Doctoral Thesis ETH Zürich No. 21837

Efficient Simulation of Wind and Wake Flows in Wind Farms Using a Preconditioned Multistage Solver

A dissertation submitted to the
ETH Zürich

for the degree of
Doctor of Sciences

presented by

Samira Jafari

M.Sc. AE.-Eng., Sharif University of Technology, Iran
born on 01.09.1984 in Zanjan
citizen of Iran

accepted on the recommendation of

Prof. Dr. Reza S. Abhari, examiner
Dr. Ndaona Chokani, co-examiner
Prof. Dr. Jakob Mann, co-examiner

2014

© 2014 Samira Jafari
All Rights Reserved

تقدیم به مادرم

Acknowledgment

I would like to thank Prof. Reza Abhari for granting me this opportunity to carry out my PhD research in the Laboratory for Energy Conversion and pushing me towards best of my abilities during all these years.

I would like to express my sincere gratitude to Prof. Jakob Mann for accepting the role of co-examiner and his valuable comments and guidance which improved the quality of my thesis.

I am very grateful to Dr. Ndaona Chokani for the helpful discussions during my PhD. His invaluable comments and guidance regarding my presentations and papers thought me how to present myself and my technical work effectively.

I would like to express my gratitude to Prof. Kazem Hejranfar from Sharif University of Technology and Prof. Patrick Jenny from ETH Zurich for their time and the invaluable and helpful discussions on development and implementation of the preconditioning technique in the solver.

I would like to thank Altug Basol for useful discussions and valuable suggestions during the development of the MULTI3. His constant help and support made the long periods of algorithm and code development quite pleasant time. I am also deeply grateful to Christian Fuchs and Bob Mischo for their constant support. They were always available to discuss all the new and old aspects of the code. I have used many pieces of algorithms which were developed by them.

I would like to express my gratitude to all LEC members for their kind help and and support and friendship during these years. Especially the members of wind group for trying to support my research via experimental studies and also for their help during the organization of the PhD seminar in summer 2012. I would like to thank Kerstin Fischer for her great help and friendship. Her humor made the difficult days of the seminar quite memorable. I would like to thank Marlene Hegner for her kind help and support on administration work during these years.

I would like to thank Luna Bozinova and Nadia Gambino for their genuine friendship and support during good and hard times. They generously shared their time with me on countless coffee breaks and lunches and after word drinks which elevated my spirit and motivation. I would also like to thank Armin Zemp, Philipp Jenny, Antriksh Singh, Kai Regina and Benoit Laveau for making the working hours in the office, on lecture notes and the conference time a pleasant memory.

I am deeply grateful to my family whose love knows no borders. I would like to thank my dearest Mahmoud for his constant emotional and technical support during these years. I have benefited a great deal from his knowledge in programming. He was the one always there for me when I desperately needed fresh set of eyes to look at my codes or on the brain storming sessions on the model development. I genuinely believe I could not do this without him.

Abstract

The simultaneous simulation of microscale wind and wind turbine wake is attempted in this work using a preconditioned multistage solver and an immersed wind turbine model. This work demonstrates a low Mach number preconditioning formulation for high Reynolds number, low Mach number atmospheric flows. The preconditioning is implemented together with multigrid approach into a multistage solver in order to provide an efficient scheme, which allows for the routine use of computational fluid dynamics in simulations of the atmospheric flow and wakes within wind farms.

Microscale wind simulations are performed using the preconditioned solver over several test cases including Askervein Hill, Kettles Hill and Bolund Hill to demonstrate the superior convergence, accuracy and robustness of the method. In addition, the performance of RANS solver is assessed in prediction of microscale wind variations over topography with varying complexity.

Next, A novel immersed wind turbine model is developed and used to simulate the evolution of single and multiple wake interactions. The model is formulated in order to reduce the stringent grid requirements for resolving the blade boundary layer and near-wake region behind the wind turbines. The model is first evaluated by comparing the predicted evolution of the velocity and turbulence intensity in the far wake with measurements performed in a wind tunnel. Predictions are also compared with full-scale measurements of a single wake at the Sexbierum wind farm. The performance of the model is also assessed in predicting wake interaction in the same wind farm by comparing the power performance with measurements in an operating wind farm. The agreement between the model results and measurement for all cases is satisfactory for both single and double wake predictions. In next step, the simulations are performed over offshore wind farm Lillgrund and sensitivity of power loss to wind direction is investigated in two first rows of the farm. The results of simultaneous wind and wake flow are also presented over two wind farms located in complex terrain in Spain and Switzerland. The recovery rate of wake and turbulence characteristics in wind farm caused by the topography are assessed. Overall the results of array loss in wind farm located in complex terrain demonstrate the sensitivity of wind farm performance in complex terrain to small variations in wind direction. The model yield acceptable results for all test cases, however, to justify the additional computational cost of RANS simulations compared to simplified engineering models, the accuracy needs to be further improved and the computational cost must be reduced. Overall, the results over various test cases demonstrates the capability of the model to resolve the wake interaction in wind farms and to estimate the array loss for different arrangements of turbines and incoming wind directions. In addition to reducing the computational time, the model employs Cartesian grids, which facilitate the grid generation process for wind farm simulations.

In the final part of the thesis, an immersed boundary method that is used in connection with the Reynolds-Averaged Navier-Stokes equations with $k-\omega$ turbulence model in order to efficiently simulate the wind flow over complex terrain is presented. With the immersed boundary method, only one Cartesian grid is required to simulate the wind flow for all wind directions, with only the rotation of the digital elevation map. Thus the lengthy procedure of generating multiple grids for conventional rectangular domain is avoided. Wall functions are employed with the immersed boundary method in order to relax the stringent near-wall grid resolution requirements, as well as to allow the effects of surface roughness to be accounted for. The immersed boundary method is applied to moderately complex terrain test case Askervein Hill and the highly complex terrain test case of Bolund Hill. The simulation results of wind speed and turbulent kinetic energy show good agreement with experiments for heights greater than 5m above ground level.

Zusammenfassung

Die Mikromassstab Simulation von Wind simultan mit der Nachlaufströmung einer Windturbine wird in dieser Arbeit mittels einem vor konditionierten mehrstufigen Solver und einem immersiertem Windturbinen Modell behandelt. Diese Arbeit zeigt die vor konditionierte Formulierung der niedrigen Mach Zahl für hohe Reynolds-Zahlen. Die vor konditionierte Formulierung wurde zusammen mit einem Mehrgitterverfahren in einem mehrstufigen Solver implementiert. Dies erlaubt die routinierte Verwendung der numerischen Strömungsmechanik für die Simulation der atmosphärischen Strömung und der Nachlaufströmung einer Windturbine. Mikromassstab Windsimulationen wurden durchgeführt mit dem vor konditionierten Solver über mehrer Testfälle wie Askervein Hill, Kettles Hill and Bolund Hill. Damit wurde die Konvergenz, Genauigkeit und Robustheit dieser Methode zeigt. Zusätzlich wurde die Leistung der RANS Solver bewertet mit der vorhersage der wind veränderungen über untersichtliche Gelände.

Weiterhin wurde ein neuartiges immersiertes Windturbinen Modell entwickelt. Dieser wurde für Simulationen der Wechselwirkungen vom die Nachlaufströmung von einzelne und mehrere Windturbinen eingesetzt. Das Modell wurde so formuliert, dass die notwendige Gitter Auflösung erzielt wird, um die Rotorblätter Grenzschicht und die Nachlaufströmung in der Nähe der Windturbine auflösen zu können. Die Evolution des Modells erfolgte mit dem Vergleich der vorhergesagten Geschwingkeit und Turbulenzintensität der fernen Nachlaufströmung mit Windkanal Messungen. Die Vorhersage wurde auch mit Messungen an einer Nachlaufströmung von einer Windturbine im Sexbierum Windpark verglichen. Die Leistungsfähigkeit des Modells ist auch bewertet in der Vorhersage von Nachlaufströmunginteraktion zwischen mehreren Windturbinen im gleichen Windpark, mittels Messungen von der Leistung der Windturbinen. Die übereinstimmung zwischen den Resultaten von Modell und Messergebnissen ist für alle Fälle gut. Diese stimmt fuer beide einzel und doppelt Nachlaufströmungvorhersage. Weiter wurden auch Simulationen über den off shore Windpark Lillgrund durchgeführt und die Sensitivität vom Leistungsverlust in Abhängigkeit von Windrichtung in zwei Reihen des Windparks untersucht. Simultan wurde der Wind und Nachlaufströmung für zwei Windparks in komplexen Terrains in der Schweiz und Spanien simuliert. Das Verhalten der Nachlaufströmung der Windturbine und die gegebenen Turbulenzeigenschaften in einen Windpark durch die Topografie wurden bewertet. Die Resultate zeigen, dass die Leistung vom Windparks die im komplexen terrains gelegen sind, abhänig von dem Terrain und kleinen Windrichtungwechseln ist. Das Modell zeigt akzeptable Ergebnisse für alle Testfälle, jedoch um die zusätzliche Rechenzeit der RANS Simulation im Vergleich zu einfachen Modellen zu rechtfertigen, bedarft die Genauigkeit

weiterer Verbesserungen und einer Reduktion der Rechenzeit. Insgesamt zeigen die Resultate der Testfälle die Fähigkeit des Modells die Interaktion der NachlaufstrÃűmung in Windparks und der Verlust für verschiedeneTurbinen Anordnungen sowie Windrichtungen, zu erfassen. Zusätzlich zur reduzierten Rechenzeit, beinhaltet das Modell ein kartesisches Gitter, welche den Gitter Erstellungsprozess fÃijr Windpark Simulationen ermöglicht.
Im Abschluss, eine Grenzschicht Methode mit die Reynolds-gemittelten Navier-Stokes Gleichungen und mit ein $k-\omega$ Turbulenzmodell ist vorgestellt. Die Windströmung über komplexen Terrain kann, mit diese methode effizient simuliert werden. Weiterhin ist nur ein kartesisches Gitter erforderlich um die Windströmung in alle Richtungen zu simulieren. Das ist möglich mit die Umdrehung der digitale topografische Karte. Damit wurde das lange Verfahren der Erstellung von mehreren Gitter im konventionellen rechteckigen Bereich umgangen. Wandfunktionen wurden zusammen mit der immersierten Grenzschicht Methode verwendet. Damit wurde die zwingende Gitterauflösung in Wandnähe erfüllt und den Effekt der Oberflächenrauigkeit berücksichtigt. Die immersierte Grenzschicht Methode wurde verwendet, um Testfälle moderater komplexer Terrains wie Askervein Hill und sehr komplexen Terrains wie Bolund Hill zu simulieren. Die Resultate der Windgeschwindigkeit und der turbulenten kinetische Energie zeigen gute Übereinstimmungen mit den experimetalen Daten die bei > 5m Bodenhöhe gemessen werden.

Contents

Contents **viii**

1 Introduction **1**
- 1.1 Modeling of Multiscale Flow for Wind Energy Application . . . 2
- 1.2 Research Objectives . . . 7
- 1.3 Thesis Outline . . . 7

2 Governing Equations **11**
- 2.1 Conservative Laws . . . 11
- 2.2 Non-dimensionalization . . . 13
- 2.3 Turbulence Model . . . 14

3 Numerical Method **17**
- 3.1 Time and Space Discretisation . . . 17
  - 3.1.1 Central Space Discretisation . . . 17
  - 3.1.2 Artificial Dissipation . . . 19
    - 3.1.2.1 Isentropic Smoothing . . . 19
    - 3.1.2.2 Non-Isentropic Smoothing (JST scheme) . . . 20
  - 3.1.3 Multistage Scheme . . . 22
    - 3.1.3.1 Hybrid Runge-Kutta . . . 22
    - 3.1.3.2 Time Step . . . 24
- 3.2 Boundary Conditions . . . 25
  - 3.2.1 Inlet Boundary Condition . . . 26
  - 3.2.2 Outlet Boundary Condition . . . 27
  - 3.2.3 Wall Boundary Condition . . . 28
  - 3.2.4 Periodic Boundary Condition . . . 28
- 3.3 Wall Function . . . 29

3.3.1 Generic Wall Function Formulation 30
3.3.1.1 Treatment of Surface Roughness 31
3.3.2 Implementation of Wall Function 32
3.3.2.1 Validation 32

4 Multigrid Method **37**
4.1 Grid Sequencing 38
4.2 Multigrid Cycle 38
4.3 Restriction and Prolongation 41

5 Local Preconditioning **45**
5.1 Preconditioning Formulation 46
5.1.1 Time Step 47
5.1.2 Artificial Dissipation 48
5.1.3 Boundary Conditions 49
5.2 Implementation 50
5.2.1 Choice of Preconditioning Parameter 51

6 Atmospheric Flow over Hilly Terrain **55**
6.1 Inflow Boundary Condition 57
6.2 Turbulence Model Coefficients 58
6.3 Flow over Kettles Hill 58
6.4 Flow over Askervein Hill 61
6.5 Flow over Bolund Hill 67
6.6 Summary 72

7 Immersed Wind Turbine Model **75**
7.1 Review of Wake Models 75
7.2 Model Formulation 77
7.3 Numerical Implementation 80
7.4 Single and Multiple Wake Simulations 83
7.5 Summary 95

8 Estimation of Array Loss in Wind Farms **97**
8.1 Lillgrund Wind Farm 97
8.2 Wind Farm in Complex Terrain 99
8.3 Summary 105

9 Immersed Boundary Method for Terrain Modeling **107**
9.1 Numerical Method 109
9.1.1 Finite Volume Lax-Wendroff Method 109
9.1.2 Multiple Grid Algorithm 111
9.2 Immersed Boundary Method 112

9.2.1 Turbulence Boundary Condition 114
9.2.1.1 Low-Reynolds Turbulence Model 114
9.2.1.2 High-Reynolds Turbulence Model 115
9.3 Validation and Results 117
9.3.1 Separating Flow over Cylinder 117
9.3.2 Flow over Two-Dimensional Hill 118
9.3.3 Flow over Askervein Hill 120
9.3.4 Flow over Bolund Hill 126
9.3.4.1 Wind Calculation for Various Directions 127
9.4 Model Accuracy Close to Ground 135
9.5 Summary 139

10 Summary and Conclusion **141**
10.1 Future Work 144

Bibliography **147**

CHAPTER

1

Introduction

The alternative energy resources has grown rapidly in recent decades aiming to reduce the consumption of fossil fuels in the presence of ever-growing worldwide energy consumption. Proponents of alternative energy argue that fossil fuels are inefficient, unsustainable, environmentally destructive, and the primary contributor to global climate change. Additionally, high economical and political cost of ensuring the security of supply of fossil fuels, the variability and the overall increase in the price over last decades are making the alternative energy resources more competitive.

The wind energy market has grown rapidly in recent decades. The EU wind energy sector installed 11.6 gigawatts (GW) of capacity in 2012, bringing the total wind power capacity to 105.6 GW. Wind energy represented 26% of all new EU power capacity installed in 2012, representing investments of between 12.8 billion and 17.2 billion Euros. It is now meeting 7% of Europe's electricity demand. The global wind energy statistics in 2012 also show the same trend of continued expansion of the market, with annual market growth of almost 10%, and cumulative capacity growth of about 19% [8].

The growth in wind energy is not limited to Europe. In year 2013, the worldwide wind capacity reached 296,255 MW by the end of June 2013, out of which 13,980 MW were added in the first six months of 2012 showing overall 6% growth. Even though this increase is less than in the first half of 2012 and 2011, when 16.5 GW and 18.4 GW were added respectively. All wind turbines installed worldwide by mid-2013 can generate around 3.5% of the world's electricity demand [10].

Despite the slight decrease in the growth rate of wind energy in 2013, the predictions show that the growing trend in the use of renewable and wind energy will continue worldwide. In the recently released International Energy Outlook 2013 (IEO2013) projects published by the U.S. Energy Information Administration's (EIA) indicates that the world energy consumption will grow by 56% between 2010 and 2040, from 524 quadrillion British thermal units (Btu) to 820 quadrillion Btu [2]. The report shows that renewable energy and nuclear power are the world's fastest-growing energy sources, each increasing 2.5% per year. In the

Reference case, the renewable share of total energy use rises from 11% in 2010 to 15% in 2040. Almost 80% of the projected increase in renewable electricity generation is fueled by hydro-power and wind power. Of the 5.4 trillion kilowatt-hours of new renewable generation added over the projection period 1.5 trillion kilowatt-hours (28%) is attributed to the wind.

1.1 Modeling of Multiscale Flow for Wind Energy Application

Following the increasing interest and investments in wind energy, many engineering techniques have been developed in order to identify the suitable sites and estimate the annual energy of the potential projects. In recent years, development of GIS based tools and their application in large-scale assessment of various factors such as land availability, road access, grid developments and cost related to construction and operation has facilitated and improved the primary estimations of the overall cost of the potential projects. However, in order to accurately estimate the revenue of a wind farm, the energy output must also be predicted reliably. The process of energy production in wind farms consists of two main processes. First, extracting the kinetic energy from the wind and conversion to mechanical energy and second the process of conversion of mechanical energy to the useful energy which is mostly electrical [153]. Even though the second phase also poses some technical challenges specially as the dimensions of the wind turbines grow that has been the dominant trend of last few years, but the real uncertainties are mainly attributed to the first mechanism.

In order to predict the amount of mechanical power produced by wind turbine, first the wind speed at the location of the turbine must be estimated. In the early years, large scale wind prediction and site assessment was mainly carried out using field techniques combined with a general understanding of the large-scale winds using meteorology and the effect of local topography in the region of interest [67]. Even though these techniques were sometimes useful in predicting the long-term wind availability over moderate terrain and flat regions with dominant and distinct climatology they often failed when more complex atmospheric and local effects were present. More advanced empirical tools were later developed using various mathematical relationships, as well as the upper-air wind data [128]. Apart from the general wind conditions at the location of a planned project obtained using above-mentioned empirical tools, an accurate measurement of the wind speed has always been a decisive factor in determining the input wind speed to the wind turbines and the economic efficiency of a project. There has been a significant technical progress in terms of near-site wind speed measurements, including the instrumentation, calibration, height selection and placement of the measurements tower. However, the innate limitations of these measurements in terms of cost, time period as well as being limited to few location and heights necessitates alternative complementary solutions.

In last few years, with the increase in computational resources and major developments in numerical weather prediction, there has been an increased focus on the application of mesoscale modeling techniques as a major focus of research in wind resource assessment and mapping [11]. The modeling tools of weather forecasting, mesoscale modeling, are used for climatological assessment of wind energy potential over region of interest with grid resolutions in order of tens of kilometers. The simulations are typically performed over a long time period, sometimes couple of decades to assess the climate in a region to the extent of a country or a continent. These models offer a number of advantages for wind resource assessment, such as the ability to simulate, with reasonable accuracy, complex wind flows in areas where surface measurements are scarce, flexibility in the choice of area of interest, resolution, height above ground and time period to be studied. Large scale data obtained using mesoscale simulations could be used together with other GIS data in order to facilitate the search for suitable sites for wind energy projects.

As the interest in use of mesoscale modeling for wind resource assessment is growing, extensive studies are being performed to improve the model predictions in terms of wind speed estimations [11] [105] [56] [69] . The extensive research in the field of numerical weather predictions mostly focuses on accurate predictions of weather fronts and low level wind predictions were not addressed adequately in this field in the past. However, in recent years several studies investigated the effect of different factors including grid resolution, the definition of surface roughness and atmospheric stratification and the part they play in the overall uncertainty of the modeling aiming to further improve the estimation of highly unsteady wind flow in the atmospheric boundary layer upstream of the wind turbines [23] [102].

Even though mesoscale simulations provide flexibility for project developments and their accuracy for wind resource assessment is in the rise, mesoscale solvers are not solely sufficient for determining the incoming wind flow at the location of a wind turbine. Mesoscale simulations are not sufficient for assessing microscale wind variations due to two main reasons. The first is the fact that the mesoscale solvers are typically built based on hydrostatic assumption in the atmosphere which filters out the vertically moving acoustic waves from the momentum equation [68]. The hydrostatic approximation is derived assuming a balance between the pressure field and gravity in vertical direction and this assumption is reasonable as long as the horizontal grid resolution is large ($\sim O(km)$). In scales relevant to microscale wind flow simulations, the hydrostatic assumption fails and hence mesoscale solvers can not provide an accurate assessment of the flow.

The second and possibly more important shortcoming of mesoscale modeling is related to horizontal grid size which could not be eliminated by using non-hydrostatic models [123]. Due to limitations of parameterization schemes as well as the computational cost, mesoscale models typically have low grid resolutions ($2-200km$) and therefore fail to represent the relevant terrain forcing of small topographical features. However these microscale structures have profound effect on the local wind distribution [132]. The variations of near-surface wind speed due to topography could be significant. The increase or decrease of wind speed over high and low elevations above 100 percent are not uncommon in nature and accurate prediction of this speed-up is very crucial in prediction of wind energy potential of a site considering the fact that both the the power and the structural load on the turbine are related to velocity cube ($\sim V^3$) and square ($\sim V^2$) respectively [59].

Even though significant improvements have been achieved during the last few years, wind resource assessment still makes the largest contribution to the overall uncertainty in the prediction of wind power generation. An error of 3% in the wind speed measurement can lead to a discrepancy of about 10% in annual energy production. It is therefore absolutely necessary to modify and improve all the available simulation and measurement techniques to achieve the best possible accuracy. In addition, the growing interest and demand in wind energy, results in future projects developments to be planned in sites with less ideal wind conditions. Consequently, many techniques used for site calibration and annual energy yield estimation used for "good quality" sites need to be revisited to decrease the model uncertainties and identify and evaluate the potential wind project sites in more complex terrains.

The second challenge in prediction of energy output from wind turbine is related to the field of aerodynamics. Once the incoming wind profile is determined, the process of converting the kinetic energy to mechanical energy by the turbine must be well understood and predicted. The system dynamics and interaction with the structure are also quite important, however that is a rather challenging task. Wind turbines, the largest rotating machines on earth, are subjected to highly turbulent and unsteady atmospheric boundary layer which makes the design and performance prediction of these machines quite difficult [85]. Wind turbines at the design point, operate while the rotating blade and non-uniform instantaneous wind velocity may create regions of separated, three-dimensional highly unsteady flow field with length scales ranging from the turbulent boundary layer around the blades to large extent of the low velocity wake region behind the turbine [36]. The combination of large Reynolds number and low velocity makes both the experimental and numerical work on the wind turbines rather costly and challenging. Numerical predictions of the three dimensional unsteady flow field in the near wake region and resultant loading distribution on wind turbine blades of a single full-scale turbine requires substantial computational resources.

Additional complexity in prediction of power performance arises considering the interaction of the turbines within wind farms. Due to several limitations such as land availability, cost, logistics, etc wind turbines usually are closely placed to each other within wind farms. Therefore, turbines often operate in the wake of upstream turbines. Due to main characteristics of wake flow, mainly meaning reduced momentum and increased turbulence, wind turbines operating in the wakes of upwind wind turbines may have 30 – 40% power losses compared to upwind turbine and up to 80% larger fatigue loads [124]. The power loss reduces the expected revenue of the wind farm and the increased loads reduce the expected life span of the downwind wind turbines in the wind farm. Hence, in last few decades, there has been a significant effort in developing analytical or semi-empirical models to simulate wind turbine wakes. These models differ in approach, level of complexity, accuracy and computational cost. To date, the vast majority of wind farms have been designed using engineering models which have been tuned and validated using experimental data. As wind farms become larger and are placed in areas with increasing variety of wind and terrain conditions, empirical correction upon empirical correction are being developed upon the basis of scarce and perhaps erroneous experimental data [9]. Therefore, while some of the models provide acceptable accuracy for single or multiple wakes in limited scenarios but due to their inherently simplifying assumption, they often fail when more complex inflow or ground features are present [43].

With the increase in available computational resources and further developments in microscale wind predictions, there has been an increased interest in using three-dimensional field models, which solve the Reynolds-Averaged Navier Stokes (RANS) equations in connection with a turbulence model [115]. This approach enables the wind flow and turbine wake to be simulated simultaneously. As the microscale wind pattern over a specific topography is affected by both the long extent of the wake and the blockage effect of the wind turbines, the placement of turbines based only on a simulated microscale wind field may not result in an optimized micrositing of wind turbines in a wind farm.

Overall, it is desirable to tackle the challenge of numerical prediction of the power output of a wind farm by focusing on the relevant scales, resolving the microscale wind and wake within the wind farm and minimizing the simplifying assumptions while keeping the computational cost reasonable. The numerical simulations of multiscale physics involved in estimations of energy output of turbines could be achieved by offline coupling of mesoscale and microscale flow solvers to obtain the wind flow upstream of the wind turbine with high horizontal and vertical resolution and employing the appropriately designed models for small scale aerodynamics around the wind turbine blades and vicinity of the rotor (Figure 1.1). In such approach, a unified wake model which copes with all types of wind farms and their interaction with the earth's boundary layer is achievable. Using Computational Fluid Dynamics, eliminates the need for simplifying assumptions in treatment of wind or wake flow over the terrain within a wind farm and can substantially improve predictions of wind flow and wake effects within wind farms, and therefore provide a more accurate assessment of the expected annual energy yield of a potential project. In addition, by reducing the computational cost and eliminating common challenges in using CFD such as grid generation, one can greatly improve the usability of such models and facilitate the path to routine use of CFD in wind energy for micrositing purposes on an industrial scale.

Due to the fact that the speed of the wind flow is quite low, typically the microscale wind simulations are performed using incompressible RANS solvers [151] [29] [117]. In these solvers an elliptic equation for the pressure is obtained by the help of the continuity equations and the incompressibility assumption which makes the characteristic of the governing equations independent of the acoustic speed [149] [19]. This eliminates the problem of large disparity between the convective and acoustic wind speed resulting in stiffness of the numerical scheme that could be observed applying the compressible Navier Stokes equations for low speed flow [156] [148]. On the other hand there are wide variety of applications in engineering that deal with many different working fluids and flow conditions. There are

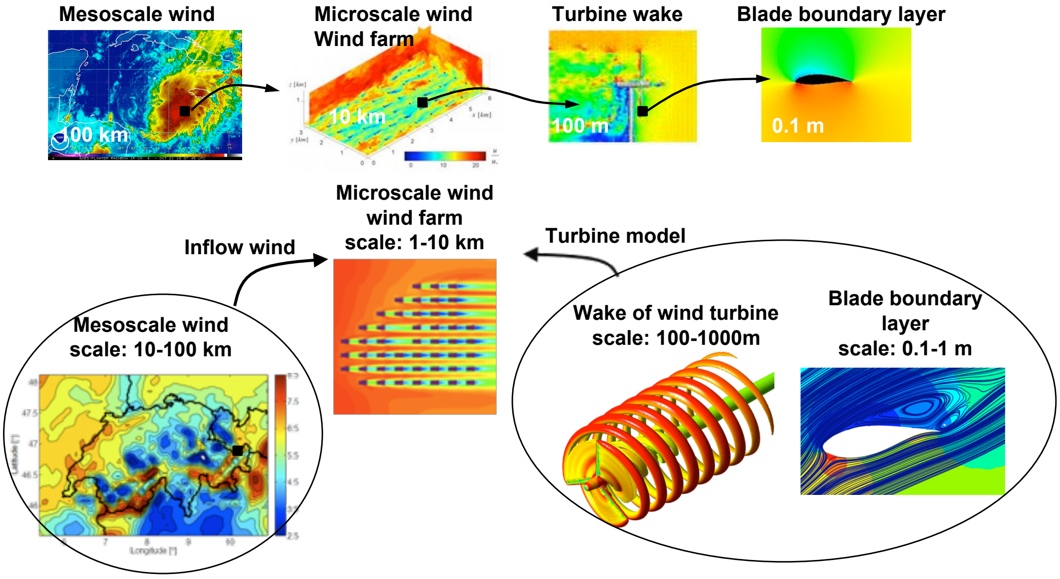

Figure 1.1: various scales involve in prediction of wind turbine performance and the selected approach to tackle the multiscale problem.

variety of application in which large region of low speed flow are embedded in a high speed flow, or cases of low speed flows with density change due to heat release or thermal conduction [19]. Therefore the development of an all purpose solver that can be used for wide range of engineering applications was also in the scope of the this work.

In this thesis the microscale wind flow simulations are made using an all purpose multistage solver. Multistage time marching CFD solvers have been successfully used for a number of years for the computation of transonic and supersonic flows, because they have low computational cost, have high accuracy in the solution, are relatively easy to implement, and are well suited for parallelization [65]. However, when these solvers are used for wind energy applications, their performance is rather poor [46] . In the region of low Mach numbers, the convergence behavior of these methods degrades due to the large disparity between the acoustic and convective wave speeds and the solution's accuracy deteriorates as a result of disproportional scaling of the artificial viscosity [84] [83]. Preconditioning techniques enable low Mach number flows to be efficiently simulated using an existing compressible solver. In this thesis, LEC's existing compressible solver, MULTI3, has been modified and improved in order to obtain an all purpose solver for incompressible and compressible fluid flow simulations. In the first step, the time integration of the solver has been replaced with a multistage Runge-Kutta scheme. Multigrid method and wall function has been integrated into the solver to improve the convergence and resolve the high Reynolds number atmospheric flow accurately. Inlet and outlet boundary conditions has been modified and a preconditioning method is implemented into a multistage compressible RANS solver and applied for the first time, to the author's knowledge, to high Reynolds number low Mach number atmospheric flow simulations.

1.2 Research Objectives

In the scope of this thesis, the simultaneous simulation of microscale wind and wind turbine wake was made using an all purpose RANS solver. The implementation of convergence acceleration including preconditioning and multigrid method, efficient wake model and numerical methods such as immersed boundary method were made in order to develop a fast and reliable Navier-Stokes solver for wind farm simulations and array loss estimation offhsore and over complex terrain. The overall research objectives could be summarized as:

- Modify and improve LEC's existing compressible solver, MULTI3, to develop an all purpose RANS solver for efficient simulation of compressible and incompressible flows that could be used for wind flow simulations using preconditioning in an efficient and robust manner
- Investigate the capabilities and limitations of RANS solver in predicting the wind flows over wide range of topography
- Develop a computationally efficient model for wind turbine wake simulations to decrease the computational cost of multiple wake simulations in wind farms while maintaining the accuracy in the prediction of far wake development.
- Demonstrate the application of the developed microscale solver and immersed wind turbine model for simultaneous simulation of wind flow in wind farms assessing the interaction of the wake and wind flow with topography in highly complex terrain.
- Apply the widely used method of immersed boundary for high Reynolds number microscale wind simulations in order to eliminate the difficulties of grid generation for wind resource and array loss assessment for multiple wind directions.

1.3 Thesis Outline

Chapter 1

A short background and the research objectives are presented in the first chapter. The background focuses on the flow modeling for wind energy application in different scales and how the current work attempts to overcome the practical challenges of use of microscale wind and wake flow simulations for more accurate assessment of wind projects.

Chapter 2

In the second chapter, the governing equations of wind flow and the turbulence closure model are described and the variables are introduced.

Chapter 3

The numerical method, called hybrid Runge-Kutta method, used in the in-house solver is described in detail in this chapter. First the spatial and temporal discretisations are discussed and then the various boundary conditions used for the flow solver are detailed. The formulations and implementation of the generalized wall function necessary for high Reynolds number wind flow simulations are also described.

Chapter 4

The implementation of the multigrid method in connection with Runge-Kutta is presented in this chapter. The concept of multigrid which employs series of coarser grids in order to improve the convergence behavior of the solver, is briefly described and then the implementations and treatments of boundary conditions are detailed.

Chapter 5

In this chapter, the implementation of low Mach number preconditioning technique is presented. The preconditioning technique further improves the degraded convergence behavior of Runge-Kutta for low Mach number flow which occurs due to the large disparity between the acoustic and convective wave speeds. Different approaches are introduced in order to improve the performance of the preconditioning for high Reynolds number atmospheric flows and the required modifications of boundary conditions are detailed.

Chapter 6

Performance of preconditioned multigrid is demonstrated for a broad range of atmospheric flow test cases including Askervein Hill, Kettles Hill and Bolund Hill and its performance, accuracy and robustness assessed for the atmospheric wind simulations. In addition the variations of local wind and turbulence due to topography are discussed.

Chapter 7

A newly developed three-dimensional field wake model that is used for simulations of wake flows in wind farms is described in this chapter. Next, predictions of single and multiple wakes, which are relevant to the flow in wind farms, are also presented using the immersed wind turbine model that is embedded in the preconditioned solver and model results are assessed against wind tunnel and full scale measurements.

Chapter 8

In this chapter, the simultaneous simulations of microscale wind and wake flow in wind farm is reported and array loss estimations are compared with available SCADA data for three wind farms. First, the simulations are performed over offshore wind farm Lillgrund. Next the simulations of wind flow in two wind farms located in complex terrain are presented.

Chapter 9

The immersed boundary method is introduced in this chapter in order to further facilitate the grid generation for the atmospheric flow simulations. The model and the detailed implementation is described. The validation of results are presented for the case of idealized flows and flow over Askervein and Bolund Hill.

Chapter 10

The chapter presents the summary of the results from all chapters and the final concluding remarks on the thesis. Suggestion on the future work and further improvements are also discussed.

CHAPTER

2

Governing Equations

This chapter describes the governing equations of wind and in general fluid flows. As mentioned in the previous chapter, in the current study, the existing solver is modified to obtain an all purpose Reynolds Averaged Navier Stokes solver and to use for atmospheric and wake flow simulations. First, the conservation laws, governing the fluid motion are presented and the non-dimensionalized equations are derived. The turbulence modeling approach used in the code is next introduced.

2.1 Conservative Laws

The three-dimensional compressible Reynolds averaged Navier Stokes equations are written in conservative form for a Cartesian coordinate system as follows:

$$\frac{\partial W}{\partial t}+\frac{\partial(F-F_v)}{\partial x}+\frac{\partial(G-G_v)}{\partial y}+\frac{\partial(H-H_v)}{\partial z}=0 \tag{2.1}$$

The state vectors of the conservative variable viscous and inviscid fluxes are:

$$W=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{bmatrix}\ F=\begin{bmatrix}\rho u\\ \rho uu+p\\ \rho vu\\ \rho wu\\ \rho Eu\end{bmatrix}\ G=\begin{bmatrix}\rho v\\ \rho uv\\ \rho vv+p\\ \rho wv\\ \rho Ev\end{bmatrix}\ H=\begin{bmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho ww+p\\ \rho Ew\end{bmatrix} \tag{2.2}$$

$$
\begin{aligned}
F_v &= \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{xx}u + \tau_{xy}v + \tau_{xz}w + q_x \end{bmatrix} \\
G_v &= \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ \tau_{yx}u + \tau_{yy}v + \tau_{yz}w + q_y \end{bmatrix} \\
H_v &= \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ \tau_{zx}u + \tau_{zy}v + \tau_{zz}w + q_z \end{bmatrix}
\end{aligned}
\tag{2.3}
$$

where the shear stresses are defined as:

$$
\begin{aligned}
\tau_{xx} &= \mu\left(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\right) \\
\tau_{yy} &= \mu\left(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right)\right) \\
\tau_{yy} &= \mu\left(\frac{4}{3}\frac{\partial w}{\partial z} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right) \\
\tau_{xy} &= \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\
\tau_{xz} &= \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\
\tau_{yz} &= \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\
\tau_{yx} &= \tau_{x_y}, \tau_{zx} = \tau_{x_z}, \tau_{yz} = \tau_{z_y}
\end{aligned}
\tag{2.4}
$$

and the heat fluxes are

$$
q_x = -\kappa\frac{\partial T}{\partial x} \quad q_y = -\kappa\frac{\partial T}{\partial y} \quad q_z = -\kappa\frac{\partial T}{\partial z}
\tag{2.5}
$$

The ideal gas relation is used to close the system of equations. This equation relates the total energy per unit mass E to pressure p.

$$
p = (\gamma - 1)\left[\rho E - \frac{1}{2\rho}\left((\rho u)^2 + (\rho v)^2 + (\rho w)^2\right)\right]
\tag{2.6}
$$

Also total enthalpy per unit mass is defined as:

$$H = E + \frac{p}{\rho} \tag{2.7}$$

where ρ is density, u, v and w are the velocity components for the Cartesian coordinates, T is the temperature, E is the total internal energy per unit mass, κ is the thermal conductivity, γ is the ratio of specific heats and μ is the total eddy viscosity which is the summation of the laminar viscosity calculated using the local temperature based on Sutherland's law and the turbulent viscosity calculated using the turbulence model described in the next sections (Eq. 2.8).

$$\mu = \mu_l + \mu_t \qquad \mu_l = 1.461 \times 10^{-6} \frac{T^{\frac{3}{2}}}{T + 110.3} \tag{2.8}$$

The thermal conductivity is obtained using different Prandtl numbers for laminar and the turbulent parts (Eq. 2.9).

$$\kappa = \frac{\gamma \mu_l}{Pr_l} + \frac{\gamma \mu_t}{Pr_t} \tag{2.9}$$

2.2 Non-dimensionalization

In practice, the governing equations are solved in a non-dimensionalized form in order to minimize the error related to numerical computations. For the non-dimensionalization of the Navier Stokes equations, the following reference quantities are defined:

μ_{ref} : reference density
c_{ref} : reference speed of sound
L_{ref} : reference length
μ_{ref} : reference viscosity

A set of non-dimensionalized quantities are defined based on reference values described above.

$$\begin{aligned}
&x^* = \frac{x}{L_{ref}} \quad y^* = \frac{y}{L_{ref}} \quad z^* = \frac{z}{L_{ref}} \\
&u^* = \frac{u}{c_{ref}} \quad v^* = \frac{v}{c_{ref}} \quad w^* = \frac{w}{c_{ref}} \\
&\rho^* = \frac{\rho}{\rho_{ref}} \quad p^* = \frac{p}{\rho_{ref} {c_{ref}}^2} \quad E^* = \frac{E}{c_{ref}}^{2} \\
&H^* = \frac{H}{c_{ref}}^{2} \quad t^* = \frac{t a_{ref}}{L_{ref}} \quad \mu^* = \frac{\mu}{\mu_{ref}}
\end{aligned} \tag{2.10}$$

As a result of the above selected non-dimensionalization, the reference Reynolds number Re_{ref} appears in the conservative form of the governing equations which scales the viscous fluxes in non-dimesionalized RANS equations (Eq. 2.11).

$$Re_{ref} = \frac{\rho_{ref} c_{ref} L_{ref}}{\mu_{ref}} \tag{2.11}$$

2.3 Turbulence Model

The two-equation $k-\omega$ turbulence model is chosen to compute the eddy viscosity μ_t. Consistent with the governing equations, the conservative formulation of the $k-\omega$ is preferred. Eddy viscosity μ_t is substituted in the source terms by the corresponding value $\rho k/\omega$. This decouples the ω equation from the k equation, and the set of coupled partial differential equations (PDE) can be treated as a sequence of ordinary differential equations (ODE). The non-dimensionalized $k-\omega$ equations reads:

$$\frac{\partial W_t}{\partial t}+\frac{\partial(F_{t,c}-F_{t,d})}{\partial x}+\frac{\partial(G_{t,c}-G_{t,d})}{\partial y}+\frac{\partial(H_{t,c}-H_{t,d})}{\partial z}=P-D \tag{2.12}$$

where subscript c and d refer to convective and diffusive parts. The source term at the right hand side consists of production and destruction terms. In the above equation, the turbulent state vector W_t is:

$$W_t=\begin{bmatrix}\rho k\\ \rho\omega\end{bmatrix} \tag{2.13}$$

and the convective and diffusive fluxes are:

$$F_{t,c}=\begin{bmatrix}\rho uk\\ \rho u\omega\end{bmatrix} G_{t,c}=\begin{bmatrix}\rho vk\\ \rho v\omega\end{bmatrix} H_{t,c}=\begin{bmatrix}\rho wk\\ \rho w\omega\end{bmatrix} \tag{2.14}$$

$$\begin{aligned}F_{t,d}&=\begin{bmatrix}\left(\frac{\mu}{Re_{ref}}+\sigma^*\mu_t\right)\frac{\partial k}{\partial x}\\ \left(\frac{\mu}{Re_{ref}}+\sigma\mu_t\right)\frac{\partial\omega}{\partial x}\end{bmatrix}\\ G_{t,d}&=\begin{bmatrix}\left(\frac{\mu}{Re_{ref}}+\sigma^*\mu_t\right)\frac{\partial k}{\partial y}\\ \left(\frac{\mu}{Re_{ref}}+\sigma\mu_t\right)\frac{\partial\omega}{\partial y}\end{bmatrix}\\ H_{t,d}&=\begin{bmatrix}\left(\frac{\mu}{Re_{ref}}+\sigma^*\mu_t\right)\frac{\partial k}{\partial z}\\ \left(\frac{\mu}{Re_{ref}}+\sigma\mu_t\right)\frac{\partial\omega}{\partial z}\end{bmatrix}\end{aligned} \tag{2.15}$$

The standard production terms defined by Wilcox [162] might give excessive results near stagnation points. Launder and Kato [78] proposed a delimited production term taking a vorticity parameter ω into account.

$$P=\begin{bmatrix}\mu_t S\Omega\\ \alpha S\Omega\end{bmatrix} D=\begin{bmatrix}\beta^*\rho\omega k\\ \beta^*\rho\omega^2\end{bmatrix} \tag{2.16}$$

where the definition of S and Ω are:

$$\Omega^2=\frac{1}{2}\left[\left(\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial u}{\partial z}-\frac{\partial w}{\partial x}\right)^2+\left(\frac{\partial v}{\partial z}-\frac{\partial w}{\partial y}\right)^2\right] \tag{2.17}$$

$$S^2=\frac{1}{2}\left[\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)^2+\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)^2\right] \tag{2.18}$$

α, β, β^*, σ, σ^* are the standard coefficients of Wilcox's turbulence model. These constants according to Wilcox are:

$$\alpha = \frac{5}{9} \quad \beta = \frac{3}{40} \quad \beta^* = \frac{9}{100} \quad \sigma = 0.5 \quad \sigma^* = 0.5 \tag{2.19}$$

and μ_t is calculated using Eq. 2.20.

$$\mu_t = \rho \frac{k}{\omega} \tag{2.20}$$

CHAPTER

3

Numerical Method

The numerical method used for the discretisation of the governing equations is presented in this chapter. The numerical model is implemented into LEC's in-house solver called MULTI3. The solver was originally developed based on Ni's Lax-Wendroff [110] method and applied by Saxer in a 3D turbomachinery Euler flow [125], who used non-reflecting boundary conditions and relative systems for rotating blade rows. Burdet extended the solver to the Reynolds Averaged Navier-Stokes equations [24] and added the turbulence models of Baldwin and Lomax and Spalart and Allmaras. Later Basol [15] extended the solver to include the turbulence model of Wilcox [162]. In this work, a multistage time integration scheme has been added to the solver. Both numerical schemes employ the finite volume method. After implementation and validation, the code has been used for the simulations of atmospheric flow over hilly terrain and the development of the immersed wind turbine model presented in the next chapters.

In this chapter, first the time and space discretisation of the governing equations are discussed. Next, the artificial smoothing and the stability of the method is analyzed and at the end the various boundary conditions imposed at domain boundaries are described.

3.1 Time and Space Discretisation

3.1.1 Central Space Discretisation

The space discretisation of the Navier Stokes equations with emphasis on convective terms are discussed here. The computational mesh is assumed to contain only hexahedral cells with the flow vector stored in the cell vertices. The viscous fluxes are calculated based on a purely central discretisation. The derivatives contained in the first order viscous fluxes F_v, G_v and H_v are calculated at the cell center using Gauss's theorem. The second order viscous fluxes are neglected since it has been shown that these terms have negligible effect on the accuracy of the whole computation, in particular in the range of high Reynolds number flows [62]. The inviscid fluxes through a cell-face surface $\overrightarrow{S}$ are split into a central part and a dissipation part. If the cell index is I, J, K, the numerical flux of cell-face $I+\frac{1}{2}, J, K$ is given by:

$$\left(\overrightarrow{F_c}\Delta S\right)_{I+\frac{1}{2},J,K} = \frac{1}{2}\left[\overrightarrow{F_c}\left(W_{I,J,K}\right) + \overrightarrow{F_c}\left(W_{I+1,,J,K}\right)\right]\Delta S_{I+\frac{1}{2},J,K} - D_{i+\frac{1}{2},J,K} \tag{3.1}$$

The first term on the right hand side of the equation corresponds to the central evaluation of the inviscid fluxes. The cell-face value of the fluxes is here obtained by averaging the fluxes at the face center using the values computed at the neighboring nodes. The second term of the right hand side is called the numerical dissipation term. It corresponds to the dissipation terms for an upwind scheme and the artificial dissipation for a central scheme. The numerical dissipation term $D_{I+\frac{1}{2},J,K}$ depends on the type of the discretisation method and is presented in detail in the next section.

