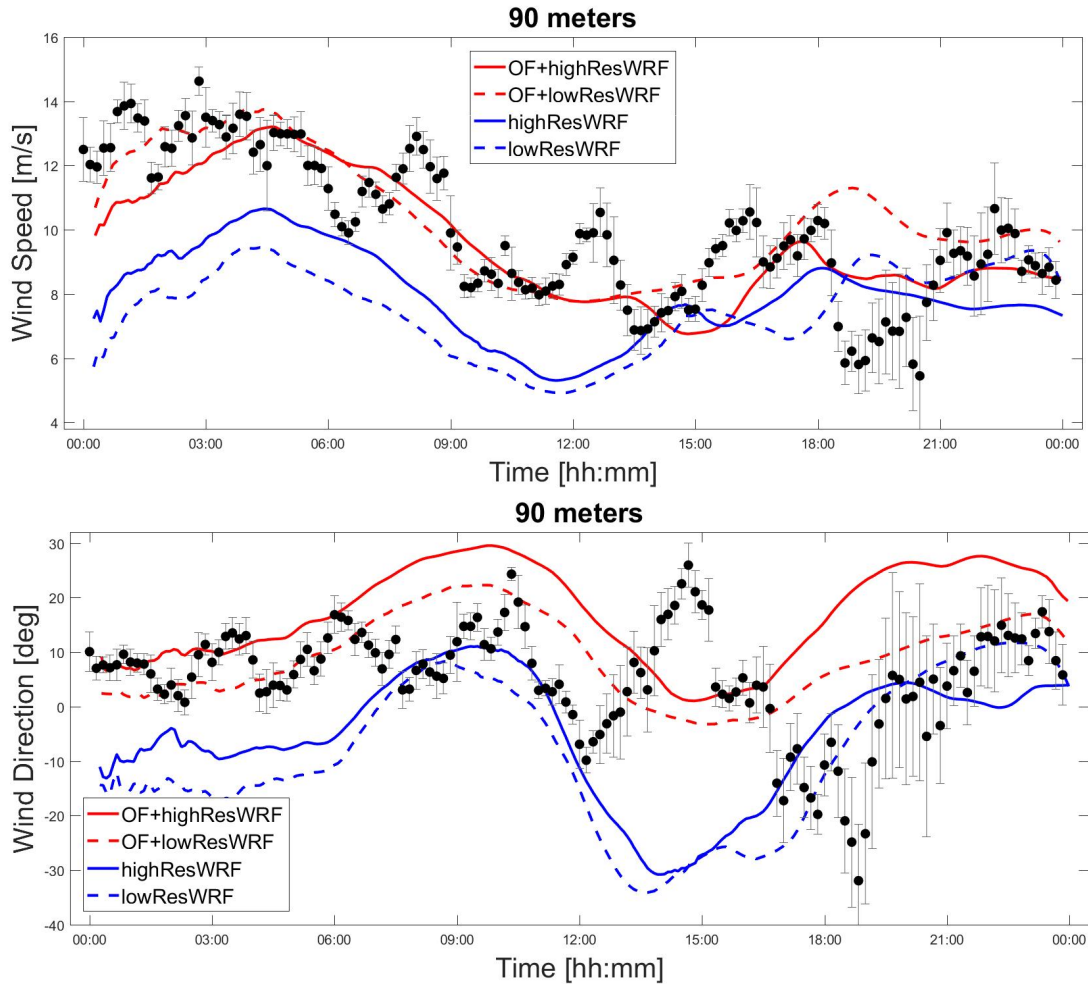


Figure 3.50: Wind speed and direction time-series 90m above the ground at the met-mast location

In order to quantify the improvements in the prediction of the wind speed and the direction, the rRMS errors for wind speed with respect to the MP5 met-mast data are given in Table 3.9 and RMS errors for wind direction are given in Table 3.10, respectively. WRF solutions with both 2.7km and 0.9km resolutions are considered.

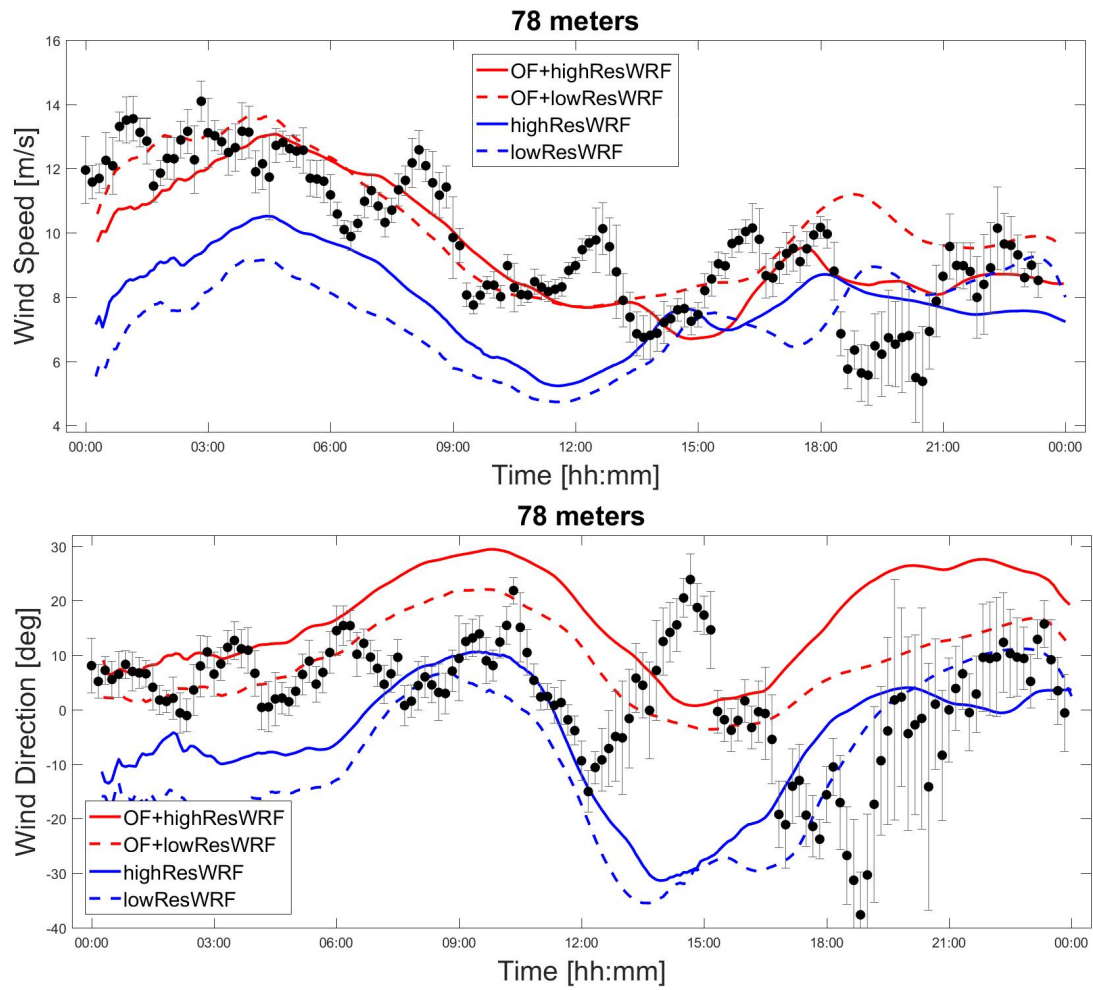


Figure 3.51: Wind speed and direction time-series 78m above the ground at the met-mast location

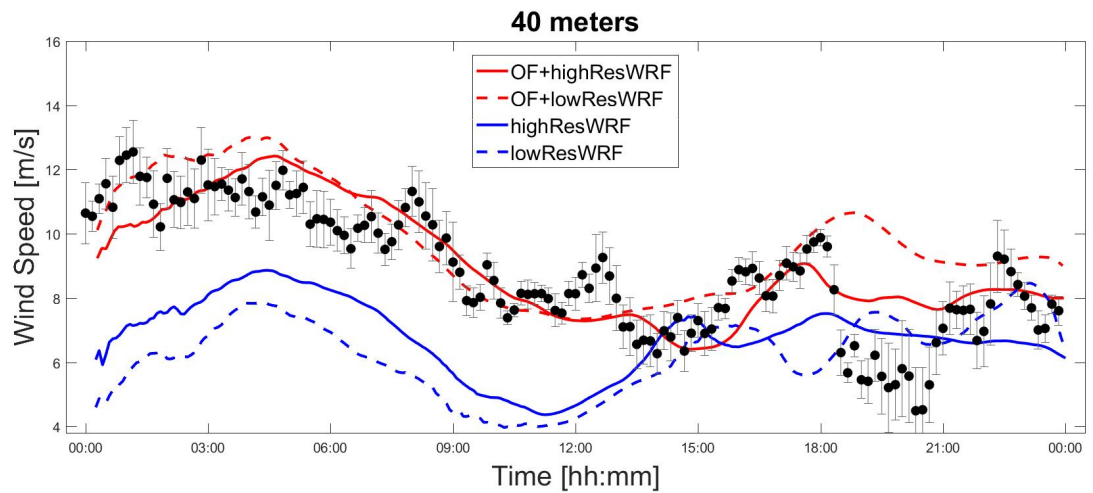


Figure 3.52: Wind speed time-series 40m above the ground at the met-mast location

Table 3.9: rRMSE of wind speed predictions with respect to the met-mast data

	40 m	78 m	90 m	102 m	118 m
highResWRF (%)	29.24	25.38	25.84	27.71	28.34
lowResWRF (%)	36.51	32.96	32.48	30.63	30.23
Improvement (%)	7.268	7.584	6.636	2.919	1.897
OF+highResWRF (%)	17.82	16.12	16.03	15.61	15.26
OF+lowResWRF (%)	29.29	25.07	23.89	22.03	20.42
Improvement (%)	11.47	8.95	7.86	6.42	5.16

Table 3.10: RMSE of wind direction predictions with respect to the met-mast data

	78 m	90 m	102 m	118 m
highResWRF (°)	17.882	18.619	22.176	18.216
lowResWRF (°)	21.069	21.351	22.218	18.139
Improvement (°)	3.187	2.732	0.042	-0.077
OF+highResWRF (°)	20.193	17.634	14.652	16.915
OF+lowResWRF (°)	13.609	11.843	10.897	10.938
Improvement (°)	-6.584	-5.891	-3.755	-5.977

It is observed that as the horizontal resolution in WRF solution increases from $2.7km$ to $0.9km$, the rRMSE of the wind speed predicted by WRF decreases at all altitudes, especially near the ground by about 7% at 40m above.(Table 3.9). The rRMSE of the wind speed predicted by the *OpenFOAM* solution coupled with the WRF solution shows the same trend. As the resolution of the WRF solution increases, the coupled *OpenFOAM* prediction improves more than 11% at 40m above the ground.

On the other hand, although the wind direction prediction also improves slightly in the WRF predictions, it unexpectedly deteriorates about 5° in the *OpenFOAM* solutions coupled with the high resolution WRF solution (Table 3.10). Further studies are needed to resolve this unexpected behaviour. It should be noted that both wind speed and direction predictions of WRF are generally improved by coupled *OpenFOAM* solution, for both high and low resolution WRF inputs.

Since the prediction of the velocity field in the coupled solution improves significantly when the maximum resolution is employed in the WRF solutions, it is concluded

that *OpenFOAM* solutions should, in general, be coupled with the high resolution WRF solutions.

3.3.6 Daily Wind Power Production Estimation

A post-processing tool that can extract the wind fields from the coupled *OpenFOAM* solutions at a given distance from the ground and estimates the wind power density (WPD) and the wind power generation is developed in order to employ for the micro-siting of wind turbines and for the wind turbine selection.

- Wind power density (WPD) is defined as the maximum wind power that can be extracted within a unit cross-sectional area (Equation 3.15). It is mainly used to select the turbine model for a given location.

$$\frac{P}{A} = \frac{1}{2} \rho v^3 \quad (3.15)$$

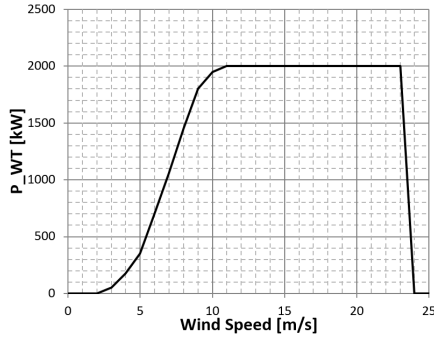


Figure 3.53: Power curve of a generic wind turbine

- Wind power production can be evaluated by using the power curve of a wind turbine and the wind speed at the turbine hub height. An example of a generic power curve is given in Figure 3.53. Wind turbine power curves are in general defined by 3 wind speeds and a rated power; the cut-in speed, the rated output speed and the cut-out speed.

The cut-in speed is the wind speed where the turbine starts to generate power. The rated output speed is the wind speed when the power output reaches the limit that the electric generator is capable of. Basically, increasing the wind speed further gives no benefit beyond this point. In between these speeds the power increases proportional to the cube of the wind speed. The cut-out speed is the speed when the turbine is stopped to avoid risks of structural damage to

the turbine. In Figure 3.53 the cut-in, rated output and cut-out wind speeds are $2m/s$, $11m/s$ and $24m/s$, respectively.

The post processing tool consists of 2 parts:

- a *ParaView* macro to extract the time dependent *OpenFOAM* solution on a copy of the terrain surface at a given height above the ground,
- Fortran routines to read the data on the extracted surface and to integrate the wind speed properly in order to obtain the wind power density distribution or the wind power production estimation according to the wind turbine power curve provided.

The *ParaView* macro generates a copy of the terrain at a given height above the ground using the grid data and the *Transform* filter. The time dependent flow variables are then extracted from *OpenFOAM* solution using *Resample with Data Set* filter. Once the extraction is done, time dependent data is written in *.csv* format. As an example, the time dependent data at $40m$ above the ground is given in Figure 3.54.