After all the flux contributions from different cell faces are calculated, an intermediate inviscid residual $R_{c_{I,J,K}}$ from all grid cells is obtained. In order to relate the cell-base to node-base residual, a distribution formula is used. Several approaches of distributing the residuals are suggested in literature. In the current implementation, a non-weighted sum due to Hall [57] is used which is proved to be the most robust. Using this approach, the residual at each node $R_{c_{I,J,K}}$ is calculated using the sum of the residuals calculated at neighboring cells. (Eq. 3.2)

$$\begin{aligned}\overrightarrow{R}_{i,j,k} = \frac{1}{8}\Big(&\overrightarrow{R}_{I,J,K} + \overrightarrow{R}_{I-1,J,K} + \overrightarrow{R}_{I-1,J-1,K} + \overrightarrow{R}_{I,J-1,K} + \\ &\overrightarrow{R}_{I,J,K-1} + \overrightarrow{R}_{I-1,J,K-1} + \overrightarrow{R}_{I,J-1,K-1}\Big)\end{aligned} \tag{3.2}$$

In fact, many other node-base variables which are computed at cell center such as the time step and volume are calculated using a similar approach (Eq. 3.3)

$$\begin{aligned}\Delta V_{i,j,k} = \frac{1}{8}\big(&\Delta V_{I,J,K} + \Delta V_{I-1,J,K} + \Delta V_{I-1,J-1,K} + \Delta V_{I,J-1,K} + \\ &\Delta V_{I,J,K-1} + \Delta V_{I-1,J,K-1} + \Delta V_{I-1,J-1,K}\big)\end{aligned} \tag{3.3}$$

Adding up all the residuals including the inviscid, viscous parts and the part related to artificial dissipation, the overall residual at cell vertices are known:

$$\overrightarrow{R}_{i,j,k} = \overrightarrow{R}_{c_i,j,k} + \overrightarrow{R}_{v_{v_i},j,k} + \overrightarrow{R}_{d_{c_i},j,k} \tag{3.4}$$

dropping the arrow bar, using the above definition, the time stepping scheme can be written as:

$$\frac{dW_{i,j,k}}{dt} = -\frac{1}{\Delta V}R_{i,j,k} \tag{3.5}$$

for each conservative flow variable. This is a system of ordinary differential equations which has to be solved at every grid point.

3.1.2 Artificial Dissipation

As described in the previous section, in order to achieve a stable scheme, artificial dissipation terms need to be added to the right hand side of Eq. 3.1 in the central discretisation. It is well-known that central discretisation without artificial smoothing is unstable due to the odd-even decoupling that occurs in neighboring nodes [19]. In addition to odd-even decoupling, the central scheme suffers from non-physical oscillations near large discontinuities such as shocks. These oscillations also contaminate the results and need to be prevented using a second order smoothing term.

In this study, since an all purpose code was developed, the detailed implementation and modification of both second-order and forth-order smoothing are described. However, in the application of the preconditioned code for atmospheric and wind flow the second order terms used in compressible flow to avoid oscillations near shocks are not used.

The dissipation terms are computed using two different approaches which differ in terms of accuracy and computational cost. In next sections both methods are described in detail.

3.1.2.1 Isentropic Smoothing

The artificial smoothing added to the RANS equations to be solved is a combination of second and fourth order smoothing formulated as:

$$D = \nu^{(2)} \left[\frac{\partial^2 W}{\partial^2 x} + \frac{\partial^2 W}{\partial^2 y} + \frac{\partial^2 W}{\partial^2 z} \right] + \nu^{(4)} \left[\frac{\partial^4 W}{\partial^4 x} + \frac{\partial^4 W}{\partial^4 y} + \frac{\partial^4 W}{\partial^4 z} \right] \tag{3.6}$$

where the coefficients $\nu^{(2)}$ and $\nu^{(4)}$ are used to scale the second and fourth order terms respectively. In the current cell-vertex implementation, unlike inviscid and viscous fluxes which are calculated at cell center and then distributed to nodes, the artificial viscosity terms are computed directly at cell vertices. The second order terms or $\nabla^2 W$ at node i is approximated as:

$$\nabla^2 W_i = \sum_{j=1}^{n_i} \omega_j \left(W_j - W_i \right) \tag{3.7}$$

where n_i is the number of the nodes surrounding node i. This number is always 8 for the internal nodes in a hexahedral mesh. The ω_j represents a weight applied to the Laplacian in order to keep the second order accuracy, even in irregular meshes. For fairly smooth mesh, the weights for the stencil is close to 1.

The fourth order dissipation terms are calculated using $\Delta^2 W$:

$$D\left(W_i\right) = \sum_{j=1}^{n} \nu^{(4)} \left(\nabla^2 W_j - \nabla^2 W_i \right) \tag{3.8}$$

where $v^{(4)}$ is the multiplication of $\epsilon^{(4)}$, a global constant that controls the amount of dissipation and a scale factor related to the spectral radius of the Jacobin matrix. In the isentropic formulation of the artificial dissipation, this scale factor is taken proportional to the node-base value for time step [92]. Therefore $v^{(4)}$ becomes:

$$v^{(4)} \approx \frac{\epsilon^{(4)}}{\Delta t} \tag{3.9}$$

The formulation of artificial dissipation described above is computationally cheap and easy to implement. However, there are two main deficiencies of current approach that needs to be addressed. The first problem is related to excessive artificial dissipation in the boundary layer where the physical viscous fluxes are rather large compared to convective forces. In the low Mach number region, either in wake flow or inside boundary layer, the numerical smoothing may slightly alter the physical viscosity level which may result in inaccuracies in modeling the near wall or wake diffusion. For Navier-Stokes equations, a possible strategy is to reduce the level of artificial dissipation in the boundary layer by scaling $\epsilon^{(2)}$ and $\epsilon^{(4)}$ based on local Mach number (Eq. 3.10). This method is implemented in MULTI3.

$$\begin{aligned} \epsilon^{(2)} &= min\left[1, \left(\frac{M}{M_\infty}\right)^2\right]\epsilon^{(2)} \\ \epsilon^{(4)} &= min\left[1, \left(\frac{M}{M_\infty}\right)^2\right]\epsilon^{(4)} \end{aligned} \tag{3.10}$$

The second deficiency of the method rises due to the fact that neither the calcualtion of Laplacian nor the scaling of the terms is directional. This might be acceptable for Euler equations where the aspect ratio of the computational cell is close to unity in the entire computational domain. However the method may result in inaccurate results when applied to Navier-Stokes equations where highly stretched meshes are typically employed. Using the current formulation, an excessive dissipation, proportional to cell aspect ratio will be added in some directions. The problem can be eliminated by defining directional derivatives and edge-based scaling of a smoothing term based on directional spectral radius instead of a cell-base time step.

3.1.2.2 Non-Isentropic Smoothing (JST scheme)

The formulation of the well-known non-linear artificial dissipation scheme of Jameson, Schmidt and Turkel (JST) [72] is presented here. In order to develop a dissipation term which is non-isentropic some modification to the previous scheme are required. Using the JST formulation, the overall node-base dissipation term is written as:

$$\begin{aligned}
D_{i,j,k}(W) = {} & \nu^{(2,x)}_{i+\frac{1}{2},j,k}\left(W_{i+1,j,k}-W_{i,j,k}\right)+\nu^{(2,y)}_{i,j+\frac{1}{2},k}\left(W_{i,j+1,k}-W_{i,j,k}\right)\\
& +\nu^{(2,z)}_{i,j,k+\frac{1}{2}}\left(W_{i,j,k+1}-W_{i,j,k}\right)-\nu^{(4,x)}_{i+\frac{1}{2},j,k}\left(\nabla^2 W_{i+1,j,k}-\nabla^2 W_{i,j,k}\right)\\
& -\nu^{(4,y)}_{i,j+\frac{1}{2},k}\left(\nabla^2 W_{i,j+1,k}-\nabla^2 W_{i,j,k}\right)-\nu^{(4,z)}_{i,j,k+\frac{1}{2}}\left(\nabla^2 W_{i,j,k+1}-\nabla^2 W_{i,j,k}\right)
\end{aligned} \tag{3.11}$$

where the edge-based scaling is given by:

$$\nu^{2,x}_{i+\frac{1}{2},j,k}=\epsilon^{(2)}\partial l_t \lambda_{i+\frac{1}{2},j,k}\Phi^x_{i+\frac{1}{2},j,k} \tag{3.12}$$

∂l_t and $\Phi^x_{i+\frac{1}{2},j,k}$ are:

$$\begin{aligned}
\partial l_t &= \frac{\sum_{s=1}^{n_c}\Delta V_s}{n_c \mid x_{i+1,j,k}-x_{i,j,k}\mid}\\
\Phi^x_{i+\frac{1}{2},j,k} &= max\left[\kappa_{i-1,j,k},\kappa_{i,j,k},\kappa_{i+1,j1k}\right]
\end{aligned} \tag{3.13}$$

where n_c is the total number of cells around the edge that connects the node number i and $i+1$. The κ is defined in Eq. 3.14.

$$\kappa_{i,j,k}=\left| \frac{p_{i+1,j,k}-2p_{i,j,k}+p_{i-1,j,k}}{p_{i+1,j,k}+2p_{i,j,k}+p_{i-1,j,k}} \right| \tag{3.14}$$

Here $\lambda_{i,j,k}$ is the absolute value of the largest eigenvalue of the Jacobian matrix and is used to scale the soothing term in each direction, replacing the Δt used in the previous formulation. λ is calculated in each direction using Eq. 3.15

$$\lambda_{i+\frac{1}{2},j,k}=\left(\mid uS^x+vS^y+wS^z\mid +c\sqrt{(S^x)^2+(S^y)^2+(S^z)^2}\right)_{i+\frac{1}{2},j,k} \tag{3.15}$$

For the fourth order smoothing coefficients we set:

$$\tau^{(4)}_{i+\frac{1}{2},j,k}=\partial l_t \lambda_{i+\frac{1}{2},j,k} max\left[0,\epsilon^{(4)}-\beta\Phi^x_{i+\frac{1}{2},j,k}\right] \tag{3.16}$$

where β is a user defined constant with typical value of 1. This dissipation model will be called the non-isentropic scalar model.

Both artificial schemes described above are used for the simulations of this thesis and the preconditioning approach is implemented and presented in connection with both schemes.

3.1.3 Multistage Scheme

The explicit Runge-Kutta time stepping procedure is used to advance the solution to the steady state. Runge-Kutta schemes also called multistage schemes advance the solution in multiple series of updates or stages. The number of stages and coefficients can be tuned in order to maximize the stability domain and the efficiency of the scheme depending on the space discretisation used. The Runge-Kutta scheme with m number of stages is expressed as:

$$\begin{aligned}
&\overrightarrow{W}^{(0)} = \overrightarrow{W}^{(n)} \\
&\overrightarrow{W}^{(1)} = \overrightarrow{W}^{(0)} - \alpha_1 \frac{\Delta t}{\Delta V} \overrightarrow{R}^{(0)} \\
&\overrightarrow{W}^{(2)} = \overrightarrow{W}^{(0)} - \alpha_2 \frac{\Delta t}{\Delta V} \overrightarrow{R}^{(1)} \\
&. \\
&. \\
&. \\
&\overrightarrow{W}^{(n+1)} = \overrightarrow{W}^{(m)} = \overrightarrow{W}^{(0)} - \alpha_m \frac{\Delta t}{\Delta V} \overrightarrow{R}^{(m-1)}
\end{aligned} \tag{3.17}$$

where α_k is the coefficient of k^{th} stage and $R^{(k)}$ is the overall residual calculated using the flow vector $\overrightarrow{W}^{(k)}$.

3.1.3.1 Hybrid Runge-Kutta

The computational cost of a classical Runge-Kutta scheme could be substantially reduced if diffusive fluxes, both numerical and artificial, are not calculated at every stage. Martinelli [100] and Mavriplis [104] proposed an approach today known as hybrid Runge-Kutta in which a blend of diffusive fluxes is used in the formulation for each stage. This effective multistage scheme can be obtained if the convective part of the overall residual and the diffusive part are stored separately (Eq. 3.18). The convective part includes the central discretisation of the convective terms described in the previous section and a diffusive part that includes both the physical and artificial dissipation terms. The evaluation of the diffusive fluxes is only performed at odd stages.

$$\overrightarrow{R} = \overrightarrow{R_c} - \overrightarrow{R_d} \tag{3.18}$$

In the hybrid Runge- Kutta method, the stage coefficients are carefully optimized to reach the same robustness of a classical Runge-Kutta scheme with higher allowable CFL number. Overall, two Runge-Kutta methods are found particularly effective. The classical 4-stage

Runge-Kutta scheme [72]:

$$\begin{aligned}
&\vec{W}^{(0)} = \vec{W}^{(n)}\\
&\vec{W}^{(1)} = \vec{W}^{(0)} - \alpha_1 \frac{\Delta t}{\Delta V} \vec{R}^{(0)}\\
&\vec{W}^{(2)} = \vec{W}^{(0)} - \alpha_2 \frac{\Delta t}{\Delta V} \vec{R}^{(1)}\\
&\vec{W}^{(3)} = \vec{W}^{(0)} - \alpha_3 \frac{\Delta t}{\Delta V} \vec{R}^{(2)}\\
&\vec{W}^{(4)} = \vec{W}^{(0)} - \alpha_4 \frac{\Delta t}{\Delta V} \left(\vec{R}^{(0)} + 2\vec{R}^{(1)} + 2\vec{R}^{(2)} + \vec{R}^{(3)}\right)\\
&\vec{W}^{(n+1)} = \vec{W}^{(4)}
\end{aligned} \tag{3.19}$$

with coefficients:

$$\alpha_1 = 0.5000,\ \alpha_2 = 0.5000,\ \alpha_3 = 1.0000,\ \alpha_4 = 0.1667 \tag{3.20}$$

and 5-stage hybrid Runge-Kutta scheme:

$$\begin{aligned}
&\vec{W}^{(0)} = \vec{W}^{(n)}\\
&\vec{W}^{(1)} = \vec{W}^{(0)} - \alpha_1 \frac{\Delta t}{\Delta V} \left[\vec{R}_c^{(0)} - \vec{R}_d^{(0)}\right]\\
&\vec{W}^{(2)} = \vec{W}^{(0)} - \alpha_2 \frac{\Delta t}{\Delta V} \left[\vec{R}_c^{(1)} - \vec{R}_d^{(1)}\right]\\
&\vec{W}^{(3)} = \vec{W}^{(0)} - \alpha_3 \frac{\Delta t}{\Delta V} \left[\vec{R}_c^{(2)} - \vec{R}_d^{(2)}\right]\\
&\vec{W}^{(4)} = \vec{W}^{(0)} - \alpha_4 \frac{\Delta t}{\Delta V} \left[\vec{R}_c^{(3)} - \vec{R}_d^{(3)}\right]\\
&\vec{W}^{(n+1)} = \vec{W}^{(0)} - \alpha_5 \frac{\Delta t}{\Delta V} \left[\vec{R}_c^{(4)} - \vec{R}_d^{(4)}\right]
\end{aligned} \tag{3.21}$$

$$\alpha_1 = 0.2500,\ \alpha_2 = 0.1667,\ \alpha_3 = 0.3750,\ \alpha_4 = 0.5000,\ \alpha_5 = 1.0000 \tag{3.22}$$

$$\beta_1 = 1.00,\ \beta_2 = 0.00,\ \beta_3 = 0.56,\ \beta_4 = 0.00,\ \beta_5 = 0.44 \tag{3.23}$$

α and β are stage coefficients, R_c is the convective flux and and R_d is the summation of the physical and artificial viscous fluxes. The stage coefficients α_k and β_k are given in Eq. 3.22 and Eq. 3.23 and R_d and R_c at even stages are calculated as:

$$\begin{aligned}
&\vec{R}_d^{(2,0)} = \beta_3 \vec{R}_d^{(2)} + \left(1 - \beta_3\right) \vec{R}_d^{(0)}\\
&\vec{R}_d^{(4,2)} = \beta_3 \vec{R}_d^{(4)} + \left(1 - \beta_3\right) \vec{R}_d^{(2,0)}
\end{aligned} \tag{3.24}$$

The stability region along the real axis is almost doubled when using the 5-stage hybrid Runge-Kutta instead of 4-stage Runge-Kutta. The overall computational cost per iteration is slightly lower since the viscous and diffusive fluxes are only calculated in 3 stages out of 5. Both formulations are implemented in the in-house code MULTI3. Overall using the 4-stage scheme, the solver was slightly more robust. However, due to the fact that the hybrid formulations showed more desirable convergence behavior, most of the results presented in next chapters are obtained using the hybrid approach.

3.1.3.2 Time Step

For every explicit time-stepping scheme, there is a distinct criteria that defines the stability of the scheme. A scheme is stable if the domain of dependence of the numerical method includes the domain of dependence of the partial differential equation [62]. This means that the time step or a distance by which the information travels within the computational domain must be bounded. This is called the Courant-Friedrchs-Lewy of CFL condition. The 1-D condition for the time step will read:

$$\Delta t = \sigma \frac{\Delta x}{\lambda^c} \tag{3.25}$$

where λ^c represents the maximum eigenvalue of the Jacobian matrix. In the physical terms λ^c is the speed by which the information propagates in the domain. σ is the so-called CFL number used to scale the time step to fulfill the stability condition. The maximum time step for a simplified and linearized set of equations can be calculated using the von-Neumann analysis. Since the governing equations of the fluid flow are highly non-linear, the application of the von-Neumann method for Runge-Kutta scheme in connection with RANS equations can not be carried out. However, in order to obtain an approximate relation, The von-Neumann analysis is applied to linearized form of Euler equation. The result of this analysis on the Cartesian grid, gives the inviscid stability criteria as:

$$\Delta t = \sigma \frac{\Delta V}{max\left(\lambda_x^c, \lambda_y^c, \lambda_z^c\right)} \tag{3.26}$$

where σ is the CFL number and λ_x^c, λ_y^c and λ_z^c are the spectral radius of convective Jacobian flux in i, j and k directions (Eq. 3.27).

$$\begin{aligned}
\lambda_x^c &= (\mid u \mid + c)\,\Delta S_x \\
\lambda_y^c &= (\mid v \mid + c)\,\Delta S_y \\
\lambda_z^c &= (\mid w \mid + c)\,\Delta S_z
\end{aligned} \tag{3.27}$$

where the areas are calculated by averaging values on two opposite side for each direction. Instead of Eq. 3.26, the mostly common used relation for inviscid time step is [19]

$$\Delta t = \sigma \frac{\Delta V}{\lambda_x^c + \lambda_y^c + \lambda_z^c} \tag{3.28}$$

For the Navier Stokes equations, similar to Euler equations, the relation for time step can be proposed as:

$$\Delta t = \sigma \frac{\Delta V}{\left(\lambda_x^c + \lambda_y^c + \lambda_z^c\right) + C\left(\lambda_x^v + \lambda_y^v + \lambda_z^v\right)} \tag{3.29}$$

where C is a constant multiplied by the viscous spectral radius. Assuming the eddy viscosity model the viscous spectral radius is written as:

$$\lambda_v^x = max\left(\frac{4}{3\rho}, \frac{\gamma}{\rho}\right)\left(\frac{\mu_l}{Pr_l} + \frac{\mu_t}{Pr_t}\right)\frac{\Delta(S_x)^2}{\Delta V} \tag{3.30}$$

$$\lambda_v^y = max\left(\frac{4}{3\rho}, \frac{\gamma}{\rho}\right)\left(\frac{\mu_l}{Pr_l} + \frac{\mu_t}{Pr_t}\right)\frac{\Delta(S_y)^2}{\Delta V} \tag{3.31}$$

$$\lambda_v^z = max\left(\frac{4}{3\rho}, \frac{\gamma}{\rho}\right)\left(\frac{\mu_l}{Pr_l} + \frac{\mu_t}{Pr_t}\right)\frac{\Delta(S_z)^2}{\Delta V} \tag{3.32}$$

The maximum allowable CFL numbers for 4-stage and 5-stage hybrid Runge-Kutta are 2.0 and 3.6 respectively.
For steady-state computations, to accelerate the convergence rate, the time step computed in each cell is locally used. This method, used to improve the convergence is called local time stepping. On the contrary, in unsteady computations, the minimum time step calculated in computational domain is used to advance the solution in time. This is called global time stepping.

3.2 Boundary Conditions

The accurate prescription of boundary conditions is very important in numerical simulation of external flows. The difficulty comes from the nature of the problem to be solved. In external atmospheric flow simulations, the artificial numerical boundaries must be placed such that on one hand the far field flow field is correctly replicated and on the other hand by limiting the domain extent the computational cost of the simulation is reduced. At different boundaries of the computational domain, the information that is carried out by waves must be reflected or passed on depending on the nature of the boundary. Mathematically the governing time dependent Navier-Stokes equations (Eq. 2.1) are a hyperbolic propagation dominated system of equations. This character is mainly attained by the inviscid part of the equation. Therefore in this region, where the wave propagation is dominant, the Navier-Stokes equations can be reduced to Euler equations. Therefore the boundary conditions of a wide range of internal and external external flows are treated as inviscid. To model the boundaries on which the shear stress or heat conduction is dominated, along with the inviscid wave propagation boundary conditions, extra boundary conditions on velocity and temperature gradient should also be specified. But this requirement makes the boundary conditions at inlet and outlet very complex and precautions should be taken to avoid non-physical solutions in cases when viscous effects vanish in the limit. Therefore, the system of equations for inlet and exit boundaries are based on Euler equations.

The additional required boundary conditions for the simulation of external or atmospheric flow depends on the selected domain type. Specifically, there are two types of domains which are used for atmospheric and microscale wind simulations. The first less known type of domain is circular domain where one half on the domain is prescribed as inlet and second half used as outlet. In this type of the domain, the inlet and outlet planes could be adjusted such that different wind directions are simulated using the same grid. This method only requires renumbering of the computational cells at the boundary. Even though this is very helpful in terms of wind simulations and micrositing, when simulations of various wind directions are required, the use of one circular computational domains amplifies the uncertainties in the definition of inflow boundary conditions. Additionally, the grid generation task over circular domains becomes more complicated since multiblock or unstructured grids are required to efficiently cover the domain. Instead, what is mostly used in wind flow simulations is a rectangular domain which is aligned with wind directions. In this type of domain the inlet and outlet boundary conditions could be imposed normal to the surface and side boundaries which are ideally located far from the region of interest are treated as periodic or symmetric.

In the following section, the boundary conditions that are implemented in this thesis are described. The inlet and outlet boundary conditions are modeled as locally one-dimensional and normal to the boundary and they are non-reflective in terms of mathematical definition. Next, the periodic and no-slip wall boundary conditions are described which could be used for both internal and external flows.

3.2.1 Inlet Boundary Condition

The characteristic equations can be derived from the Euler equations. For subsonic flows which will be dealt in this thesis, four characteristics enter and one leaves the physical domain, Therefore, four characteristic variables are prescribed based on freestream values and one characteristic is extrapolated from the interior of the physical domain. Therefore the number of boundary conditions to be specified at the inflow is four. As a result:

$$\left[\rho - \frac{p}{\bar{c}^2}\right]_b = \left[\rho - \frac{p}{\bar{c}^2}\right]_{inlet} \tag{3.33}$$

$$[u_s]_b = [u_s]_{inlet} \tag{3.34}$$

$$[u_t]_b = [u_t]_{inlet} \tag{3.35}$$

$$\left[u_n + \frac{p}{\overline{\rho c}}\right]_b = \left[u_n + \frac{p}{\overline{\rho c}}\right]_{inlet} \tag{3.36}$$

$$\left[-u_n + \frac{p}{\overline{\rho c}}\right]_b = \left[-u_n + \frac{p}{\overline{\rho c}}\right]_{RK} \tag{3.37}$$

The subscript denoting $inlet$ corresponds to the physical boundary conditions to be specified. b corresponds to the variables to be corrected at the inlet node according to wave

propagation and RK corresponds to variables that were already calculated from the internal domain solved by Runge Kutta scheme. u_n, u_s and u_t are the velocity normal and tangential to the boundary. c is the speed of sound and $\overline{c}$ and $\overline{\rho}$ represent a reference state which is set equal to the values calculated at the boundary node by the scheme in the current implementation.
If the total pressure, temperature and flow angles at inlet are specified the required boundary conditions are complete. First the static pressure is calculated using the equation which is obtained by combining the isentropic relation and the fifth characteristic relation (Eq. 3.37). This equation is solved iteratively using a Newton-Raphson method. Once p_b is found, the freestream static pressure p_{inlet} is calculated using both Eq. 3.36 and Eq. 3.37. Using the p_{inlet} and $p_{t_{inlet}}$ the absolute velocity and static temperature are known. The velocity components at boundary are obtained using the given flow angles and two other characteristic equations 3.34 and 3.35. Finally the five primitive variables at inlet are specified as:

$$p_b = \frac{1}{2}\left(p_{inlet} + p_{RK} + \overline{\rho c}\left(u_{n inlet} - u_{nRK}\right)\right) \tag{3.38}$$

$$\rho_b = \rho_{inlet} + \frac{p_b - p_{inlet}}{\overline{c}^2} \tag{3.39}$$

$$u_{nb} = u_{n inlet} + \frac{p_{inlet} - p_b}{\overline{\rho c}} \tag{3.40}$$

$$u_{sb} = u_{s inlet} \tag{3.41}$$

$$u_{tb} = u_{t inlet} \tag{3.42}$$

3.2.2 Outlet Boundary Condition

For the outflow boundary condition, one eigenvalue becomes positive. Therefore, the number of boundary conditions to be specified at the outflow is one which corresponds to pressure. The resultant eigenvalue equations become:

$$\left[\rho - \frac{p}{\overline{c}^2}\right]_b = \left[\rho - \frac{p}{\overline{c}^2}\right]_{RK} \tag{3.43}$$

$$[u_s]_b = [u_s]_{RK} \tag{3.44}$$

$$[u_t]_b = [u_t]_{RK} \tag{3.45}$$

$$\left[u_n + \frac{p}{\overline{\overline{\rho c}}}\right]_b = \left[u_n + \frac{p}{\overline{\overline{\rho c}}}\right]_{outlet} \tag{3.46}$$

$$\left[-u_n + \frac{p}{\overline{\overline{\rho c}}}\right]_b = \left[-u_n + \frac{p}{\overline{\overline{\rho c}}}\right]_{RK} \tag{3.47}$$

where the subscript $outlet$ corresponds to physical boundary condition to be specified. In case of atmospheric flow simulations, the static pressure is a good choice since the outlet must be placed close to region where terrain and therefore flow gradients are zero and uniform distribution of pressure at outlet is expected. Once the static pressure is known, the rest of the variables are calculated as:

$$\begin{aligned}
&p_b = p_{outlet} \\
&\rho_b = \rho_{RK} + \frac{p_{outlet} - p_{RK}}{\overline{c}^2} \\
&u_{nb} = u_{nRK} + \frac{p_{out} - p_{RK}}{\overline{\rho c}} \\
&u_{sb} = u_{sRK} \\
&u_{tb} = u_{tRK}
\end{aligned} \tag{3.48}$$

3.2.3 Wall Boundary Condition

The relative velocity between the air flow and the terrain surface is zero. This condition which is called the no-slip condition is expressed for a stationary wall as:

$$\overrightarrow{V} = 0 \tag{3.49}$$

The temperature at the terrain surface or the heat flux in case of non-neutral atmosphere is specified using:

$$T = T_{wall} \tag{3.50}$$

or

$$-k\frac{\partial T}{\partial n} = q_{wall} \tag{3.51}$$

where n denotes the direction normal to the terrain surface and q_{wall} is the heat flux at the ground.

In this thesis, atmospheric stability is not considered and therefore the terrain is modeled as adiabatic. The heat transfer at the wall is assumed to be zero and therefore:

$$\frac{\partial T}{\partial n} = 0 \tag{3.52}$$

At a solid boundary with no-slip condition, the momentum equation reduces to a form that the normal pressure gradient becomes zero.

$$\frac{\partial p}{\partial n} = 0 \tag{3.53}$$

Pressure and temperature at the wall surface are calculated using Eq. 3.52 and 3.53.

3.2.4 Periodic Boundary Condition

As mentioned before, using a rectangular domain for atmospheric flow simulations requires a proper definition of the side boundary conditions, which are located far from the

region of interest to reduce the effect of the artificial boundary and retain accuracy.
In a cell vertex scheme, the periodic nodes depicted in Figure 3.1 are treated as internal nodes during the computations (Eq. 3.54) and at the end the contribution of the other side of the boundary is added (Eq. 3.55). The partial volume of both sides also needs to be added up. Ideally, since the rectangular domain is aligned with the wind direction, there should be no inflow/outflow across the periodic boundaries.

$$W_{1'} = W_1 \quad W_{2'} = W_2 \tag{3.54}$$

$$R_{1,tot} = R_1 + R_{2'} \quad R_{2,tot} = R_2 + R_{1'} \tag{3.55}$$

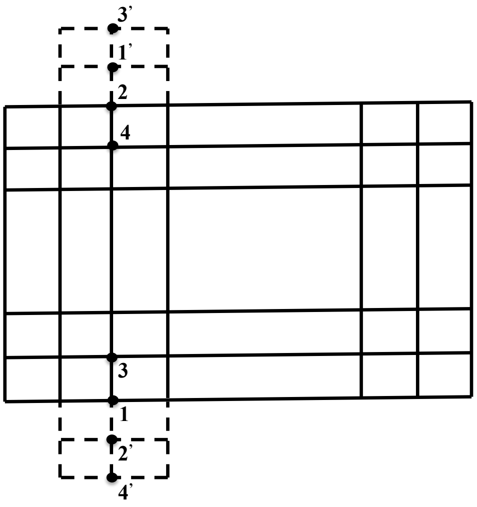

Figure 3.1: Periodic boundary condition for a cell vertex scheme in a hexahedral domain. The dummy parts are denoted as primed.

3.3 Wall Function

For simulation of atmospheric flow and microscale wind similar to many other high Reynolds number engineering problems, wall functions must be employed in order to accurately represent the flow and turbulent field close to the no-slip ground. Wall functions or high Reynolds number turbulent boundary conditions are typically used in engineering applications in order to reduce the stringent grid requirements close to the ground. In such cases, the standard treatment of the low Reynolds number turbulence models referred to as integration method are replaced by generic formulations of turbulent boundary layer profile close to the ground.

The low Reynolds turbulence models require the first node above the wall to be located in

viscous sub-layer ($y^+ < 1$) whereas using the generic wall function formulation, the accuracy is maintained with first node located in the buffer or logarithmic layer. This reduces the number of the required grid points in the boundary layer and overall mesh size as well as the cell aspect ratio near the wall. The latter greatly improves the robustness of the solver by reducing the numerical stiffness.
In wind flow simulations, in addition to very high Reynolds number of the flow ($10^7 - 10^{10}$) which makes the use of low Reynolds number turbulence models practically impossible, the use of wall function allows one to efficiently account for surface roughness which is known to have great impact on microscale wind variations.

3.3.1 Generic Wall Function Formulation

The wall function approach relies on the characteristics of turbulent boundary layers. It has been shown that under viscous flow condition, Turbulent boundary layer profiles remain the same if expressed in suitable non-dimensionalized form. This empirical formulation is called law of the wall. In turbulent flow, three regions with different velocity profiles are distinguished. the laminar sub-layer, buffer layer and logarithmic layer. The edge of logarithmic layer depends on Reynolds number and its extent grows as Reynolds number increases. For the case of atmospheric flow, the edge of boundary layer within which the law of the wall holds is up to 300 (m). The universal laws of the viscous and logarithmic layers are derived using friction velocity u_τ and laminar viscosity μ_l for non-dimensionalization. The velocity, viscosity and turbulent variables in plus units are:

$$u^+ = \frac{u}{u_\tau} \quad y^+ = \frac{y u_\tau}{\nu} \quad \mu^+ = \frac{\mu_t}{\mu} \quad k^+ = \frac{k}{{u_\tau}^2} \quad \omega^+ = \frac{\nu\omega}{{u_\tau}^2} \tag{3.56}$$

The equations of law of the wall are derived using the incompressible RANS equations and simplifying assumptions of flow over flat plate with zero pressure gradient [163]. The law of the wall for velocity, k and ω for viscous and logarithmic layers are expressed as:

$$u^+ = y^+ \quad k^+ = K{y^+}^{3.23} \quad \omega^+ = \frac{7.2}{\beta^* {y^+}^2} \tag{3.57}$$

$$u^+ = \frac{1}{\kappa} ln\left(y^+\right) + B \quad k^+ = \frac{1}{\sqrt{\beta^*}} \quad \omega^+ = \frac{\sqrt{k^+}}{{\beta^*}^{\frac{1}{4}} \kappa y^+} \tag{3.58}$$

where $K = 0.0022$, $B = 5.1$ and $\kappa = 0.41 - 0.43$.
In order to obtain a generalized wall function formulation, the constrain of the first node being located in viscous or logarithmic layer must be removed. Unfortunately, unlike viscous and logarithmic layer there is no analytical formulation available for buffer layer. A simple yet effective solution is presented by Kalitzin et al. [75] where the non-dimensionalized values of u, k and ω in buffer layer are obtained using a look-up table built based on a solution of fully resolved boundary layer obtained using the developed solver MULTI3. The

formulation obtained by a functional fit to profiles of u^+, k^+ and ω^+ must give consistent values with Eq. 3.57 and Eq. 3.58 at the edge of viscous ($y^+ = 5$) and logarithmic ($y^+ = 30$) layers respectively. The detailed deviation and formulation of u^+, k^+ and u^+ functions in buffer layer can be found in [109].

3.3.1.1 Treatment of Surface Roughness

For many applications in engineering including atmospheric flow simulations, the assumption of a fully smooth wall in description of turbulent boundary layer is quite unrealistic. Surface roughness increases the shear stress and breaks up the viscous sub-layer structure and hence alters the flow field within the boundary layer. In order to formulate the effect of surface roughness, the aerodynamic roughness height k_s is defined. The non-dimensionalized ${k_s}^+$ is called the roughness Reynolds number and indicates different regimes under which roughness affects the turbulent boundary layer profile in different mechanism. The flow is considered fully rough if ${k_s}^+ > 55-90$ which is always valid for microscale wind simulations over terrain.
In atmospheric flow the effective roughness height z_0 of a specific land cover is determined by measuring average wind speed u with height z in neutral atmospheric conditions well above the roughness elements. The surface roughness practically alters the shear at the wall and shifts the velocity profile. The logarithmic law of the wall holds on a rough surface with appropriate shift [163]:

$$u^+ = \frac{1}{\kappa} ln(z^+) + B - \frac{\Delta U}{u_\tau} \tag{3.59}$$

$$\frac{\Delta U}{u_\tau} = B - C + \frac{1}{\kappa} ln({k_s}^+) \tag{3.60}$$

with $C = 8.5$ for a fully rough surface. Using the roughness height definition of z_0 as:

$$z_0 = k_s e^{(-8.0\kappa)} \tag{3.61}$$

we obtain the logarithmic profile of velocity over rough terrain as:

$$\frac{u}{u_\tau} = \frac{1}{\kappa} ln\left(\frac{z}{z_0}\right) \tag{3.62}$$

The equivalent roughness height z_0 for different terrain type is given in [161]. The surface roughness can vary significantly over land and a transition from smooth to very rough surface may occur. This transition can significantly affect the flow field. High local pressure and velocity fluctuations can be observed in the vicinity of an abrupt roughness change. In this scenario, the assumption behind the derivation of wall function formulation and equivalent sand grain roughness definition break down and therefore larger errors in model prediction are expected. Additionally the model accuracy deteriorates as the dimensions and complexity of roughness elements increase. More advanced methodologies are presented in literature to attempt to represent the complex flow structure and turbulence behind large canopies and forest edge accurately [133] and [164]. However, roughness modeling using equivalent roughness height remains to be the most robust and effective approach in microscale wind simulations and wind farm modeling.

3.3.2 Implementation of Wall Function

There are many ways to implement wall-functions into a Navier Stokes code with classical approach but usually the main steps are as follows [75] (i) The u-velocity component requires a condition that ensures the correct shear at the wall. This can be done either by adding a source term to the momentum equation or prescribing an artificial viscosity at the wall. (ii) The turbulent kinetic energy needs to be set at the first node and (iii) ω (or ϵ) also needs to be corrected based on k (Eq. 3.58).
In the current work, the shear is corrected at the wall adjacent cell by defining an artificial viscosity at the wall. The correct value of the shear is obtained by solving Eq. 3.57 or Eq. 3.58 on smooth wall depending on the y^+ value or Eq. 3.62 for the rough terrain iteratively using a Newton-Raphson method. In the case of variable roughness, the corresponding roughness height z_0 in Eq. 3.62 is used for each node on the terrain. Then an effective viscosity is prescribed at the wall node as:

$$\mu_{eff} = \tau_w \frac{z_p}{u_p} \tag{3.63}$$

where subscript p denotes the wall adjacent node. Since the viscous fluxes are calculated at cell center in our cell vertex scheme, the laminar viscosity at wall nodes are altered such that the average laminar viscosity at wall adjacent cells satisfies the condition given in Eq. 3.63. The correct friction velocity then is used to compute the correct k and ω at the node above the wall.
Since the viscous time step is affected by the laminar viscosity (Eq. 3.30), in order to insure the stability of the scheme, the new time step needs to be calculated at the wall using modified value of viscosity at the wall adjacent cell. This greatly improves the robustness of the scheme especially when used in connection with variable roughness maps.

3.3.2.1 Validation

The implementation of wall function formulation is validated for flow over smooth and rough flat plates. In order to separate the effect of numerical error in validation of wall function with errors due to wall modeling $\delta-$grids are used (Fig. 3.2).

Figure 3.3 and 3.4 show the profiles of u^+,μ^+,k^+ and ω^+ for different y^+ values using a generalized wall function. It is observed that the values of all flow quantities agree well with the fully resolved profile and the empirical relations.
Next, the prediction of flow over fully rough flat plate is evaluated. The case of fully rough flat plate presented here is one of the cases experimentally studied by Ligrani and Moffat

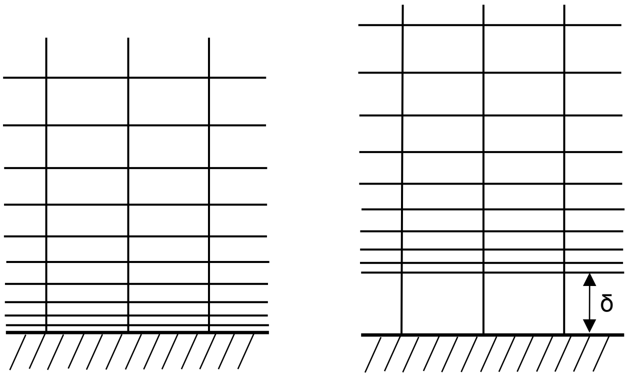

Figure 3.2: Schematic plot of fine and δ grid used for validation of wall function formulation.

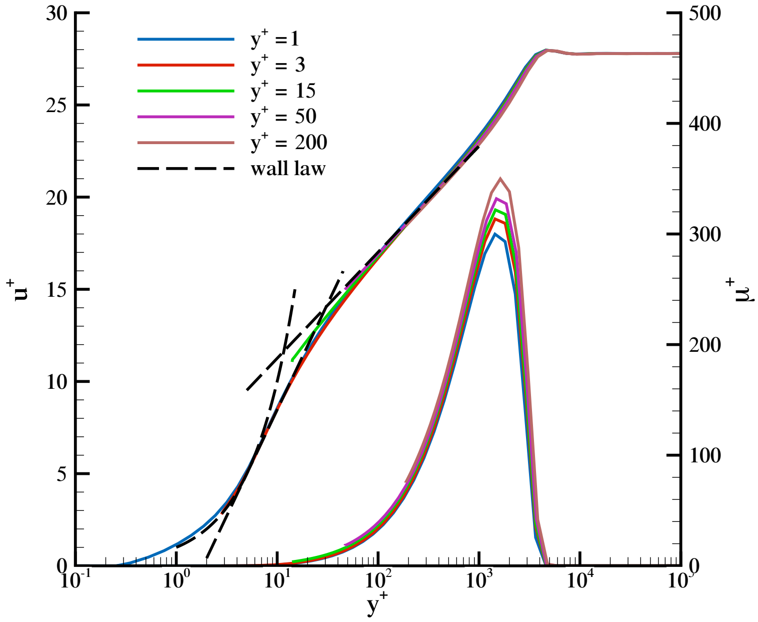

Figure 3.3: Predicted profiles of velocity and eddy viscosity obtained using wall function with different y^+ values. The results are compared with fully resolved profile and the available empirical formulations.

[86]. In the measurement set-up, the ground is covered uniformly distributed elements with equivalent aerodynamic roughness height of $k_s = 0.79mm$ $(z_0 = 2.97x10^-2mm)$. The roughness Reynolds number is $Re_k = 61.4$ which indicates a fully rough boundary layer. The free stream wind speed in the simulation 26.8 m/s, is the same as in the measurements. The extent of the computational domain is $5 \times 1m$ in the x and y directions. There are 200 grid points in the flow direction with local clustering at the leading edge of the plate and 56 grid points along $y-$axis. In order to separate the numerical error due to the grid resolution from the modeling errors arising from use of wall functions, a δ-grid is used [75]. The grid is generated such that the first node is located in logarithmic layer with $y^+ = 100$. The free stream Mach number in the simulation is 0.08 and the Reynolds number is roughly 10^7. The simulations are performed with $CFL = 3.5$ and a fourth order smoothing coefficient of

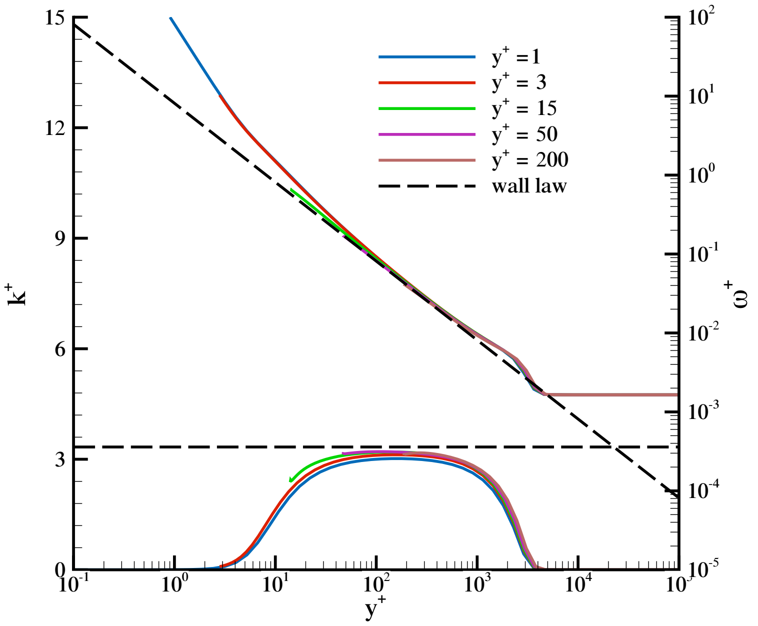

Figure 3.4: Predicted profiles of k and ω using wall function with different y^+ values.

0.003.

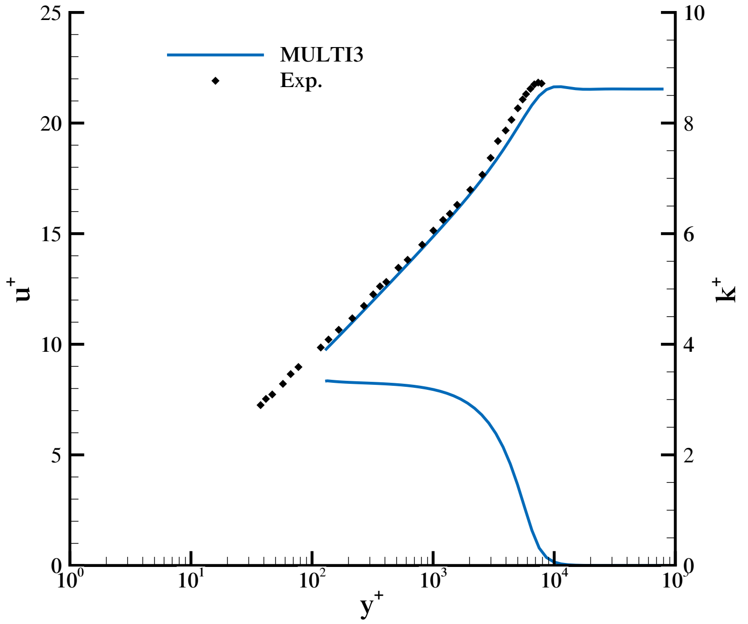

Figure 3.5: Velocity and turbulent kinetic energy profile in fully rough boundary layer.

The comparison of velocity profiles with measurements is also shown in Figure 3.5. The agreement with measurements is quite satisfactory except for the region close to the edge of the boundary layer. The predictions of skin friction are also compared with empirical relations given by Schlichting [127] and Mills [108] (Figure 3.6).

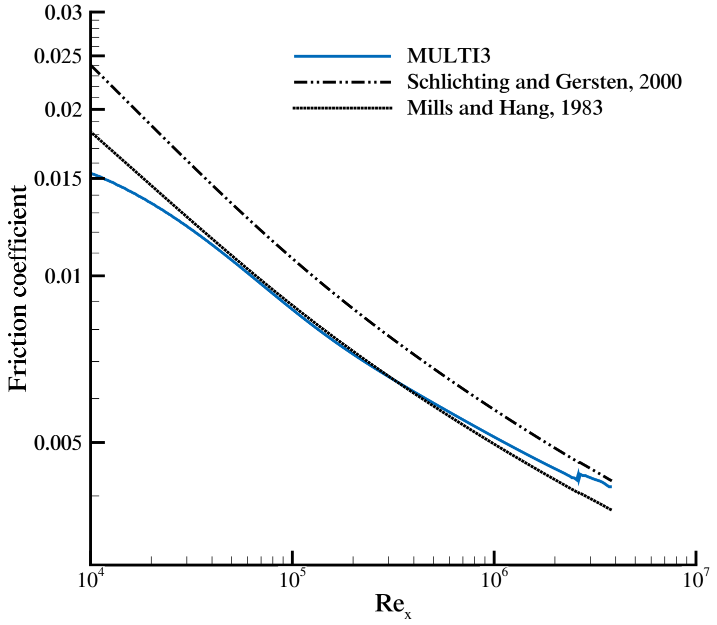

Figure 3.6: Predictions of skin friction coefficient over rough flat plate compared to empirical formulas.

More detailed validations of the wall function implemented in MULTI3 for three dimensional flows can be found in [109].