The Fortran routines first read the *.csv* files containing the day-long flowfield data at a specific altitude such as shown in Figure 3.54. The average wind power density is then calculated via integration of $u(t)^3$ using Equation 3.16 and given for the date 01.01.2015 in Figures 3.55, 3.56 and 3.57 at $40m$, $90m$ and $118m$ above the ground, respectively.

$$AverageWPD = \frac{1}{86400sec} \int_{t=0}^{t=24h=86400sec} \frac{1}{2} \rho u(t)^3 dt \quad (3.16)$$

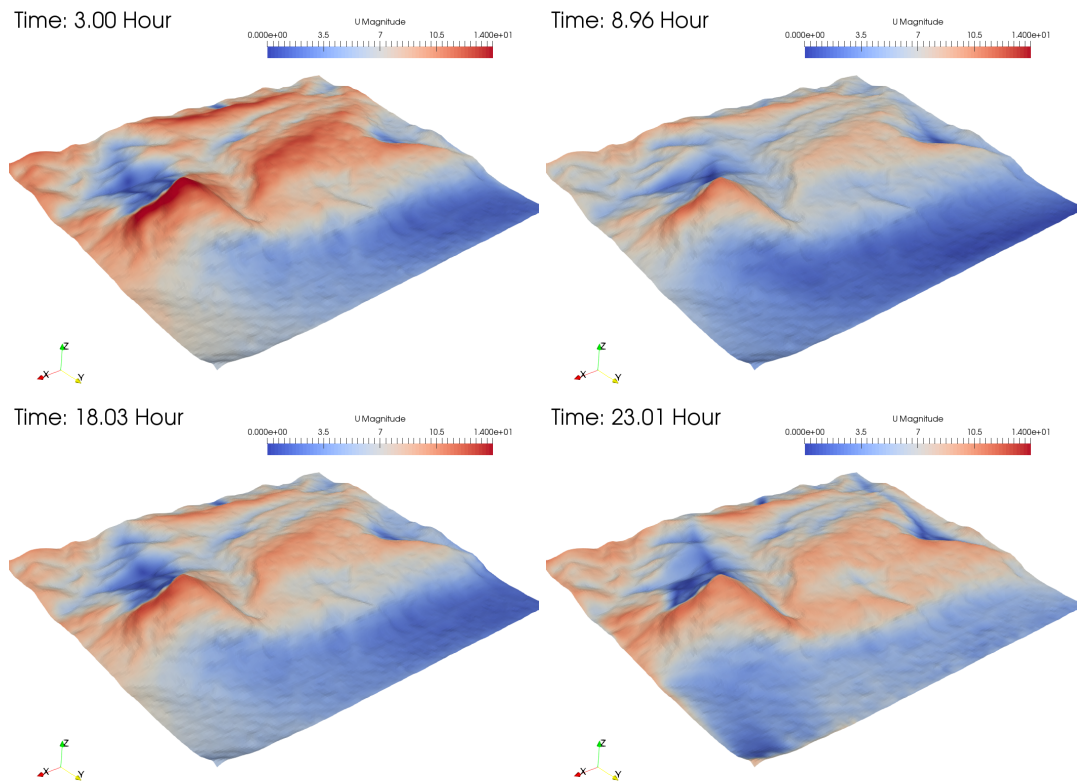


Figure 3.54: Unsteady OpenFOAM solutions coupled with WRF at 40m above the ground

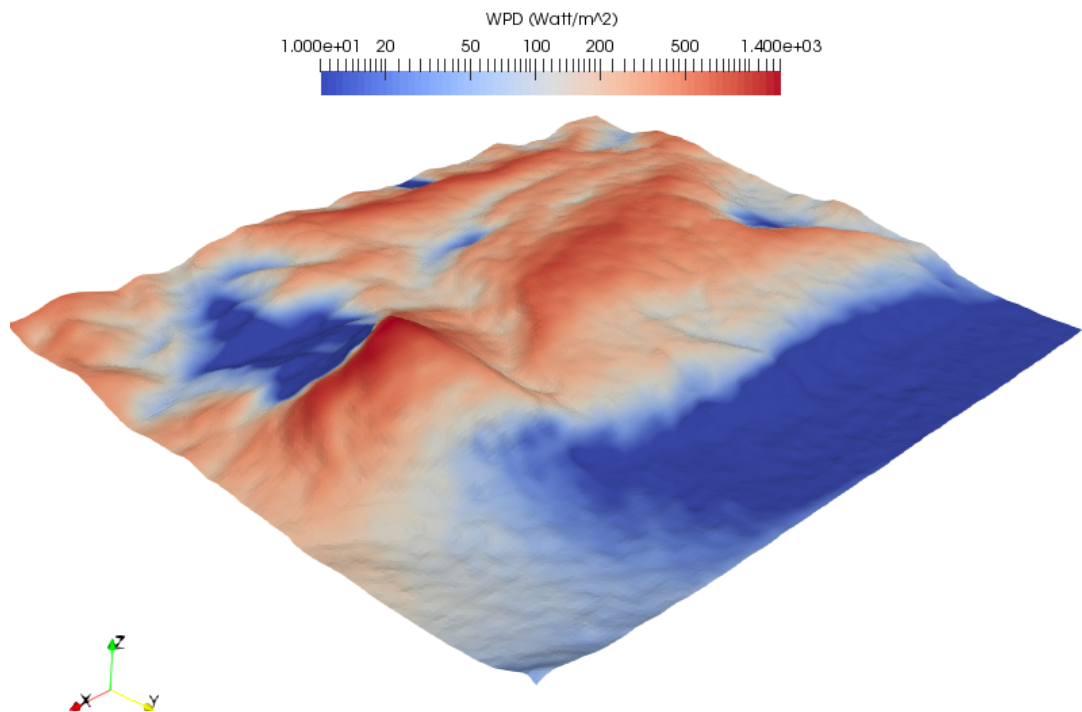


Figure 3.55: Wind power density at 40m above the ground

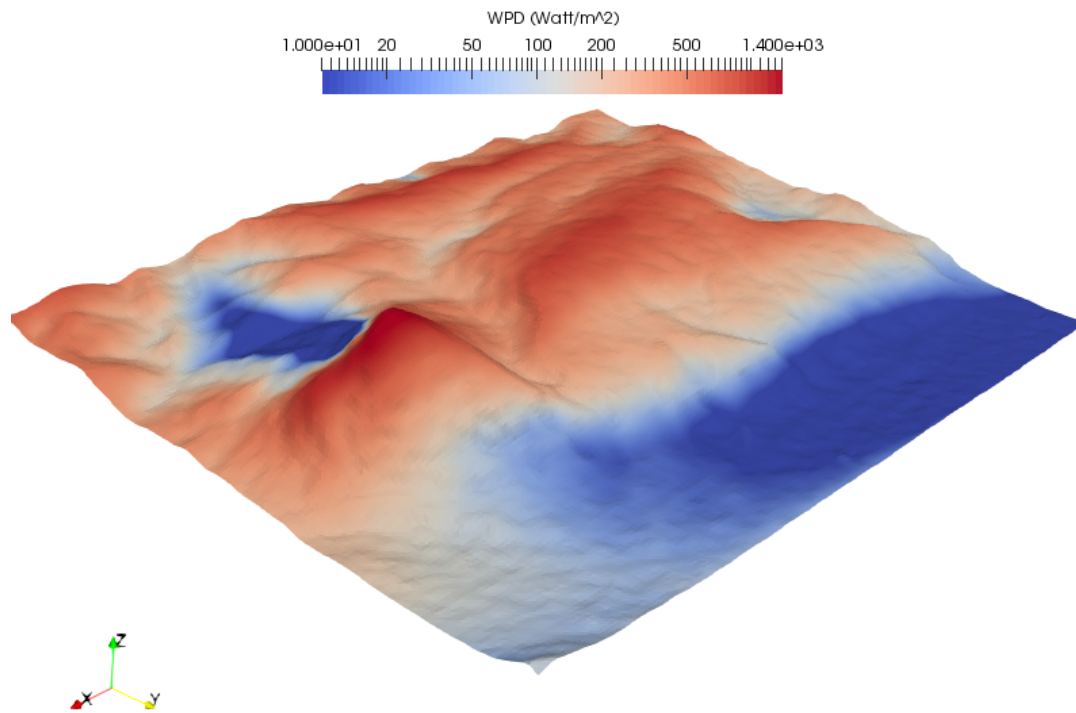


Figure 3.56: Wind power density at 90m above the ground

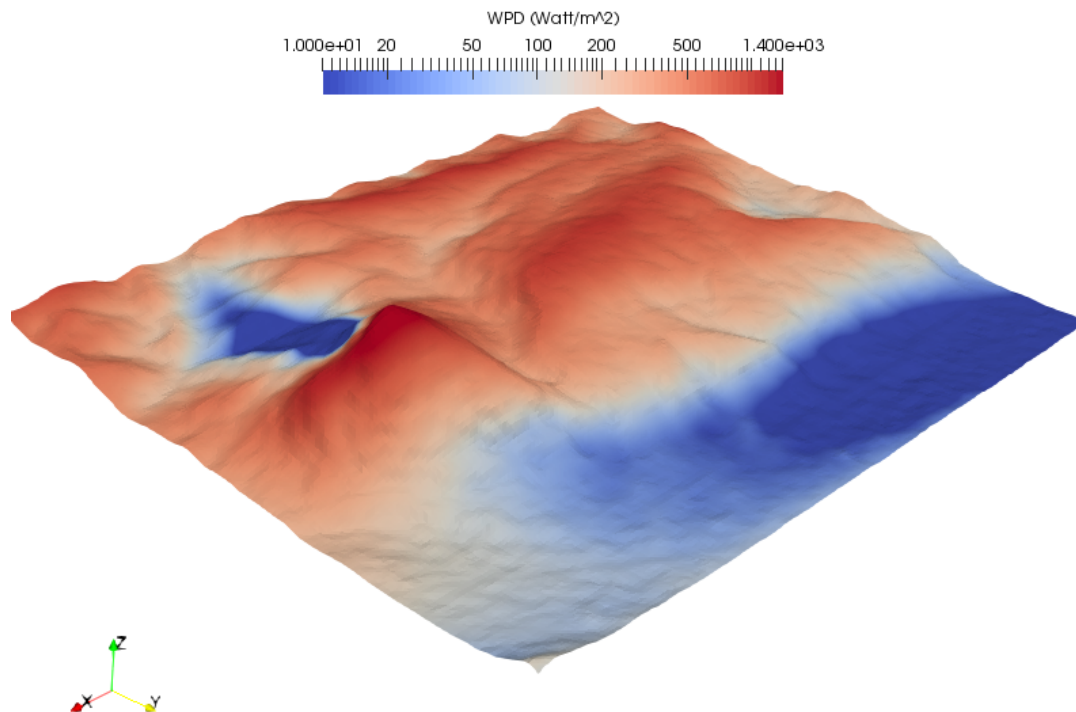


Figure 3.57: Wind power density at 118m above the ground

The trapezoid rule is used in the time integration. Thus the average wind power density for a time step is calculated by averaging the time adjacent average wind power densities, $\frac{1}{2}\rho u(t)^3$ and $\frac{1}{2}\rho u(t + \Delta t)^3$. It is important to note that the simulation times are not uniformly spaced as the time stepping is based on the Courant number.

It is important to note that the legend of the WPD distributions is logarithmic since the wind energy potential is proportional to the cube of wind speed. It is seen that the wake region behind the mountain becomes narrower as the distance above the ground increases, as expected. The best locations for the maximum wind power generation are observed to be at the mountain top, the ledge at the south-west part (where the wind turbines are located) and the valley to the west of the mountain. More variations in the WPD distributions are observed at 40m above the ground with respect to WPD distributions at 90m and 118m above the ground.

The daily wind power production E_{WT} is similarly estimated by integrating the power output of a wind turbine based on its power curve in time, $P_{WT}(u, t)$, as given in Equation 3.17:

$$E_{WT} = \int_{t=0}^{t=24h} P_{WT}(u(t), t) dt \quad (3.17)$$

The daily wind power production estimation for the date 01.01.2015 with the generic wind turbine power curve in Figure 3.53, is given in Figures 3.58, 3.60 and 3.61 at 40m, 90m and 118m above the ground, respectively. It can be seen from Figures 3.58, 3.60 and 3.61 that the power production increases significantly as the wind speed increases with the distance above the ground. One interesting thing to note is that at 118m and 90m levels the generic turbine generates the maximum power it can generate, which is around 48000 KwH , in the most of the southern parts of the domain. Whereas at 40m more variation is seen and the maximum power is only produced at the south-west ledge and at the mountain top. The ledge at the south-west part of the solution domain's topography where the daily wind power production estimation shows a sharp increase at both 40m and 118m above ground, is where the wind farms are mainly located in Alaiz, as shown in Figure 3.59. It is also seen that there are regions in the North-West corner where no power is produced due to the wind speed being smaller than the cut-in speed. The area where no power is produced,

decreases with the height above the ground.