CHAPTER

4

Multigrid Method

In the numerical simulation of all fluid flows, reducing the computational time is of primary importance. In addition to parallelization of the solver using OpenMP, numerical algorithms are implemented into the solver to accelerate the convergence to steady state. The multigrid method is known to be one of the most effective strategies to dramatically reduce the solution time for range of fluid flow and solvers. In this chapter, the algorithm and implementation of the multigrid method into our in-house RANS solver MULTI3 is discussed. This is accomplished within the scope of this thesis. An optimization of the solver and multigrid solver is carried out and the best type of cycle and details of restriction and prolongation method are presented and discussed. All the test cases in this thesis are simulated using multigrid, demonstrating the effectiveness and robustness of the method. On average, compared to a single-grid solver, a multigrid solver converges to steady-state in half to one-sixth the time.

The multigrid is used to accelerate the convergence of partial differential equations. The theory was originally developed for linear set of equations and first was applied to elliptic differential equations. Since then it has been used in connection with variety of equations and schemes. Considering the compressible Navier-Stokes equations, explicit methods typically apply a kind of update to the spatially discretised equations and this step is repeated until the final solution is reached. At each time step and update cycle, the solver decreases/smooths the error of the solution but in fact the damping characteristic of the solver is different for different error frequencies. Most explicit solvers damp the high frequency errors of the solution much more rapidly and efficiently while low frequency errors seem to persist longer in the solution and delay the convergence to steady-state. Considering an error in a one-dimensional grid, the frequency of the error wave will double by omitting the every other node in the grid (Figure 4.1). In other words, a low frequency error on the fine grid appears as high frequency error on the coarse grid. Therefore, coarser grids could be used in order to accelerate the elimination and damping of the low frequency errors. However, the solution must be finally constructed on the fine grid in order to maintain the desired accuracy.

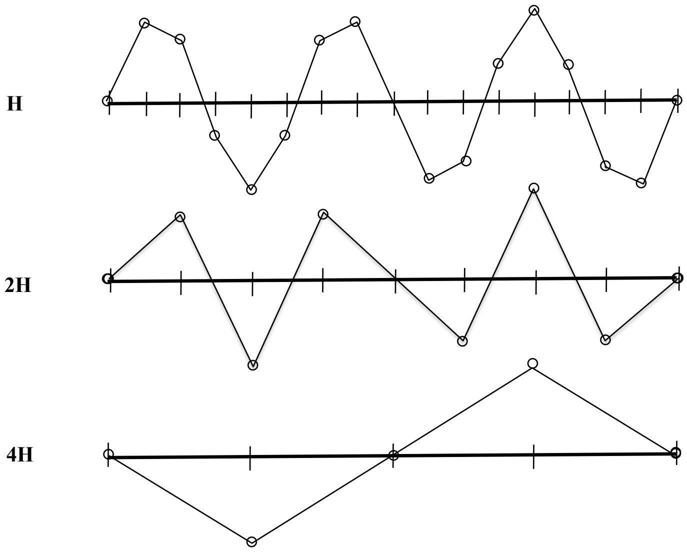

Figure 4.1: Discrete representation of a wave on three different grid level [90].

4.1 Grid Sequencing

One of the simplest algorithms for accelerating the convergence to steady state is grid sequencing in which coarse grid levels are used to generate a good initial guess for the final solution on the finest level. In order to use grid sequencing, first coarse grid levels must be constructed. It is convenient to perform this step such that the same solver and subroutines could be used in all levels. The coarse grid could be simply constructed by removing every second node on the fine grid in a cell-vertex solver. Once the coarse grid and all the required variables and arrays are constructed, the solution procedure starts with an initial guess on the coarsest level. The update algorithm is repeated until the final solution is reached. This solution is then prolonged to the next finer level and the whole procedure is repeated until the finest grid is reached. Grid sequencing will not damp out all the low frequency error modes and additional computations must be performed on the finest level to obtain the final solution, but it is very effective in accelerating the convergence by generating a good initial guess. In addition, it provides a suitable ground for the implementation of main multigrid algorithm described in the next section.

4.2 Multigrid Cycle

Multigrid approach employs coarser grid levels in order to accelerate the damping of the errors on the fine grid using one or more constructed coarse grid levels. The effectiveness of

multigrid method strongly depends on how effective the basic scheme is in damping high frequency errors.
In the literature, considerable effort has been focused on applying multigrid method to different kind of solvers. It is specifically proven to be very effective in connection with the explicit Runge-Kutta scheme proposed by Martinelli [100] described in section 3.1.3.1. Maksymiuk et al. [97] present a comparison of two schemes for solving viscous flow around airfoils. They conclude that the Runge-Kutta solver using multigrid is considerably faster than an approximate factorization implicit method without multigrid. Hulshoff [66] and Arnone [7] have solved rotor and cascade flows with a Runge-Kutta method accelerated using multigrid.
A detailed and extensive description and analysis of multigrid cycle in connection with Runge-Kutta scheme is given in [71]. This approach was used for implementation of multigrid into our in-house code MULTI3. The construction of different grid levels were described in the previous section. In order to detail the algorithm here, the simplest multigrid cycle is assumed with only two grid levels. In a multigrid cycle, transferring the information from fine to coarse grid is called restriction and transferring from coarse to fine grid is called prolongation. The detailed descriptions of restriction and prolongation methods for a cell-vertex scheme are described in the next section.
A discretized explicit time difference equation can be expressed as [31] [19]:

$$\Delta\overrightarrow{W} = \frac{\Delta t}{\Delta V}\overrightarrow{R} \tag{4.1}$$

where $\overrightarrow{W}$ is the flow variables and $\frac{\Delta t}{\Delta V}\overrightarrow{R}$ represents the right hand side of the equation where $\overrightarrow{R}$ is the residual. The algorithm below describes the simplest multigrid cycle using two grid levels:

- Form the right hand side of the differential equation

$$S_1^0 = RHS(W_1^0) \tag{4.2}$$

- Perform one (or two) full cycles of hybrid Runge-Kutta on the fine grid

$$W_1^i = ITER(W_1^0, S_1^0) \tag{4.3}$$

- Form the residual vector for the new solution

$$R_1^i = RSTRCT(W_1^i) \tag{4.4}$$

- Restrict the solution W_1^i to the coarse level

$$W_2^0 = RSTRCT(W_1^i) \tag{4.5}$$

- Restrict the residual R_1^i to the coarse level

$$R_2^r = RSTRCT(R_1^i) \tag{4.6}$$

- Form the right hand side of the equation using the restricted solution of the coarse level W_2^0

$$S_2^0 = RHS(W_2^0) \tag{4.7}$$

- Form F_2, the forcing function that will be used on coarse grid to maintain the accuracy of the fine solution

$$F_2 = R_2^r - S_2^0 \tag{4.8}$$

- Update the solution on the coarse grid using the residual of the fine level plus the forcing function

$$W_2^+ = ITER(W_2^0, S_2^0 + F_2) \tag{4.9}$$

- Prolong the difference between the new solution W_2^+ and the initial restricted solution to the fine level

$$W_1^{corr} = PRLNG(W_2^+ - W_2^0) \tag{4.10}$$

- Apply the correction to the solution on the fine grid

$$W_1^+ = W_1^i + W_1^{corr} \tag{4.11}$$

Note that in the update process, on the coarse grid, the term of forcing function term is also added to the right hand side of the equation. The algorithm is also illustrated in Figure 4.2.

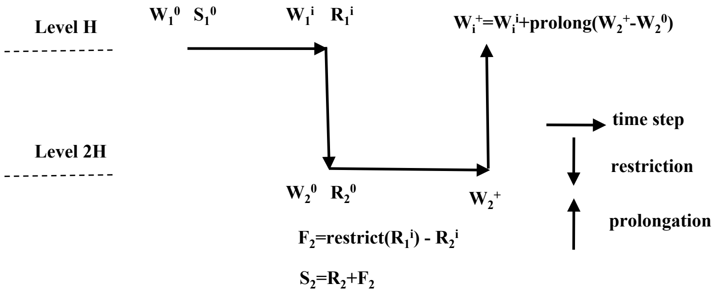

Figure 4.2: Illustration of the multigird method and algorithm for two level grid.

The basic multigrid scheme described above consists of two grid levels only. In practical applications, more coarse grid levels are used in order to accelerate the convergence further. If multiple grid levels are present, all the steps except the two last steps are repeated until the coarsest grid is reached. It is important to note that the forcing function in consecutively coarser grids is formed using the difference between the calculated residual and the "corrected" residual from the finer level which already includes the forcing function from

previous grid levels [19]. Using this algorithm ensures that the accuracy on all grid levels is controlled by the fine grid.
The multigrid cycle as described, is called V-cycle (Figure 4.3). Another typical multigrid cycle called W-cycle is also shown in figure 4.3 which proves to be very effective for transonic flows. In this thesis, a V-cycle with 3 or 4 levels is chosen as a very simple, yet effective, approach.

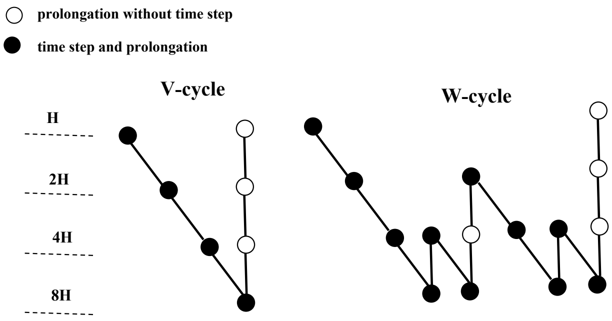
○ prolongation without time step
● time step and prolongation
V-cycle
W-cycle
H
2H
4H
8H

Figure 4.3: Illustration of two widely used multigrid cycles

4.3 Restriction and Prolongation

As shown in Figure 4.2 in a multigrid cycle, both the solution and residual vectors need to be transferred between different grid levels. The restriction and prolongation must fulfill the requirement of:

$$O_R + O_P > Q_E \tag{4.12}$$

where the O_R and O_P represent the order of accuracy of prolongation and restriction and O_E shows the order of accuracy of numerical scheme which is equal to 2 for hybrid Runge-Kutta in connection with the central discretisation used in this thesis. If this condition is not satisfied the error introduced by theses two processes will contaminate the solution on the fine grid and must be avoided.
Restriction process on a cell-vertex solver could be performed using two different approaches. Since the coarse grid is constructed by removing every other node on the fine grid, there will be a node on the fine grid for every computational point on the coarse level. Therefore restriction could be performed by a simple injection for the flow variables. For transferring the node based residuals, however, special treatment is required. In a standard central discretisation for three-dimensional flow, 8 fine grid cells construct one super cell on the

coarse grid. Therefore the weighted averaged residual on a coarse grid in 3D reads:

$$
\begin{aligned}
R^{2H}_{i,j,k} = R^{H}_{i,j,k} + \frac{1}{2} & \left(R^{H}_{i+1,j,k} + R^{H}_{i-1,j,k} + R^{H}_{i,j+1,k} + R^{H}_{i,j-1,k} + R^{H}_{i,j,k+1} + R^{H}_{i,j,k-1} \right) \\
+ \frac{1}{4} & \left(R^{H}_{i+1,j+1,k} + R^{H}_{i-1,j+1,k} + R^{H}_{i+1,j-1,k} + R^{H}_{i-1,j-1,k} \right. \\
& \quad + R^{H}_{i,j,k+1} + R^{H}_{i+1,j,k+1} + R^{H}_{i-1,j,k+1} + R^{H}_{i+1,j,k-1} \\
& \quad \left. + R^{H}_{i,j+1,k+1} + R^{H}_{i,j-1,k+1} + R^{H}_{i,j+1,k-1} + R^{H}_{i,j-1,k-1} \right) \\
+ \frac{1}{8} & \left(R^{H}_{i+1,j+1,k+1} + R^{H}_{i-1,j+1,k+1} \right. \\
& \quad + R^{H}_{i-1,j-1,k+1} + R^{H}_{i+1,j-1,k+1} \\
& \quad + R^{H}_{i+1,j+1,k-1} + R^{H}_{i-1,j+1,k-1} \\
& \quad \left. + R^{H}_{i-1,j-1,k-1} + R^{H}_{i+1,j-1,k-1} \right)
\end{aligned}
\tag{4.13}
$$

Using the above mentioned formula for cell-vertex scheme poses a challenge in the proper treatment of the boundary conditions, especially the symmetric or periodic boundary conditions described in section 3.2.4. In this thesis, to preserve the accuracy of the calculated residuals in periodic surfaces and to ensure that the residual transfer remains conservative, an alternative approach is used. During the restriction of the residuals to the fine grid, the node-based residuals are first added to obtain the cell-based residuals. In the next step, the cell-based residuals of eight cells constructing the super-cell are added to compute the overall cell-based residual on the coarse grid. Finally, this residual is distributed to the eight surrounding nodes using a similar algorithm to that employed on the fine grid. Using such algorithm, no special treatment is required at the periodic boundaries and the scheme remains fully conservative.

In successful implementations of multigrid algorithm, appropriate treatment of boundary conditions is of crucial importance. The physical boundary conditions need to be updated after every iteration, restriction or prolongation at every grid level before computing the residual or applying the correction. The residuals at nodes located at physical boundaries including wall, inlet and outlet must be set to zero before the restriction process. This is not applied to computational nodes located at periodic boundaries since they could be treated as internal nodes using the algorithm described above.

The prolongation of the solution correction must be also performed from the coarse level to fine grid levels. This is achieved using a linear interpolation. If the computational node on the fine grid has a corresponding node on the coarse level, the value is directly copied on the fine grid. Otherwise, a weighted averaged interpolation is applied as detailed (Figure 4.4). Prolongation using a distance weighted-averaged is another alternative for cell-vertex solvers.

The application of the multigrid method including the update cycle performed on coarse levels, restriction and prolongation all require additional computational time compared to the basic solver. This needs to be taken into account in order to assess the overall performance of the method. In the current implementation of a V-cycle, a 2-level multigrid has

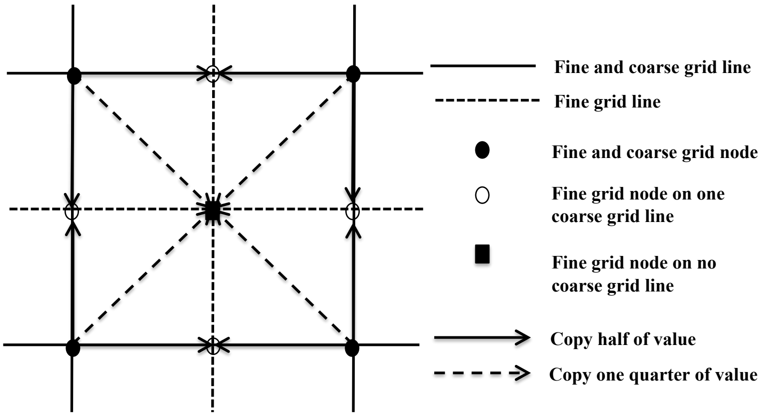
Fine and coarse grid line
Fine grid line
Fine and coarse grid node
Fine grid node on one coarse grid line
Fine grid node on no coarse grid line
Copy half of value
Copy one quarter of value

Figure 4.4: Algorithm used for linear interpolation of the correction from coarse to fine grid level during prolongation [31].

31% additional cost per iteration. This increases to 34% and 35% for the three and four levels respectively. As described in section 4.1, all the arrays in coarse grid levels are constructed such that same subroutine could be used for all levels.

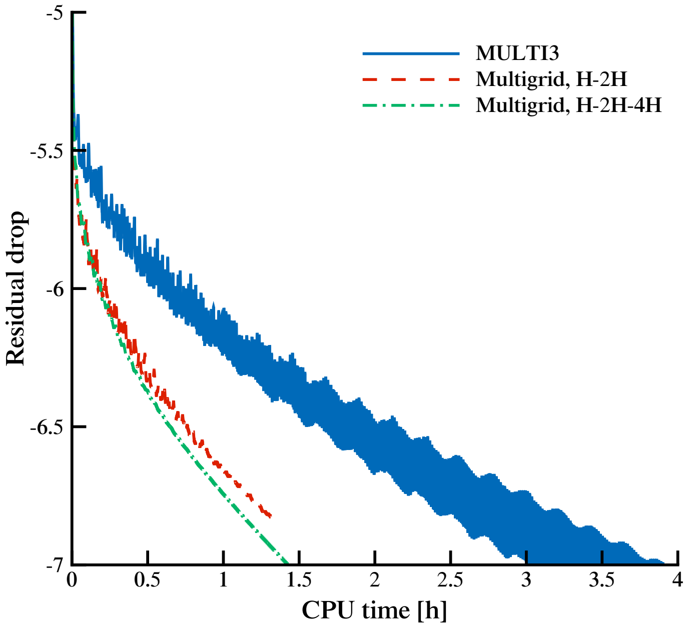

Figure 4.5: Convergence acceleration using multigrid method in connection with explicit multistage scheme.

The performance of multigrid is assessed for the case of rough flat plate described in chapter 3. Figure 4.5 shows the convergence history with 2 and 3 levels of multigrid using a V-cycle in connection with hybrid Runge-Kutta. As observed using 2 and 3 level coarse grids, results in 2x and 2.5x reduction in CPU time. The improvement of convergence rate is marginal once the fourth layer is added. Therefore in most cases in this thesis, the number of levels is restricted to 3 or 4. Figure 4.6 shows the profiles of velocity and turbulent

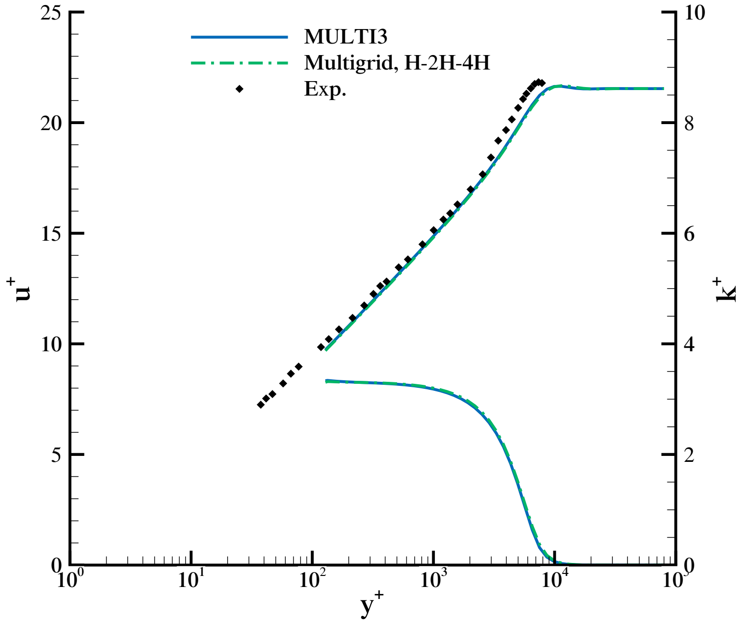

Figure 4.6: Distribution of velocity and turbulent kinetic energy in boundary layer over rough flat plate obtained with hybrid Runge-Kutta and multigird algorithm.

kinetic energy using the multigrid method. As expected, the overall accuracy of scheme is not affected by use of the coarse grid corrections.

CHAPTER

5

Local Preconditioning

In low Mach number flows, due to the disparity between the acoustic and convective wave speeds, the governing equations become stiff. By analyzing the eigenvalues of Navier-Stokes equations in a three-dimensional flow, the ratio of the largest and lowest eigenvalues in the system increases and therefore the equations become so called ill-conditioned (Eq. 5.1).

$$Condition\ \ Number = \frac{\mid \lambda_{max} \mid}{\mid \lambda_{min} \mid} = \frac{\mid V \mid + c}{\mid V \mid} \gg 1 \tag{5.1}$$

As described in section 3.1.3.2, the largest allowable time step in a compressible solver is controlled by the largest eigenvalues in the system, introducing disparity in wave propagation system. In other words, the convective waves in low Mach number flow, advance only a small fraction of grid size in every time step and as a result the convergence rate deteriorates rapidly [156]. In addition, it is shown by several authors [84] [83] [65] that the ratio of artificial smoothing terms (described in section 3.1.2) to physical fluxes does not scale properly in the region of low Mach number flow. The result of this analysis is particularly important because it shows that in low Mach number region, in addition to poor convergence behavior, the disproportional scaling of the artificial dissipation terms contaminates the solution and destroys the scheme's accuracy.

In order to overcome the inefficiency of compressible solvers for low Mach number flow, pressure based incompressible Navier-Stokes solvers are often used for simulations of low speed fluid flow. These solvers convert the governing equations to an elliptic type and eliminate the time step restriction by the acoustic waves. Even though this class of solver is widely used in engineering applications, including low speed environmental flows, but their application is restricted to globally incompressible flows with constant density. The incompressible Navier-Stokes equations are not applicable for low speed flows where the change in density due to heat release or heat transfer is significant or in many problems in energy conversion systems where low and high Mach number flows exist side by side. Therefore it is necessary to develop numerical algorithms which are applicable over all

Mach numbers. This is achieved by using a preconditioning method in connection with the compressible Navier-Stokes equations. There are several formulations and algorithms for preconditioning presented in literature. An excellent review on different schemes is given by Turkel [149].
The main issue addressed in this thesis, is the application of local preconditioning technique in connection with atmospheric flow simulations where due to large values of computed eddy viscosity compared to aerodynamic flows with medium Reynolds number, the local cell Reynolds number varies greatly in the computational domain, ranging from the highly inviscid flow in some parts of the computational domain to very low Reynolds number in viscosity dominated flows.

5.1 Preconditioning Formulation

In low Mach number flow regions, due to the disparity between the convective and acoustic eigenvalues, the convective part of the governing equations become stiff. In order to solve this problem, preconditioning, which improves the convergence behavior of the governing equations at low Mach numbers, is employed. This is achieved by multiplying the time derivatives of the Navier Stokes equations by a well-chosen preconditioning matrix P^c [19]. Therefore, the governing equations in quasi-linear Jacobian form becomes:

$$P^c \frac{\partial W}{\partial t} + A_x \frac{\partial W}{\partial x} + A_y \frac{\partial W}{\partial y} + A_z \frac{\partial W}{\partial z} = R_v(W) \tag{5.2}$$

where P^c is the preconditioning matrix in conservative variables, A_x, A_y and A_z are Jacobian matrices and R_v is the sum of the viscous flux Jacobians. It is evident that if P^c is nonsingular, the steady state solutions of the non-preconditioned and preconditioned equations are the same. Since the conservative variables are not well-suited for preconditioning analysis, we replace the conservative variables by entropy variables $Q = [dp/\rho c, du, dv, dw, dp - c2d\rho]$. Entropy variables are chosen since they result in much simplified formulations [65]. The replacement of variables and multiplication of Eq. 5.2 by $P^{e^{-1}}$ results in the system of preconditioned Navier Stokes equation (Eq. 5.3).

$$\frac{\partial Q}{\partial t} + P^{e^{-1}} \left[A_x^e \frac{\partial Q}{\partial x} + A_y^e \frac{\partial Q}{\partial y} + A_z^e \frac{\partial Q}{\partial z} \right] = P^{e^{-1}} \left[R_v(Q) \right] \tag{5.3}$$

From the range of preconditioning matrices proposed in the literature, [149] the low Mach number preconditioning of Weiss-Smith is used in this study [160]. Even though the Weiss-Smith preconditioner does not result in the most optimum condition number, the robustness of the implementation and the resultant scheme makes it a popular choice. The matrix and parameters in the Weiss-Smith preconditioning are defined in Eq. 5.4 in terms of the entropy variables.

$$P^e = \begin{bmatrix} \epsilon & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.4}$$

where ϵ is related to local Mach number as:

$$\epsilon = \begin{cases} \dfrac{M^2}{1-3M^2} & M < 0.5 \\ 1 & M \geq \end{cases}$$

It is also necessary to express the preconditioning matrix in the conservative variables since most compressible codes, including ETH's in-house code MULTI3, uses conservative variables.

$$P^c = I - (\gamma - 1)\frac{1-\epsilon}{c^2}\begin{bmatrix} \phi & -u & -v & -w & 1 \\ u\phi & -u^2 & -uv & -uw & u \\ v\phi & -uv & -v^2 & -vw & v \\ w\phi & -uw & -vw & -w^2 & w \\ h\phi & uh & vh & wh & h \end{bmatrix} \tag{5.5}$$

where Φ, and h enthalpy are defined as:

$$\phi = \frac{1}{2}\left(u^2 + v^2 + w^2\right) \quad h = E + \frac{p}{\rho} \tag{5.6}$$

The analytical formulation given in Eq. 5.5 is used to calculate the preconditioned residuals in the Runge-Kutta time integration algorithm.

5.1.1 Time Step

As described in section 3.1.3.2, the maximum allowable time step for multistage explicit methods are defined based on the absolute value of the maximum eigenvalue of the Jacobian matrix. Since preconditioning alters the eigenvalues of the governing equations, the maximum allowable time step of the scheme will be affected. In fact, preconditioning is designed such that the wave propagation speed in the system is altered. The preconditioning brings the values of acoustic and convective speeds closer to each other in order to obtain the ideal condition number for the governing equations. Therefore, it is expected that the time step in the preconditioned solver is restricted by the convective speed rather than acoustic waves. Therefore, the formulations of eigenvalues in the calculation of time step given in Eq. 5.7 are replaced with modified values of:

$$\lambda_{1,2,3} = | q_n | \quad \lambda_{4,5} = \frac{1}{2}(1+\epsilon)\, q_n + \sqrt{4c^2\epsilon\, | S | + (1-\epsilon)^2\, q_n} \tag{5.7}$$

where q_n is convective wave speed defined as:

$$q_n = un_x + vn_y + wn_z \tag{5.8}$$

As shown in Eq. 5.3, the viscous eigenvalues are also affected by the preconditioning and therefore need to be modified. However, the formulation and calculation of viscous eigenvalues are rather complicated and computationally expensive and are often left unchanged in the calculation of the time step.

5.1.2 Artificial Dissipation

As mentioned before, in addition to poor convergence behavior, the accuracy of multistage compressible solvers deteriorates in regions of low Mach number flow. One can demonstrate that by analyzing the order of the different terms in an Euler equation for low speed flow. It is shown that [70] [46] the value of entropy variables in the low Mach number limit becomes:

$$O\left(Q^e\right) = \begin{bmatrix} \frac{dp}{\rho c} \\ du \\ dv \\ dq \\ d\rho - c^2 d\rho \end{bmatrix} = \begin{bmatrix} O(M) \\ O(1) \\ O(1) \\ O(1) \\ O(1) \end{bmatrix} \tag{5.9}$$

Since the inviscid Jacobian matrices in entropy variables are given as:

$$A_x^e = \begin{bmatrix} c & u & 0 & 0 & 0 \\ u & c & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 \\ 0 & 0 & 0 & c & 0 \\ 0 & 0 & 0 & 0 & c \end{bmatrix} \quad A_y^e = \begin{bmatrix} c & 0 & v & 0 & 0 \\ 0 & c & 0 & 0 & 0 \\ v & 0 & c & 0 & 0 \\ 0 & 0 & 0 & c & 0 \\ 0 & 0 & 0 & 0 & c \end{bmatrix} \quad A_z^e = \begin{bmatrix} c & 0 & 0 & w & 0 \\ 0 & c & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 \\ w & 0 & 0 & c & 0 \\ 0 & 0 & 0 & 0 & c \end{bmatrix} \tag{5.10}$$

in the low Mach number region, $(u \ll c,\ v \ll c\ w \ll c)$ one can obtain the order of convective terms as:

$$A_x^e \frac{\partial Q}{\partial x} + A_y^e \frac{\partial Q}{\partial y} + A_z^e \frac{\partial Q}{\partial z} = \begin{bmatrix} O\left(\frac{1}{M}\right) \\ O(1) \\ O(1) \\ O(1) \\ O(1) \end{bmatrix} \tag{5.11}$$

Now knowing that the value of dissipation terms is related to the maximum eigenvalue of the spectral radius of the Jacobian matrix multiplied by second or fourth order derivatives (Eq. 3.6) the order of dissipation terms become:

$$D = \begin{bmatrix} O(1) \\ O\left(\frac{1}{M}\right) \\ O\left(\frac{1}{M}\right) \\ O\left(\frac{1}{M}\right) \\ O\left(\frac{1}{M}\right) \end{bmatrix} \tag{5.12}$$

By comparing Eq. 5.12 and Eq. 5.11, it is clear that the artificial dissipation terms do not properly scale in the low Mach number regime and therefore require special treatment. Hosseini [65] shows that introducing any low Mach number preconditioner matrix P^c eliminates the problem and scales the artificial viscosity terms back to the order of the convective fluxes given in Eq. 5.11.

To ensure the stability and improve the accuracy of the solver, the formulations of both isentropic and non-isentropic artificial viscosity are modified in the multistage scheme. The isentropic smoothing is scaled by the spectral radius of Jacobian matrix via the cell-based time step. Therefore, no direct modification of the scheme is required except that the time step used in the artificial dissipation terms is obtained using the modified eigenvalues in Eq. 5.7. For the non-isentropic case, a similar approach is used but since the spectral radius is directly used in this formulation, Eq. 5.7 is used instead of Eq. 3.15. It must be noted that since the isentropic smoothing is scaled by the local Mach number, further reduction using preconditioning sometimes results in inefficient damping of oscillations and poor convergence. This might be the reason why in some cases the solver using isentropic smoothing and preconditioning diverges using CFL number of 3.6.

5.1.3 Boundary Conditions

In most CFD solvers for simulation of compressible flows, the implementation of inlet and outlet boundary conditions is typically based on characteristic variables as described in section 3.2. In the formulation of preconditioning, the change in the eigenvalues of the governing equations also necessitates a change in the characteristic-type boundary conditions. Since the characteristic variables are defined based on speed of sound, in the region of low Mach number flow, when $u \ll c$, the characteristic boundary conditions are not well posed. Therefore, it is necessary to modify the formulation in the preconditioned equations. The characteristic equations corresponding to modified acoustic modes are given by:

$$\Delta u_n + \frac{u'_n \pm c' - u_n}{\epsilon c} \frac{1}{\rho c} \Delta p = 0 \tag{5.13}$$

where Δu_n and Δp are the differences between variables calculated to be corrected according to wave propagation and variables calculated from the internal domain using

$$u' = \frac{1}{2} u_n (1+\epsilon) \quad c' = \frac{1}{2}\sqrt{u_n^2 (1-\epsilon)^2 + 4\epsilon^2 c^2 \mid S \mid^2} \tag{5.14}$$

$$u_n = uS_x + vS_y + wS_z \quad \mid S \mid = \sqrt{S_x^2 + S_y^2 + S_z^2} \tag{5.15}$$

It is also observed that a simplified set of boundary conditions yield the same results in low Mach number flow regions [60]. In the simplified formulation, the simplified inlet boundary condition is defined as:

$$u_b = u_\infty \quad v_b = v_\infty \quad w_b = w_\infty \quad p_b = p_i \quad \rho_b = \rho_\infty \tag{5.16}$$

where all flow variables are specified from the upstream, except for the pressure which is extrapolated from the domain. The simplified outlet boundary condition is as follows:

$$u_b = u_i \quad v_b = v_i \quad w_b = w_i \quad p_b = p_\infty \quad \rho_b = \rho_i \tag{5.17}$$

5.2 Implementation

In order to implement the preconditioning into a multistage solver, first the preconditioning matrix must be expressed in conservative variables. For Weiss-Smith preconditioning this formulation is given in Eq. 5.5. To solve the preconditioning equations, first the viscous and inviscid fluxes are computed using the basic scheme. The modified artificial viscosity described in section 5.1.2 is added to the physical fluxes to form the overall residual of the preconditioning equations. As expressed in Eq. 5.3 this residual must be multiplied by matrix P^c. Since matrix P^c has a special structure and all rows could be obtained by multiplying the first row with a scalar value, the multiplication process could be simplified and written as [148]:

$$P^c \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{bmatrix} - \frac{(1-\epsilon^2)(\gamma-1)}{c^2} \left(\phi R1 - uR_2 - vR_3 - wR_4 + h\right) \begin{bmatrix} 1 \\ u \\ v \\ w \\ h \end{bmatrix} \tag{5.18}$$

Using Eq. 5.18, the multiplication of the residuals by the preconditioning matrix at every stage of the Runge-Kutta scheme could be performed quickly, avoiding additional computational cost. Once the preconditioning residual is formed, it is multiplied by the modified time step using Eq. 3.30 and corresponding stage coefficient to form the right hand side of the equation. The right hand side is subtracted from the old flow vector to obtain the solution at the new time step. Next, similar to the basic scheme, boundary conditions are updated. Note that all the boundary conditions based on characteristics must be modified in connection with preconditioning.

To integrate the multigrid method into the preconditioned solver, no major modification of the algorithm is required. The update cycle on the coarse grids are modified similar to the fine grid algorithm described above. The forcing function is formed by restricting the preconditioned residual instead on the unpreconditioned one.

The formulations of wall and turbulent boundary conditions remain unchanged in connection with preconditioning. However it is important to note that since the viscous time step is affected by the laminar viscosity, in order to insure the stability of the scheme, the new time step needs to be calculated at the wall using the modified value of viscosity at wall adjacent cell. Otherwise the modified viscosity in connection with preconditioning could result in too large time steps violating the stability criteria. This is necessary especially since the viscous eigenvalues are left unchanged. This greatly improves the robustness of the preconditioned scheme in connection with the wall function especially when applied for terrains with variable roughness. The remaining challenge in the implementation of preconditioning is the choice of preconditioning parameter ϵ which is addressed in the next section.

5.2.1 Choice of Preconditioning Parameter

As shown by Turkel [148], for efficient low Mach number preconditioning, the first component of the preconditioning matrix (ϵ in Weiss-Smith preconditioning) must be proportional to M^2. However, the preconditioning parameter initially defined by Eq. 5.1 needs to be modified in order to avoid singularities in the matrix when the Mach number approaches to zero. In general, even though the stiffness of the governing equations is substantially improved using preconditioning, the robustness of the solver is often reduced as $M \rightarrow 0$. Various numerical test cases show that the preconditioning parameter, ϵ is the most important factor that affects the robustness.

$$\epsilon = min\left(1, max\left(\epsilon_w, \sigma\frac{|\Delta p|}{\rho c^2}\right)\right) \tag{5.19}$$

A very efficient way of averaging and limiting ϵ, which addresses the singularity in the vicinity of the stagnation point (Eq. 5.19), is proposed by Darmofal [45] In Equation 5.19, ϵ_w is obtained from Eq. 5.1, Δp is the maximum pressure gradient inside the computational cell and σ is a user defined constant with typical value of 1.0. In addition to the above mentioned approach, it is also necessary to stop ϵ from reaching too small values specifically close to no-slip walls. Therefore:

$$\epsilon = min\left(1, max\left(\epsilon_w, \sigma\frac{|\Delta p|}{\rho c^2}, M_{lim}^2\right)\right) \tag{5.20}$$

where M_{lim} is specified as 10^{-5}. The preconditioning formulation described above, is suited for the efficient solution of the hyperbolic system of Euler equations in regions of low Mach number. However, in order to maintain the efficiency of the approach in viscous dominated flows where the equations have more a parabolic character, special treatment is required. In atmospheric flow simulations, the Reynolds number is typically very high ($\sim 10^7 - 10^{10}$) and therefore the characteristics of the governing equations are much like those of the Euler equations. However, due to large values of calculated eddy viscosity in the atmospheric boundary layer at higher heights above ground, or in regions of separated flow, the local cell Reynolds number often becomes too small in parts of the domain and thus the formulation used in Eq. 5.20 results in too large a time step in these regions. Therefore, an additional limiter for the preconditioning parameter is defined in the boundary layer as:

$$\epsilon = min\left(1, max\left(\epsilon_w, \sigma\frac{|\Delta p|}{\rho c^2}, M_{lim}^2, \beta M_{ref}^2\right)\right) \tag{5.21}$$

where M_{ref} represents the free stream Mach number and β is a user-defined variable with typical values of 0.1-5.0. The scaling of variable β could be defined locally depending on cell Reynolds number. The alternative approaches are also used mainly based on defining the local Mach number based on the diffusion velocity [152].
Significant additional improvement is achieved if the preconditioning parameter in every vertex or cell is related to the value of ϵ in the neighboring nodes (Eq.5.22).

$$\epsilon_{i,j,k} = max\left(\epsilon_{i+1,j,k}, \epsilon_{i-1,j,k}, \epsilon_{i,j+1,k}, \epsilon_{i,j-1,k}, \epsilon_{i,j,k+1}, \epsilon_{i,j,k-1}\right) \tag{5.22}$$

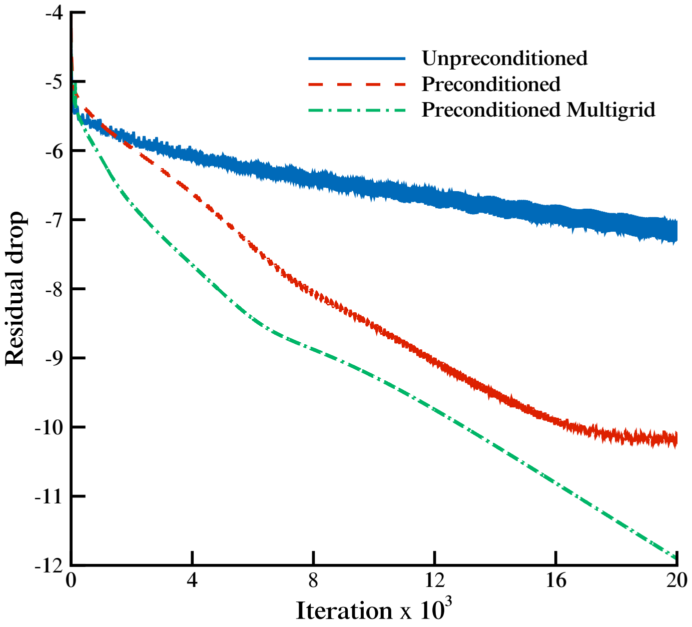

Figure 5.1: Convergence history for flow over fully rough flat plate. Preconditioned multigrid results in more than 5x speed up in convergence.

Figure 5.1 shows the convergence history (RMS residual) for the solution with and without preconditioning with the preconditioned characteristic-type inlet and outlet boundary conditions. It is observed that use of preconditioning results in more than 5x speed-up in convergence. The comparison of velocity profiles with measurements is also shown in Figure 5.2. The agreement with measurements is quite satisfactory. The predictions of skin friction are also compared with empirical relations given by Schlichting [127] and Mills [108] (Figure 5.3). The performance of preconditioning in connection with different boundary conditions for atmospheric flow simulations are investigated further in the next chapter.

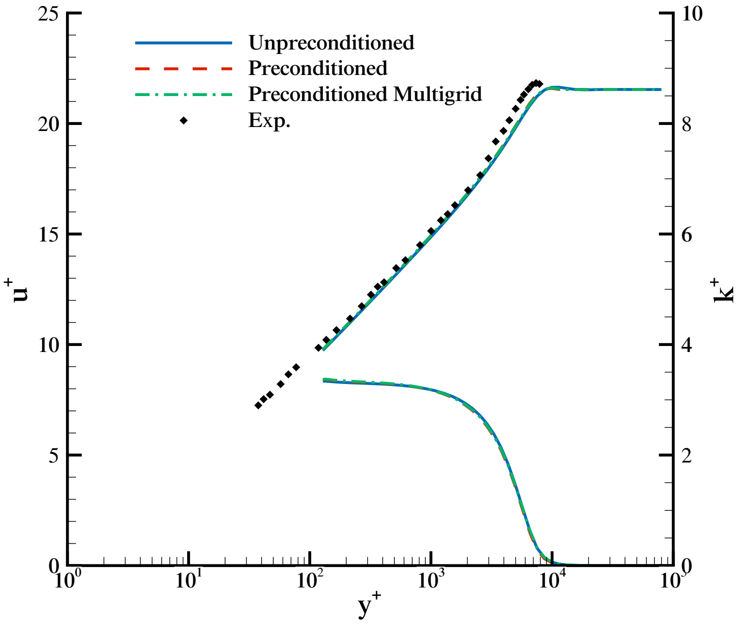

Figure 5.2: Velocity and turbulent kinetic energy profile in fully rough boundary layer obtained using preconditioning

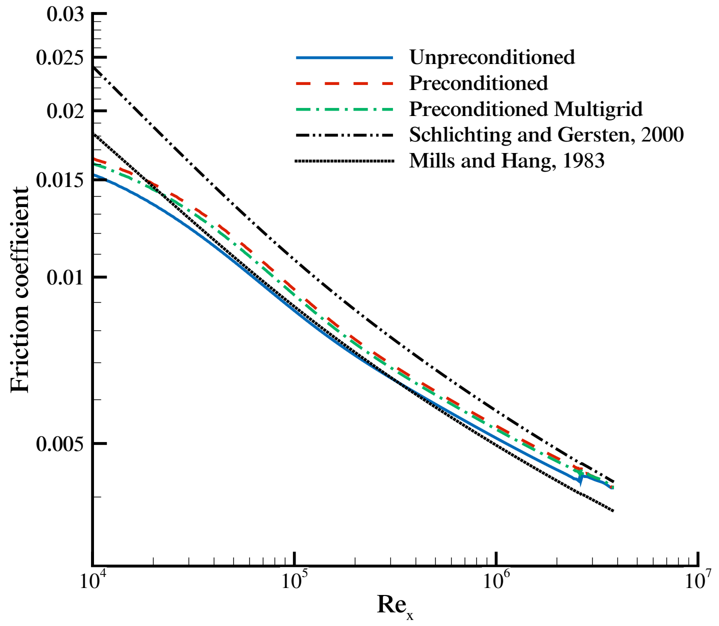

Figure 5.3: Predictions of skin friction coefficient over rough flat plate compared to empirical formulas using preconditioning Runge-Kutta.

CHAPTER

6

Atmospheric Flow over Hilly Terrain

It is well known that atmospheric boundary layer flows are significantly influenced by the underlying topography. In order to assess the wind resource for the planning and analysis of wind farms, measurements and/or the simulation of atmospheric flow over specific terrain are required [122] [51] [99]. In the case of gentle topography and low hills, analytical solutions and simple linear models that yield predictions with acceptable accuracy have been developed [138]. The key paper on linear models was presented by Jackson and Hunt [67] where they proposed a formula for fractional speed-up as a function of upstream undisturbed wind velocity in the near surface of low two-dimensional hill. Mason and Sykes [101] extended the previously developed theory to three dimensions and later its application to three dimensional atmospheric flow was demonstrated by Walmsley et al. [158] and Taylor et al. [142]. In the three-dimensional code of Walmsley, the linear formula is derived in wave-number space and the flow equations are solved using the Fourier transform. The typical formulation of wind speed-up using the linear theory is given as a function of hill height, horizontal extent of the hill, the surface cover or roughness and finally a function that represents the precise shape of the hill [35].

The successful application of linear models for real case scenarios have been reported by several authors. The results show that linear models are able to resolve flow deficit both upwind and summit of isolated hills of moderate slop. These models have very low computational cost and can be used to obtain accurate flow predictions over hills with sufficiently gentle slopes and attached flows [20].

The linear models however, due to simplified assumptions in their formulation, often fail to predict the separation region downstream of steep hills [111]. The formulation of linear models is only valid for small flow perturbations, which makes them unreliable in highly complex terrain. Some empirical corrections have been added to the models to improve their accuracy in separated flow regions, but these relations are often site-specific and do not overcome the inherent limitations of the model. Even though the poor predictions of wind flow on the lee-side of an isolated hill may not seem particularly important for wind energy applications, in real case scenarios and for the purpose of wind farm micrositing, this model inefficiency may result in major inaccuracies even in planning a farm located on moderately sloped terrain consisting of array of hills where acceleration and deceleration of flow occur over the extent of the farm. Additionally, in recent years, wind turbines are being installed in increasingly complex topographies where in addition to flow separation, high vertical wind velocities, strong shear, three-dimensional effects and unsteadiness becomes important and are not present in simplified linear formulations. Moreover, the accurate prediction of flow downstream of steep topography is important in other applications of atmospheric flow simulations such as the study of pollution dispersion or calculation of drag on man-made or natural obstacles. Therefore non-linear models started merging as a suitable alternative from the work of Raithby et al.[119].

With the increase in computational resources and major developments in computational fluid dynamics, there has been an increased focus on the application of non-linear models for wind resource assessment. Raithby et al. pioneered the use of three-dimensional CFD in atmospheric flow simulations over Askervein Hill [119] followed by Beljaars [17]. Raithby used the three-dimensional RANS equations which long has been used for simulation of environmental water flows to air flow simulations over Askervein Hill. He compared the results obtained with second order closure $k-\epsilon$ turbulence model of measurements of 1983 over the hill. Whereas the agreement of speed-up over hill was satisfactory, rather large discrepancies were observed in prediction of turbulent quantities. Raithby attributed the error to both deficiencies of turbulence models as well as the inaccuracies of the measurement. In his study however the simulations were performed using a rather coarse grid with 20×20 grid points in horizontal and 19 grids in vertical directions with first node located 0.75 m above ground level. This also might have contributed to the overall error.

Following Raithby, RANS methods with second order closure turbulence models have been used for atmospheric flows over moderately sloped and complex terrains by several authors [49] [103] [80] [29] [22]. Over the years the various aspects on microscale wind modeling has been addressed and the effect of several parameters on accuracy of predictions have been investigated. Primarily the effect of the inflow boundary condition and the stratification of the flow has been studied by Coppin et al. [35]. They found that the assessment of incoming flow shear and atmospheric stability very important in accurate prediction of wind speed at hilltop. In most studies the models were applied to one or two cases and were not extensively validated for a broad range of applications. In an attempt to evaluate the performance of RANS solver for microscale wind simulations, Kim et al. [80] used the non-linear flow model in connection with $k-\epsilon$ turbulence model to assess wind variations over four different test cases. Measurement data was available at all sites which had different levels of terrain complexity. The selection of the test cases was based on the criteria of well-defined geometry and upstream wind profile. The simulations overall showed good agreement with measurements and the comparison with linear models demonstrated the superiority of non-linear models at predictions of wind flow at lee-side of the hill.