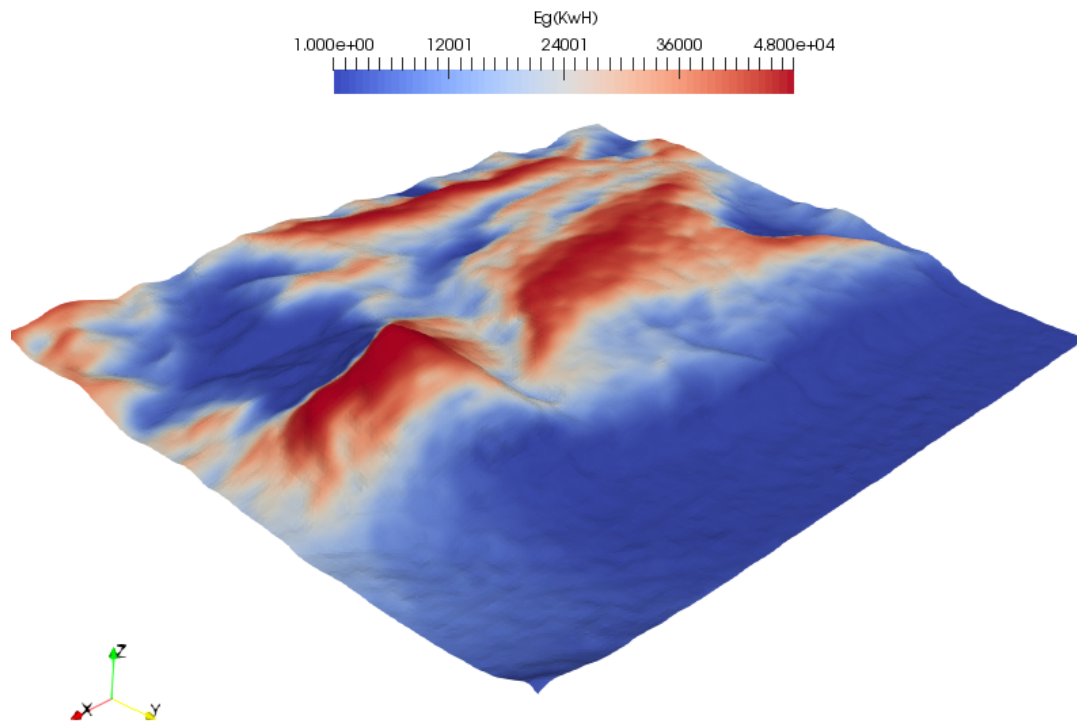


Figure 3.58: Daily wind power production estimation at 40m above the ground



Figure 3.59: Wind turbines in the Alaiz region

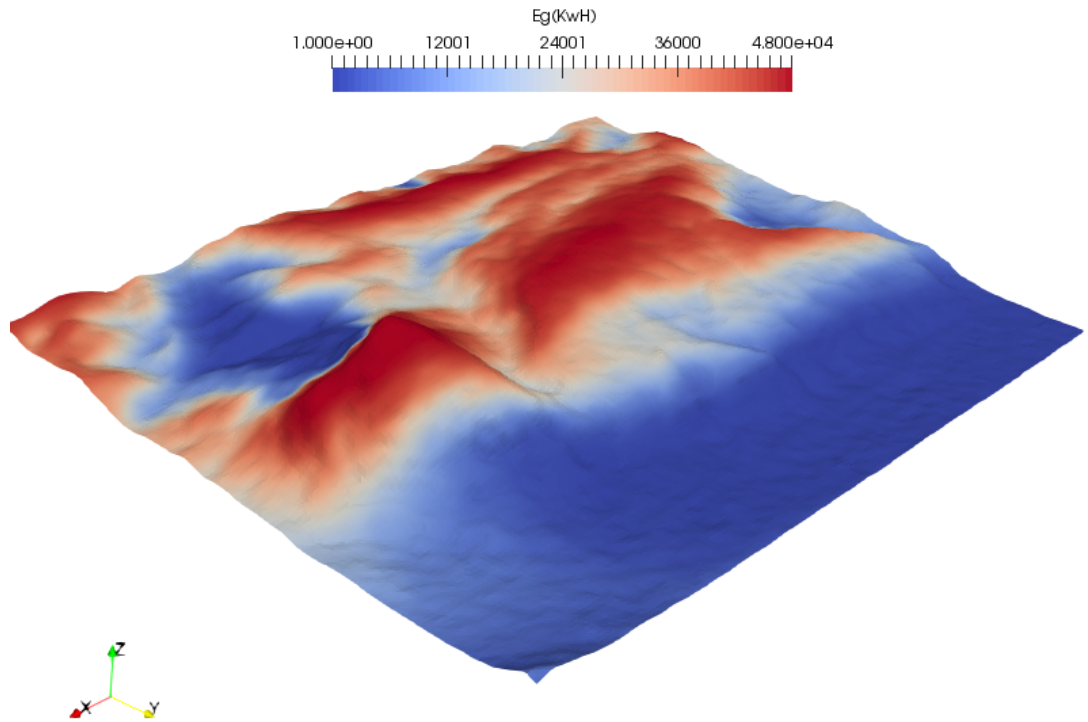


Figure 3.60: Daily wind power production estimation at 90m above the ground

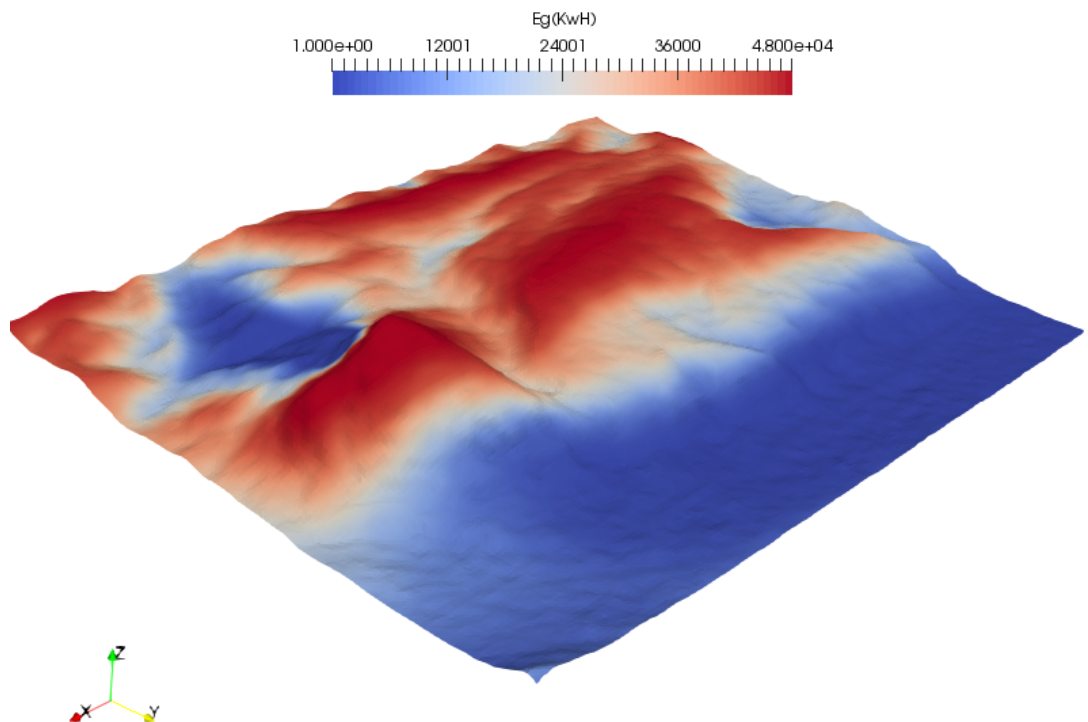


Figure 3.61: Daily wind power production estimation at 118m above the ground

CHAPTER 4

CONCLUSION

In this work, atmospheric boundary layer flows are computed with *OpenFOAM* coupled with the numerical weather prediction model, WRF, on a terrain fitted, high resolution grid. The WRF solution, which is based on a global model and accounts for all the atmospheric physics on a meso-scale coarse grid, provides the spatially varying unsteady boundary conditions for the *OpenFOAM* solver, *pimpleFOAM*. The coupling is achieved through the development of a new boundary condition class in *OpenFOAM*, *timeVaryingMixed*, which is implemented with a parallel solution algorithm. The Coriolis acceleration is successfully added into *OpenFOAM* solutions which are performed in a rotating reference frame. A spatially varying roughness model is also utilized in the $k - \epsilon$ turbulence model in order to account for the effect of surface canopy on the high resolution topography.