The effect of horizontal grid resolution was later addressed by Castro et al, [29]. Castro studied the effect of vertical and horizontal grid resolution on the accuracy of prediction over

Askervein Hill. 12 different grids were used in connection with two different maps of the hill and surrounding areas. The effect of variable roughness was also studied over the hill where the roughness height was varied between 0.01 and 0.03 m as a linear function of hill height. As expected by introducing variable roughness, the velocity close to ground increased improving the agreement between measurement and predictions.

Undheim [150] also addressed the effect of grid resolution and the accuracy of the digital elevation map (DEM) of the area. He included the downstream hills around Askervein hill in his simulations performed with high resolution grid and DEM and could predict the speed-up with very good accuracy at all locations especially in the downstream wake region. He also investigated the effect of variations in wind direction in predicted wind speeds to assess the effect of small fluctuations in wind directions which could normally occur during the measurements.

While most of the RANS studies of wind flow were focused on few reliable measurements of flow over moderately sloped terrain, the real challenges and uncertainties of various modeling methods over complex terrain were not quantitatively addressed until recently. In 2007 a measurement campaign was designed to addressed the lack of high fidelity experimental data over highly complex terrain. The measurements were performed over a small hill with 12 m height and 170 m length with a circular leading edge with almost vertical slope [18]. The results of Bolund experiment were used to perform a blind comparison with wide range of numerical models including linear, second order closure and LES models. The results of the blind comparisons further indicated the superiority of RANS solvers in real complex terrain test cases [16].

More detailed modeling of turbulence is possible using large eddy simulations (LES) applied to two-dimensional hills and relatively simple flows [5] [117]. Large eddy simulations have also been used for three-dimensional cases including Askervein Hill [130]. However, the LES for atmospheric flow seems to be in early stages due to immature modeling techniques and large computational cost. The recent results of the Bolund Hill complex terrain test case, show that there are still challenges in achieving sufficiently accurate predictions using large eddy simulations [16].

In this chapter, the predictions of the preconditioned solver are assessed for the flow over three dimensional hills and the performance of the multigrid method, preconditioning and different boundary conditions on the performance and accuracy of the numerical scheme for wind predictions are assessed.

6.1 Inflow Boundary Condition

The modeling of atmospheric boundary layer in wind engineering is typically limited to the surface layer where the well-known logarithmic profiles for velocity and turbulent quantities withholds. The equations in the surface layer are derived using Monin Obukhov similarity theory assuming homogeneous and steady-state conditions. Using the logarithmic equations and neglecting the Coriolis forces, height independent equations are obtained for turbulent kinetic energy and wind direction. For atmospheric flow simulations, the variations of wind velocity, k and ω at inlet plane are specified based on Eq. 6.1.

$$u = \frac{u_\tau}{\kappa} ln\left(\frac{z}{z_0}\right) \quad k = \frac{u_\tau^2}{\sqrt{\beta^*}} \quad \omega = \frac{u_\tau}{\sqrt[4]{\beta^*}\kappa z} \tag{6.1}$$

The inlet boundary conditions for all atmospheric flow simulations in this study are obtained using Eq. 6.1. Typically the values of surface roughness are obtained over a topography using standardized values for different land covers and u_τ is obtained using the measurements of turbulent kinetic energy or velocity of undisturbed flow.

6.2 Turbulence Model Coefficients

The standard coefficients of turbulence models are often empirically determined based on wind tunnel measurements of various flows and therefore in order to employ the turbulence closure models for atmospheric flow simulations, the coefficients of these models need to be revisited. A detailed description and derivation of modified coefficients for $k-\omega$ turbulence model for atmospheric simulations can be found in [118]. The first modification is required to match the profiles of turbulent kinetic energy in the surface layer. Various measurement show the value between 0.17 and 0.18 for $\frac{u_\tau^2}{k}$ in neutral atmosphere [37] [112] [135] and therefore the value for β^* becomes:

$$\beta^* = 0.033 \tag{6.2}$$

The range of values for von-Karman constant is between 0.37 and 0.41 [52]. The value of 0.41 is used in this thesis since it better represents the wind over surface with relatively small roughness height [53]. The remaining coefficients are obtained using the simplified equations of k and ω for the decaying homogeneous isotropic turbulence and also the simplified momentum and turbulence equation for incompressible constant pressure boundary layer flow. Finally the modified coefficients for $k-\omega$ turbulence model for simulation of atmospheric flow become:

$$\alpha = 0.3706 \quad \beta = 0.0275 \quad \beta^* = 0.033 \quad \sigma = 0.5 \quad \sigma^* = 0.5 \tag{6.3}$$

6.3 Flow over Kettles Hill

Simulations are first performed over the moderately complex terrain case, Kettles Hill. The Kettles Hill experiments are comprised of field measurements of boundary-layer flow over a low hill. The data were collected over two time periods, February 1981 reported in [141] and March 1984 in [107]. The Kettles Hill site was selected after a nationwide search for a hill that was isolated in flat terrain with uniform surface cover. Further criteria for selection were that the mean slope that should be around 0.2, terrain micro structures or man-made obstacles were absent, and that wind speeds and directions were constant. Kettles Hill is located approximately 8 km to the east of Pincher Creek in Alberta. The height of the hill is approximately 100 meters and the elevation is about half of the hill height 600 meters to the west. The wind in this area is predominantly from the west and the main axis of the hill lies along this west-east line (Fig. 6.1).

As shown in Figure 6.1, the coordinate system is chosen to be aligned with the hill. The x-axis goes from true west to east and the y-axis from south to north. The center of the coordinate system was placed at the hilltop. To the west (negative x-axis), the hill matches the condition of being relatively isolated with a flat and uniform fetch. In this direction the terrain has no change in elevation for about 18 km. During the first two measurement runs in 1984, the ground was covered in large parts (90% and 60%) by snow. A following measurement run was then conducted with a snow cover of 10%. For all the following measurements, the surface on the hill was covered partially by short grass and mud. Despite these non-ideal conditions an experimental estimation of the surface roughness between 0.006 and 0.01 m was made. The surface roughness of 1 cm is used for the simulations. The digital elevation map of the area was used to create the computational grid for wind direction of 255^o. The resolution of the grid is 20 m in the refined zone around the area where the measurements were made. The extent of the computational domain is $8 \times 8km$. The first cell height is 0.5 m with a total number of 71 grid points to cover the domain height of 5000 m. The inlet boundary condition is specified based on the logarithmic profile with friction velocity of 0.5 m/s in order to duplicate the measurements of undisturbed wind of the run $5b$ of 1981.

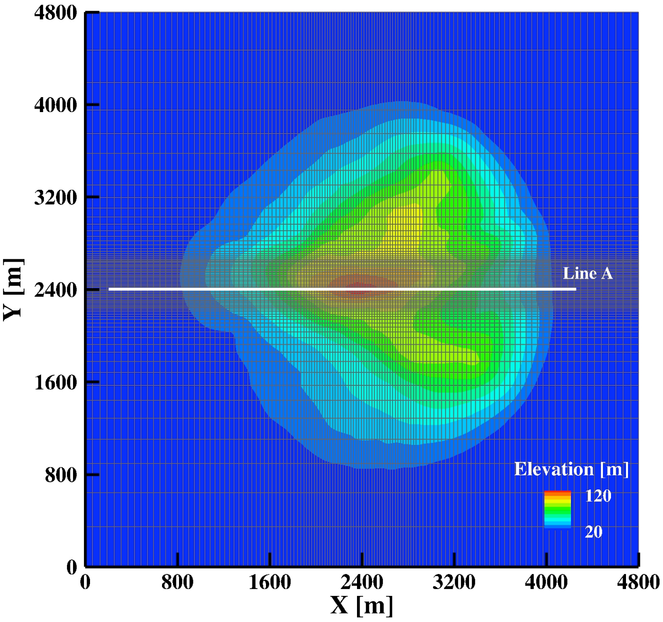

Figure 6.1: Computational grid superimposed on digital elevation of Kettles Hill. Line A goes through hilltop and along which measurements were performed.

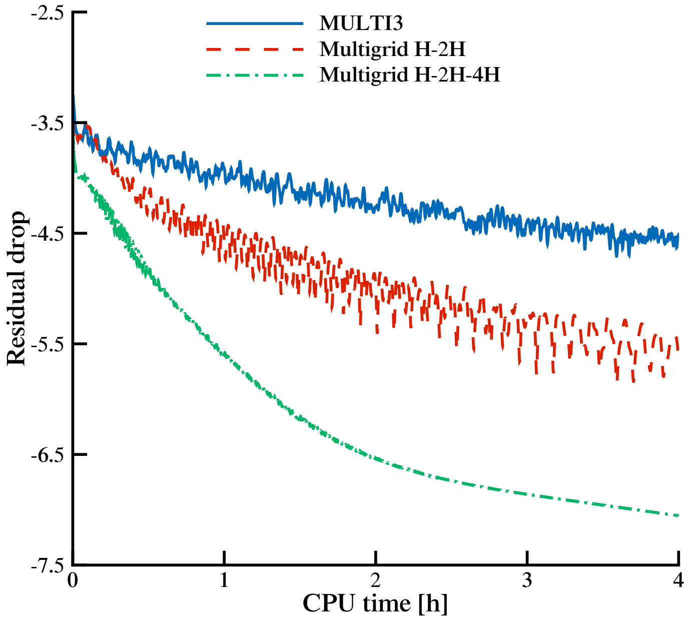

Figure 6.2: Convergence history for flow over Kettles Hill. $4x$ *and* $6x$ *speed-up in CPU time is obtained using 2 and 3 level V-cycle multigrid.*

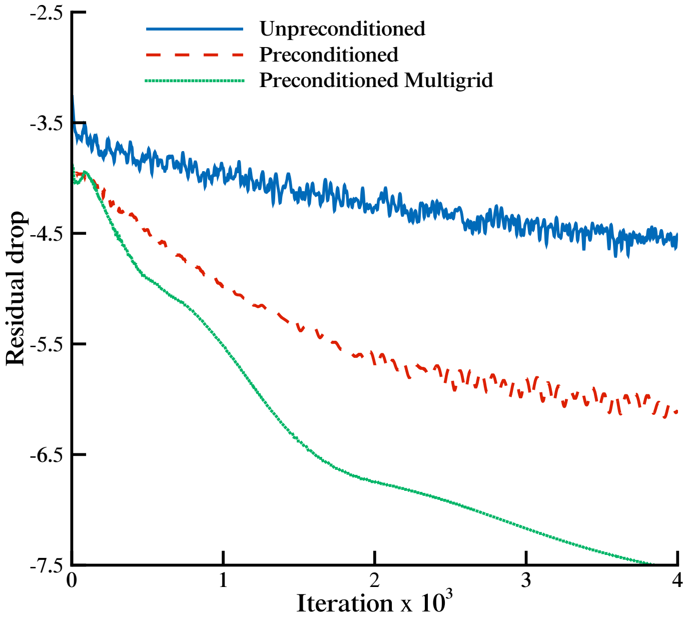

Figure 6.3: Convergence history for flow over Kettles Hill. $11x$ *speed-up in CPU time is obtained using preconditioned multigrid to reach residual of* 10^{-7}.

The simulations are first performed using the unpreconditioned hybrid Runge-Kutta in connection with different number of cycles of multigrid method. Figure 6.2 shows the convergence history of simulations performed over Kettles Hill with 2 and 3 levels of coarsening in the $V-$cycle described in chapter 4. Considering the computational overhead of the cycle, $4x$ and $6x$ speed-up is obtained respectively using the multigrid approach. Additional coarsening has marginal effect on the speed-up and therefore the number of cycles used for most of the simulations is limited to 3.

Figure 6.3 shows the convergence history from the non-preconditioned and preconditioned schemes together with the results obtained using preconditioned multigrid with two level coarsening. The computational overhead is 3% for preconditioning and 34% for preconditioned multigrid with two level coarsening. Accounting for the computational cost of the algorithm, there is $11x$ speed-up in computational time to reach the residual value of 10^{-7} using preconditioned multigrid.

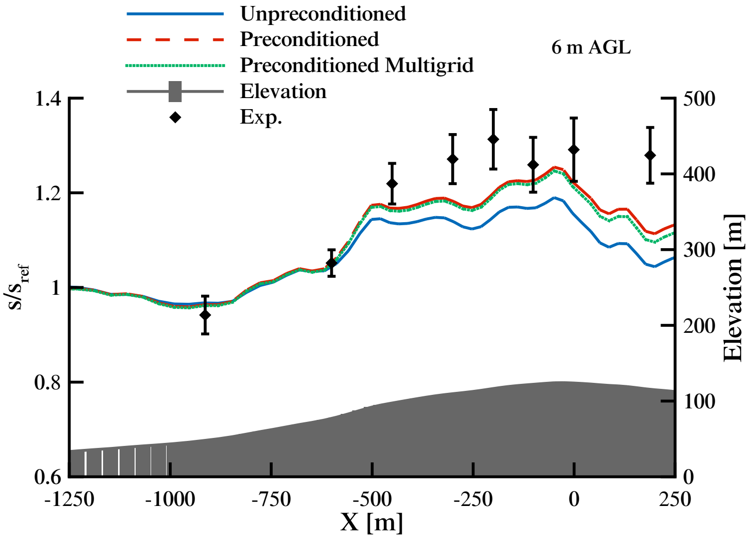

Figure 6.4: Prediction of wind flow over Kettles Hill compared to measurements. Predictions are improved at hilltop using preconditioning.

Figure 6.4 and 6.5 show the comparison of normalized wind speed along the hill for all three cases. The bars on the plot show the recorded variations of the wind flow during the measurements. Numerical results show good qualitative agreement with measurements, however the quantitative agreement at hilltop is relatively poor. As expected, the predictions at both heights are improved using the preconditioning.

6.4 Flow over Askervein Hill

The simulations are next performed over Askervein Hill with moderately complex topography. The hill is located in the Isle of Uist in Scotland. A detailed survey on the measurements

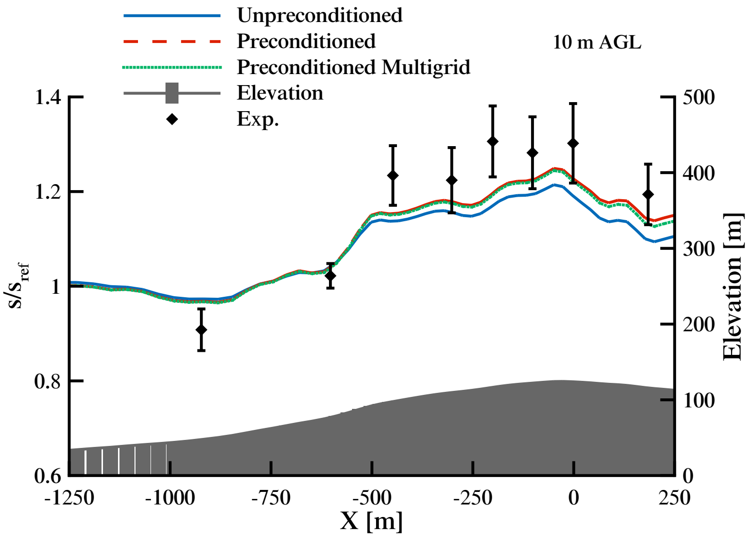

Figure 6.5: Predictions of wind flow over Kettles Hill at 10 m above ground level.

is reported in [140]. The detailed measurements over this hill have been used by various researchers for evaluation of numerical models [29] [150]. The digital elevation map of the isolated hill with resolution of 23 m is used for grid generation. The dimension of the computational domain is 6 × 6 km. The grid resolution in the clustered region around the hill is 23 m in x and y directions and the distance of the first node above ground is 0.5 m. These dimensions result in a total of 1 million computational nodes. Figure 6.6 shows the Cartesian grid superimposed on the digital elevation map of the hill. The roughness height is 0.03 m and is constant over the entire hill.

The inlet boundary condition for velocity and turbulent quantities are specified using the value of 0.65 for friction velocity. This results in approximately 9 m/s wind velocity at 10 m above ground level. The simulations are performed using both the characteristic and simplified inlet/outlet boundary conditions.

The effect of preconditioning on the convergence rate of the solution over Askervein Hill is shown in Figure 6.7. It is observed that the preconditioned solver converged to the final solution in 15000 iterations (2 hours on single Opteron 8356 CPU). This is reduced to 1 hour using the preconditioned multigrid with two level coarsening. The results also show that the preconditioning has similar performance in terms of convergence acceleration in connection with both the characteristic and simplified boundary conditions that are described in section 5.1.3.

The predicted speed-ups along lines A-A, AA-AA compared to experiment are shown in Figures 9.13 and 9.14. The speed-up (ΔS) is defined based on the undisturbed velocity upstream of the hill at the reference station using Eq. 6.4.

$$\text{speed-up} = \frac{u(z) - u_{ref}(z)}{u_{ref}(z)} \tag{6.4}$$

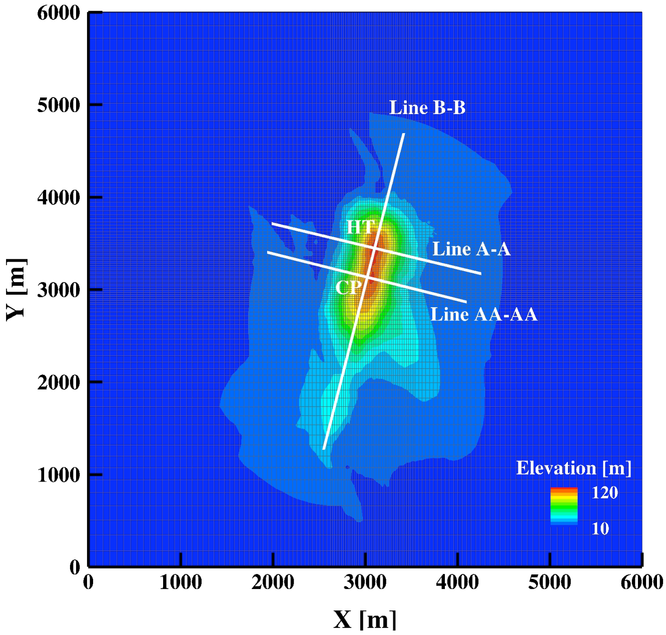

Figure 6.6: Computational grid superimposed on the digital elevation map of Askervein hill.

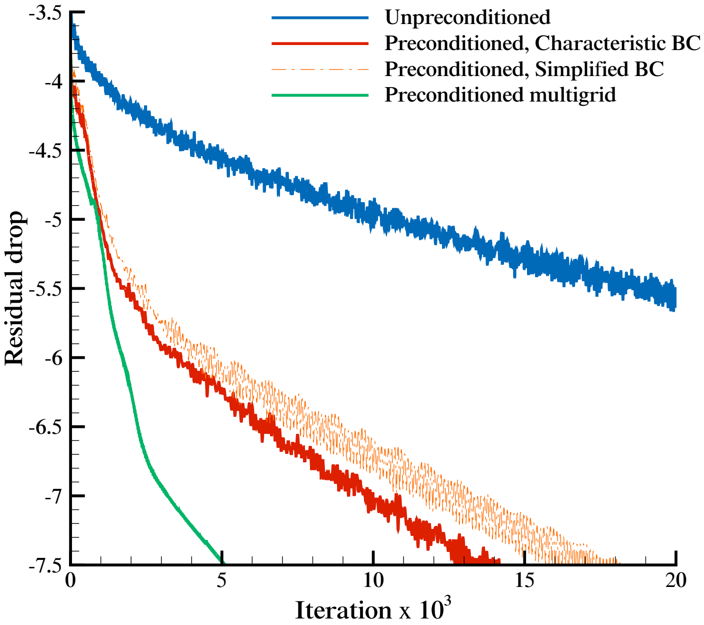

Figure 6.7: Convergence history for flow over Askervein Hill using different type of boundary condition and acceleration methods.

The predictions along line A-A and AA-AA with the basic scheme underestimate the speed-up at hill top by 14% and 6% along both lines. The results are generally in good agreement with experiment. The agreement is slightly improved at hilltop using the preconditioned solver and the error is 8% and 1% respectively. However along line A-A, the wind speeds are overestimated at the leeward slop of the hill using preconditioned solver. It is also observed

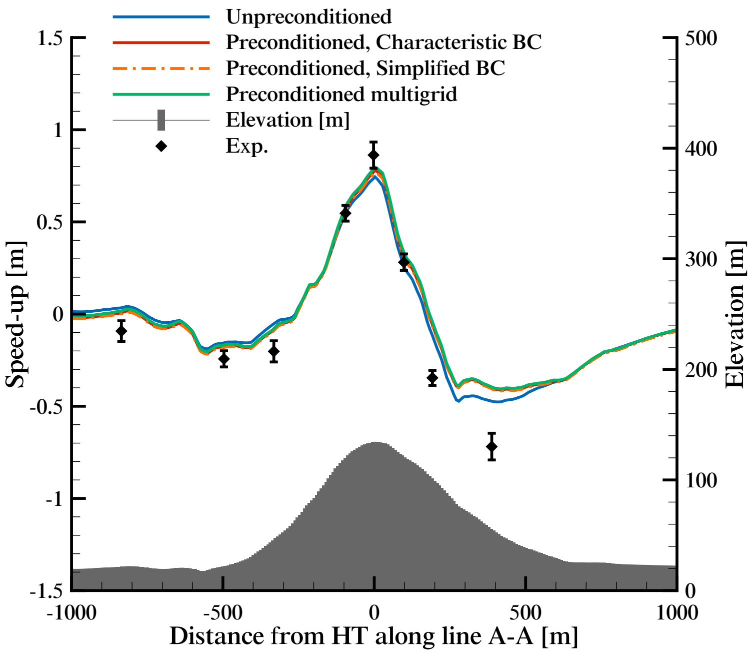

Figure 6.8: Predictions over Askervein Hill compared to experiment along line A-A. Good agreement is obtained at hilltop using preconditioning.

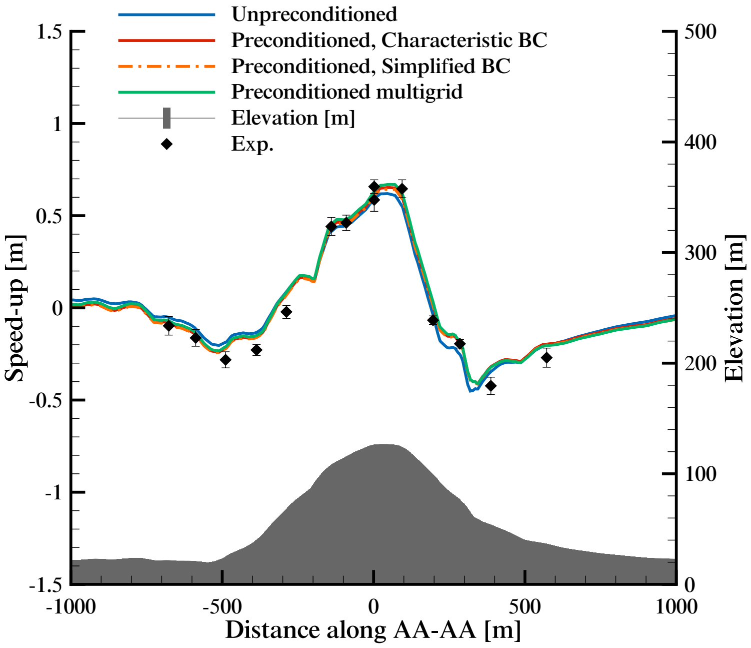

Figure 6.9: Predictions over Askervein Hill compared to experiment along line AA-AA. Preconditioning improves the convergence behavior and the accuracy of the solver.

that simplifying the characteristic boundary condition affects the results neither up- nor downstream of the hill. It must be noted however, the absolute value of wind is slightly different at the reference station at 10 m above ground (9.4 m/s with characteristic boundary condition compared to 9.24 m/s using the simplified formula).

Similar to the current results, the majority of models tend to under predict the speed-up at hilltop at 10 m AGL. Different explanations are suggested for this discrepancy. The non-uniform surface roughness seems to be the most important reason addressed in the literature. Flow separation is also observed in some of the flow models but not all. It is evident that higher grid resolution, variable roughness definition and transient solutions improves the accuracy in capturing the separation zone. However this is not captured in current study. Transient simulations performed over Askervein Hill using ANSYS CFX considering the above-mentioned criteria did not show any evidence of a separation zone or unsteadiness [145]. This suggests that inclusion of the downstream hills and topography alters the details of the flow in the deceleration zone which might be lost in the simulations of an isolated hill, but several authors such as Castro et al. [29] only focus on the isolated hill.

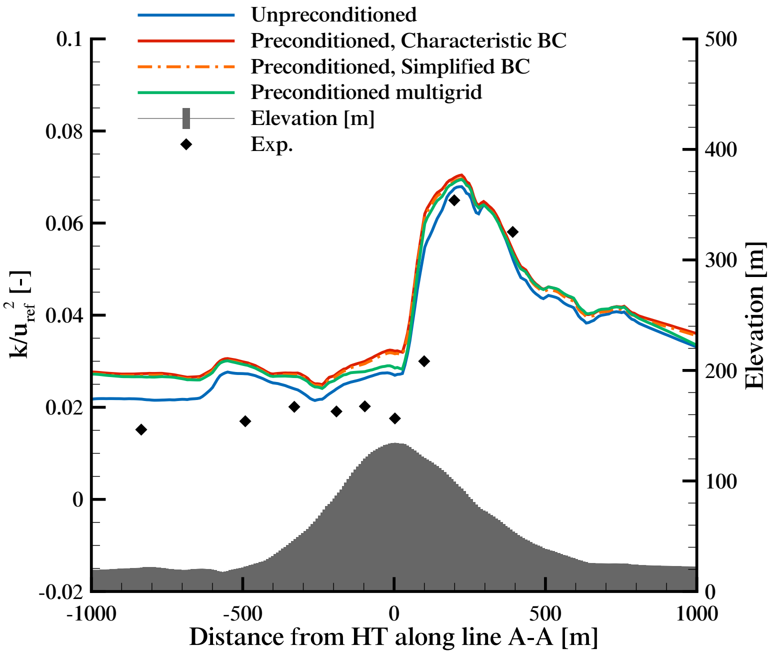

Figure 6.10: Predictions of turbulent kinetic energy at 10 m above ground level over Askervein Hill along line A-A compared to measurements.

The predictions of turbulence kinetic energy is also compared with the measurement in Figure 6.10 and 6.11. The turbulent kinetic energy is slightly over estimated at the upstream of the hill which [to the author's knowledge] is also observed in most of the RANS simulations over Askervein Hill. However, the agreement at hilltop is very good with 7% and 8% error at location x=200 m and x=390 m respectively. The difference in prediction of k using preconditioned and unpreconditioned solvers are mainly due to the change in scaling of artificial viscosity which can have significant effect on the shear and turbulent production terms in the vicinity of the wall and within the wake downstream of the hill.

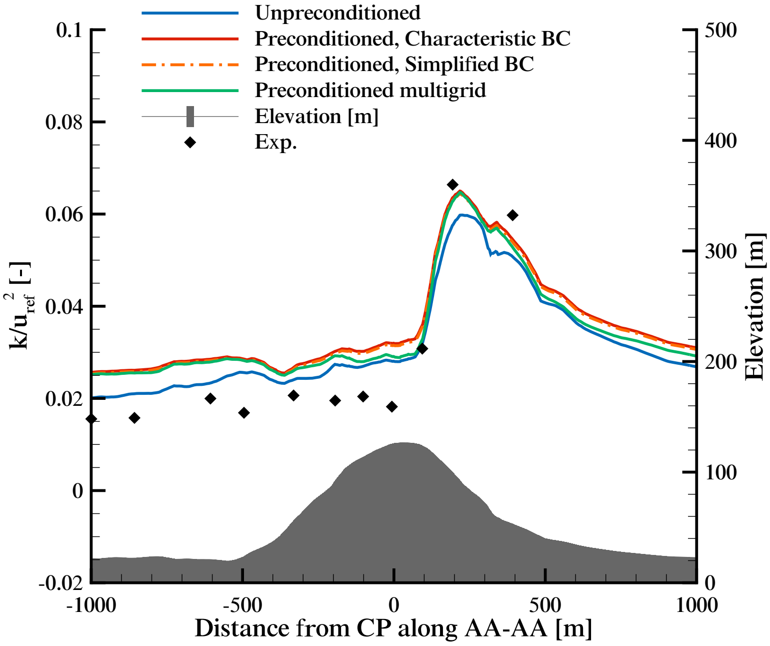

Figure 6.11: Predictions of turbulent kinetic energy at 10 m above ground level over Askervein Hill along line AA-AA compared to measurements.

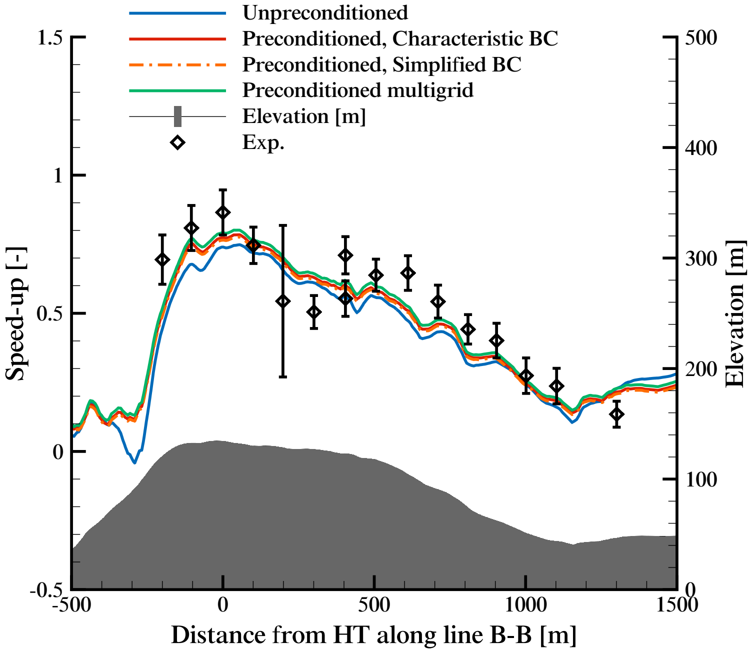

Figure 6.12: Predictions of wind speed at 10 m above ground level over Askervein Hill along line B-B compared to measurements.

The effect of topography on the predicted and measured speed-up and turbulent kinetic energy along line B-B is also shown in Figure 9.15 and 6.13. Unfortunately no data is available for turbulent parameters along this line however the quantitative agreement of speed-

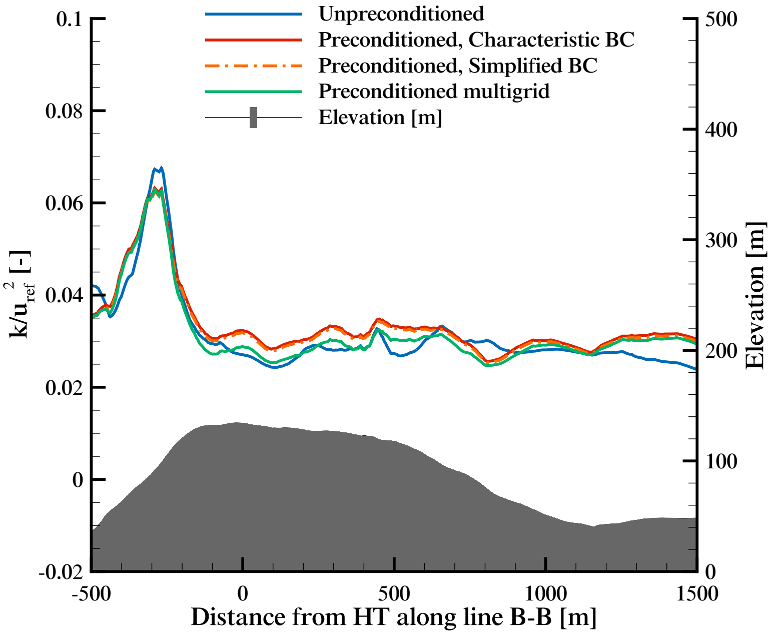

Figure 6.13: Predictions of turbulent kinetic energy at 10 m above ground level over Askervein Hill along line B-B compared to measurements.

up with measurements is satisfactory.

6.5 Flow over Bolund Hill

Finally simulation results are compared for atmospheric flow over the complex terrain test case Bolund Hill. Bolund is a 12 m high, 130 m long and a 75 m wide coastal hill located at Roskilde Fjord Denmark. The geometrical shape of the hill which is shown in Figure 6.14, includes a vertical escarpment which makes the flow prediction challenging due to the presence of the turbulent shear flows, separation zones and flow detachment. Additionally, sharp changes in surface roughness and the wind passing from water to grass well represent the typical characteristics of a complex terrain. The site and layout of Bolund provides well-defined boundary conditions, in particular for the inlet flow that is always a challenge in field experiments. The measurements were performed during a three month period in 2007 and 2008. Nine masts collected the mean flow and turbulent measurements upstream and over the hill. Measurements from instrumentation at Bolund that include sonic, cup anemometers and LIDAR are described in [18].

The simulations are performed for wind directions of 270^o and 239^o. The Figure 6.15 shows the computational grid superimposed on the digital elevation map of the terrain. The inlet boundary conditions are imposed at the inlet plane using a friction velocity of 0.4 based on the measurements at the reference station. The variable roughness map shown in Figure

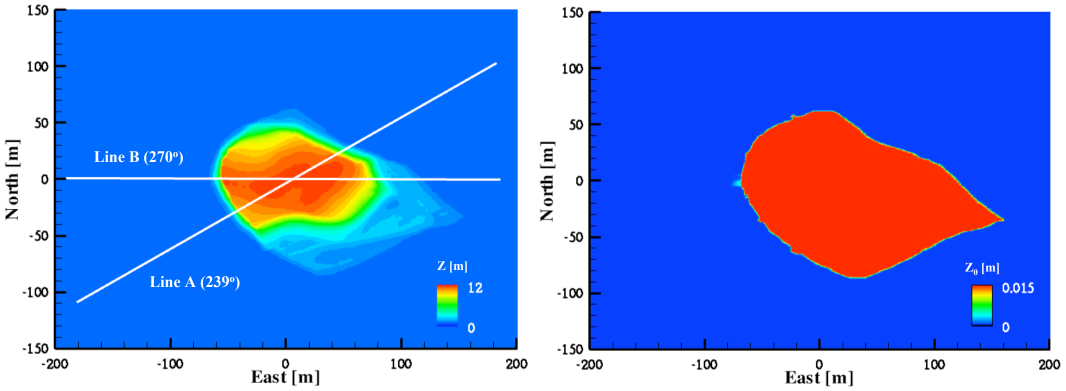

Figure 6.14: Elevation (left) and roughness (right) maps for Bolund Hill.

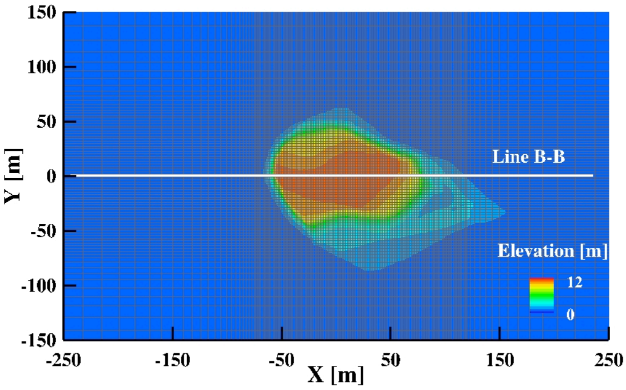

Figure 6.15: Computational grid superimposed on digital elevation map of Bolund Hill.

6.14 is used for the simulations.

Figure 6.16 shows the convergence history for the flow over Bolund Hill for wind direction of 270^o using unpreconditioned solver using both the characteristic and simplified boundary conditions. The simulations with the preconditioning technique are shown in the same figure. The Bolund Hill is a challenging test case for preconditioning due to presence of a stagnation point at the leading edge of the hill and also a separation region downstream of the hill. The simulations are all performed using CFL=2.0 and a smoothing coefficient of 0.003. The convergence behavior with preconditioned solver is much improved. It is also observed that simplifying the inlet and outlet characteristic boundary conditions does not influence the convergence behavior significantly.

Figure 6.17 and 6.18 show the normalized wind speed along line B-B shown in Figure 6.14 at 2 and 5 m above ground level. The wind speeds are normalized using the velocity predicted at mast M_0 which is located upstream of the hill and records the undisturbed wind for direction 270^o. It is observed that at 2 m above ground level the difference in normalized wind for the two type of boundary conditions is negligible. Unlike the two previous cases, it seems that preconditioned solver does not result in an improved agreement with measurement at 2 m above ground level. The unpreconditioned solver predicts the speed-up

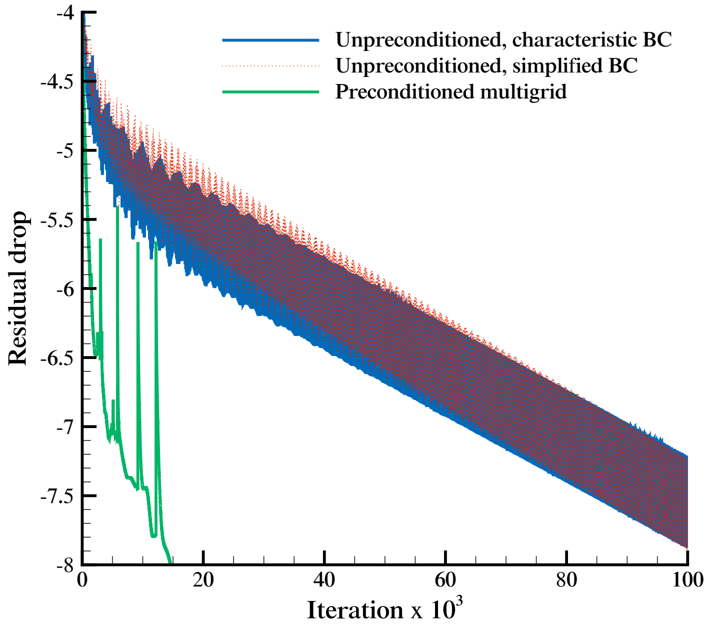

Figure 6.16: Computational history over Bolund Hill. The use of simplified boundary condition does not affect the convergence behavior or the results. The preconditioning accelerates the convergence significantly.

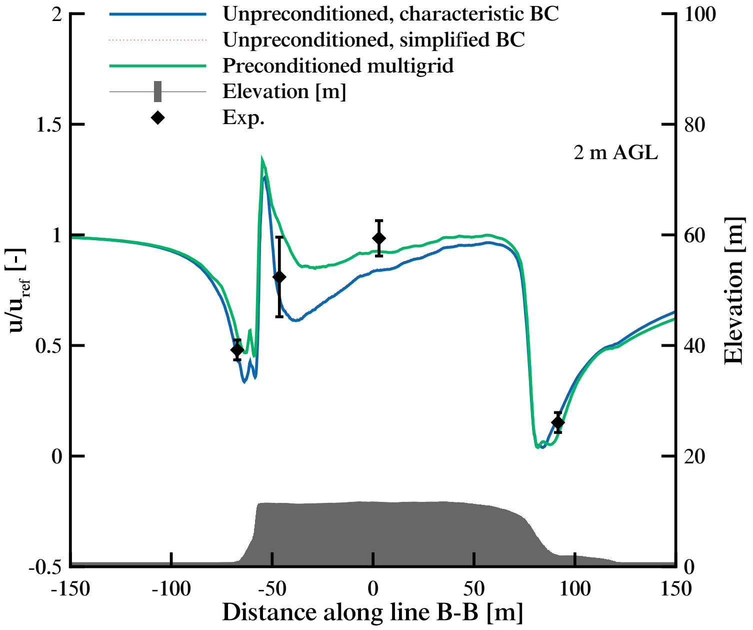

Figure 6.17: Normalized wind speed over Bolund Hill at 2 m AGL.

at x=-75 quite accurately. In general the size of separation region has reduced using preconditioning. The effect is also visible at 5 m above ground level. At the downstream of the hill however at 2 and 5 m above ground level both unpreconditioned and preconditioned formulations yield similar results.

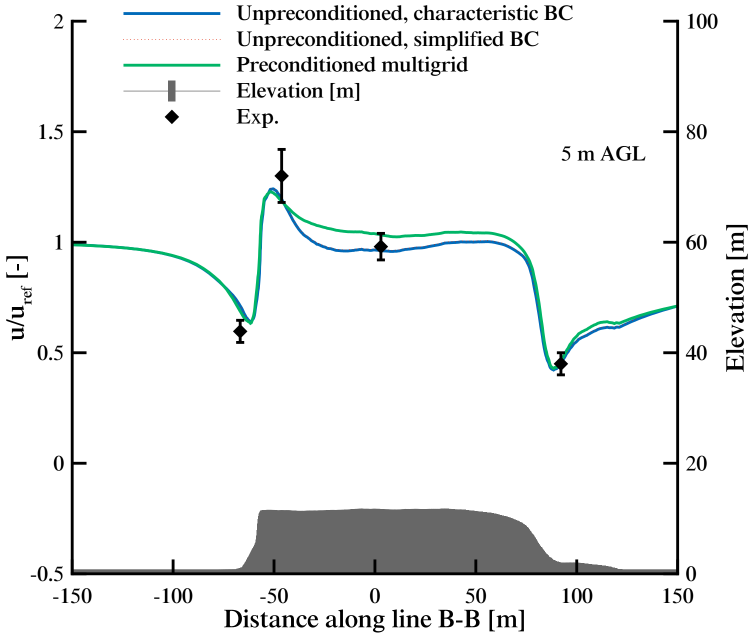

Figure 6.18: Normalized wind speed over Bolund hill at 5 m AGL.

The predictions of turbulent kinetic energy are also compared to measurements along line

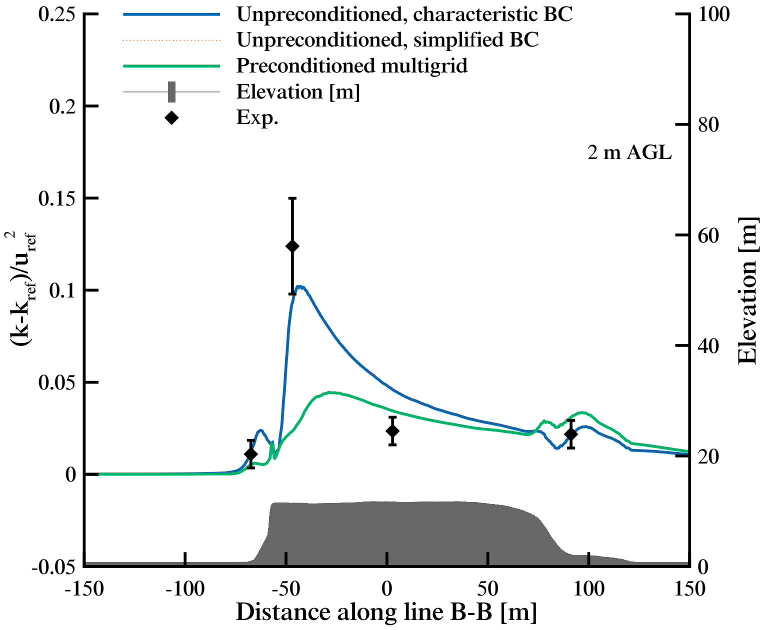

Figure 6.19: Normalized turbulent kinetic energy over Bolund Hill at 2 m AGL.

B-B at 2 and 5 m above ground level in Figure 6.19 and 6.19. In this case, due to the reduced size of the predicted separation zone, the peak of the turbulent kinetic energy at x=-75 m is underestimated using the preconditioned solver. This could be due to the effect of artificial dissipation on the production of turbulent kinetic energy. The effect of artificial dissipa-

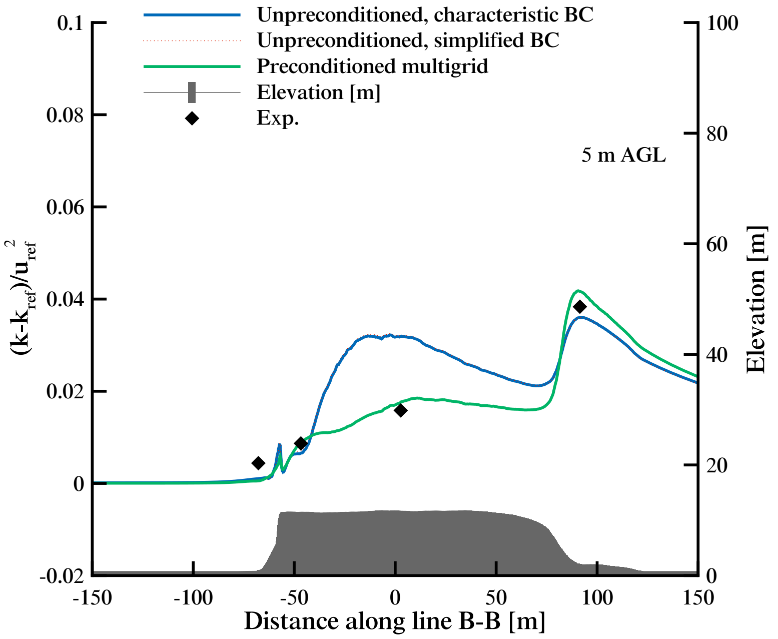

Figure 6.20: Normalized turbulent kinetic energy over Bolund hill at 5 m AGL.

tion on the turbulent flow prediction can be profound [15] especially near the wall using wall functions in connection with coarse grids [109] [75]. Numerical experimentation on flat plate shows that the value of the turbulent kinetic energy could be over estimated at the first node above ground by $20-40\%$. In addition, in the wake flow the effect of excessive artificial dissipation and coupling between mean flow and turbulent quantities can result in an underestimation of the production or decay of the turbulent kinetic energy. Therefore even though the agreement at 2 m above ground with measurement is better predicted with the unpreconditioned scheme, that could be due to error compensation and coupling of the numerical errors of the basic scheme with the modeling errors of the turbulence closure. At 5 m above ground level, the agreement between measurements and simulations is much improved both at the center of the hill and also downstream.