Two daily validation studies, one in Mut wind farm in Mersin, Turkey on the date 04.04.2010 and other for Alaiz wind field in Navarra providence, Spain on the date 01.01.2015 are performed. In the Mut case study, the new boundary condition class developed, *timeVaryingMixed*, is validated and its parallel implementation is assessed. In the implementation of *timeVaryingMixed*, all the boundary cells are collected in a single partition. The parallel performance of the current solution algorithm is therefore limited, and needs to be improved for more efficient computations.

In the Alaiz case study, a grid independence study for various vertical and horizontal resolutions is performed. In addition, the performance of various turbulence models is assessed. It is concluded that 25 vertical levels in the atmospheric boundary layer and a horizontal resolution of 38m for the computational grid with standard k-epsilon model that utilizes a spatially varying roughness model may successfully be used for

the simulations.

Unsteady *OpenFOAM* solutions coupled with WRF are investigated near the distinct regions of the complex topography in Alaiz, and it is shown that time dependent, complex terrain effects such as recirculation zones, separation and re-attachment can be captured and accounted for using the methodology developed which is not possible with the current wind energy prediction methodologies.

It is also observed in the comparison of velocity profiles at the met-mast location that the coupled solution captures the diurnal variation in the vertical velocity profile even though a neutral atmosphere is assumed in the *OpenFOAM* solution. As expected, it is attributed to the unsteady coupling with the WRF solution, which accounts for density variations and resolves the diurnal cycle.

The effects of Coriolis acceleration are investigated via taking the time series of wind speed and direction at 5 different heights above the ground on the MP5 Met-Mast location with and without the Coriolis acceleration implementation into *PimpleFOAM* solver. It is seen that Coriolis acceleration does not change the wind speed significantly, but mainly effects the wind direction by deflecting it 1.8° on average to the right of flow for the region. It is also seen that the amount of deflection does not vary significantly with increasing distance from the ground.

Similarly, effects of the spatially varying roughness model for the topography is investigated at the MP5 Met-Mast location and a location with a lower surface roughness value. It is seen that the presence of surface roughness retards the flow and slows down the wind speed 4.3% at 40 meters above the MP5 mast location, yet its influence on wind speed fades away at 118m as the distance above the ground decreases, as expected. The wind direction is not effected significantly by the surface roughness. Investigating a location with a much lower surface roughness, it is seen that the retardation of wind speed near the ground is reduced.

Time series of wind speed and direction obtained from the WRF coupled *OpenFOAM* solutions are then compared with pure WRF results and observation data from the MP5 Met-Mast location at 118m, 102m, 90m, 78m and 40m above the ground level. It is observed that most accurate of the coupled *OpenFOAM* results for both

direction and velocity magnitude with respect to the experimental data are observed during the night-time. Difference between the *OpenFOAM* solutions and met-mast data increases in day time. This drop in accuracy in day-time is thought to be due to the atmospheric stability effects due to heating and buoyancy which are neglected using the incompressible fluid approximation in *OpenFOAM* solutions. By taking the time dependent relative RMS errors of wind speed with respect to the MP5 Met-Mast data, it is shown that the methodology developed improved the wind speed results at all distances above ground level, especially near the ground. Average improvement is found to be 11.13%. Though slightly, wind direction estimations are also improved using the coupled solution methodology.

Sensitivity analysis on the WRF resolution are performed using $2.7km$ and $0.9km$ horizontal resolution WRF predictions. It is observed that as the horizontal resolution in WRF solution increases from $2.7km$ to $0.9km$, WRF wind speed predictions are improved, especially near the ground by about 7% at $40m$ above. The improvement in wind speed predictions is more significant for coupled *OpenFOAM* solutions with more than 11% at $40m$ above the ground. On the other hand, although the wind direction prediction also improves slightly by increasing the horizontal resolution in the WRF predictions, it unexpectedly deteriorates about 5° in the *OpenFOAM* solutions coupled with the high resolution WRF solution. Further studies are needed to resolve this unexpected behaviour. It should be noted that both wind speed and direction predictions of WRF are generally improved by coupled *OpenFOAM* solution, for both high and low resolution WRF inputs.

Lastly, daily wind energy and wind power density estimations are done via a post-processing tool that can extract the wind fields from the unsteady WRF coupled *OpenFOAM* solutions at a given distance from the ground. It is seen that the ledge at the south-west part of the solution domain's topography where the daily energy estimation shows a sharp increase at $40m$ above ground, is where the wind farms are mainly located in Alaiz.

The improvement on the wind speed estimations over WRF about 12% is achieved which roughly translates into 40% improvement in the power production using the novel coupled solution methodology developed. The inclusion of atmospheric stabil-

ity effects by using the Boussinesq approximation for density in `pimpleFOAM`, could further increase the accuracy of energy estimations. Also, other models for physical processes can be implemented with minimal effort and tested as the methodology is mostly independent of the flow solver used. The coupled solution methodology developed is shown to be especially useful for daily wind energy predictions where the unsteady variations in the wind speed are critical.

BIBLIOGRAPHY

- [1] Wrf source codes and graphics software download page. http://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog.html. (Accessed on 10/10/2017).
- [2] G. Ahmet. Micro-siting of wind turbines using navier-stokes solutions coupled with a numerical weather prediction model. *PhD thesis, Middle East Technical University*, 2014.
- [3] K. W. Ayotte. Computational modelling for wind energy assessment. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(10):1571–1590, 2008.
- [4] P. Beaucage, M. C. Brower, and J. Tensen. Evaluation of four numerical wind flow models for wind resource mapping. *Wind Energy*, 17(2):197–208, 2014.
- [5] A. Beljaars, J. Walmsley, and P. Taylor. A mixed spectral finite-difference model for neutrally stratified boundary-layer flow over roughness changes and topography. *Boundary-Layer Meteorology*, 38(3):273–303, 1987.
- [6] B. Blocken, T. Stathopoulos, and J. Carmeliet. Cfd simulation of the atmospheric boundary layer: wall function problems. *Atmospheric environment*, 41(2):238–252, 2007.
- [7] G. Botta, R. Castagna, M. Borghetti, and D. Mantegna. Wind analysis on complex terrain—the case of acqua spruzza. *Journal of wind engineering and industrial aerodynamics*, 39(1-3):357–366, 1992.
- [8] J. Boussinesq. Théorie de l’écoulement tourbillant. *Mem. Présentés par Divers Savants Acad. Sci. Inst. Fr*, 23(46-50):6–5, 1877.
- [9] Z. Boutanios, C. Miller, and H. Hangan. Computational analysis of the manitoba september 5 1996 storm: mesoscale wrf-arw simulations coupled with microscale openfoam cfd simulations. In *The 5th International Symposium on*

Computational Wind Engineering, Chapel Hill, North Carolina, USA (to be submitted), 2010.