Figure 6.21 and 6.22 show the turbulent length scale at 2 and 5 m above ground level for wind direction 270^o with rather similar trends to the wind speed variations. The values for freestream turbulent length scale is estimated as 0.35 m using the unpreconditioned code compared to 0.5 m of the preconditioned Runge-Kutta at 2 m above ground level.

Simulations are also performed for the direction of 239^o when wind passes across the steep slopes upstream and downstream of the hill along line A-A. Similar to the case 270^o the wind blows from sea to land and therefore the incoming profile is estimated using the friction velocity of $0.4 m/s$ and roughness height of 0.0003 m. The wind speed-up along line A-A is shown in Figures 6.23 and 6.24 using the preconditioned code. The average error between the measurements and simulations at all points is 16% and 9% at 2 and 5 m above ground level respectively.

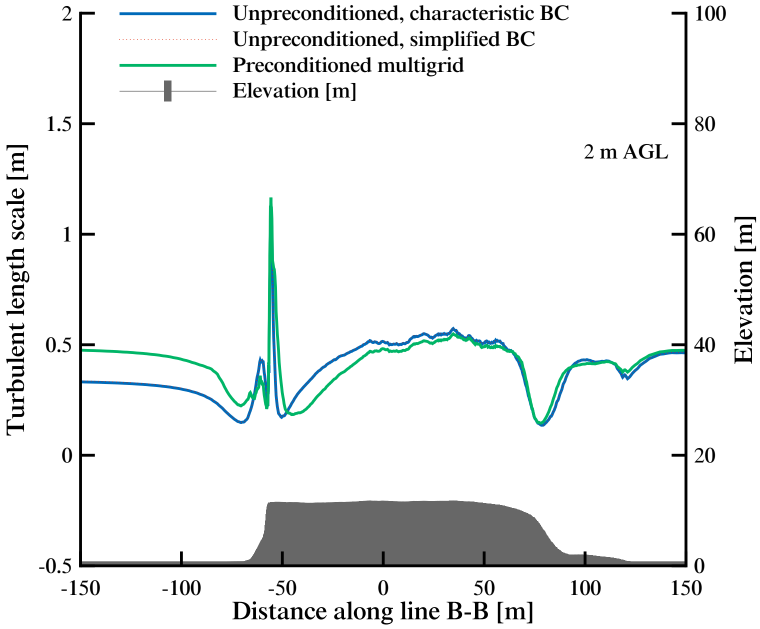

Figure 6.21: Normalized turbulent length scale over Bolund Hill at 2 m AGL.

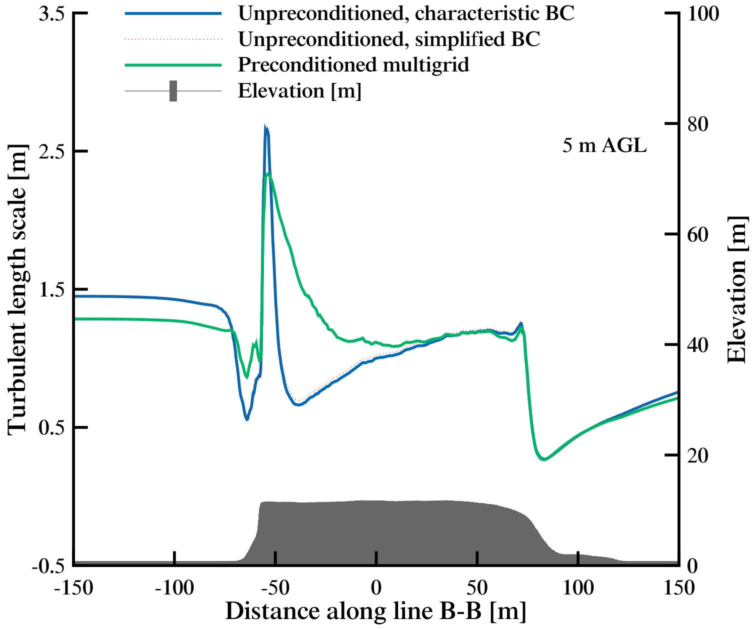

Figure 6.22: Normalized turbulent length scale over Bolund hill at 5 m AGL.

6.6 Summary

The performance of the preconditioned multistage method was assessed for the flow over Kettles Hill, Askervein Hill and Bolund Hill. In general, the robustness and accuracy of the method was demonstrated for the high Reynolds number low Mach number atmospheric flow with stagnation points and separated flow regions. The use of preconditioned multi-

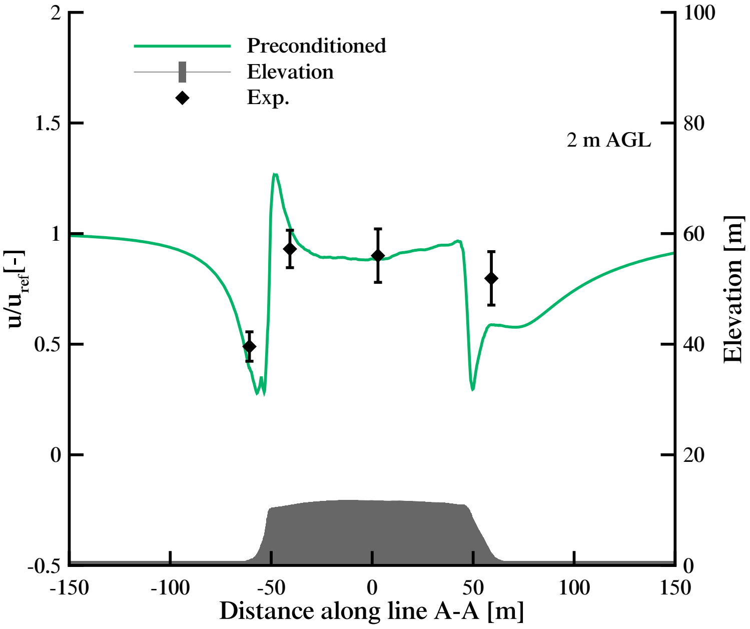

Figure 6.23: Normalized wind speed over Bolund Hill at 2 m AGL for wind direction 239^o.

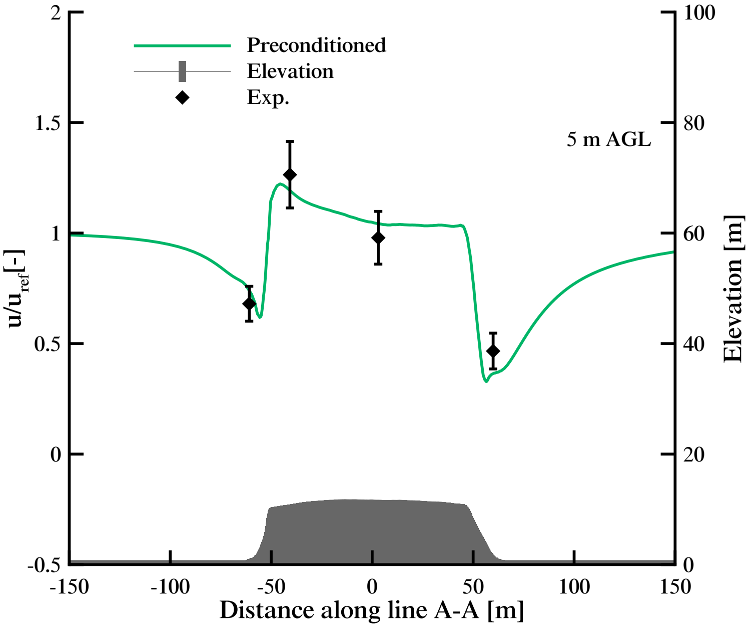

Figure 6.24: Normalized wind speed over Bolund Hill at 2 m AGL for wind direction 239^o.

grid resulted in 9-12x speed-up to decrease the residual to 10^{-7}. The performance of the simplified inlet/outlet boundary conditions was also assessed against the characteristic boundary conditions used in connection with compressible solvers. No difference was observed in the results, neither in terms of convergence nor in terms of accuracy by replacing the preconditioned boundary conditioned by the simplified formulations and inlet and outlet of the domain.

The accuracy of the prediction of speed-up with measurements was good compared to measurements over the three hills. The average error between measurements and simulations at all points in prediction of speed-up was 5% and 40% over Kettles Hill and Askervein Hill at 10 m above ground level and 8.5% over Bolund Hill at 5 m above ground level. In all cases the use of preconditioning resulted in an improvement in accuracy of predictions except for velocity and turbulent kinetic energy predictions over Bolund Hill at 2 m above ground level.

CHAPTER

7

Immersed Wind Turbine Model

In this chapter the formulation of a newly developed immersed wind turbine model which stems from previous works in the modeling of film cooling jets in turbomachinery, is discussed and the detailed implementation in connection with preconditioned multigrid is described. The immersed wind turbine model was developed in order to reduce the grid requirements for wind turbine wake flow simulations using a RANS solver. This facilitate the path for use of Computational Fluid Dynamics in micrositing.

7.1 Review of Wake Models

Wind turbines operating in the wakes of upwind wind turbines may have 30 – 40% power losses compared to the upwind turbines and fatigue loads up to 80% larger than the upstream turbines [124]. The power loss reduces the expected revenue of the wind farm and the increased loads reduce the expected life span of the downwind wind turbines. Hence, in recent decades there has been a significant effort to develop models to simulate wind turbine wakes. The simplest models use empirical or quasi-analytical expressions, to estimate the downstream evolution of the wake. The first attempt to model the flow behind wind turbine was reported by Lissaman [87] where he tried to assess the array efficiency within arrays of turbines. Several analytical and empirical models have since been developed to assess the velocity, extent of the wake [63] [88] [39] [93] [95] [77] and turbulence intensity behind a single or cluster of turbines [50] [63] [41] [40] [82] . Some studies tried to extend the models to account for environmental parameters such as ground effect or atmospheric stability [94]. Analytical and empirical models are still widely used by industry for prediction of the wake losses in an initial estimation of the energy yield of a wind farm that is under development. However, due to the inherent simplifying assumptions, the estimates often have high uncertainties.

The second group of widely used wake models are kinematic models which are based on self-similar velocity profiles that are obtained from experimental or theoretical work on co-flowing jets. Kinematic models usually neglect the expansion region of the wake immediately downstream of the rotor and provide different formulations describing the wake flow in near, transitional and far wake region [43]. Different kinds of profiles are used for estimation of the velocity in each region. The two most commonly used profiles are top-hat and Gaussian profiles [77] [155]. Kinematic models despite their simplicity, often provide acceptable accuracy when used in connection with properly chosen parameters [48] [4] [154]. However, taking into account the effect of ground and topography is not quite feasible using such models. More detailed reviews of analytical and kinematic models can be found in [43].

The last and most complex type of wake models, developed and used in recent decades, are called field models. Field models are typically three-dimensional models that calculate the magnitude of the flow at every point of the flow field. Field models require larger computational resources, but provide a more detailed description of the processes governing the evolution of the wake. The simplest models solve the linearized momentum equations in the main flow direction using a constant eddy viscosity [129]. The first numerical model for solving the turbine operating in atmospheric boundary layer using more sophisticated turbulent closure schemes was developed by Taylor [139]. Since then several field models using $k-\epsilon$ turbulence closure have been developed and used for single or multiple wake simulations [3] [157] [42].

The necessity for use of field models rises when more complex inflow or ground features are present [43]. It is evident from the prior literature that the interaction of topography with the wake and wind flow has not been treated satisfactorily [153]. With the increase in available computational resources and further developments in microscale wind predictions, there has been an increased interest in using three-dimensional field models, which solve the Reynolds-Averaged Navier Stokes (RANS) equations in connection with a turbulence model. This approach enables the wind flow and turbine wake to be simulated simultaneously and allows one to (i) account for the change in inflow wind due to terrain effects, (ii) assess the effect of the elevated turbulence on the wake evolution and (iii) investigate the interaction of the wake with the adverse or favorable pressure gradients caused by the local terrain. As the microscale wind pattern over a specific topography is affected by both the long extent of the wake and the blockage effect of the wind turbines, the placement of turbines based only on a simulated microscale wind field may not result in an optimised micrositing of wind turbines in a wind farm.

Several wind farms in complex terrain have been studied such as Castello Ranch in Altamont Pass [131], Alta Mesa [113] and in Greece on the Samos Island [61] and farm in complex terrain reported in [116]. In these studies the focus has been on loads, structure, near wake regions, far wake and the array loss estimation. The simulation of wind turbine wakes located in complex terrain is also reported by several authors [6] [30] [115] [14] [116]. In these studies, the turbines have been modeled using either an actuator disk or an actuator line approach. In both approaches, the turbine is modeled by imposing the aerodynamics loads on the computational grids either over a disk representing the rotor in the actuator disk approach or along lines representing the blades in the actuator line approach [134]. The load distributions are often obtained using Blade Element Momentum theory. However, in order to achieve accurate simulations of the flow field, a correction of the load distributions is often required. Using the actuator line approach, unsteady simulations must be performed which results in higher computational cost. The application of actuator line model in connection with Large Eddy simulations reported by several authors [165] [146] [91] [32] provides very unique insight to the flow and turbulent field in wind farms. However, due to the large computational cost, their application for micrositing purposes is not yet feasible. However application of actuator disk model in connection with RANS for prediction of wakes in complex terrain shows promising results [116].

In the current study, the simulations of single and multiple wind turbine wakes is performed using a computationally efficient immersed turbine model which is designed to combine the approaches of kinematic and field models. The model provides an alternative to actuator disk model. The representation of wind turbines using an immersed body was inspired by the similar works for modeling of film cooling in turbomachinery [25] [26] [27] [28]. This new wind turbine model can be embedded in a RANS solver with grid requirements comparable to those used in microscale wind flow simulations. The model allows for the simulation of multiple wake interactions and the effect of topography on the evolution of the far wake. Furthermore, using the immersed turbine model, a single computational grid can be used to evaluate different arrangements of turbines within a wind farm. The current approach facilitates the path to routinely using CFD to study wake interactions and the optimized micrositing of wind turbines within wind farms.

In this chapter, first the immersed wind turbine model is described. Then the model is evaluated against several test cases of stand-alone turbines in wind tunnel and field experiments and multiple turbines in wind farms and the results are compared to available measurements.

7.2 Model Formulation

The immersed turbine model is developed and incorporated into the in-house RANS solver with the $k-\omega$ turbulence model. The unsteady wind flow past a wind turbine produces a broad range of flow structures downstream of the turbine. The nature of these structures depends on the characteristics of the inflow wind, the turbine geometry and the design and operating conditions. In this work, the modeling approach is to anchor the near-wake region using analytical models, full CFD simulations and available experimental data. This is achieved by introducing an immersed body around the wind turbine in the main RANS computational domain [25]. Burdet and Abhari [25] reported the successful use of the same approach for modeling of film cooling in turbomachinery application. In this study, in order to avid resolving small scale flow structures of the film cooling jet near the hole, the jet flow was represented by an immersed body (Figure 7.1). Similar approach is adopted here for modeling of wind turbines. The immersed body represents both the blockage and the momentum extraction of the turbine without requiring a detailed solution of the boundary layers of the blades and the rotor plane. Furthermore, unlike models which operate based on the introduction of the body forces at rotor plane and therefore require a grid refinement at the location of rotor, introducing the immersed body, eliminates the need for such refinement at the rotor plane to some degree as discussed later in detail. The shape of the immersed body is specified as an axisymmetric streamtube around the rotor. Figure 7.1 shows a schematic of such a streamtube superposed on the Cartesian grid.

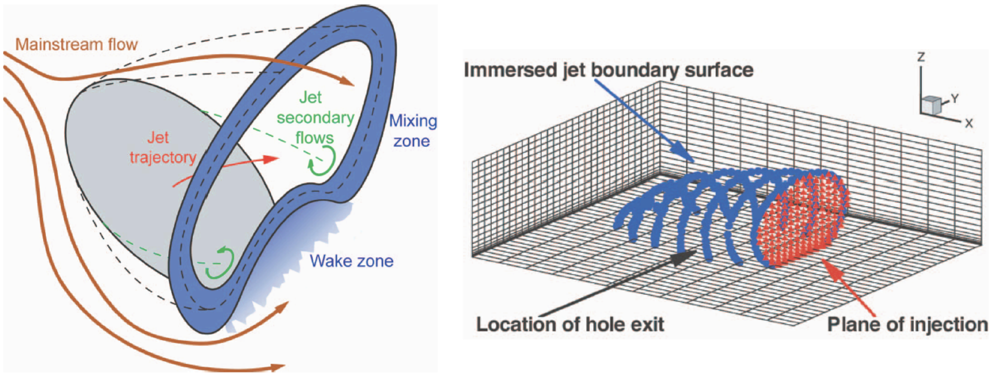

Figure 7.1: Schematics of near-hole macroflow features where the coolant jet boundary surface is immersed in the computational mesh [25].

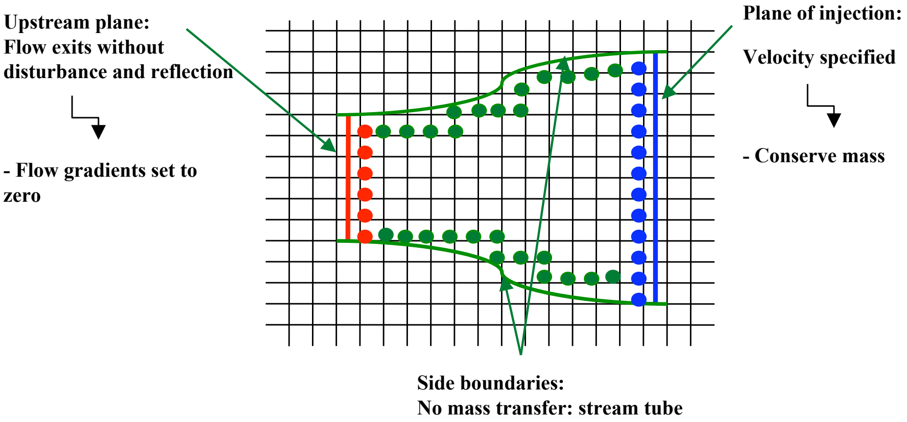
Upstream plane:
Flow exits without disturbance and reflection
- Flow gradients set to zero
Plane of injection:
Velocity specified
- Conserve mass
Side boundaries:
No mass transfer: stream tube

Figure 7.2: Schematic of the immersed wind turbine model superposed on the Cartesian grid. The immersed body is a streamtube around the rotor plane of a wind turbine. The predicted near-wake velocity field is mapped at the outlet plane (Plane O) onto the RANS computational domain. The far wake region is resolved on the computational grid by the RANS solver.

The inlet boundary of the immersed body (plane I) is located upstream, where the undisturbed flow approaches the rotor plane. The boundary condition at this plane is defined such that wind flow enters the body without disturbance. The side boundaries of the immersed body are specified to be a streamtube, and thus there is no mass transfer across them. The outlet boundary (plane O) is located downstream of the turbine at the end of the inviscid expansion of the wake.

The geometry of the immersed body is dictated by the turbine dimensions and the operating point. The radiuses of the inlet and outlet planes are defined using the rotor diameter and are related to the power or thrust coefficient (C_p or C_T) using Blade Element Momentum Theory. First the induction factor is determined using the thrust coefficient (Eq. 7.1) The velocity field at plane O is defined using Eq. 7.2 as:

$$C_T = \begin{cases} 4a(1-a) & \text{if } a < 0.4 \\ 0.89 - \frac{0.20-(a-0.143)^2}{0.643} & \text{if } a \geq 0.4 \end{cases} \tag{7.1}$$

$$u(y,z) = u_{ave}\left(1 - \frac{V_d}{u_{ave}}(y,z)\right) \tag{7.2}$$

where u_{ave} is the average velocity at the inlet plane and V_d represents the velocity deficit distribution (Eq. 7.3).

$$V_d(y,z) = V_{d_{max}} exp\left[-\left(\frac{(y-y_0)^2}{2\sigma_y} + \frac{(z-z_0)^2}{2\sigma_z}\right)\right] \tag{7.3}$$

The Gaussian profile is used to specify the deficit since it best represents the velocity field at the end of the near wake [126] [21]. The value of $V_{d_{max}}$, the maximum deficit, is calculated based on mass conservation between the inlet and outlet planes at each iteration during the computation. The values of constants in Eq. 7.3 are specified as $\frac{y_0}{D} = 0, \frac{z_0}{D} = -0.08, \sigma_y = \sigma_z = 0.8$. These values are specified since they best fit the typical wake flow structure observed in various experimental data and full CFD simulations of sub- and full-scale wake measurements. The value of z_0 defines the downward shift of the velocity center line observed in measurements and simulations [136] [137] [42]. The value of these coefficients remains unchanged for the range of validation cases and require no further tuning or case specific adjustment. The turbulence intensity and eddy frequency at the outlet plane are defined as a function of freestream turbulence and the turbine operating condition following the approach first proposed by Ainslie [3] for the eddy viscosity. The turbulence quantities are defined based on Prandtl's free shear model and formulations suggested by Crespo and Hernandez's for the turbulence parameters in wake [41]. Turbulent kinetic energy , k is defined based on the local velocity imposed at the outlet plane field as:

$$k = k_\infty + \Delta k \quad \Delta k = \frac{1}{2}(u_\infty - u_{local})(u_{local} - u_{min}) \tag{7.4}$$

where u_{min} is the minimum velocity in the profile and u_{local} is the velocity at each position in the outlet plane calculated at each point using Eq. 7.2. The formulations for ω following Crespo and Hernandez's approach is defined as:

$$\Delta\omega = a\frac{u_\infty}{z_{hub}}\left(\frac{\Delta k}{u_\infty^2}\right)^b \tag{7.5}$$

where a is calculated as 0.21 using the relation between ω, ϵ, k and taking into account the modification to the turbulent coefficients for atmospheric flow. However, obtaining a unique value for a which works for all kind of full-scale and wind tunnel wake flows proved to be difficult. This is partially due to the fact that there is a large variability in the measured turbulence length scale within the wake of a wind turbine [153], especially in field measurements. This could be partially due to variability in incoming atmospheric turbulence as well as the atmospheric stability. In full scale measurements, the effect of wake meandering also influences the overall mixing process. Overall a value between 0.58-5.8 is chosen for the coefficient a. In the case of wind tunnel measurements and flow predictions in Sexbierum wind farm discussed later the upper value of 5.8 is chosen, whereas for the full scale wind farms and assessment of array loss presented in the next chapter, a value of 0.58 is used for all cases. These formulation for k and ω with both range of coefficients results in turbulence length scale in the wake being smaller than those in freestream as it is observed in measurements [144] [64] [155]. All model boundary conditions are implemented on the Cartesian grid using an implicit immersed boundary method. This approach simplifies the flow field around the wind turbine to a level that is resolved in RANS computations with a comparable grid size to what is required for microscale wind simulations. The details of model implementation are described in the next section.

7.3 Numerical Implementation

As described in the previous section, the approach in this thesis for the simulation of the wake flow was to model the near wake region via introducing an immersed boundary method and to resolve the far wake on the computational grid. The inclusion of the immersed body in the computational domain must be performed in a computationally efficient manner such that it does not require substantially higher grid resolution or time-intensive grid generation process. There are several approaches available in the realm of Computational Fluid Dynamics for the inclusion of three-dimensional bodies within a given computational mesh such as global grid reshaping or overlapping grid approach. In this work, the immersed boundary method approach [12] is used since it allows one to accurately to handle any kind of complex geometry on the Cartesian grid with relatively high computational efficiency and acceptable accuracy [24]. Using the immersed boundary method, no modification to the existing Cartesian grid is necessary and different boundary conditions of the immersed body are imposed by introducing a source term also called forcing function. This is achieved in this work using the definition of ghost and physical nodes in the computational domain. In this approach, similar to a ghost cell definition in cell-based algorithm for implementation of e.g. wall boundary conditions, ghost nodes lying within the immersed turbine are used to impose the desired Dirichlet or Neumann boundary condition at the location of the boundary. A detailed overview of the immersed boundary method and formulation of the method for wall modeling is presented in chapter 9. Here the global procedure of imposing model boundaries are discussed.

In the implementation process, first the shape of the immersed body is defined using the prescribed operating condition and the induction factor. Using Eq. 7.1, first the diameter of inlet and outlet planes are defined (Eq. 7.6) and then a fifth-order polynomial is fitted to define the side boundaries of the immersed surface. In Eq. 7.8 G is the function describing the shape of the streamtube. The values for the polynomial coefficients are obtained using the inlet and outlet diameters and assuming zero gradients of the body at the edges. The model extends from one diameter upstream where the flow is not yet affected by the potential field of the turbine to one diameter downstream where the time dependent signature of turbine blades as well as the inviscid expansion of the wake start to vanish (see Figure 7.3).

$$D_1 = \sqrt{1-2a}D \quad b = 1-2a \quad D_2 = \sqrt{1/b}D_1 \tag{7.6}$$

$$G\left(x,y,z\right) = \left(y-y_h\right)^2 + \left(z-z_h\right)^2 - R\left(x\right)^2 \tag{7.7}$$

$$R\left(x\right) = ax^5 + bx^4 + cx^3 + dx^2 + ex + f \tag{7.8}$$

In the next step, different zones near the immersed body are detected and the nodes located in the boundary are labeled as ghost nodes (Figure 7.5). Then the location of the mirror and surface points are prescribed and the flow field at those locations are determined using a first or second order interpolation. At the location of inlet plane, the gradients of the flow quantities are set to zero satisfying boundary condition expressed in Eq. 7.9. The boundary at the outlet plane are defined similarly using Eq. 7.10. The definition of different normal and directions to the immersed surface are shown in Figure 7.4.

$$W_{IBM}^{Inlet\ plane} = \begin{bmatrix} \frac{\partial \rho}{\partial s} \\ \frac{\partial u}{\partial s} \\ \frac{\partial v}{\partial s} \\ \frac{\partial w}{\partial s} \\ \frac{\partial p}{\partial s} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.9}$$

$$W_{IBM}^{Outlet\ plane} = \begin{bmatrix} \frac{\partial \rho}{\partial s} \\ u \\ v \\ w \\ \frac{\partial p}{\partial s} \end{bmatrix} = \begin{bmatrix} 0 \\ u_{outlet} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.10}$$

At the side boundaries a slip boundary condition is imposed. For this purpose, the velocity is decomposed into the normal V_n and tangential components V_b and V_t to the streamtube defining the side surface, and then the slip boundary condition is imposed using Eq. 7.11.

$$W_{IBM}^{Side\ plane} = \begin{bmatrix} \frac{\partial \rho}{\partial n} \\ V_n \\ \frac{\partial V_b}{\partial n} \\ \frac{\partial V_t}{\partial n} \\ \frac{\partial p}{\partial n} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.11}$$

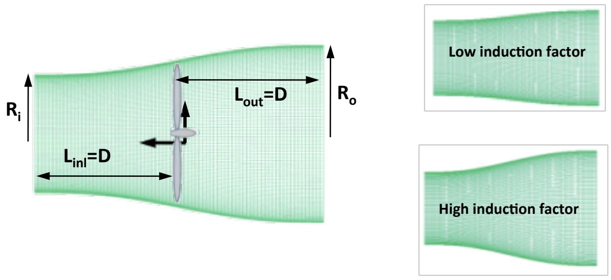

Figure 7.3: The three-dimensional shape on the model and side boundaries. The figure illustrates how induction factor affects the shape of the immersed turbine.

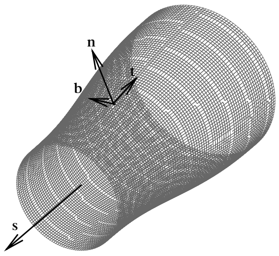

Figure 7.4: The three-dimensional shape on the model. Vectors s, n, b and t illustrate the normal and tangential vectors used in Eq. 7.9, 7.10 and 7.11.

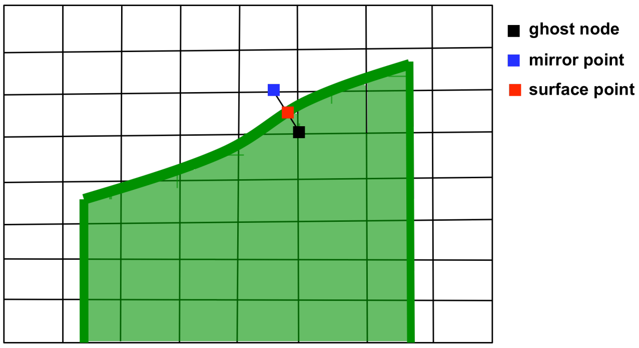

Figure 7.5: Schematic graph depicting ghost, mirror and surface points. Boundary conditions of the model are imposed are the Cartesian grid using an immersed boundary method.

The categorization of the nodes at the boundary of turbines is performed once at the beginning of the computations. As the solution evolves at every iteration, the boundary conditions of the immersed turbine are updated using the calculated velocity and turbulent field in the physical domain upstream of the turbine. Therefore, any modifications of the velocity or turbulent quantities upstream of the turbine related to terrain or upstream wake are taken into account automatically in the calculated model output at the outlet plane. At each iteration, the maximum velocity deficit at outlet plane is defined based on the incoming mass flow to the immersed turbine (Figure 7.6)

In the implementation of the model, the rate of the change of flow properties inside the

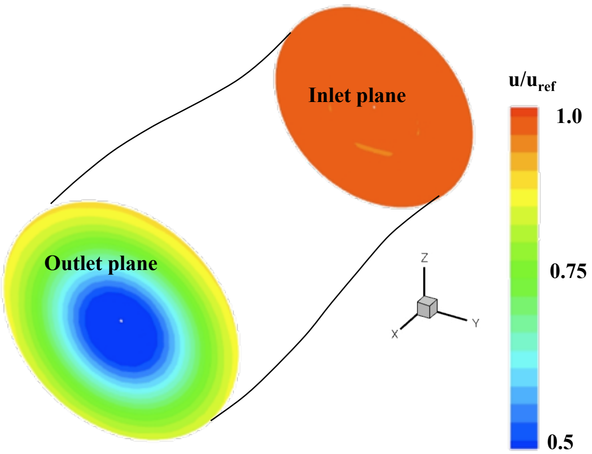

Figure 7.6: The mass flow definition at the outlet plane using the mass conservation within the immersed body.

bodies are set to zero as the internal nodes have no influence on the global flow field and can be treated as *dead* cells. Therefore no special treatment is required in connection with preconditioning. It must be noted that however, the scaling of the preconditioning parameter ϵ discussed in section 5.2.1 is very crucial to the successful and robust use of preconditioning for wake flow simulations, where reduced velocity and large values of computed viscosity makes it a difficult case for preconditioning solver. However, special treatments are required in connection with the other convergence acceleration method, multigrid.
Since the immersed wind turbine model, imposes extra boundary conditions on the main physical domain, they must be addressed in the multigrid cycle when corrections and residuals are transferred between levels. Unlike the other boundary conditions, such as at wall or inlet which are imposed on all levels, the conditions at the boundaries of the immersed wind turbines could not be updated at all levels due to insufficient grid resolution on coarse levels in the representation of the shape of the immersed body (Figure 7.7). The alternative approach is to freeze the solution at the boundaries of the immersed turbine at the coarse levels. This is crucial in successful application of multigrid in connection with the immersed wind turbine model implementation. Figure 7.8 shows how such a modification affects the convergence for a single wake simulation in connection with multigrid.

7.4 Single and Multiple Wake Simulations

For validation, predictions using the immersed turbine model are compared with measurements performed in the wind tunnel of Marchwood Engineering Laboratory [126] [58]. In the experiment, single wake measurements are performed in the wake of a model turbine operating in a simulated atmospheric boundary layer. The rotor diameter and the hub height of the turbine are 0.27 and 0.3 m respectively. Velocity and turbulence measurements are performed at three different tip speed ratios $\lambda = 2.9$, 4.0 and 5.1 which corre-

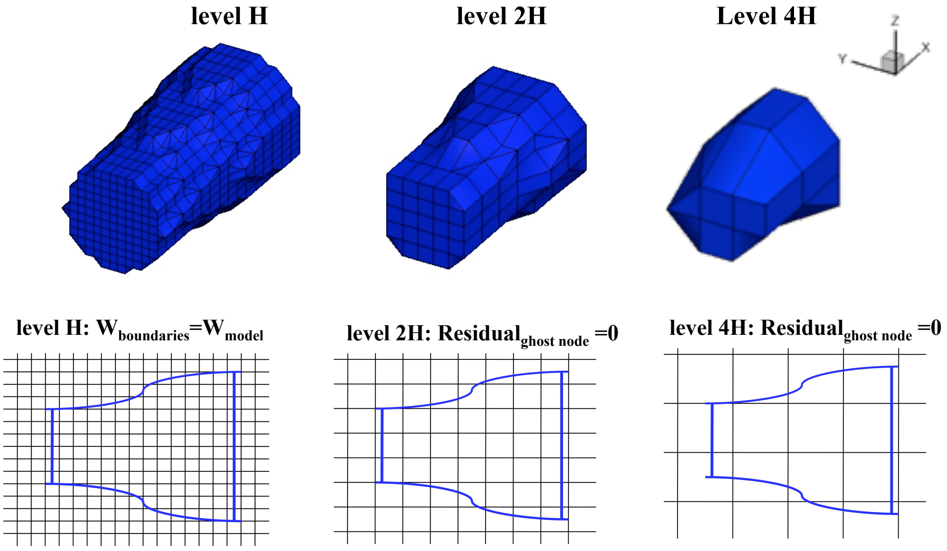
level H
level 2H
Level 4H
Z
Y
X
level H: W_boundaries=W_model
level 2H: Residual_ghost node =0
level 4H: Residual_ghost node =0

Figure 7.7: Ghost nodes within the immersed body on different grid levels.

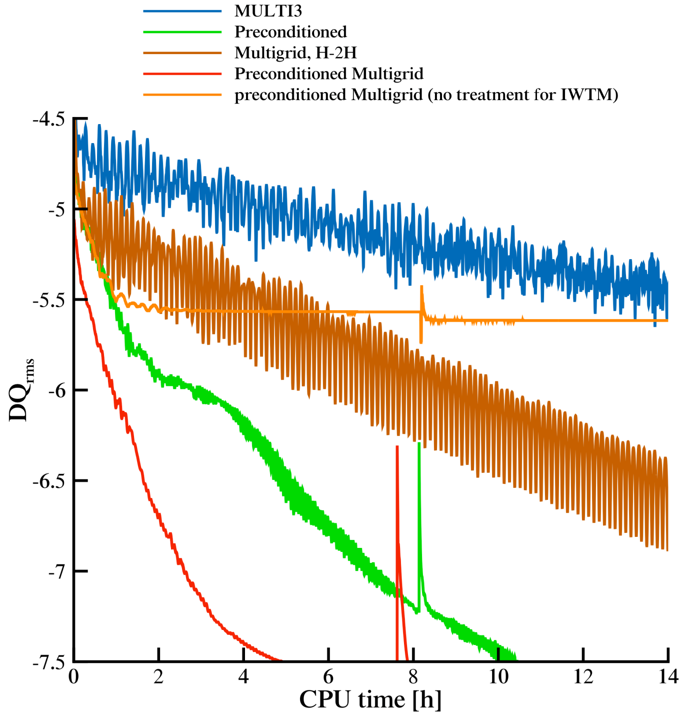

Figure 7.8: Residual drop for a single wake simulations using the immersed wind turbine model in connection with preconditioning and multigrid.

spond to measured axial force coefficients of $C_{D_{ax}} = 0.62$, 0.79 and 0.85. These coefficients are used in the model to prescribe the geometry of the immersed body. The incoming velocity profile in the simulations is specified based on Eq. 6.1 with values of $z_0 = 7.56 \times 10^{-6} m$ and $u_\tau = 0.16 m/s$ to reproduce the free stream velocity measured in the wind tunnel.

The extents of the computational domain are 67D, 22D and 20D in the x, y and z direc-

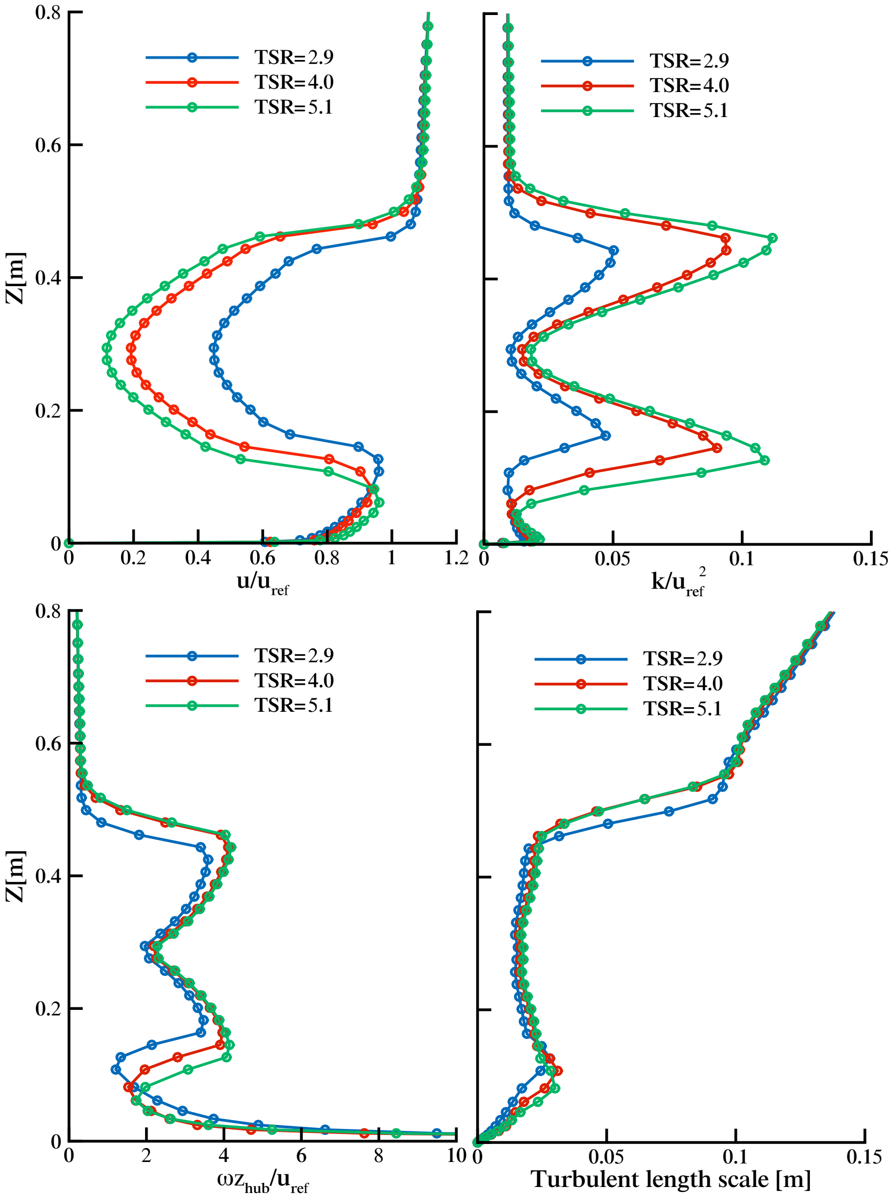

Figure 7.9: The model output for the velocity and turbulent quantities to be mapped at one diameter downstream for three different operating condition. As expected the velocity deficit and generated turbulence by the rotor increases as the thrust coefficient increases.

tions respectively. The grid is locally clustered around the immersed body and in the wake with a resolution of 0.12D in the x direction and 0.06D in the y and z directions. The effect of reducing the resolution by a factor of two in all directions was also studied. There was less than 2% difference in the velocity profiles, and therefore the results on the above grid are considered grid independent.

Figure 7.9 shows the model output for velocity and turbulent quantities at three different operating conditions at 1D downstream at the outlet plane. As expected, the value for velocity deficit and turbulent kinetic energy increases as the loading on turbine increases.

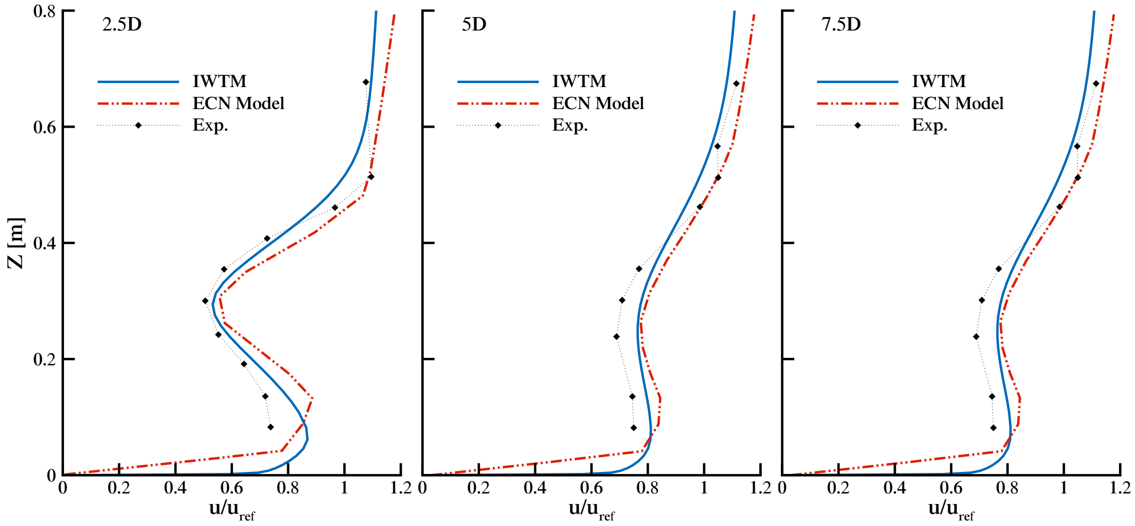

Figure 7.10: Predicted velocity profiles downstream of single turbine compared to measurements in wind tunnel for operating condition of TSR=2.9 compared to measurements [58] and simulations [126].

Predictions of wind speed compared with measurements at three downstream locations for tip speed ration of 2.9 are shown in Figure 7.10. The wind speeds are normalized with respect to the reference wind speeds. Overall good agreement is observed between the predictions of velocity deficit and wake recovery and the experimental measurements. The results are also compared with numerical solution obtained ECN's wake model [126]. The velocity deficit predictions are also compared at two other operating conditions in Figure 7.11. The error in the predicted velocity deficit at the wake centreline is 4%, 17% and 36% for TSR of 2.9, 4.0 and 5.1. As expected, in both the simulations and the experiment the deficit increases when the tip speed ratio and the axial thrust coefficient are increased from trip speed ratio of 2.9 to 4.1. The trend continues for tip speed ratio of 5.1 at one diameter distance, however due to higher generated turbulence and mixing, the deficit diminishes at 2.5D distance in the simulations. The largest difference is observed for tip speed ratio of 5.1 where the turbine is heavily loaded. The evolution of the profiles is predicted well however for all three cases. At 7.5D downstream of the turbine the difference between the measured and predicted profiles is less than 0.13 at all heights.

In order to further assess the performance of the immersed turbine model, the profiles of turbulent kinetic energy are also compared with measurements in Figure 7.12 and 7.13 for tip speed ratio of 2.9, 4.0 and 5.1 respectively. The turbulence intensity is calculated based on the hub height velocity upstream of the turbine. As shown in Figure 7.12, two peaks appear in the profiles of turbulence intensity, corresponding to the blade wake below and above the height of the nacelle. In the current formulation of turbulent kinetic energy, these peaks do exist primarily at the profile of turbulent kinetic energy that are introduced using Prandtl's equation (Eq. 7.4). The peak in the upper half is larger due to the higher wind shear. The turbulence intensity downstream of the turbine increases in both measurements and simulations when the drag force on the rotor increases, going from a tip speed ratio of

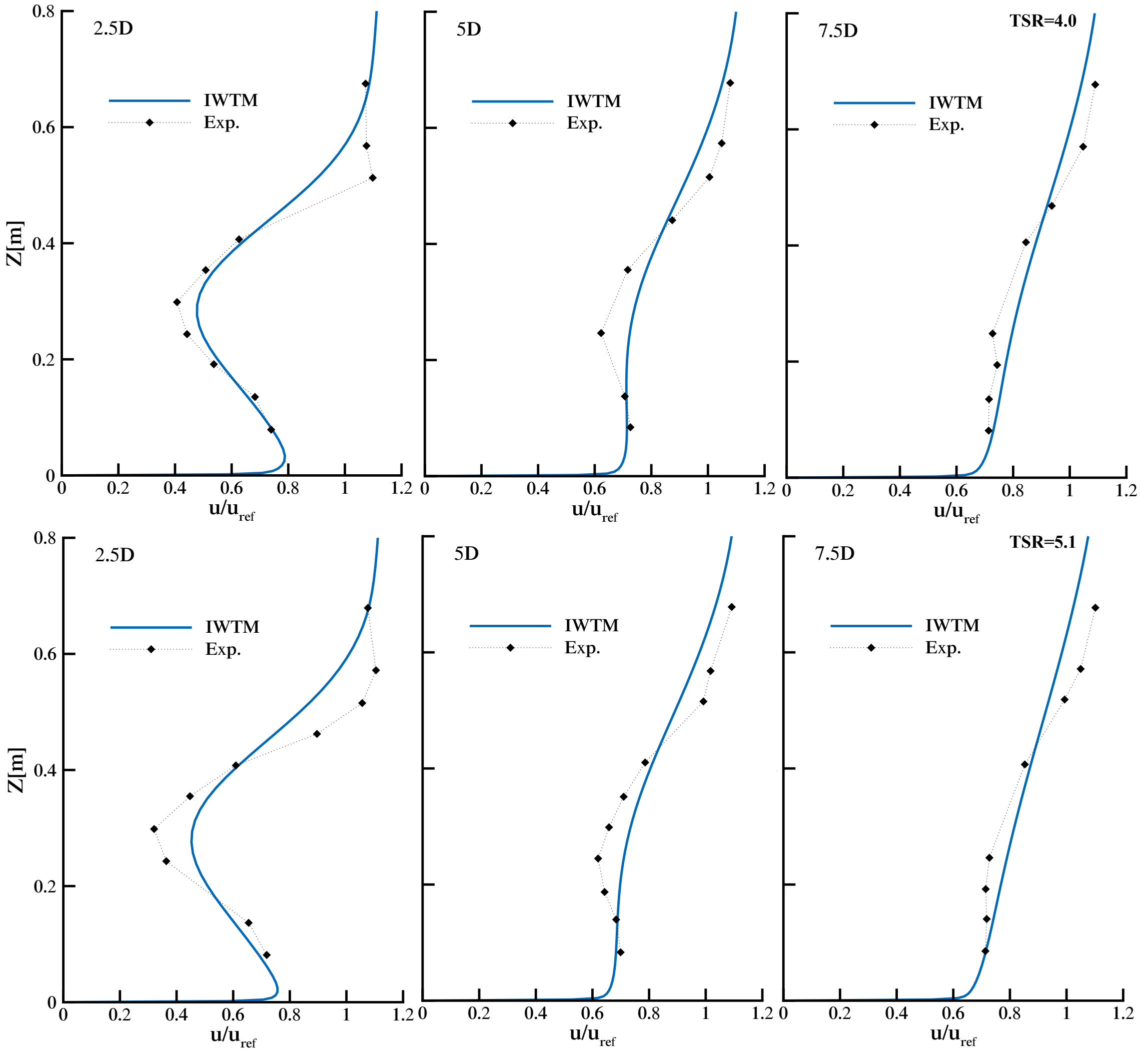

Figure 7.11: Predicted velocity profiles downstream of a single turbine compared to measurements in a wind tunnel for three operating conditions TSR=4.0 and TSR=5.1.

2.9 to 5.1. However this increases is largely overestimated in the simulations. The turbulent kinetic energy and turbulence intensity at 2.5D downstream are over-predicted by the model. The agreement between the predictions and the experiment is not satisfactory for tip speed ratio of 5.1 where the errors are the largest.

The performance of the immersed model is next assessed with regard to predictions of velocity deficit, wake interactions, recovery of the wake and profiles of turbulent properties of full-scale wind turbines. The measurements of single and double wakes from the Sexbierum wind farm are chosen for this purpose [33] [34]. This Dutch experimental wind farm is located in the north of the Netherlands approximately 4 km distance from the sea. The layout of the wind farm is shown in Figure 7.14. The wind farm is located in flat homogeneous terrain covered by grass. The location and topography of the wind farm makes it

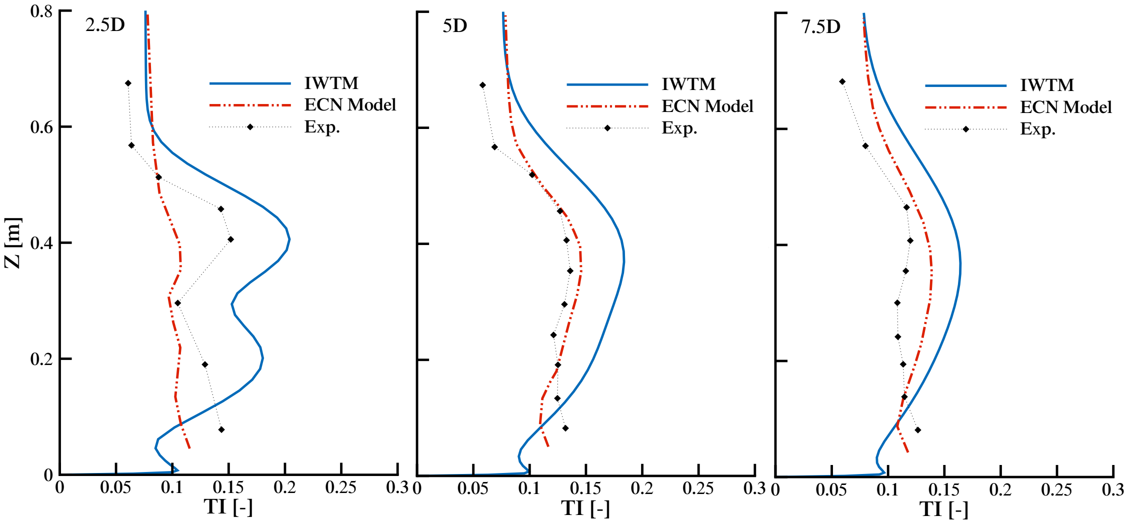

Figure 7.12: Predicted turbulence intensity profiles downstream of a single turbine compared to measurements in a wind tunnel for tip speed ratio of 2.9 compared to measurements [58] and simulations [126]

possible to reduce the uncertainties in defining the inflow boundary conditions. The rotor diameter of all turbines in the farm is 30.1 m and the hub height is 35 m. The turbines have 300 kW rated power. In order to obtain maximum efficiency, the tip speed ratio of the turbines is kept constant up to a wind speed of 10 m/s. Therefore the thrust coefficient of the turbines is relatively constant ($C_T = 0.75$) for wind speeds of 6-10 m/s.

Single wake measurements are performed at different up- and downstream positions of the turbines using mobile masts instrumented with 3-component propeller and cup anemometers. The free stream hub height velocity is 10 m/s and turbulence intensity is 10%.

The extents of the computational domain are 94D, 46D and 24D in x, y and z directions. A Cartesian grid with local clustering around the turbine and in the wake is used for the simulations. The grid resolution is 0.04D in all directions. Simulations were also performed with a coarser grid with 0.08D resolution to investigate the effect of grid in prediction of wake recovery. Both grids are also clustered close to the ground in the z direction in order to capture the large gradients close to the rough terrain (Figure 7.15). The black body in Figure 7.15 denotes the immersed wind turbine model.

Figure 7.16, 7.17 and 7.18 show the predicted velocity downstream of the turbine compared to measurements. The results obtained using two different grids are shown in Figure 7.16. Coarsening the grid has no effect in prediction of the wake recovery. In Figure 7.17 and 7.18 results obtained using the current model are compared with predictions where the turbines are modeled using an actuator disk model [120]. The results show that the velocity deficit is underestimated in predictions using both models. The centreline velocity is predicted with a 4% error using IWTM. However, the lateral extent of the wake is captured quite accurately using the newly developed model. This has significant importance for studying the wake effects in wind farms. Under or overestimation of the wake extent may lead to inaccurate

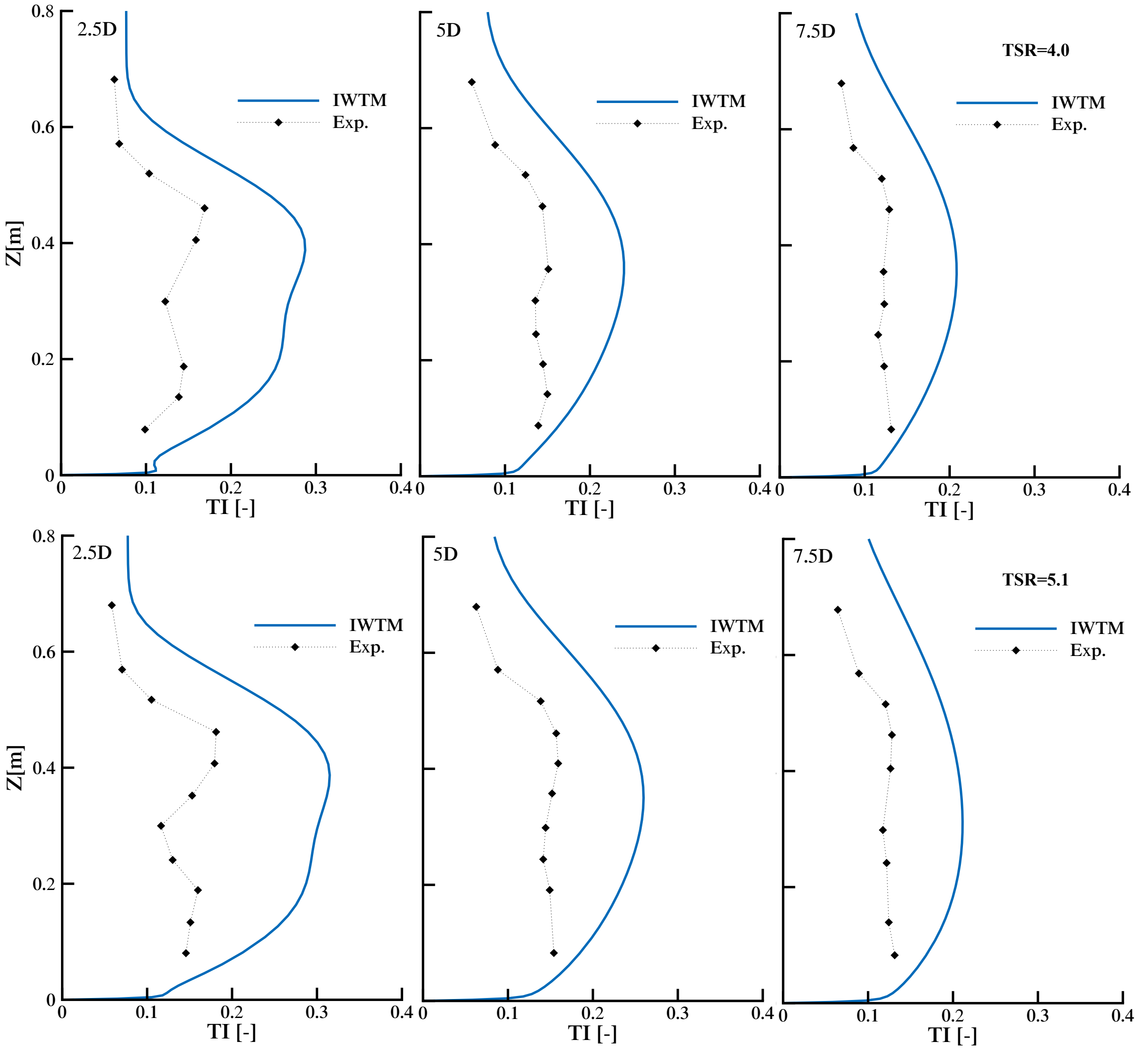

Figure 7.13: Predicted turbulence intensity profiles downstream of a single turbine compared to measurements in a wind tunnel for three operating conditions TSR=4.0 and TSR=5.1.

loss prediction in an array of turbines.

The predictions are also compared to measurements at downstream distance of 8D, where the predictions are even more relevant in the context of micrositing. The measurements of both velocity deficit and turbulent intensity show a nonaxisymmetric profiles which are not captured in the simulations. At 8D downstream, the velocity deficit at wake centerline is predicted as 0.15 compared to measured value of 0.25. The turbulence intensity predictions are also compared with measurements at 2.5 and 8D downstream of the turbine (Figure 7.19 and 7.20). Similar to the velocity profiles, the turbulence intensity is predicted well at the outer edges of the wake, but the local minima of the profile at the centreline is not present in the simulations. The agreement of the turbulent kinetic energy at 8D is less good where the measurements do not show a coherent symmetrical distribution inside the wake. This could be due to wake meandering or small changes in the upstream wind direction during

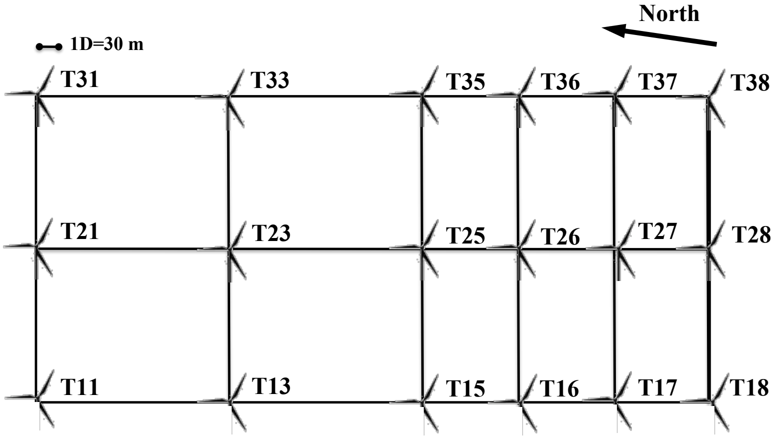

Figure 7.14: Layout of the Sexbierum wind farm located in the north of the Netherlands.

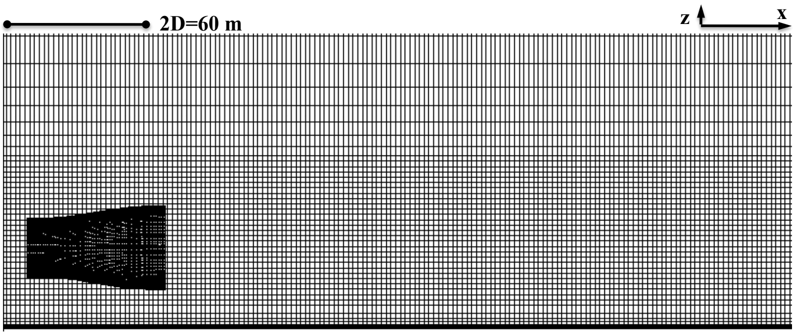

Figure 7.15: The view of the computational grid with local clustering in the z direction and around the immersed turbine used for the single wake simulations at the Sexbierum wind farm.

the measurements, which are not accounted for in the simulations. However, the level of turbulence intensity seems to be captured well by all models at around 8 – 10%.

Figure 7.21 shows a side view of the contours of wind speed and turbulent kinetic energy downstream of turbine modeled with IWTM. It can be seen that the atmospheric boundary layer and far wake region are resolved on the computational grid. The downwind shift of the minimum velocity from the rotor axis as well as the wake expansion could be observed in the results. It is also seen that the point of maximum turbulent intensity also moves upward from the turbine axis as expected.
Next the wake interaction is studied using the immersed model. The simulations are done based on double wake measurements undertaken at the Sexbierum wind farm. The data are from turbines T36, T37 and T38, see Figure 7.14, for a southerly wind direction, when turbine 37 operates in the wake of T38 and turbine 36 is in the wake of both upstream turbines. The turbines are separated by a distance of 5D. The simulations are done for seven different wind directions, -15^o, -10^o, -5^o, 0^o, 5^o, 10^o and 15^o, where 0^o indicates the direction parallel to the turbines T36, T37 and T38. The simulations are performed for a wind speed of 10 m/s and the same computational grid is used for all directions.

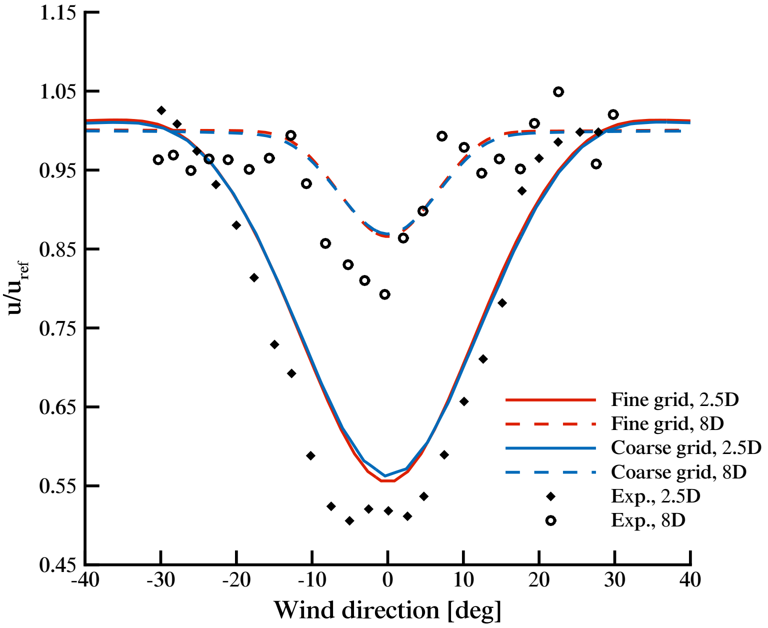

Figure 7.16: Predictions of the velocity profiles at 2.5D and 8D downstream of the turbine using two different grid resolutions.

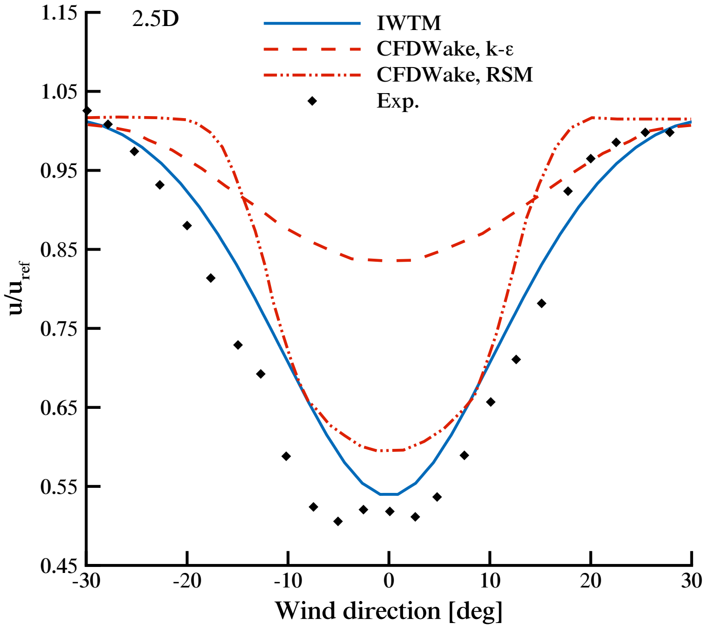

Figure 7.17: Predictions of the velocity profiles at 2.5D downstream of a full-scale wind turbine compared to measurements and simulations performed using actuator disk mode [120]

Figure 7.22 shows a plane view of the contour plots of velocity at hub height for the 5^o and 15^o wind directions when the turbines are partially immersed in the wake of the upstream turbine. The prediction of the power loss for T37 and T36 versus wind direction is com-

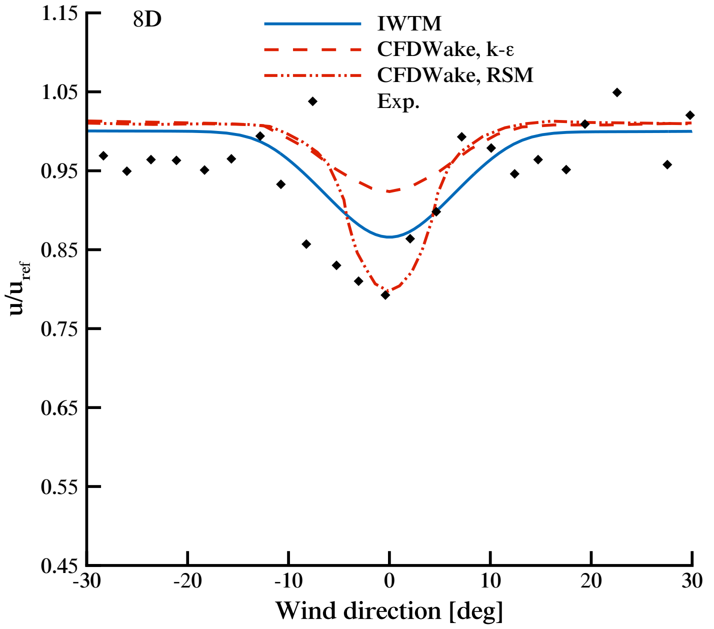

Figure 7.18: Predictions of the velocity profiles at 8D downstream with measurements [33] and simulations [120]

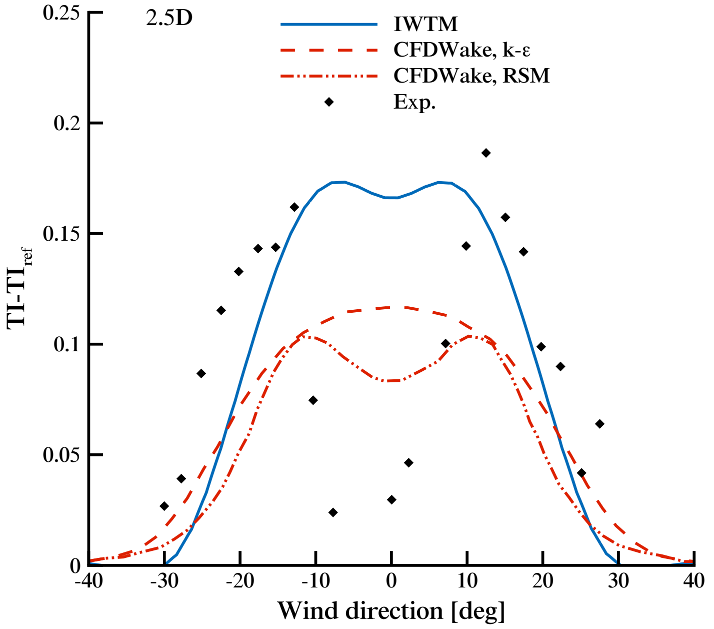

Figure 7.19: Predictions of turbulence intensity at 2.5D downstream of a full-scale wind turbine compared to measurements and simulations performed using actuator disk mode [120]

pared with measurements in Figure 7.23. Since the power and thrust coefficients of the turbines are reported as constant for a wind speed below 10 m/s, the power ratio of the turbines in the simulations is calculated as the ratio of the velocity cubed. The average ve-

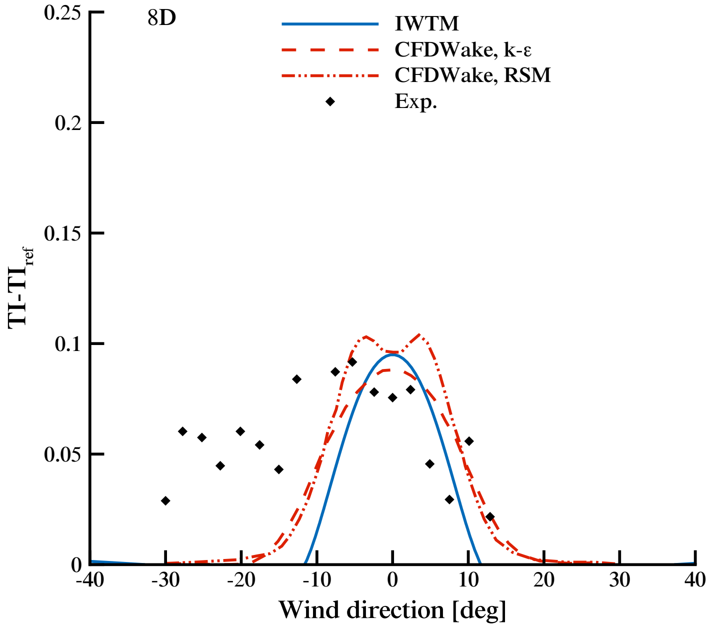

Figure 7.20: Predictions of turbulence intensity profiles at 8D downstream with measurements [33] and simulations [120]

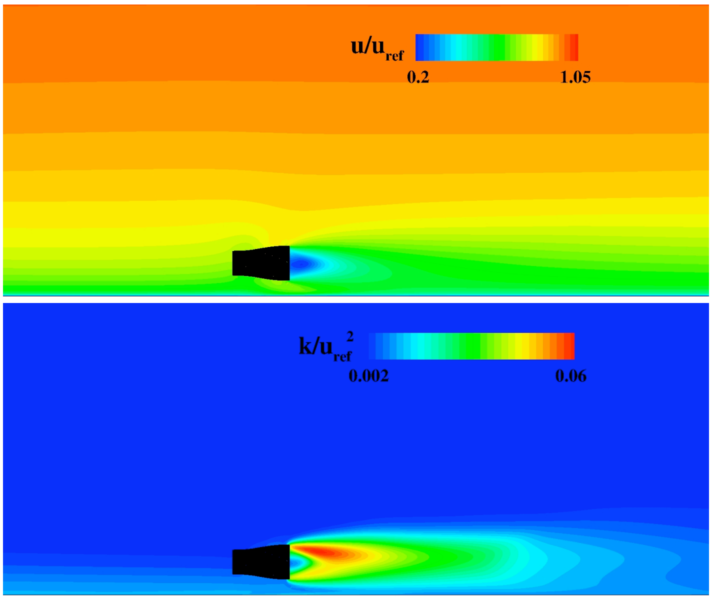

Figure 7.21: Side view of contour plots of wind speed and turbulent kinetic energy downstream of the turbine in operation in an atmospheric boundary layer simulated using IWTM.

locity over the inlet plane in the IWTM is used as the reference velocity at the hub height of the turbines. Results show that for the 0^o wind direction, where the downstream turbines are fully immersed in the wake, the power loss of T36 is predicted well by the model with less than 7% error, whereas the power deficit of T37 is overestimated by 13%. In general, it is observed in both simulations and experiments that the largest drop in power output

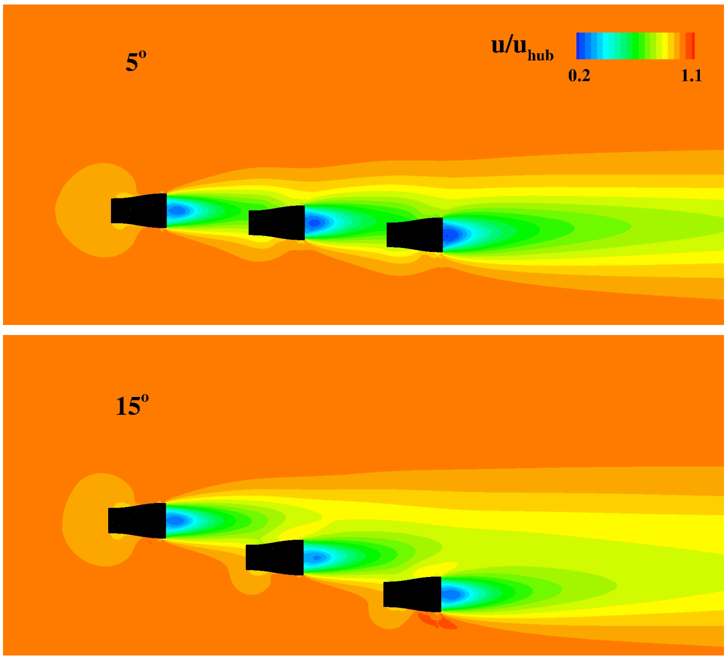

Figure 7.22: Plane view of contour plots of wind speed showing the wake interactions for two different wind directions.

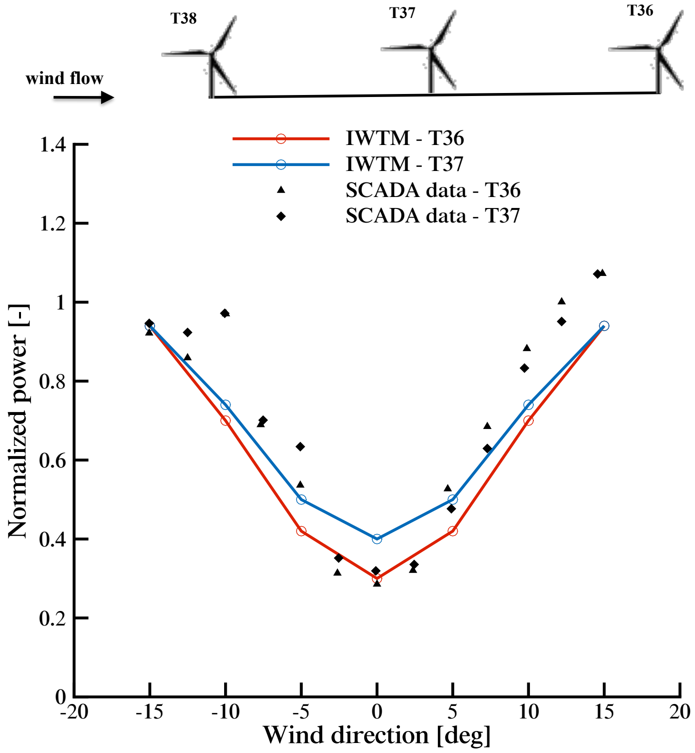

Figure 7.23: Power versus wind direction compared to measurements for two downstream turbines, T37 and T36 at the Sexbierum wind farm.

occurs between the first row and the second row whereas the power loss between the second row and third row is much less reduced. The power from second to third row in the simulations decreases by 10% where as this reduction is only 3% in the measurements. The small change in the power loss in subsequent rows in cluster of turbines occurs because the recovery in the wind speed between T37 and T36 is approximately equal to the velocity drop across the rotor plane of turbine T37. The same trend is observed in measurements performed in several operating wind farms [73] [44]. Even though the power loss for both turbines is overestimated by the model for directions between 5 and 15 degrees, the wake extent is captured quite well. In both measurements and simulations, when the angle between the turbine axis and wind is larger than 15 degrees, the downstream turbines are out of the wake's shadow effect and the turbines see the undisturbed wind.

7.5 Summary

The formulations and implementation of immersed wind turbine model is verified and validated for single and double wake interaction compared to wind tunnel test and full-scale wake measurements. The results show that the model is capable in reproducing the mean and turbulent field behind the wind turbines. The performance of the model for different operating conditions was tested compared to wind tunnel measurements. The increase of velocity deficit and turbulent kinetic energy is obtained downstream of the turbine as thrust coefficient increases as expected based on the model formulations. The trend is visible at 2.5D downstream for tip speed ratio of 2.9 and 4.1, corresponding to axial drag coefficients of 0.62 and 0.79. At tip speed ratio of 5.1 however, the excessive generated turbulence results in faster recovery of centerline velocity than measured. The dissipation of turbulent kinetic energy is also poorly predicted for tip speed ratio of 5.1. The average turbulence intensity at rotor plane decreases from 12% at 2.5D to 10% at 7.5D in measurements whereas this decrease is 18% in the predictions. The model is also tested for single and double wake measurement in Sexbierum wind farm. Overall 5% and 30% error is observed in predictions at 2.5D and 8D downstream of the turbine. The extent of the wake is predicted well by the model, with predicted 100% recovery at 28^o at 2.5D. The average values of turbulent intensity are also predicted well by the model. The reduction of turbulent intensity in the nacelle wake is captured by the model even though the reduction in the measurements is more pronounced. The Power loss in the row of three turbines are also estimated using the model and compared to measurements. The predicted power loss in the first and second downstream turbine is predicted as 67% and 60% compared to measured 72% and 69%.

CHAPTER

8

Estimation of Array Loss in Wind Farms

In this chapter, the wake interaction in wind farms is studied using the immersed wind turbine model and the array loss estimations are compared with SCADA data. The simulations are performed based on double and multiple wake measurements undertaken at the Lillgrund offshore wind farm [44], Spanish wind farm located in complex terrain [116] and Mont Crosin wind farms located in Switzerland.

8.1 Lillgrund Wind Farm

First the performance of wind turbine in an offshore Lillgrund wind farm is studied. The Lillgrund wind farm is located in the *Ö*resund, a body of water between Copenhagen Denmark, and Malmo Sweden. The farm consists of 48 Siemens SWT-2.3-93 turbines with a rated power of 2.3 MW, rotor diameter of 93 m and a rotor hub height of 65 m. Figure 8.1 shows a plane view of the turbines in the wind farm. The farm has a relatively dense turbine configuration and therefore of great interest is an investigation of how wakes affect the production of the farm.

The numerical simulations are performed including over two first rows of the farm for different wind directions. The dependency of the power of the turbines to wind direction and the prediction of the wake extent are compared to the measurements. The performance of turbine 22 (C_{07}) for different wind directions is simulated. Six turbines in the first and second row of the farm are included in the simulations. The simulations are performed over the wind direction range of $195^o - 245^o$ with 5 degrees interval. The influence of power level on relative power is reported in [44]. The insensitivity of the relative power to power level implies that turbines are operating efficiently with a constant thrust coefficient. A thrust coefficient of 0.79 is used in the simulations based on the manufacturer's power curve. The extents of the computational grid are 95D, 80D and 17D in x, y and z directions respectively. The computational grid used for the wind farm simulations are shown in Figure 8.2.

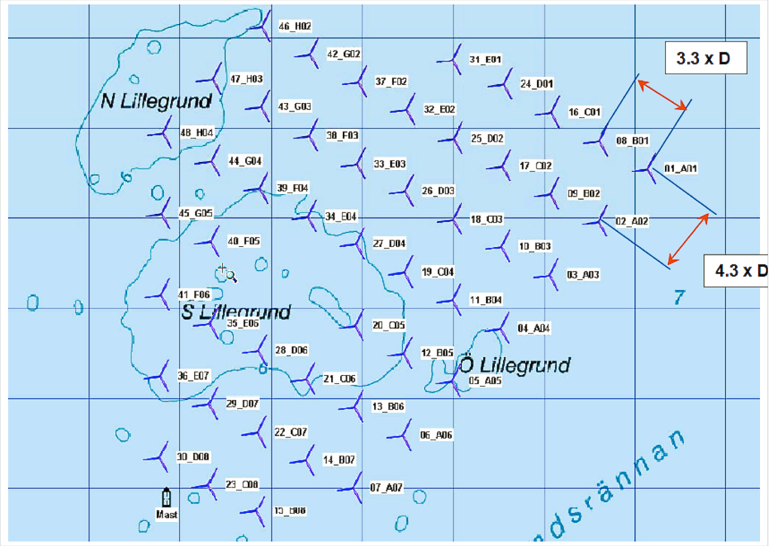

Figure 8.1: Layout of Lillgrund wins farm [44].

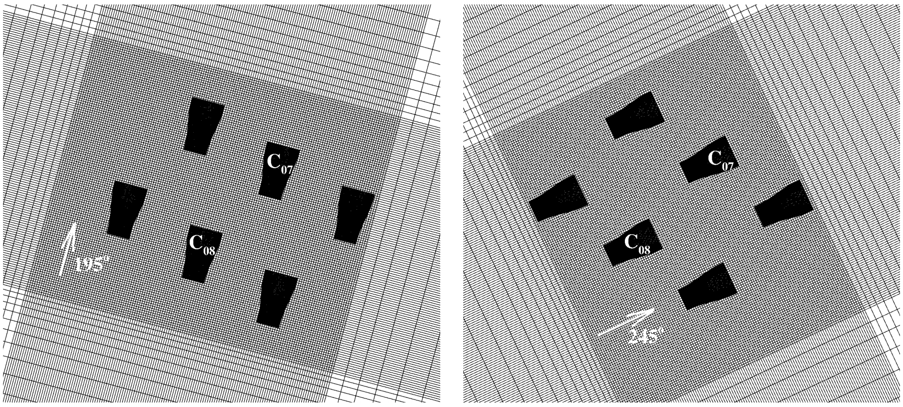

Figure 8.2: Same Cartesian grid used for simulations of all wind directions.

The black symbols denote the immersed wind turbines. As mentioned before using the immersed wind turbine model all direction can be modeled using a single grid. A surface roughness of 0.1 mm, hub height velocity of 9 m/s and turbulent intensity of 6% are used in the simulations [32].

The normalized power of turbine 22 (C_{07}) versus wind direction is shown in Figure 8.3. The power of turbine C_{07} is normalized by the upstream turbine 23 (C_{08}). The variations of the power are captured very well by the model with average error of 8% except for 230^o and 235^o wind directions where turbine 23 is still partially immersed in the wake of upstream turbine (C_{08}) whereas based on measured data it operates in undisturbed wind flow.

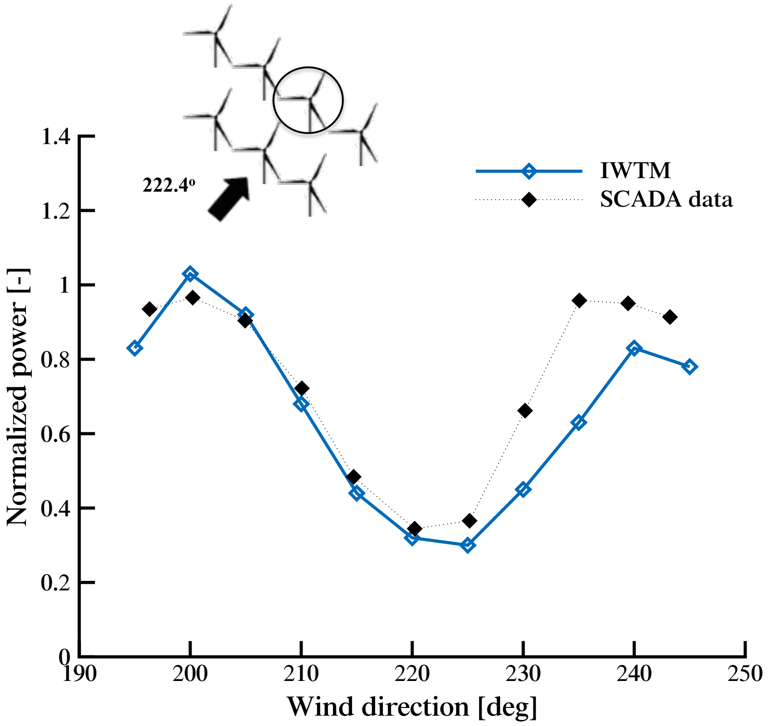

Figure 8.3: Predicted effect of wake on power of turbine C07 (shown with circle) versus wind direction.

8.2 Wind Farm in Complex Terrain

As discussed in chapter 6, the local wind flow and turbulence are greatly influenced by the underlying topography in onshore wind farms. Steep hills often cause large wind speed-ups, flow turning, changes in wind sheer and flow separation. In the previous sections, we have also shown that how the interaction of wind turbine with upstream wake flow greatly impacts the performance of the wind turbine. In order to assess the effect of topography and wake flow simultaneously, simulations of wind turbine over complex topography is attempted in this thesis. The first wind farm which is studies is a medium sized wind farm with 43 turbines with moderately complex topography located in Spain. Figure 8.4 shows the layout of the turbine. The performance of this turbine was reported within the Upwind Integrated Project [116]. Politis et al. performed simulations of this wind farm using two different solvers in connection with a actuator disk model and compared the array loss predictions with results from more established tools such as WAsP. In this work, the simulations are performed using the immersed wind turbine model. The bin for wind direction in experimental data reduction and WAsP analysis is $327^o \pm 5^o$ and therefore simulations are performed for three different wind directions of 322^o, 327^o and 332^o. The arrow in Figure 8.4 shows the wind direction of 327^o where the array losses are most significant for the layout of this farm.

The extent of the computational domain is $15 \times 13 \times 1.8 km$ in x,y and z directions. The inlet plane is located 6 km upstream to resolve the effect of topography on the incoming wind flow. The outlet boundary condition is located downstream of the last row and periodic

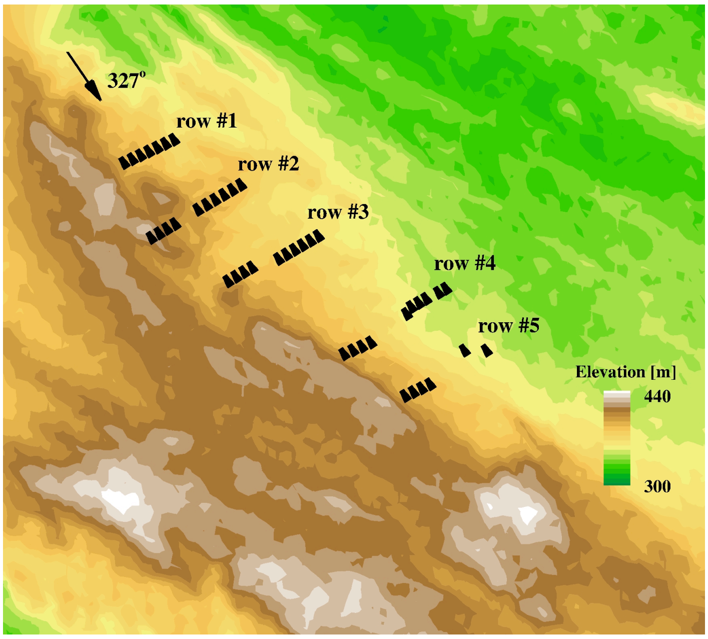

Figure 8.4: Layout of the wind farm located in complex terrain superimposed on the digital elevation map of the area. The measurement data and simulations are reported for wind direction of 327^o *where the turbine are operate in maximum wake coverage of upstream rows.*

boundary condition are located 6 km away of the edge of the wind farm. A Cartesian grid is generated with clustering around the turbines. The grid resolution is 0.08D in all directions in the clustered region. The mesh is additionally clustered close to the ground with the first point being 2 m above the ground level. These resulted in overall total of 8-9 million grid points. The thrust coefficients of the turbines are specified based on provided data. The inlet boundary conditions are defined using Eq. 6.1 for flow and turbulent quantities. The roughness is assumed constant over the topography as 0.01 m. The velocity at hub height is specified as 8.4 m/s.

The simulations are also performed for an identical wind farm with similar layout located on flat terrain in order to evaluate the possible effects of terrain on the array loss and wake interaction in the wind farm. These simulations are performed using same computational grid and boundary conditions to isolate the effect of local topography on the predictions.

Figure 8.5 shows the array loss predictions in rows 1-4 in the wind farm. The power of individual turbines are normalized using the average power of first 7 turbines in the first row. Figures shows the results from three directions using immersed wind turbine model. The numerical results from CFDWake [116] using the actuator disk model are also shown in the same figure. The RMS error of prediction of normalized power in all rows is compared with two other CFD solvers and WAsP in Table 8.1 for wind direction of 327^o. The RMS error in predictions is better than predictions obtained by WAsP in all rows. The highest error is observed in row # 3 where losses in turbines 5-8 is poorly predicted. As is observed in the contour plot, last three turbines operate almost out of the wake region (Figure 8.7). Figure 8.5 also shows the sensitivity of the power loss to wind direction in this wind farm. The sen-

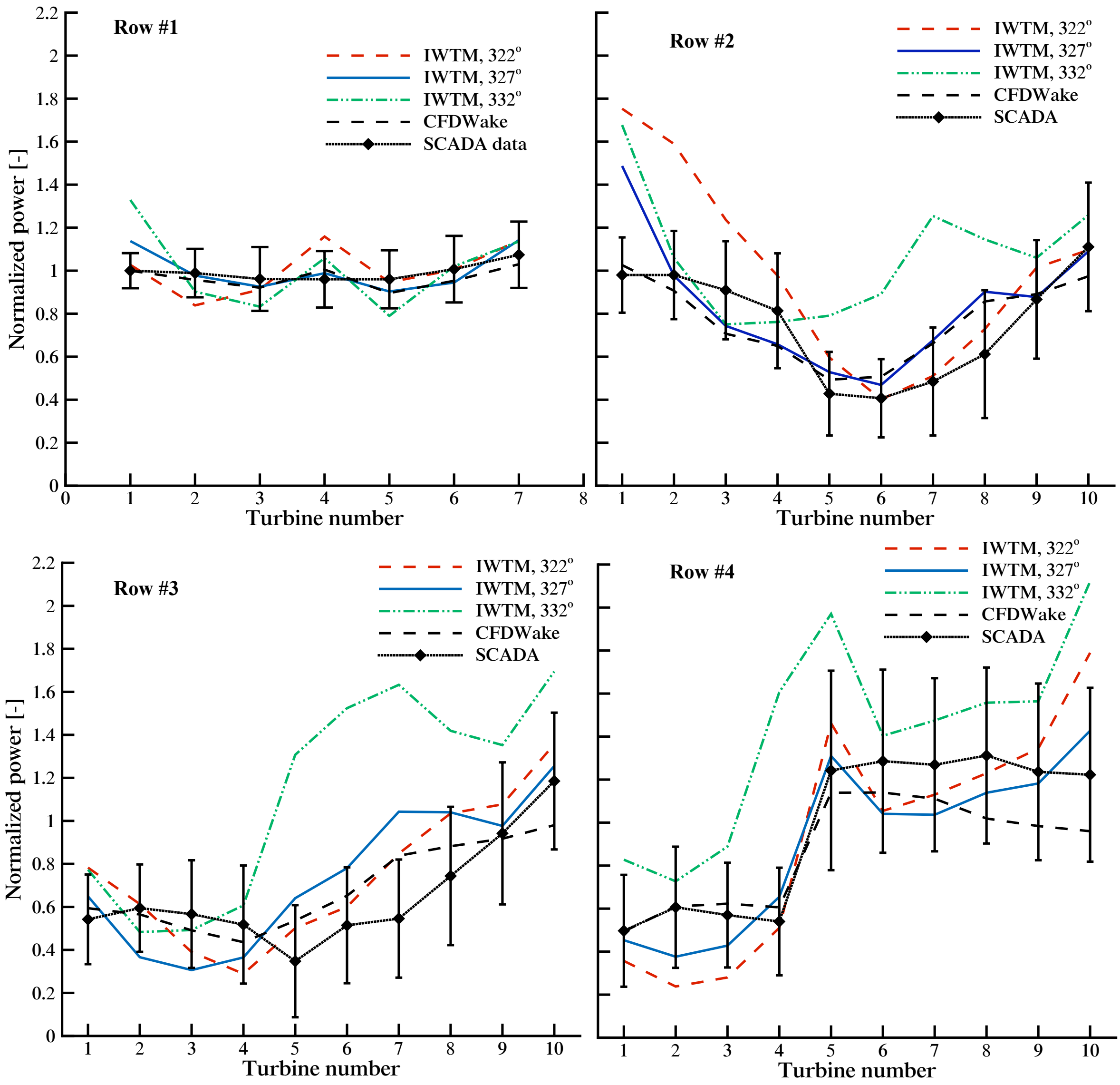

Figure 8.5: Normalized power for turbines in 4 rows compared to SCADA data. The results show the predictions for three directions 322^o, 327^o and 332^o. The numerical results obtained using CFDWake solver and actuator disk model are also presented in the figure [116]

sitivity is rather large especially for wind direction of 332^o. In addition to the change in the wake extent and orientation, for the farms located in complex terrain, the change in wind direction also alters the upstream topography and therefore undisturbed wind as well as the wake evolution. Hence, in complex terrain, there is even larger sensitivity of array loss to change in wind direction. Based on current results, averaging band of 10^o for assessing array loss proves to be too large even in moderately complex topography.