- [10] A. J. Bowen and N. G. Mortensen. Exploring the limits of wasp the wind atlas analysis and application program. In *1996 European Wind Energy Conference and Exhibition*, pages 584–587, 1996.
- [11] A. J. Bowen and N. G. Mortensen. Wasp prediction errors due to site orography. Technical report, 2004.
- [12] A. J. Bowen and T. Saba. The evaluation of software for wind turbine siting in hilly terrain. In *Proc. 9th Int. Conf. on Wind Engineering, New Delhi, India*, 1995.
- [13] E. Cantero, F. Borbon, R. Chavez, and J. Sanz. Wp2: Planning of alaiz experiment, 2016.
- [14] S. Cao and T. Tamura. Effects of roughness blocks on atmospheric boundary layer flow over a two-dimensional low hill with/without sudden roughness change. *Journal of wind engineering and industrial aerodynamics*, 95(8):679–695, 2007.
- [15] T. Cebeci and P. Bradshaw. Momentum transfer in boundary layers. *Washington, DC, Hemisphere Publishing Corp.; New York, McGraw-Hill Book Co., 1977. 407 p.*, 1977.
- [16] M. J. Churchfield, G. Vijayakumar, J. G. Brasseur, and P. J. Moriarty. Wind energy-related atmospheric boundary layer large-eddy simulation using open-foam: Preprint. Technical report, National Renewable Energy Laboratory (NREL), Golden, CO., 2010.
- [17] B. C. Cochran and R. R. Damiani. Harvesting wind power from tall buildings. In *Wind Power*, 2008.
- [18] R. Damiani, B. Cochran, K. Orwig, and J. Peterka. Complex terrain: A valid wind option? *American Wind Energy Association*, 2008.
- [19] R. G. Derickson and J. A. Peterka. Development of a powerful hybrid tool for evaluating wind power in complex terrain: atmospheric numerical models

- and wind tunnels. In *Proceedings of the 23rd ASME Wind Energy Symposium*, page 15, 2004.
- [20] H. Gopalan, C. Gundling, K. Brown, B. Roget, J. Sitaraman, J. D. Mirocha, and W. O. Miller. A coupled mesoscale–microscale framework for wind resource estimation and farm aerodynamics. *Journal of Wind Engineering and Industrial Aerodynamics*, 132:13–26, 2014.
 - [21] C. J. Greenshields. Openfoam user guide. *OpenFOAM Foundation Ltd*, version, 3(1), 2015.
 - [22] S. Ivanell, J. N. Sørensen, and D. Henningson. Numerical computations of wind turbine wakes. In *Wind Energy*, pages 259–263. Springer, 2007.
 - [23] P. Jackson and J. Hunt. Turbulent wind flow over a low hill. *Quarterly Journal of the Royal Meteorological Society*, 101(430):929–955, 1975.
 - [24] H. Jasak. Openfoam: open source cfd in research and industry. *International Journal of Naval Architecture and Ocean Engineering*, 1(2):89–94, 2009.
 - [25] P. Jiménez, J. Dudhia, J. González-Rouco, J. Montávez, E. García-Bustamante, J. Navarro, J. Vilà-Guerau de Arellano, and A. Muñoz-Roldán. An evaluation of wrf’s ability to reproduce the surface wind over complex terrain based on typical circulation patterns. *Journal of Geophysical Research: Atmospheres*, 118(14):7651–7669, 2013.
 - [26] J. C. Kaimal and J. J. Finnigan. *Atmospheric boundary layer flows: their structure and measurement*. Oxford university press, 1994.
 - [27] H.-G. Kim and V. Patel. Test of turbulence models for wind flow over terrain with separation and recirculation. *Boundary-Layer Meteorology*, 94(1):5–21, 2000.
 - [28] P.-Å. Krogstad and P. E. Eriksen. “blind test” calculations of the performance and wake development for a model wind turbine. *Renewable energy*, 50:325–333, 2013.

- [29] L. Landberg, L. Myllerup, O. Rathmann, E. L. Petersen, B. H. Jørgensen, J. Badger, and N. G. Mortensen. Wind resource estimation—an overview. *Wind Energy*, 6(3):261–271, 2003.
- [30] L. Landberg and S. J. Watson. Short-term prediction of local wind conditions. *Boundary-Layer Meteorology*, 70(1):171–195, 1994.
- [31] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. *Computer methods in applied mechanics and engineering*, 3(2):269–289, 1974.
- [32] M. Marquis, J. Wilczak, M. Ahlstrom, J. Sharp, A. Stern, J. C. Smith, and S. Calvert. Forecasting the wind to reach significant penetration levels of wind energy. *Bulletin of the American Meteorological Society*, 92(9):1159–1171, 2011.
- [33] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, 32(8):1598–1605, 1994.
- [34] N. Mortensen, D. Heathfield, L. Myllerup, L. Landberg, and O. Rathmann. Getting started with wasp 9, risø-i-2571 (en). *Risø National Laboratory, Technical University of Denmark*, 2007.
- [35] D. Morton, O. Nudson, D. Bahls, G. Newby, A. Region, and S. Center. Pushing wrf to its computational limits. In *2010 Alaska Weather Symposium, Fairbanks, Alaska, March*, 2010.
- [36] R. D. Moser, J. Kim, and N. N. Mansour. Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$. *Physics of fluids*, 11(4):943–945, 1999.
- [37] E. S. Politis, J. Prospathopoulos, D. Cabezón, K. S. Hansen, P. Chaviaropoulos, and R. J. Barthelmie. Modeling wake effects in large wind farms in complex terrain: the problem, the methods and the issues. *Wind Energy*, 15(1):161–182, 2012.
- [38] S. Reid. Modelling of channelled winds. In *Proc. BWEA Conference, Warwick, UK*, pages 19–21, 1995.