In order to investigate the effect of topography on wake evolution in more detail, power loss predictions for two directions, 322^o and 327^o, are also compared to a similar farm located

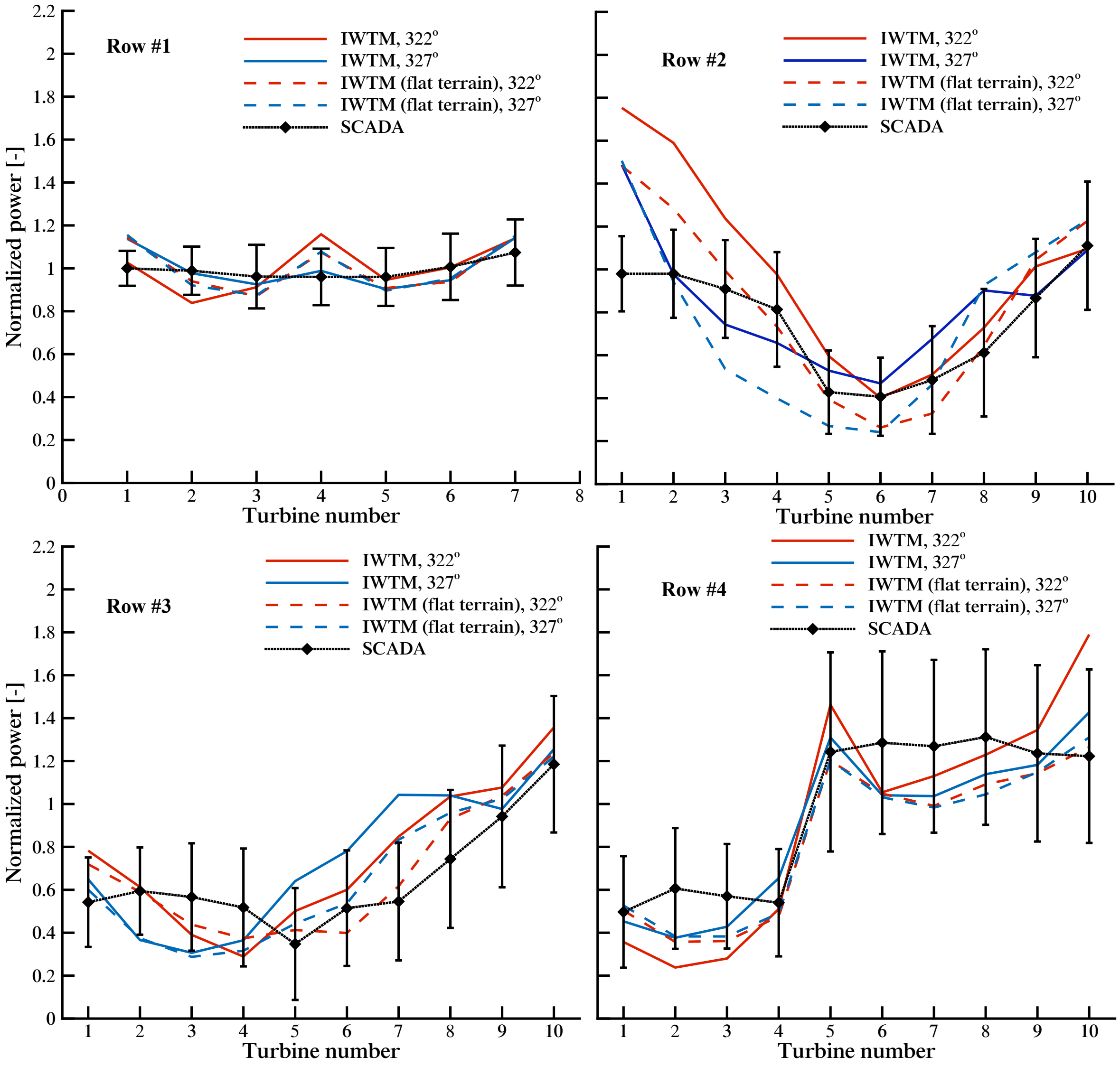

Figure 8.6: Normalized power for two wind direction in the wind farm located over moderately complex topography and a similar farm located over flat terrain.

on flat terrain. Results are compared in Figure 8.6. As expected, the sensitivity of power to wind direction is eliminated for the case of flat terrain in the first row. The small fluctuations of power between seven turbines are due to the change in the hub height of the turbines which results in different computed averaged velocities at the inlet plane of the model. The maximum average difference between two cases in observed in row # 3 for turbines 5-8 where the predictions have the largest uncertainty. The difference between two cases shows that the over prediction of the power in this series of turbines is terrain-related. In general, the array loss is larger for the case of flat terrain in all rows which confirms a more rapid mixing and recovery of the wind turbine wake in interaction with topography.

Next the aerodynamics and performance of another wind farm that is located in complex terrain is examined.This second wind farm studied is the 23.7MW Mont Crosin wind farm

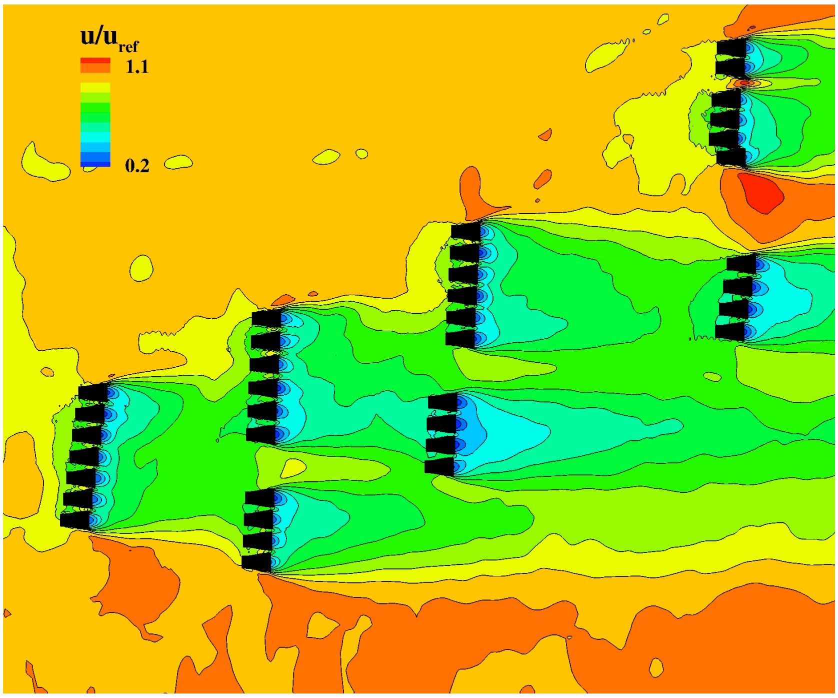

Figure 8.7: Velocity contours in wind farm located in complex terrain.

Table 8.1: Comparison of mean RMS error of different models for each row. The results obtained using immersed wind turbine model is compared with model results obtained by [116]

Model	Row #1	Row #2	Row #3	Row #4
IWTM	0.038	0.215	0.293	0.165
CFDWake	0.049	0.23	0.294	0.119
CRES-flowNS (a)	0.062	0.300	0.369	0.120
CRES-flowNS (b)	0.090	0.427	0492	0.192
WAsP	0.095	0.329	0.405	0.265

that is comprised of 16 turbines and located in western Switzerland. The terrain and the positions of the turbines are shown in Figure 8.8. To analyze the wake effects, 9 closely placed turbines on the centre of the hill are included in the simulations. Turbines 1-4 are small turbines (Vestas V44 and Vestas V47) with hub height of 45 m and turbines 10-14 (Vestas V90) have hub height of 95 m. The terrain is moderately complex and there is an approximately 500 m change in elevation in less than 3 km distance towards south from the location of the turbines.

The analysis of 18 months of SCADA data using our in-house Blade Element Momentum Code [81] shows that the thrust coefficient is 0.8 for the relevant wind speed of 5-9 m/s. The land is mainly covered by short grass and the roughness height of 0.03 m is used in the sim-

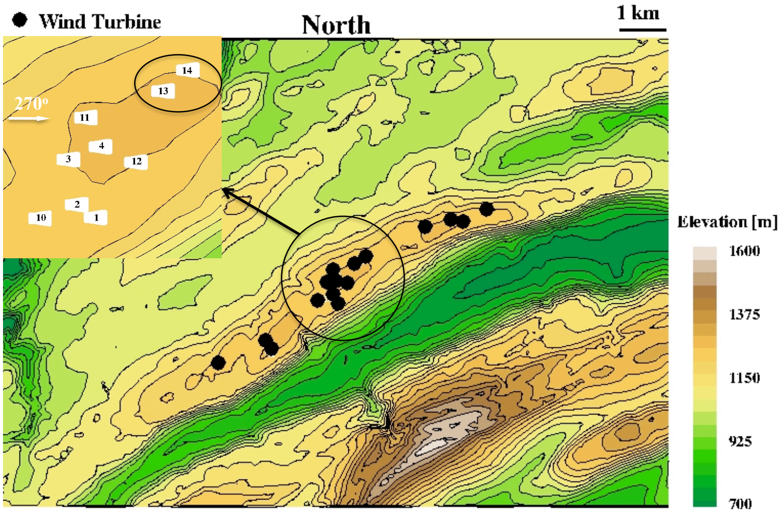

Figure 8.8: Digital elevation map of the terrain where Mont Crosin wind farm is located. Turbines are shown as black circles.

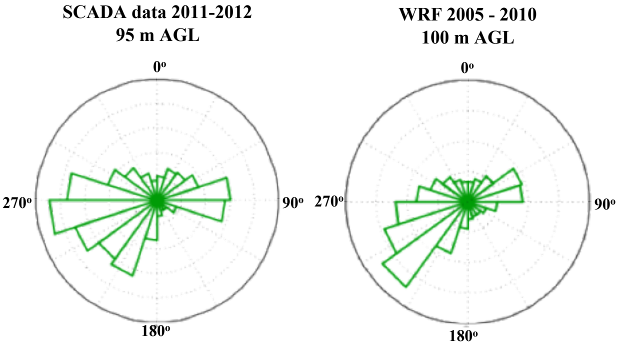

Figure 8.9: Distribution of wind direction on the site.

ulations. The turbulence intensity is specified based on ETH Zurich measurements using a nacelle mounted probe over a span of a year at this wind farm [98].
Figure 8.9 shows the distribution of wind direction obtained based on SCADA data from turbine 10. The wind rose obtained from mesoscale simulations, which are performed over Switzerland using Weather Research Forecast Model (WRF) are also shown [69]. It can be observed that the dominant wind direction is from the southwest quadrant. Thus wake effects are quite significant for this site due to the turbine arrangement. Based on the wind rose, the wind directions between 180^o and 270^o are simulated. The extent of the computational domain is 20 km x 14 km x 4 km in x, y and z directions. The grid with local clustering around the turbines is used. The horizontal resolution of the grid is 10 m.

Simulations are performed for 6 directions of 170^o, 195^o, 205^o, 225^o, 250^o and 260^o. A quantitative comparison of the effect of wind direction on the output power of turbine 14 relative to power output of turbine 13 to measurements is shown in Figure 8.10.

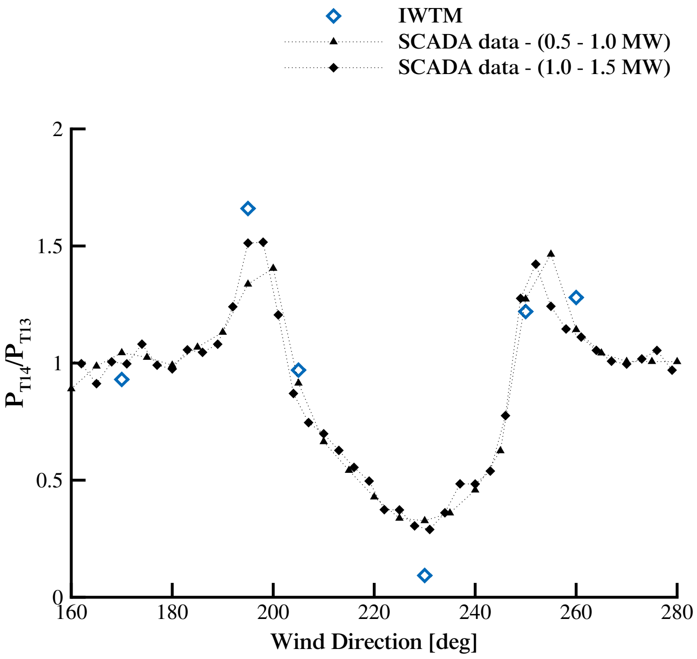

Figure 8.10: Effect of wake shadow for different wind directions, on power ratio of turbine 14 with respect to turbine 13.

8.3 Summary

The performance of immersed wind turbine model in estimation of array loss was presented and assessed in this chapter. The performance of offshore wind farm Lillgrund is first analyzed. Tow first rows of the wind farm were simulated and dependency of the power of downstream turbine to variations in wind direction was estimated. The results were compared to SCADA data collected from the operating farm. Overall, the average RMS error between simulations and measurements is 0.13. The minimum power of the turbine fully immersed in the upstream wake is predicted as 0.31 compared to measured value of 0.35. Next the performance of wind farm located in complex terrain is investigated and array loss estimations are compared to SCADA data. The entire wind farm consisting of 43 turbines with mixed hub heights is simulated using the preconditioned solver and immersed wind turbine model. The predictions are also compared with other RANS solvers and WAsP engineering tool. Overall, the predictions of power loss are improved compared to WAsP at all row in the farm located in moderately complex terrain. The relative error in power predictions by immersed wind turbine model is 5.9 %, 21.5 %, 40.1 % and 16 % in row number 1 to 4 respectively.The array loss estimations in this farm are also compared to a similar farm located on flat terrain and the effect of wake-terrain interaction on power performance in onshore wind farms is quantified. Last, the dependency of power output to wind direction in Mont Crosin wind farm located in complex terrain is evaluated for a pair of turbines. The mean RMS error between simulations and measurements is 0.11. The predictions are in good agreement with measurements except for the case of full immersion in the upstream wake where the power loss is largely over estimated.

CHAPTER

9

Immersed Boundary Method for Terrain Modeling

The capability of RANS models has been demonstrated for atmospheric and wake flows in the previous chapters. It was also discussed that in order to achieve sufficient accuracy in wind prediction, further improvements are required in several areas including turbulence modeling, surface roughness, boundary conditions and grid generation. This penultimate chapter focuses on the grid generation problem, and demonstrates alternative modeling approaches for the simulation of flow over complex terrain that avoid the difficulties related to grid generation for complex geometries and simulation of multiple wind directions.

Most flow solvers used for atmospheric flow modeling employ structured or unstructured body-fitted grids over the specified topography. Since most of the solvers use finite difference or finite-volume methods with relatively low (that is first or second-order) accuracy, fine meshes are required to obtain good results. Moreover, in highly complex terrain, it is necessary to use fine meshes irrespective of the order of the discretisation scheme; otherwise the poor representation of the terrain results in reduced accuracy of the predictions. Apart from the fact that some solvers are not capable of handling complex geometries, the grid generation process often requires significant computational time to achieve a satisfactory balance between the desired grid size and the skewness of the elements. The latter is more problematic over steep landscapes. Moreover, most algorithms use a terrain following coordinate transformation, which introduces additional numerical error in the solver when used over very steep terrain.

In addition to the above considerations, in wind resource assessment for the micrositing of wind turbines, multiple wind directions must be investigated. In order to cover the wind rose of interest using conventional rectangular domains, often grids must be generated for each wind direction. This is a lengthy and tiresome procedure. On the other hand, the use of one circular computational domain amplifies the uncertainties in the definition of inflow boundary conditions. To overcome these problems, in this study, an immersed boundary method has been used in a our RANS solver in order to model the flow over any arbitrary topography using a single Cartesian grid. The method does not incur significant additional

costs. Furthermore, changes in surface geometry simply require modification of the orientation of the topography without any further modification of the code or the grid.
The immersed boundary method (IBM) was first introduced by Peskin for low Reynolds number biological flows [114]. However, later its application was successfully extended to simulate the flow over arbitrary complex geometries using RANS solvers, large eddy or direct numerical simulations [96] [12] [76]. In the IBM, the presence of the surface is modeled by an external body force acting on the grid nodes in the vicinity of the surface. Peskin [114] modeled the force acting on the flow by moving solid boundaries with a spring system. Later, several authors extended the method by using *feedback forcing* in the momentum equation to set the desired boundary condition [54]. However, in this approach, the computational cost is significantly increased for transient flows [121]. To overcome this problem, the *direct forcing* approach was introduced. In this method the forcing function due to the immersed surface is included in the equations being solved through the use of ghost nodes [47]. This approach is used here. In addition to higher stability, the advantage of this method is that the main part of the solution algorithm is calculated only once for a specific simulation and remains unchanged during the remainder of the computation. This significantly decreases the computational overhead.
As discussed above, the immersed boundary method has become very popular in the CFD community and is used extensively in Direct Numerical and Large Eddy Simulation methods. The application of the method in connection with RANS however is not very well established. One reason for the lack of widespread use is extreme grid requirement in the boundary layer, normal to the wall, in RANS methods ($y^+ < 1$). The implementation of the IBM in a RANS solver was first presented by Majumdar [96]. Several authors later used IBM in connection with RANS but the application was still limited to relatively low and moderate Reynolds flows [111] [74]. While the immersed boundary method has been applied to wind predictions using LES [89], in this study the implementation of the method with a RANS solver and its application to wind flow complex terrains is presented for the first time.
Furthermore to facilitate the efficient application of IBM for atmospheric wind simulations using RANS with Cartesian grids, following Tullio [147] wall functions are used to relax the stringent near wall grid resolution requirements. In the following section, the numerical method including the details on the solver and implementation of IBM and wall functions are presented. In a next section, the implementation for basic solver is validated for laminar separating flow over cylinder. Then, the wind flow simulations using IBM over a two-dimensional hill are compared with simulations using a grid-aligned with geometry. The immersed boundary method is finally applied to simulations for the moderately complex terrain and complex terrain test cases, Askervein Hill and Bolund Hill described in chapter 6.

9.1 Numerical Method

The immersed boundary method is implemented into our solver in connection with different discretisation than Runge-Kutta. The method uses a central discretisation in space similar to the Runge-Kutta scheme described in section 3.1, but the space discretisation is based on Ni's Lax-Wendroff [110]. Ni described an explicit time marching staggered grid technique to solve the Euler and Navier-Stokes equations. In contrast to the standard Lax-Wendroff method, second order time derivatives are replaced by expressions of second order spatial derivatives. The method of Ni was applied by Saxer in a 3D turbomachinery Euler flow [125], who used non-reflecting boundary conditions and relative systems for rotating blade rows. Burdet extended the solver to the Reynolds Averaged Navier-Stokes equations [24] and added the turbulence models of Baldwin and Lomax and Spalart and Allmaras. Later Basol [15] extended the solver to include the turbulence model of Wilcox [162].
In the scope of this work, a multiple-grid algorithm was added to the solver in order to further accelerate the convergence of the scheme. In the next sections, a brief description of the scheme will first be presented and then the details of multiple grid algorithm will be discussed.

9.1.1 Finite Volume Lax-Wendroff Method

Ni's Lax-Wendroff is a cell-vertex finite volume integration technique of governing Navier-Stokes equations. The method consists of a Lax-Wendroff time integration procedure and a cell-vertex formulation of the inviscid terms and cell-center formulation of viscous terms over control volumes. The formulation of the method starts with a Taylor series expansion of the variable vector W:

$$W^{n+1} = W^n + \Delta t \left(\frac{\partial W}{\partial t}\right)^n + \frac{\Delta t^2}{2}\left(\frac{\partial^2 W}{\partial^2 t}\right) \tag{9.1}$$

$\left(\frac{\partial W}{\partial t}\right)^n$ can be replaced by the sum of the derivations of the viscous and inviscid fluxes at time step n. If we call the $\delta W = W^{n+1} - W^n$ residual, one can obtain:

$$\begin{aligned}\delta W = \Delta t \left(\frac{\partial F}{\partial x} + \frac{\partial G)}{\partial y} + \frac{\partial H}{\partial z} - \frac{\partial F_v}{\partial x} - \frac{\partial G_v)}{\partial y} - \frac{\partial H_v}{\partial z}\right)^n - \\ \frac{\Delta t^2}{2}\frac{\partial}{\partial t}\left(\frac{\partial F}{\partial x} + \frac{\partial G)}{\partial y} + \frac{\partial H}{\partial z} - \frac{\partial F_v}{\partial x} - \frac{\partial G_v)}{\partial y} - \frac{\partial H_v}{\partial z}\right)^n\end{aligned} \tag{9.2}$$

In the above formulation, the rate of change of fluxes could be also linearized and expressed based on space derivatives. After replacement and linearization of the flux calculation, the node based- residual is given as:

$$\delta W = \Delta W_i^n - \Delta t\left(\frac{\partial}{\partial x}(\Delta F^n) + \frac{\partial)}{\partial y}(\Delta G^n) + \frac{\partial}{\partial z}(\Delta H^n)\right)$$
$$- \left(\Delta W_v^n - \frac{\Delta t}{2}\left(\frac{\partial}{\partial x}(\Delta F^n) + \frac{\partial)}{\partial y}(\Delta G^n) + \frac{\partial}{\partial z}(\Delta H^n)\right)\right) \tag{9.3}$$

where ΔW_i^n and ΔW_v^n are the first order change of the inviscid and viscous fluxes. The rate of change in Eq. 9.10 is calculated on the cell center and needs to be distributed to the nodes in a cell vertex algorithm. The detailed derivation and formulation of the scheme on a general curvelinear grid can be found in [143] and will not be discussed here. However, to describe the multiple grid algorithm, the formulation of Ni's Lax-Wendroff is given in detail for 1D Euler equation [110]. The Taylor series expansion for the 1D flow is written as:

$$\delta W_i = \left(\frac{\partial W}{\partial t}\right)^n \Delta t \tag{9.4}$$

where index i refers to the node number. By defining:

$$\Delta W_I = \left(F_{i-1}^n - F_i^n\right)\frac{\Delta t}{\Delta x} \tag{9.5}$$

$$\Delta W_{II} = \left(F_{i1}^n - F_{i+1}^n\right)\frac{\Delta t}{\Delta x} \tag{9.6}$$

$$\Delta F = \left(\frac{\partial F}{\partial W}\right)\Delta W \tag{9.7}$$

we can now write:

$$\delta W_i = \frac{1}{2}\left(\Delta W_I + \frac{\Delta t}{\Delta x}\Delta F_I\right) + \frac{1}{2}\left(\Delta W_{II} + \frac{\Delta t}{\Delta x}\Delta F_{II}\right) \tag{9.8}$$

where I and II refer to the cells surrounding node i (Figure 9.1). The Eq. 9.8 can be interpreted as a formulation that describes how the cell-based residuals ΔW_I and ΔW_{II} are distributed to the cell vertices. Therefore, the rate of change in the flow vector W in node i has contribution of both cells I and II:

$$\delta W_i = (\delta W_i)_I + (\delta W_i)_{II} \tag{9.9}$$

The simplified equations above provide a sufficient basis for describing the multiple grid algorithm used in connection with Ni's Lax-Wendroff for convergence acceleration.

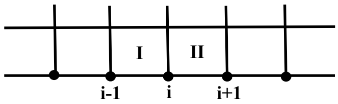

Figure 9.1: Control volumes used for one-dimensional model

9.1.2 Multiple Grid Algorithm

Following a rather similar approach of classical multigrid method described for hybrid Runge-Kutta in chapter 4, the multiple grid method tries to combine the fast convergence rate of coarse grids with the high accuracy of fine grids. Similar to Runge-Kutta, in Ni's Lax-Wendroff coarse grids are used to propagate the fine grid corrections more rapidly in the computational domain and accelerate the convergence to steady-state. However the details of the algorithm and implementation differ.

In the multiple grid approach, once the coarse grid levels are constructed, the multigrid cycle starts with a complete iteration and update of the basic scheme on the fine level. The cell based residuals ΔW are calculated on the fine grid and distributed to the nodes using the "distribution" formula (Eq. 9.8) to the surounding nodes. The fine level solution is updated using the computed node based residual. In the next step, both the flow variables and residuals are restricted to the coarse level. The restriction of the flow variables is performed using a simple injection. Unlike the Runge-Kutta scheme, the restriction of the residuals are also performed using a simple injection. In Ni's Lax-Wendroff the node based residuals are restricted to the coarse level shaping the cell-based residual at the center of the super cell (Eq. 9.10).

$$\Delta W_I^{2H} = \delta W_i^H \tag{9.10}$$

Once the cell-based residual on the coarse grid is formed, the distribution formula similar to the one used on fine grid is applied using the larger time step computed on the coarse grid. The final node based residuals are prolonged to the fine grid using a linear interpolation described in section 4.3 and added to the solution on the fine grid level.

In the presence of multiple coarse grid levels, the residual is transferred to the coarsest level and distribution is not performed until the coarsest level is reached. As mentioned before, in the multiple grid algorithm, the coarse grids are used to distribute the correction of the fine grid to larger distances and therefore greatly accelerate the convergence to steady state. Since the inviscid part of Navier-Stokes equations dominate the wave propagation in the computational domain, only the distribution of inviscid fluxes is performed on the coarse levels and viscous and dissipation fluxes are only computed on the finest grid. The boundary conditions are also only updated on the fine grid. Therefore in general, the computational overhead due to multigrid method is smaller (25% per iteration) compared to the Runge-Kutta scheme where all flux calculations and boundary conditions are updated at all grid levels.

Figure 9.2 shows the convergence history for the flow over a two-dimensional hill using the multiple grid algorithm with different grid levels in connection with Baldwin-Lomax turbulence model. Using 3 levels of grid, results in $4.5x$ speed-up in convergence.

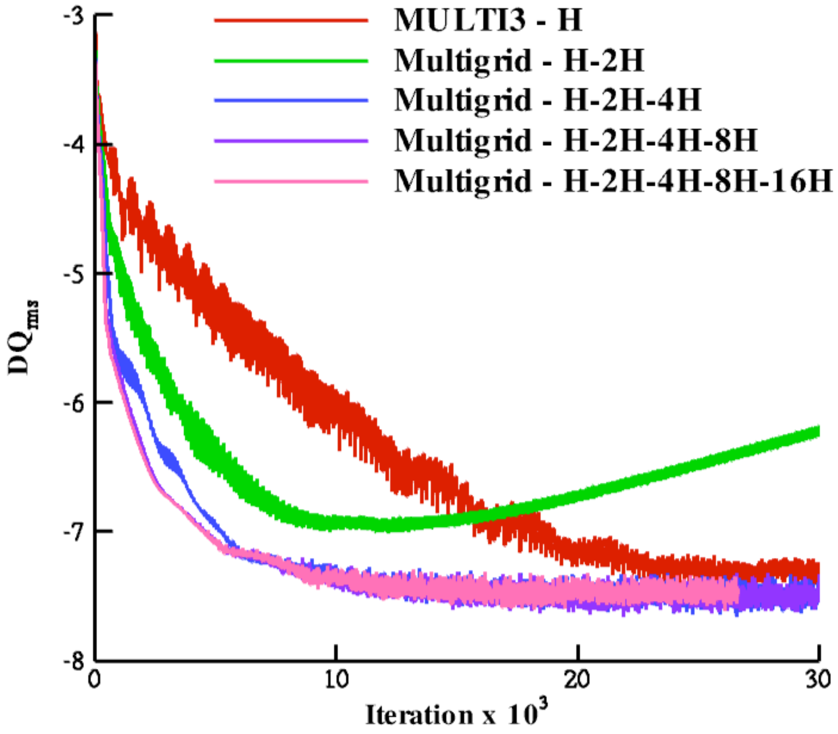

Figure 9.2: Convergence history for flow over two-dimensional hill using multiple grid algorithm.

9.2 Immersed Boundary Method

As mentioned before, in this study, the direct forcing approach is used to simulate the no-slip boundary condition at solid boundaries of the computational domain. In this approach, the solid surface divides the whole computational domain into three regions, (i) physical cells, (ii) interfacial cells and (iii) ghost cells (Fig. 9.3). Ghost cells are dead

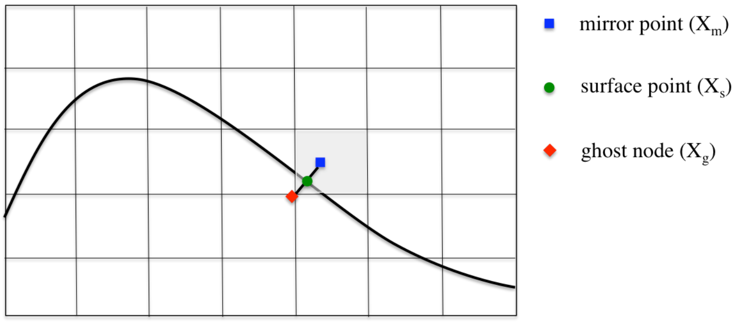

Figure 9.3: Schematic diagram of the computational stencil used for the implicit IBM. The immersed surface (shown as a black line) divides the domain into physical, interfacial and ghost cells. Illustration of the implicit boundary condition at the point X_s which is on the immersed boundary. The computed flow at the mirror point X_m is used to specify the flow at the ghost point X_g such that the boundary condition is implicitly fulfilled on the immersed surface

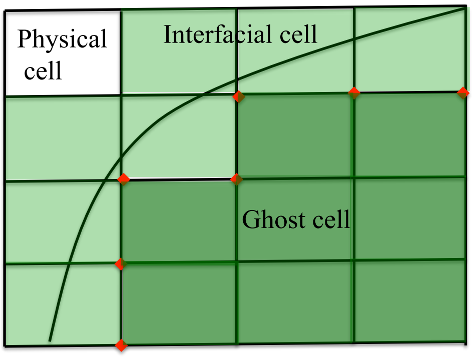

Figure 9.4: Schematic diagram depicting physical, interfacial and ghost cells separated by the immersed boundary

cells within the computational domain (Fig. 9.4). These cells do not carry any meaningful physical values. Instead, the grid nodes in the vicinity of the surface (ghost nodes) are used to impose the desired boundary condition at the surface. With the known location of ghost nodes, each ghost node is linked to the immersed surface through a surface point, X_s. The surface point is the point on the virtual surface that is closest to the ghost node. The normal line connecting the ghost nodes and surface points also specifies the mirror points (X_m) that are located within the physical domain (see Eq. 9.11). Note that $\lambda = 1$ for a linear implementation.

$$X_m = \frac{(1+\lambda)\,X_s - X_g}{\lambda} \tag{9.11}$$

The overall idea of the immersed boundary method is to impose the proper flow field on X_g based on the flow vector on X_m, so that the desired boundary condition on the immersed surface, X_s, is fulfilled. Since the mirror points are not necessarily located at computational nodes, the full immersion of the boundary condition necessitates the interpolation of the flow field to corresponding points within the physical domain. Here a weighted average interpolation based on inverse distance is used for this purpose. The boundary condition is imposed for the flow variable, which is either density, velocity components or pressure. After determining the flow variable at a mirror point, Q_m, the flow variable at the ghost node Q_g is specified to set the required no-slip condition of Q_s at the surface. If the immersed boundary condition is of a Dirichlet type, the specified flow field is set at the ghost nodes accordingly. For a no-slip boundary condition with linear interpolation, Eq. 9.12 gives the flow variable at the ghost node:

$$Q_g = (1+\lambda)\,Q_{IBC_{Dirichlet}} - \lambda Q_m \tag{9.12}$$

where Q_{IBC} is the desired boundary condition for a specific variable at the immersed surface. For the velocity, Q_{IBC} is set to zero. Furthermore, a Neumann boundary condition may be required to specify the gradients at the wall. This boundary condition is given as following:

$$Q_{IBC_{Neuman}} = \frac{\partial Q}{\partial n} \Longrightarrow Q_g = \lambda Q_m - (1+\lambda)\,Q_{IBC_{Neuman}} \tag{9.13}$$

which can be reduced to Eq. 9.14 for the pressure at the no-slip wall.

$$Q_g = Q_m \tag{9.14}$$

In order to initiate a search for ghost nodes in the vicinity of the immersed surface, it is necessary to define the entire immersed surface. The topographical data is read as discrete points from a Digital Elevation Map model. Bicubic interpolation is used to specify the entire terrain. The search for mirror and surface nodes is undertaken once at the beginning of the computation. A gradient descending algorithm is used to find the surface points that have the minimum distance to the ghost node (see Fig. 9.3). Depending on the initial guess, as few as 10 iterations are required. In general, the computational time required for the geometric calculations is less than 0.01% of the overall computational time. Similar to classic boundary conditions, interpolation of the flow field at mirror points and specifying the desired boundary condition at ghost nodes is undertaken at every iteration. The implementation of the wall function in connection with turbulence models for atmospheric flow simulations is discussed in the following section.

9.2.1 Turbulence Boundary Condition

9.2.1.1 Low-Reynolds Turbulence Model

The immersed boundary method developed in the present work can be used with either Baldwin-Lomax or $k-\omega$ turbulence models. More complete details of these models can be found in [13] and [163] respectively, but salient features of the models with regard to the implementation of IBM are given below.

In the two-layer algebraic Baldwin-Lomax turbulence model, the turbulent eddy viscosity is given by:

$$\mu_T = \begin{cases} \mu_{inner} & \text{if } y \le y_m \\ \mu_{outer} & \text{if } y > y_m \end{cases} \tag{9.15}$$

where y is the normal distance from the wall and y_m is the distance at which the inner and outer eddy viscosity are equal. Inner layer eddy viscosity is calculated from Prandtl's mixing length theory and depends on the absolute value of the vorticity and the mixing length. The characteristic mixing length is defined as below as:

$$l_{mix} = 0.4y[1 - e^{\left(-\frac{y^+}{26}\right)}] \tag{9.16}$$

where the non-dimensionalized distance y^+ is given by Eq. 3.56.

In computations using body-fitted grids, y is defined as the minimum distance between a grid point and the closest point on the terrain (no-slip surface). On the other hand, in the IBM implementation, the minimum distance to the immersed surface must be used. The minimum distance is determined either as the distance between the node within the domain and all surface points or using the minimum of distance function, the algorithm used for the calculation of the surface points. The wall shear is estimated using the computed velocities at the mirror points (Eq. 9.17).

$$u_\tau = \sqrt{\frac{\mu (U_m - U_s)}{\rho_s (X_m - X_s)}} \tag{9.17}$$

In the $k-\omega$ turbulence model, transport equations are used to solve the turbulent kinetic energy, k and the specific dissipation rate, ω. The eddy viscosity is then obtained using Eq. 2.20. The boundary condition for k is:

$$k_{wall} = 0 \tag{9.18}$$

and is applied at a grid point located on the wall for the classical body-fitted grid and at ghost nodes for the IBM implementation. For ω, the boundary condition follows [106],

$$\omega = \frac{60\mu}{\rho \beta_1 \Delta {y_0}^2} \tag{9.19}$$

where $\beta_1 = 0.075$, Δy_0 is the distance of the first node above the wall for the classical body-conformal grid and for the IBM implementation, either the distance between the mirror and surface points, maximum value of the prescribed quantity or the grid spacing in the direction normal to the immersed wall. The best results over several test cases were obtained using the latter approach. μ_T is set to zero for all nodes below the immersed surface for both turbulence models. It should be noted that the k and ω behavior close to the boundary is strongly non-linear. Hence, the accuracy of the implemented boundary conditions is strongly dependent on the grid resolution. This dependency can be substantially reduced using wall functions. A detailed description of the wall function implementation in connection with IBM is given in the next session.

9.2.1.2 High-Reynolds Turbulence Model

Wall functions are employed in connection with the immersed boundary method in order to relax the stringent near-wall grid resolution requirements. Because of the strongly nonlinear behavior of the turbulent kinetic energy, k, and the turbulent eddy frequency, ω, close to the solid boundary, the application of the developed IBM for high Reynolds number flows such as wind simulations over terrain will not be efficient since a huge number of cells are required to resolve high Reynolds number wall bounded flows. Local grid refinements alleviate such a difficulty, but are not fully satisfactory especially in three dimensions. To improve both the accuracy and computational time, wall functions are required to provide accurate and reliable solutions.
The use of wall functions in connection with turbulence models is very well suited for atmospheric flow simulations where resolving the viscous sub-layer is not of interest even for

body-fitted grids. In addition, surface roughness can be modeled using empirical relations developed for high Reynolds boundary conditions for turbulence models as discussed in chapter 3. In high-Reynolds type boundary conditions for turbulence models, the approach is to specify the variables at the first near-wall node based on the universal law of the wall (Eq. 3.58) similar to body-fitted grid described in section 3.3

To implement wall functions into a Navier-Stokes code in connection with immersed boundary method, the u-velocity component requires a condition that ensures the correct shear at the wall. Similar to simulations with a grid-aligned with geometry, this can be done either by adding a source term to the momentum equation or prescribing an artificial viscosity at the wall. Turbulent kinetic energy and ω also need to be corrected.

In the immersed boundary method, the ghost nodes are used to impose the boundary conditions at the wall for Navier-Stokes equations and low Reynolds number turbulence models, whereas the interfacial nodes are used to implement the wall function (Fig. 9.5). In the current implementation the correct shear at the wall is first calculated. In the shear equation (Eq. 3.58), u at wall adjacent node is replaced by difference in velocities at the interface node and the surface point. To apply the correction similar to body-fitted grid, an artificial viscosity is prescribed at the interfacial cell to correct the shear to ensure that the wall shear is correct. Since the current solver is a vertex-based method, to modify the averaged artificial viscosity at the cell center, the laminar viscosity at ghost node is modified. The shortest distance between the interfacial node and immersed wall, y, is determined using the same algorithm as used for locating mirror points.

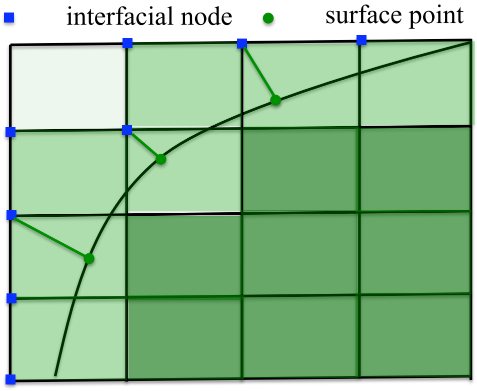

Figure 9.5: interfacial nodes and corresponding surface points are used to implement wall function in connection with immersed boundary method

As discussed before, surface roughness can significantly affect the predicted speed-up of the wind flow over a topography. Therefore it is necessary to use modified wall functions based an equivalent sand-grain roughness. Specifically instead of using Eq. 3.58, the wall friction velocity is determined from Eq. 3.62.

The roughness height, z_0, at the interfacial node, is determined from a two-dimensional map of land cover. For a given land cover, the corresponding roughness height can be specified. Required surface distribution of roughness height determined by bilinear interpolation.

9.3 Validation and Results

In the next sections, the implementation of immersed boundary method is validated for laminar flow over cylinder and low Reynolds number turbulent flow over two-dimensional hill. Next, the method is applied for simulations of wind flow over Askervein Hill and Bolund Hill and the performance for microscale wind simulations over complex terrain is assessed.

9.3.1 Separating Flow over Cylinder

The accuracy of the method for the laminar solver is examined for a low Reynolds number laminar flow around a circular cylinder. Figure 3 shows the comparison between the surface pressure coefficient calculated using the present approach and experimental measurements [55]. Velocity streamlines are also shown in Figure 9.6. The agreement between the pressure coefficient predicted by IBM and the experimental data is quite good. The separation occurs at$\theta = 129.8^o$ compared to a measured value from Coutanceau and Bouard [38]

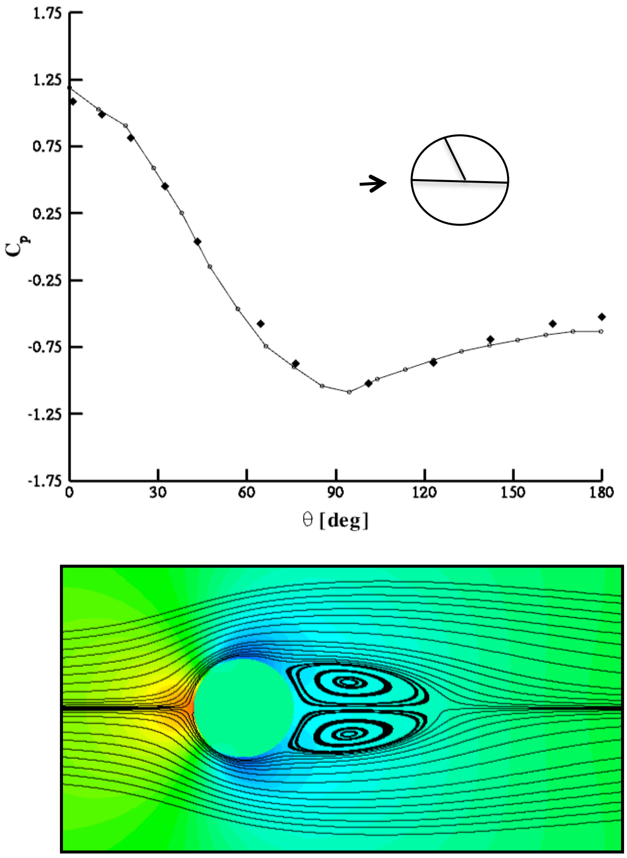

Figure 9.6: Simulation of Reynolds number Re=40 flow over a circular cylinder. Upper plot: comparison of IBM solution to experimental results. Lower plot: streamlines superimposed on pressure contours.

equal to 126.2^o. The implementation of the IBM for the vertex base solver, separate from turbulence modeling is validated through this case.

9.3.2 Flow over Two-Dimensional Hill

In the following the performance of the immersed boundary method is first examined in the flow over a two-dimensional hill. Results from IBM are also compared with those from a body-fitted grid to isolate the effect of wall boundary condition on the results. A three-dimensional test case of flow over Askervein Hill is also examined. The 2D test case is the sinusoidal hill defined by Kim and Lee [79], schematically shown in Figure 9.7 . The shape of the hill is defined using Eq. 9.20 where $H = 11.5\ cm$ and $L = 7\ cm$.

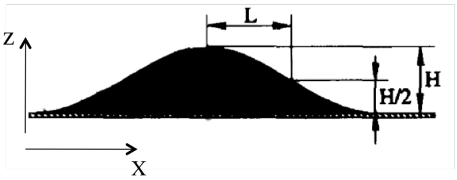

Figure 9.7: Definition of the geometry for the 2D hill, L is the upwind half-length of the hill at the one-half height of the hill.

$$z(x) = \frac{H}{2}\left(1 + cos(\frac{\pi x}{2L})\right) \tag{9.20}$$

The chord length and the height of the curved hill are 0.46 meters and 0.07 meters respectively. The unit Reynolds number based on the free stream velocity ($U_\infty = 7m/s$) is 7.5×10^5 per meters. For simulations with a classical body-fitted grid, an H-grid with clustering in the x and z directions is used (Fig. 9.8). Also shown in Figure 5 is the corresponding Cartesian grid for the IBM simulations.