- [39] O. Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Proceedings of the Royal Society of London*, 56(336-339):40–45, 1894.
- [40] A. Sempreviva, I. Troen, and A. Lavagnini. Modelling of wind power potential in sardinia. In *Proc. European Wind Energy Association Conference and Exhibition, Rome Italy*, pages 7–9, 1986.
- [41] W. C. Skamarock, J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, W. Wang, and J. G. Powers. A description of the advanced research wrf version 2. Technical report, National Center For Atmospheric Research Boulder Co Mesoscale and Microscale Meteorology Div, 2005.
- [42] J. Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3):99–164, 1963.
- [43] J. Sørensen, W. Shen, and X. Munduate. Analysis of wake states by a full-field actuator disc model. *Wind Energy*, 1(2):73–88, 1998.
- [44] T. Stovall, G. Pawlas, and P. Moriarty. Wind farm wake simulations in openfoam. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 825, 2010.
- [45] R. B. Stull. *An introduction to boundary layer meteorology*, volume 13. Springer Science & Business Media, 2012.
- [46] X. P. Tapia. Modelling of wind flow over complex terrain using openfoam. *Msc thesis, University of Gävle*, 2009.
- [47] P. Taylor, J. Walmsley, and J. Salmon. A simple model of neutrally stratified boundary-layer flow over real terrain incorporating wavenumber-dependent scaling. *Boundary-Layer Meteorology*, 26(2):169–189, 1983.
- [48] T. H. Terrain. Pollutant dispersion over two—dimensional hilly terrain.
- [49] I. Troen and E. L. Petersen. European wind atlas. *Roskilde: Riso National Laboratory*, 1989, 1989.

- [50] J. Walmsley, P. Taylor, and T. Keith. A simple model of neutrally stratified boundary-layer flow over complex terrain with surface roughness modulations (ms3djh/3r). *Boundary-Layer Meteorology*, 36(1):157–186, 1986.
- [51] C. Wang and S. Jin. Error features and their possible causes in simulated low-level winds by wrf at a wind farm. *Wind Energy*, 17(9):1315–1325, 2014.
- [52] W. Wang, D. Barker, J. Bray, C. Bruyere, M. Duda, J. Dudhia, D. Gill, and J. Michalakes. User’s guide for advanced research wrf (arw) modeling system version 3. *Mesoscale and Microscale Meteorology Division–National Center for Atmospheric Research (MMM-NCAR)*, 2007.
- [53] D. C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA journal*, 26(11):1299–1310, 1988.
- [54] F. J. Zajackowski, S. E. Haupt, and K. J. Schmehl. A preliminary study of assimilating numerical weather prediction data into computational fluid dynamics models for wind prediction. *Journal of Wind Engineering and Industrial Aerodynamics*, 99(4):320–329, 2011.
- [55] H. Zhang, Z. Pu, and X. Zhang. Examination of errors in near-surface temperature and wind from wrf numerical simulations in regions of complex terrain. *Weather and Forecasting*, 28(3):893–914, 2013.
- [56] X. Zhu and M. G. Genton. Short-term wind speed forecasting for power system operations. *International Statistical Review*, 80(1):2–23, 2012.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Leblebici, Engin

Nationality: Turkish (TC)

Date and Place of Birth: 01.03.1986, Antalya

Marital Status: Single

Phone: +90 554 7298399

EDUCATION

Degree	Institution	Year of Graduation
M.S.	Aerospace Engineering Department, METU	2012
B.S.	Aerospace Engineering Department, METU	2008
High School	Ataturk Anatolian High School	2004

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2008 - 2018	Aerospace Engineering Department, METU	Graduate Research Assistant
2015 - Present	ERA-NET Project No: 215M385	Researcher
2016 - 2016	TAI DKTM Project	Researcher
2012 - 2015	TUBITAK Project No : 112M104	Researcher

PUBLICATIONS

International Conference Publications

- E. Leblebici and I.H. Tuncer, *Downscaling Methodology Based on Coupled Unsteady WRF and OpenFOAM Solutions*, Wind Energy Science Conference 2017, WESC 2017, 2017, Lyngby, Denmark.
- E. Leblebici and I.H. Tuncer, *OpenFOAM Solutions Coupled with WRF for Wind Power Estimation*, European Congress on Computational Methods in Applied Sciences and Engineering, AIAA Aviation 2017, 2017, Denver, Colorado, USA.
- E. Leblebici and I.H. Tuncer, *Development of Parallel OpenFOAM - WRF Coupling Methodology for Wind Power Production Estimations*, The Fifth Symposium on OpenFOAM in Wind Energy, SOWE 2017, 2017, Pamplona, Spain.
- E. Leblebici and I.H. Tuncer, *Coupled Unsteady OpenFOAM and WRF Solutions for an Accurate Estimation of Wind Energy Potential*, European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2016, 2016, Crete Island, Greece.
- E. Leblebici and I.H. Tuncer *Development of OpenFOAM - WRF Coupling Methodology for Wind Power Production Estimations*, , The Fourth Symposium on OpenFOAM in Wind Energy, SOWE 2016, 2016 Delft, the Netherlands.
- E. Leblebici and I.H. Tuncer, *Wind Power Estimations using OpenFOAM Coupled with WRF*, EAWE 11th PhD Seminar on Wind Energy in Europe, 2015. Stuttgart, Germany.
- E. Leblebici, G. Ahmet and I.H. Tuncer, *Micro Scale Atmospheric Flow Solutions Coupled with a Mesoscale Weather Prediction Model*, 27th International Conference on Parallel Computational Fluid Dynamics, ParCFD2015, 2015. Montreal, Quebec, Canada.
- G. Ahmet, E. Leblebici, T. Sevine and I.H. Tuncer, *CFD Coupled with WRF for Wind Power Prediction*, EAWE 10th PhD Seminar on Wind Energy in Europe, 2014. Orléans, France.

- E. Leblebici, G. Ahmet and I.H. Tuncer, *Unsteady Atmospheric Turbulent Flow Solutions Coupled With A Mesoscale Weather Prediction Model*, RUZGEM'2013 Conference on Wind Energy Science and Technology, Ankara, Turkey, Oct. 3-4, 2013
- G. Ahmet, E. Leblebici and I.H. Tuncer, *Terrain Fitted Turbulent Flow Solutions Coupled with a Mesoscale Weather Prediction Model*, EAWC 9th PhD Seminar on Wind Energy in Europe, Visby, Sweden, Sept 18-20, 2013
- E. Leblebici, G. Ahmet and I.H. Tuncer, *Wind Potential Estimations Based on Unsteady Turbulent Flow Solutions Coupled with a Mesoscale Weather Prediction Model*, 7th Ankara International Aerospace Conference, Ankara, Turkey, Sept. 11-13, 2013
- E. Leblebici, G. Ahmet and I.H. Tuncer, *Atmospheric turbulent flow solutions coupled with a Mesoscale weather prediction model*, III. South-East European Conference on Computational Mechanics, SEECCM'2013 (Special Interest Conferences of the ECCOMAS and IACM), Kos Island, Greece, June 12-14, 2013
- E. Leblebici, G. Ahmet, and I.H. Tuncer, *Atmospheric Turbulent Flow Solutions Coupled with a Mesoscale Weather Prediction Model*, EAWC 4th Scientific Conference; The Science of Making Torque from Wind, Oldenburg, Germany, Oct.9-11 , 2012
- E. Leblebici, G. Ahmet, and I.H. Tuncer, *Terrain Modeling and Atmospheric Turbulent Flow Solutions Based on Meteorological Weather Forecast Data*, 6th Ankara International Aerospace Conference, Ankara, Turkey, Sep. 14-16, 2011