For the simulations the upstream boundary is $3C$ upstream of the windward foot of the hill. The inflow streamwise velocity is a fully developed turbulent boundary layer with a thickness of 0.25 m. The outflow boundary is located 5C downstream of the leeward foot of the hill. A quantitative assessment of the immersed boundary method is given in Figures 9.9 . With the $k-\omega$ turbulence model, the predicted pressure coefficients with the immersed boundary method and body-conformal grid are in excellent agreement with each other, as well as with the experiment. However, the predictions of the Baldwin-Lomax turbulence model are in poor agreement with the experiment at the leeward corner of the hill. Numerical results show that there is a separated flow in this region, which results in an over-prediction of the minimum pressure at the hillcrest. However, for the Baldwin-Lomax turbulence model, a comparison of IBM and GA results shows the maximum difference observed for the pressure coefficient is less than 0.03 and they are in fairly good agreement. Profiles of the non-dimensionalized velocity speed-ups defined by Eq. 9.21 at the hilltop

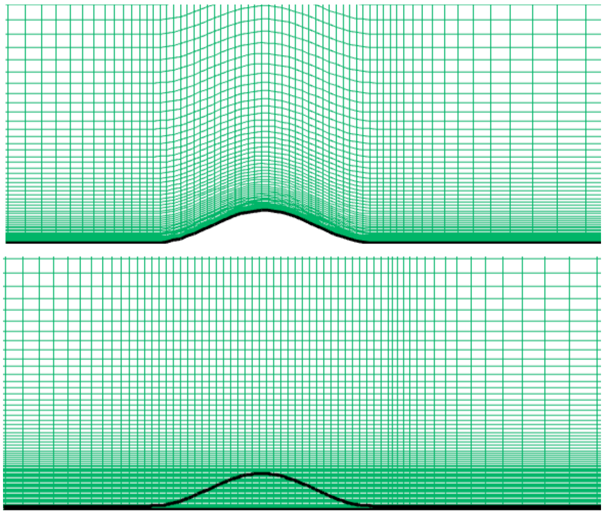

Figure 9.8: Typical H-grid (upper plot) and non-uniform Cartesian grid (lower plot) used for the grid-aligned with flow and immersed boundary method simulations, respectively.

are presented in Figure 9.9.

$$\sigma = \frac{u(z) - u_{ref}(z)}{\frac{H}{L} u_{\infty}} \tag{9.21}$$

Good agreement is observed between the IBM and the body-fitted grid for both turbulence models in the outer region ($z/L > 0.2$) of the boundary layer. Small discrepancies are observed closer to the wall; these discrepancies are attributed to differences between the grids and the resolution of the boundary layer. Overall, it is seen that both the pressure and velocity speed-up are better predicted using the $k - \omega$ turbulence model. In general, the results demonstrate that the immersed boundary technique is capable of resolving the flow field

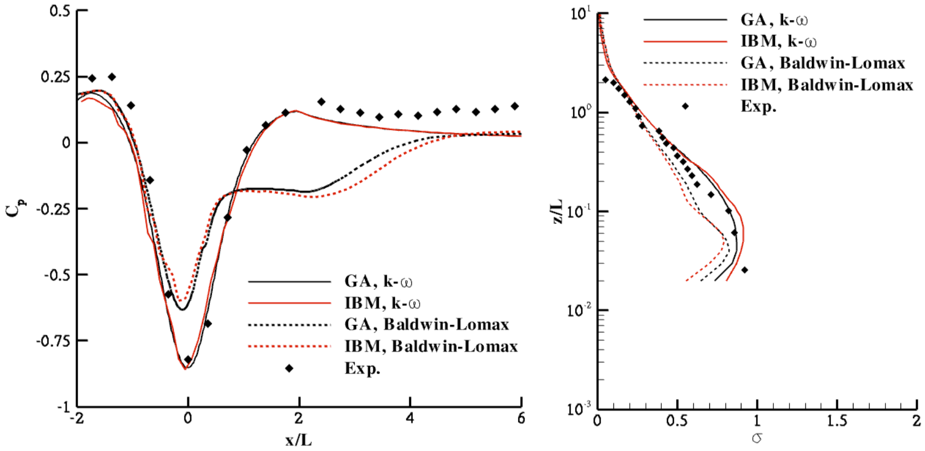

Figure 9.9: Comparison of the pressure coefficient over the hill (left) and non-dimensionalized speed-ups at the hilltop (right).

in high Reynolds number flows.

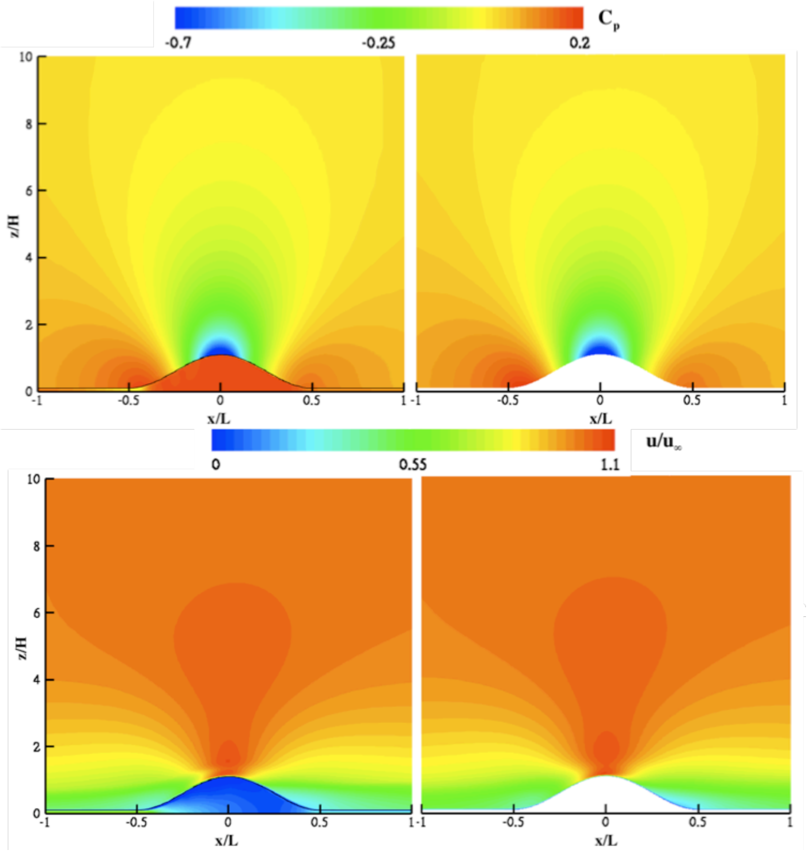

Figure 9.10: Comparison of pressure coefficient (upper plots) and axial velocity (lower plots) over the two-dimensional hill in grid-aligned with geometry and immersed boundary method simulations. Note, for the immersed boundary method the flow properties below the surface have no physical meaning.

Figure 9.10 shows a comparison between the pressure coefficient and the non-dimensionalized axial velocity over the hill with IBM and the body-fitted grid. The overall qualitative agreement between the IBM and the solution obtained with the classical body-fitted grid is quite satisfactory.

9.3.3 Flow over Askervein Hill

The flow over a three-dimensional hill is also investigated using the newly developed method. The Askervein Hill test case described in chapter 6 is chosen for this purpose. As is shown in the lower plot of Figure 9.11 the digital elevation map of the area, downstream of Askervein Hill is a hill with a similar elevation. Another hill of slightly higher altitude is located to the east. There are two maps available for Askervein Hill. Map A with 63 m resolution which also includes the surrounding topography and map B covering only the isolated hill with 23 m resolution. Other than the resolution, there are few differences between the map A and B including the elevation of the hilltop, the elevation along line B-B and the topography downstream of the hilltop. In the current study, simulations are done using both maps (Fig. 9.11).

For grid generation, as mentioned above, digital elevation maps of the terrain are used.

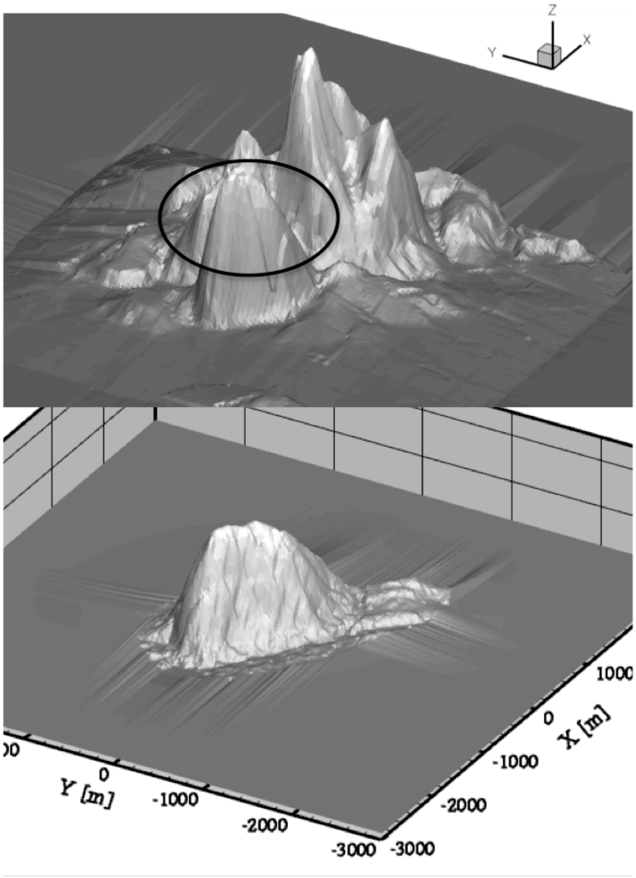

Figure 9.11: Digital elevation maps of Askervein Hill. Note that the z direction is stretched by a factor 10 compared to the x and y directions. Lower plot shows the actual map of Askervein Hill with its surrounding topography (map A). Upper plot shows map B which omits the elevated topographic features that are downwind of Askervein Hill

The maps are rotated by 60 degrees, so that the flow direction is collinear to the x-axis going through the centre point (CP), (Fig. 9.12). The non-uniform Cartesian grid has a resolution of 16 m in the x direction and 20 m in the y direction for the clustered region defined by an area spanning 700 m by 500 m around the center between CP and HT. The size of the entire domain is 8,000 m by 8,000 m in x and y directions for map A and 6,000 m by 6000 m for map B; for both maps the domain is 1,000 m in the z direction and the cell height is 3.0m for the entire hill. These dimensions result in a total of 0.6 million computational nodes. The velocity and turbulent kinetic energy inflow profiles are specified following Kim et al. [80]. The roughness height is set to 0.03 m for the entire terrain. The available experimental data include measurements of the speed-ups along the lines A-A, AA-AA, B-B and at the hilltop

(HT) as shown in Figure 10.

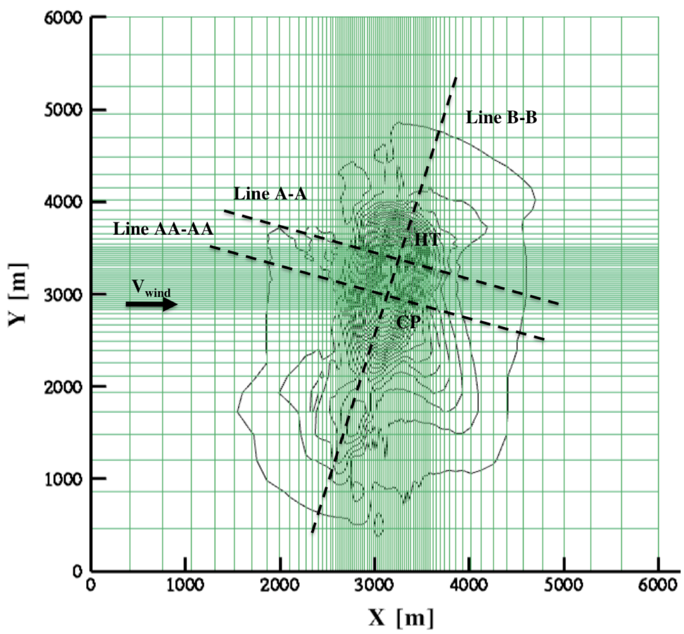

Figure 9.12: Askervein Hill topography map superposed with the non-uniform Cartesian grid used for the simulations (map B). The grid is clustered around the hilltop (HT) and the center point (CP) of the computational domain. Also shown is the orientation of the lines (A-A, AA-AA and B-B) along which measurements are available, are also shown.

The predicted speed-ups along lines A-A and AA-AA and B-B are compared to experiment in Figures 9.13 and 9.14 and 9.15. Also shown in the lower portion of the figures is the elevation change along the respective lines of measurements. The speed-up is defined based on the undisturbed velocity upstream of the hill at the reference station. The results are also compared to simulations of Walmsley and Taylor [159] and Undheim et al., [151]. The prediction over line A-A is generally in good agreement with experiment. However, the computed speed-up does not follow the steep decrease of ΔS on the lee side of the hill using either map A or B. At x=0, speed-up is 0.69 compared to the 0.86 of the experiment in the simulations for the isolated hill. Including the downwind hills does not affect the flow downstream but changes the maximum speed-up significantly. This difference is thought to be due to limitations in turbulence and roughness modeling. Local minima are observed at about x =-700, -350 and -200 m resulting from the changes in topography. The latter is more pronounced in the current simulation, but has also been observed by Undheim et al. [151]. The over-prediction of the low velocity in the wake, downstream of the hill, where non-linear effects are dominant, can be due to turbulence modeling uncertainties in the prediction of flow separation.

The predictions over line AA-AA, Figure 9.14 , match well with the experiment except at $x = 400$ m and $x = 600$ m, where the speed-up is over predicted. The local minimum at

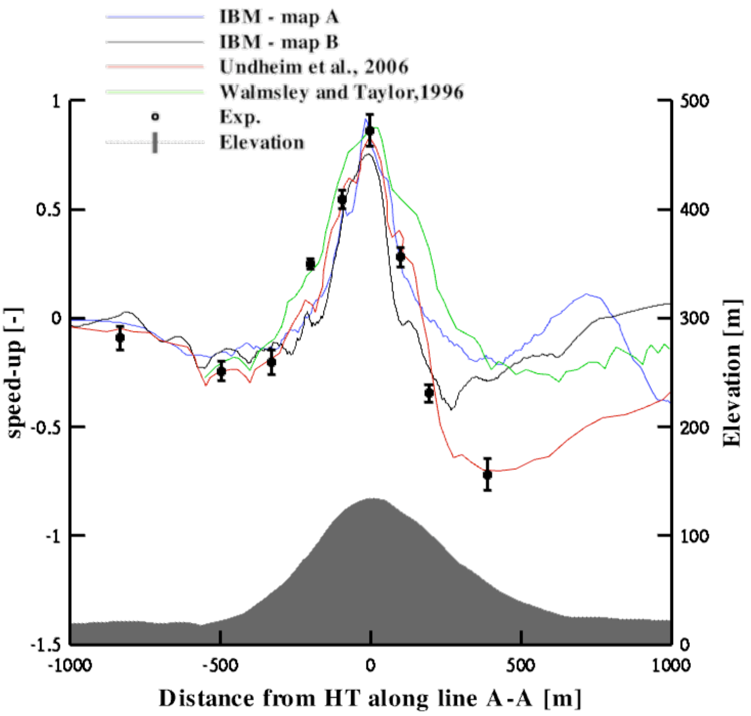

Figure 9.13: Comparison of the predicted and measured non-dimensionalized speed-ups along line A-A on Askervein Hill. The predictions are from simulations using the immersed boundary method.

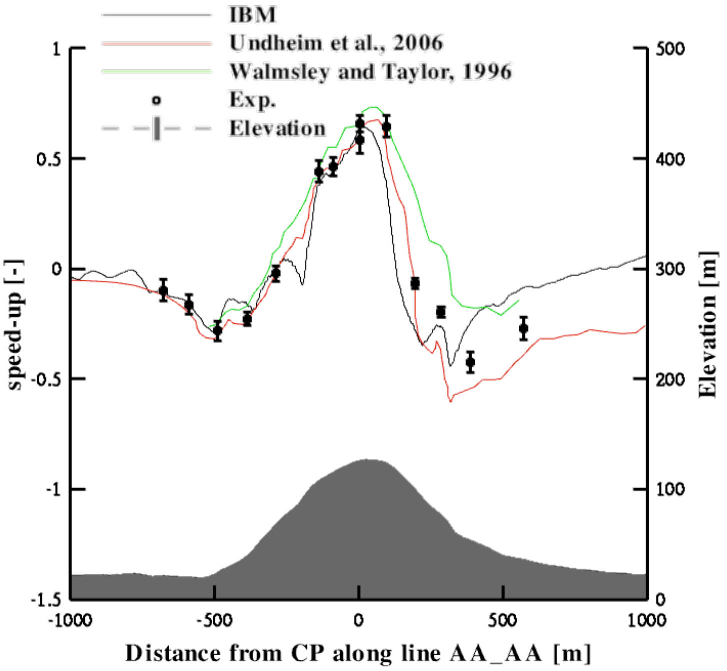

Figure 9.14: Comparison of the predicted and measured non-dimensionalized speed-ups along line AA-AA shown in Figure 10 on Askervein Hill.

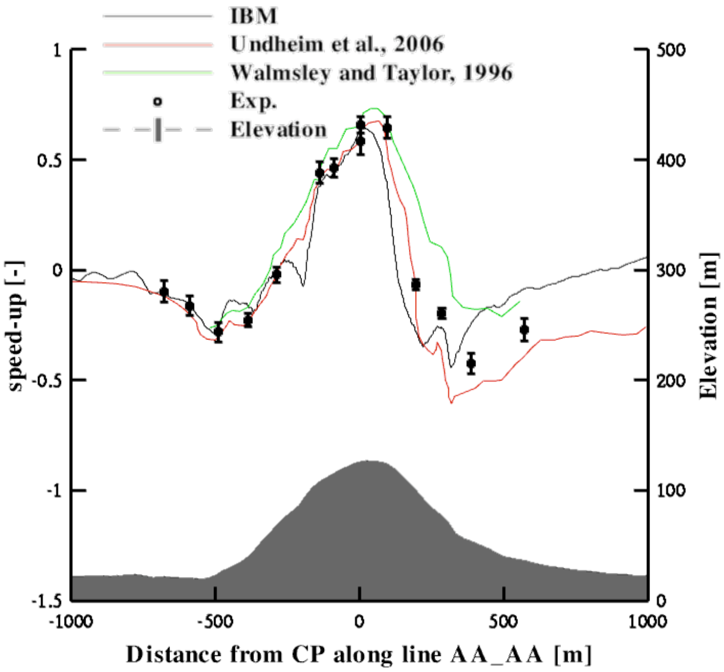

Figure 9.15: Comparison of the predicted and measured non-dimensionalized speed-ups along line B-B on Askervein Hill (map A).

$x = -500\ m$ in the field measurements due to the change in topography (see the elevation map) is captured correctly with the IBM method. The observed increase in velocity downstream of CP ($x = 0\ m$) as well as the sharp change in velocity at $x = 450\ m$ are consistent with the elevation change and are also observed in other simulations [80] [151].

The effect of topography on the predicted and measured speed-ups, along line B-B, are compared in Figure 9.15. The predictions are obtained using map A, and show good qualitative agreement with the measured wind speed-up.

The predicted speed-up at the hilltop is shown in Figure 9.16 and compared to the simulations of Undheim et al. [151]. Except for region very close to the surface ($z < 5m$), the IBM prediction matches well with Undheim's simulations, as well as with the experimental data. The difference close to the surface can be reduced by using finer grids or a higher order interpolation close to the surface. More details of the accuracy of the method close to the ground will be discussed in section 9.4.

The near surface velocity vectors, 5 m above the ground are also shown in Figure 9.17. In Figure 9.17, the velocity vectors are superimposed on the contour plot of velocity field. Accelerated and decelerated flow regions can be observed, as expected, at the hilltop and downstream of the hill.

In general, the immersed boundary method is observed to perform well in flowfield predictions for full-scale scenarios. At potential hub heights, where the measured speed-up is more than 60%, the predicted speed-up is within 1% of the measurement. Thus it is evident

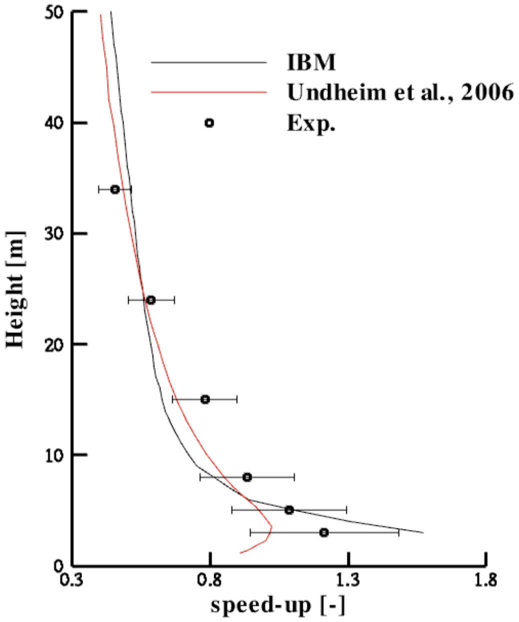

Figure 9.16: Comparison of the predicted and measured non-dimensionalized speed-up at hilltop.

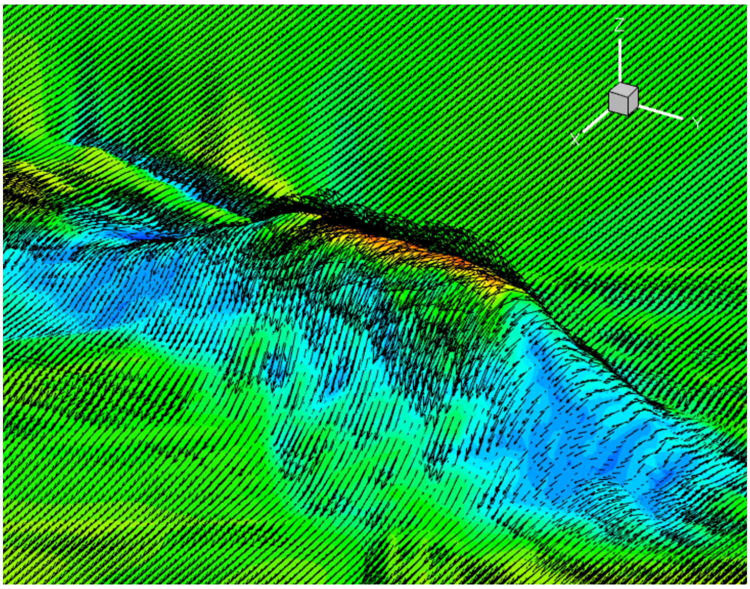

Figure 9.17: Near surface velocity vectors superimposed on velocity contours over Askervein Hill. Note that the z direction is stretched by a factor 5 compared to the x and y directions.

that the immersed boundary method that has been developed in the present work can provide desired accuracy that the wind industry requires for the optimum micrositing of wind turbines.

9.3.4 Flow over Bolund Hill

The immersed boundary method implemented in MULTI3 is applied to the complex terrain test case, Bolund Hill described in chapter 6. The simulation is done for four wind directions, 90^o, 239^o, 270^o and 301^o. The extent of the computational domain is 1500, 750 and 300 meters in the x,y and z directions respectively where x is along the wind axis. u is the velocity along x axis, and s is the measure of the wind speed. Cartesian non-uniform grids with local clustering around the hill are used. Four grids with different resolution and grid density, whose characteristics are detailed in Table 9.18, are used. Figure 9.19 shows a side view of the first computational grid close to the vertical escarpment. The two-dimensional map of surface roughness is used.

Grid	**Δx_{min}[m]**	**Δy_{min}[m]**	**Δz_{min}[m]**	**No. nodes**
a	1.1	1.1	0.4	**1×10^{6}**
b	1.1	1.1	0.8	**7×10^{5}**
c	2.2	2.2	0.4	**4.5×10^{5}**
d	2.2	2.2	0.8	**3×10^{5}**

Figure 9.18: Summary of characteristics of the computational grids used in the simulations.

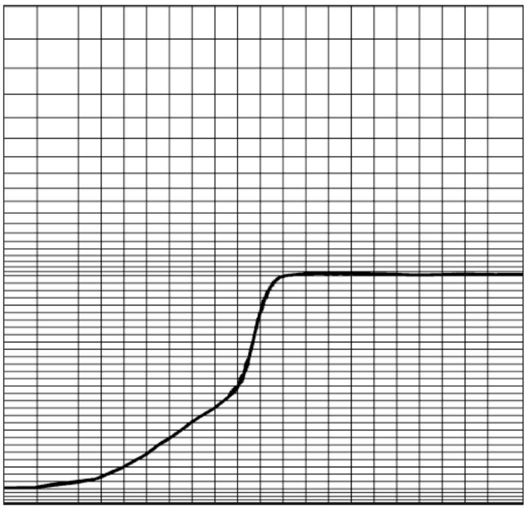

Figure 9.19: Side view of the computational grid A at the leading edge of the hill.

The simulations are carried out with Wilcox's standard $k-\omega$ turbulence model without any modifications of the constants. Since the upper boundary is located far enough, 300 m above see level; it is treated as a slip wall. Locating the upper boundary further away, at 500 m did not change the results. The side boundary conditions are specified as periodic.

As suggested in the comparison, the measured profile is specified at the inlet plane, even though the reference station is located further away.

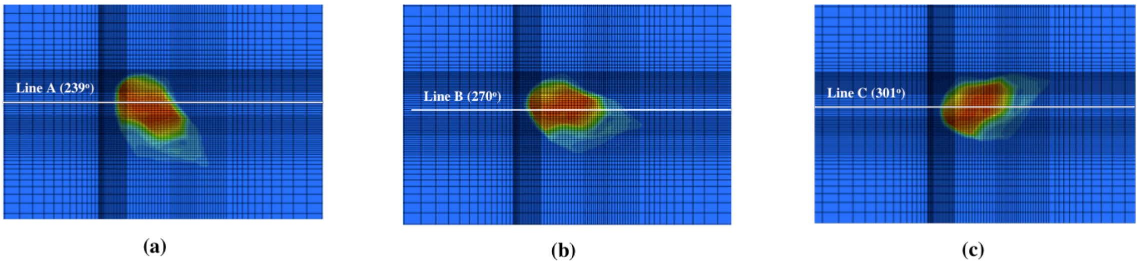

Figure 9.20: Non-uniform Cartesian grid superimposed on digital elevation map of Bolund Hill for simulations with three wind directions, 239^o, 270^o and 301^o. The same grid is used for all directions and only the rotated digital elevation map is provided as an input to the solver

9.3.4.1 Wind Calculation for Various Directions

In wind applications, velocity speed-ups at hub height are of interest for the all wind directions. To simulate the flow with different directions, one can use circular computational domains. In this case, one grid can be used for the all computations. However, due to uncertainties in defining the inflow conditions, rectangular grids-one for each are often used. In this case, to cover the whole wind rose, several grids must be created. On the other hand, using the immersed boundary method, depending on the geometry, as few as one rectangular grid can be used for the whole wind. In other complex terrain cases more rectangular grids may be required. This number of grids will nevertheless be substantially reduced. In order to simulate altered wind directions using IBM, only the information of the rotated digital elevation map needs to be provided as an input to the solver. In some of the scenarios, where the topography is surrounded by flat terrain, such as at Bolund Hill, even the initial and boundary conditions, including those at the inlet remain unchanged (Figure 9.20).

To demonstrate this capability of using one grid for multiple wind directions, the flow with three different wind directions are simulated for Bolund. Figure 9.20 shows the non-uniform Cartesian grid superimposed on the digital elevation map for 239^o, 270^o and 301^o. The two-dimensional roughness map is also rotated accordingly. The grid A is used for all three simulations. Figure 9.21 shows the speed-ups, 5 m above ground, along lines A, B and C that are indicated in Figure 9.12. The simulations for 239^o and 270^o wind direction cases are compared to the experimental data. The information of the surface elevation along the lines A, B and C are also shown in the same figure, so that one can relate the distribution of speed-up to the elevation change in the topography. The results show that on the single Cartesian grid, wind variations are captured for all the three directions with acceptable accuracy.

Wind flow simulations over Bolund Hill are also performed for the 90^o wind direction. For this wind direction, the digital elevation map of the terrain shown in Figure 9.27 is rotated by 180^o (see Figure 9.22). A new grid with characteristics similar to grid A but with clus-

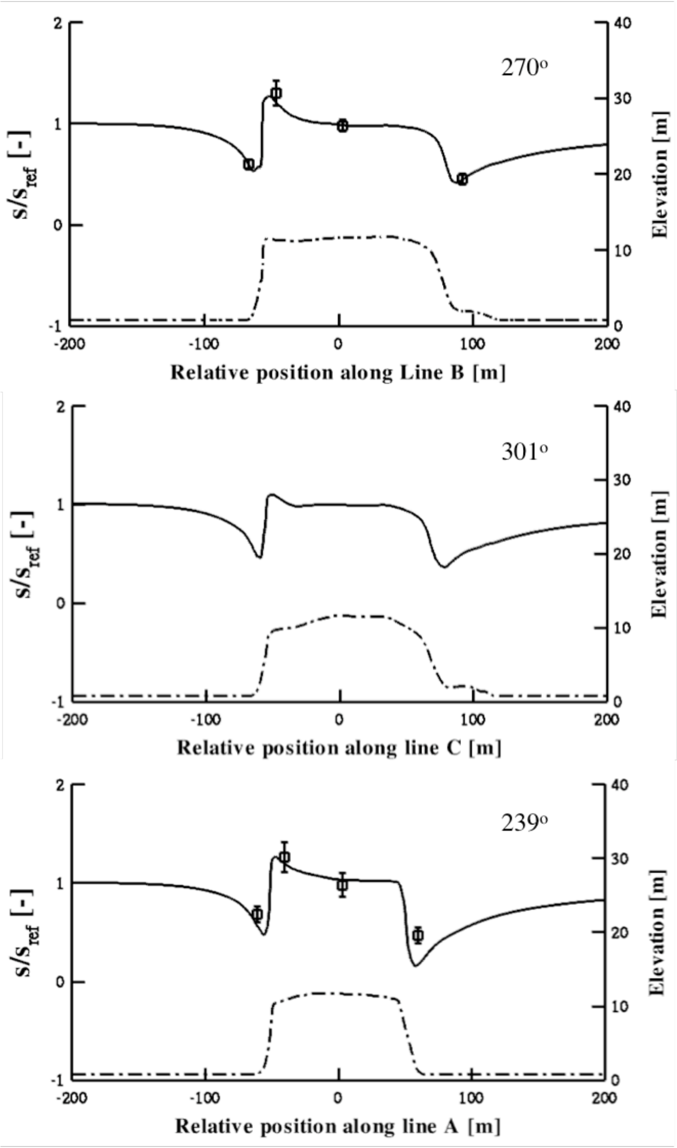

Figure 9.21: Comparison of normalized wind speed (solid line) over the Bolund Hill for three different wind directions, 239^o, 270^o and 301^o, 5 m above ground. The elevation of the topography is shown as a dashed-dotted line

tering reversed in x direction is used. The inlet boundary condition and surface roughness map are also modified for this case since wind is blowing over the land covered with grass at the inlet boundary.

Figure 9.23 shows the streamwise distribution of the normalized wind speed at 2 and 5 m above the ground for 90^o wind direction. The wind speed for this direction is normalized using the undisturbed wind speed at mast 9. The predicted wind speeds at masts 8, 6 and 3 matches the experiment but the agreement at mast 7 for both heights is relatively poor. This can be due to poor prediction of the separation region downstream of the hill. The

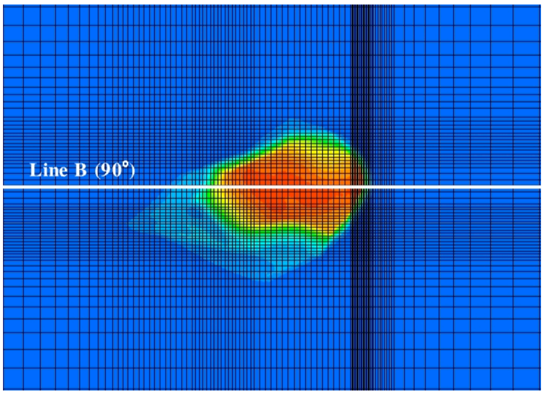

Figure 9.22: Digital elevation of Bolund Hill superimposed on non-uniform Cartesian gird generated for westerly wind directions.

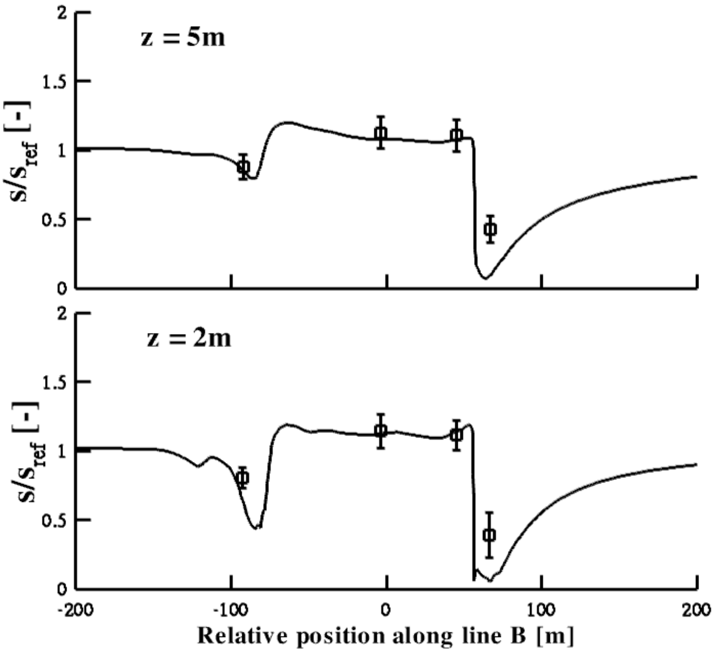

Figure 9.23: Comparison of normalized wind speed over the Bolund Hill for 90^o *wind direction along line B, 2 and 5 m above ground*

separation zone, which can be observed at 2 m and not 5 m AGL, is under predicted in the simulations. Uncertainties in turbulent modeling can be considered as the main reason for the poor behavior. Moreover, the assumption of steady flow conditions used in RANS may not be strictly valid in separation zones, because these regions are inherently unsteady.

To further analyze the performance of the immersed boundary method, a detailed comparison of the flow field with experimental data is performed. The 270^o wind direction is also used to study the grid dependency. Figure 9.24 shows the comparison of the speed-up between the present results and the experimental data at 2 and 5 m above ground, on all

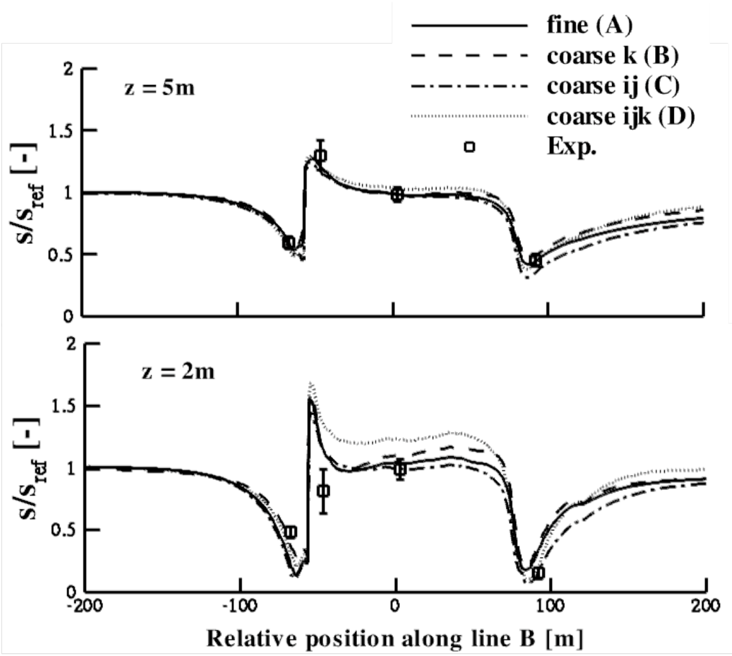

Figure 9.24: Comparison of normalized wind speed for the 270^o *wind direction at 2 and 5 m above the surface along line B.*

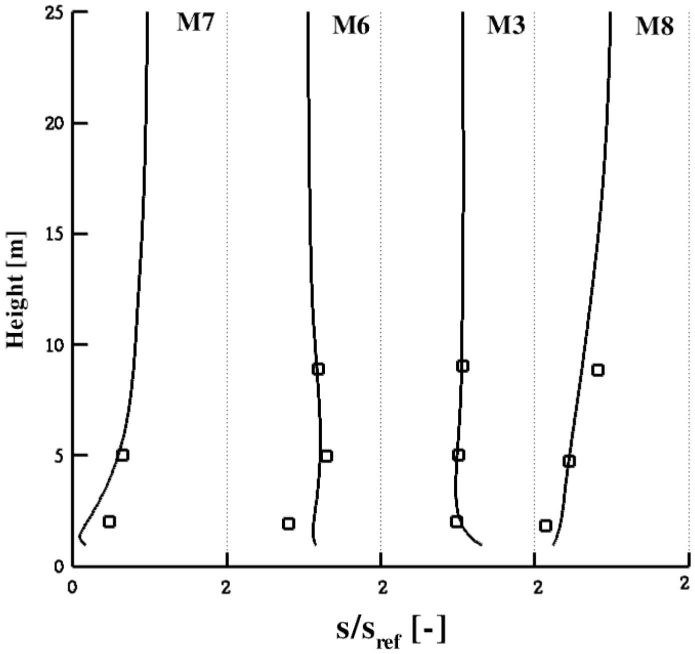

Figure 9.25: Vertical profiles of wind speed at four mast locations. M3, M6, M7 and M8 are the meteorological masts along line B [17] for the 270^o *wind direction.*

four grids. As is observed in Figure 9.24, coarsening the grid in z direction does not affect the results significantly. On the otherhand, reducing the resolution in x-y direction destroys the accuracy of the method in regions close to the solid boundary. However this error diminishes 5 m away from the wall where all grids predict the speed-up fairly well. The largest deviation from the measurement is observed for mast 7 at two meters above ground where the simulations show small regions of separated flow. At this point the predicted speed-up is 0.20 compared to measured 0.49.

Vertical profiles of normalized wind velocity obtained with the first grid at four mast stations are also compared in Figure 9.25. Flow separation occurs at the windward escarpment, leeward slope and southern side of the hill. The difference between predictions and experiment is the largest close to the surface. This difference may arise from several possible sources. Firstly, as mentioned before, there are uncertainties due to turbulence modeling when flow separation occurs. Secondly, the modeling of surface roughness may be another source of uncertainty. This is particularly the case when there is an abrupt change in the roughness height. A third possible source of error is the use of immersed boundary method and the interpolation scheme that is used to impose the boundary conditions at the no-slip wall. A weighted-average interpolation based on an inverse distance is used to specify the flow quantities of the ghost nodes. For the moderate Reynolds number flow over the 2D hill, 9.7, the scheme yields satisfactory results. However for the higher Reynolds numbers that are typical of atmospheric flows; the large gradients in the cell adjacent to the terrain are not well resolved. This can explain the observed trend in the gradient of wind speed in Figure 9.25 at the grid points close to the ground, which is opposite to the predictions that are made using classical body-fitted grids. A higher order interpolation may reduce the differences seen near the surface. Nevertheless, the differences between experiment and predictions are substantially reduced away from the ground at heights relevant for wind energy applications.

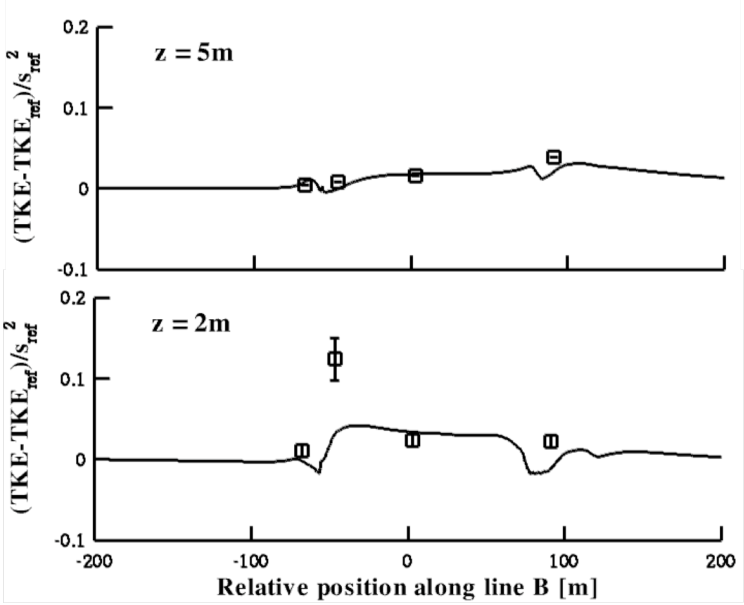

Figure 9.26: Comparison of predicted non-dimensionalized turbulent kinetic energy to experiments for the 270^o wind direction case at 2 and 5 m above the surface along line B.

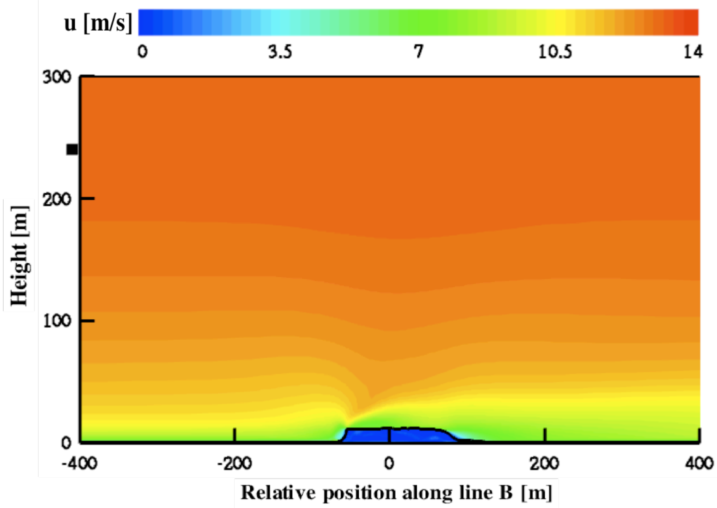

Figure 9.27: Flowfield of velocity contours over Bolund Hill along line B for the 270^o *wind direction*

The streamwise evolution of turbulent kinetic energy obtained using grid A is shown in Figure 9.26. The quantitative agreement between the predictions and measurements at 5 m above ground level is generally good. At 5 m height, the predicted turbulent kinetic energy matches well with the experiment, except at x=90 m where the prediction is 50% lower than the measurement. However, the prediction of turbulent kinetic energy is not good close to the ground (2m A.G.L.) both quantitatively and qualitatively. This further shows the relatively weak performance of the method in near wall regions. Horizontal velocity contours are also shown in Figure 9.27 in a vertical plane through the line B for 270^o wind direction.

For wind direction 239^o, vertical profiles of wind speed at four different mast locations are shown in Figure 9.28. The predictions, similar to those of the 270^o direction shown in Figure 9.25, deviate from the experiment adjacent to the terrain. The vertical profile in the deceleration zone (mast M4) is not well predicted. The profiles of turbulent kinetic energy at mast 1 and 2 are also shown in Figure 9.29. The prediction at mast 2 matches well with experiment at all heights, but at mast 1 the values are under predicted in the simulations. This is also observed in other RANS simulations performed using classical body-fitted grids [16].

For micrositing of wind turbines, other than wind speed and turbulent kinetic energy, local variations of wind inclination and turning are of interest. This helps the developers to assess the "quality" of the wind in addition to its availability. Wind inclination and turning angles are also calculated over Bolund Hill for the 239^o wind direction and compared to experiment. Figures 9.30 and 9.31 show the wind inclination and turning angles at four mast location along line A. As can be seen in both figures, the quantitative and qualitative agreements of both angles at mast 1 are predicted well in the simulations. At mast 2 where separation occurs, the variation of turning and inclination angles are not well predicted. Similarly poor agreement is observed in all of RANS results presented in the blind comparison [16]. This further indicates the poor behavior of turbulence models in regions

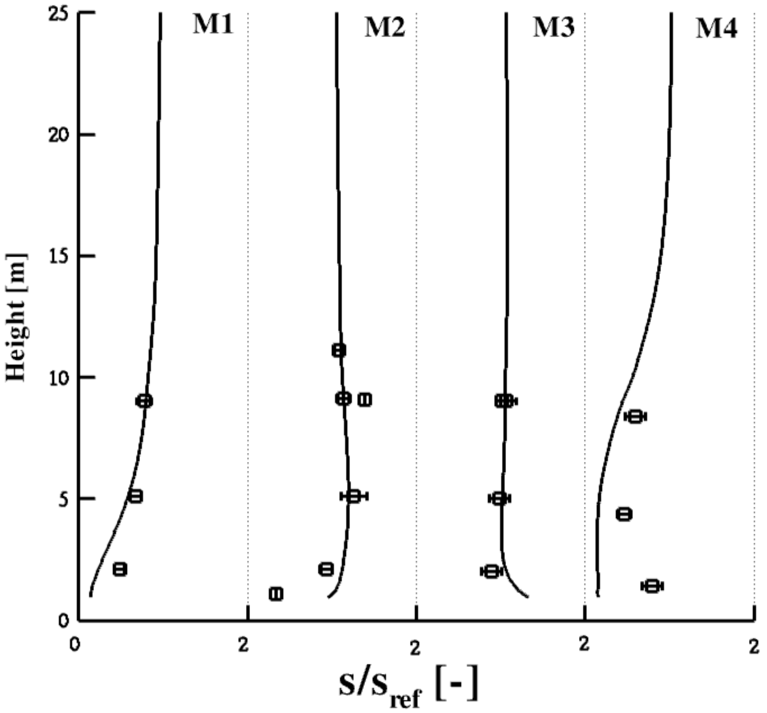

Figure 9.28: Vertical profiles of wind speed at four mast locations. M1, M2, M3 and M4 are the meteorological masts along line A for wind direction 239^o.

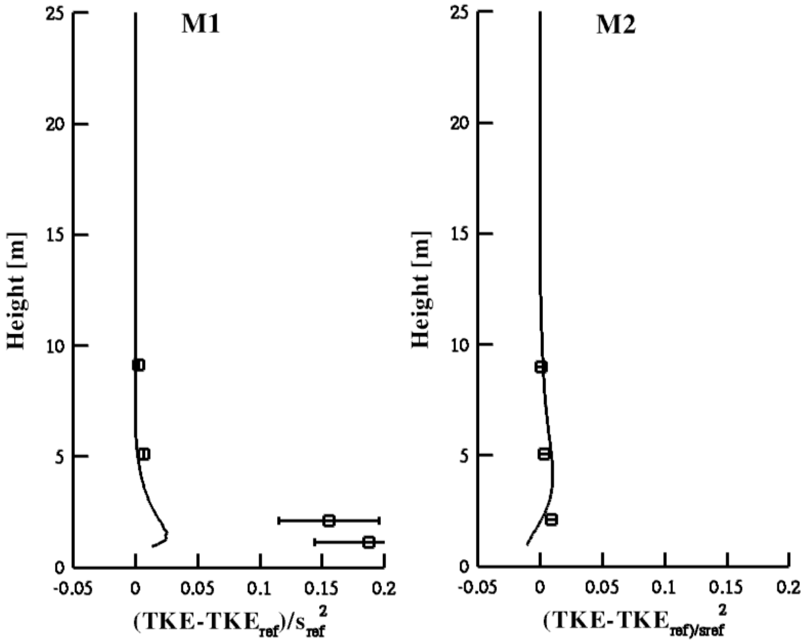

Figure 9.29: Vertical profiles of non-dimensionalized turbulent kinetic energy at mast 1 and 2 for 239^o *wind direction.*

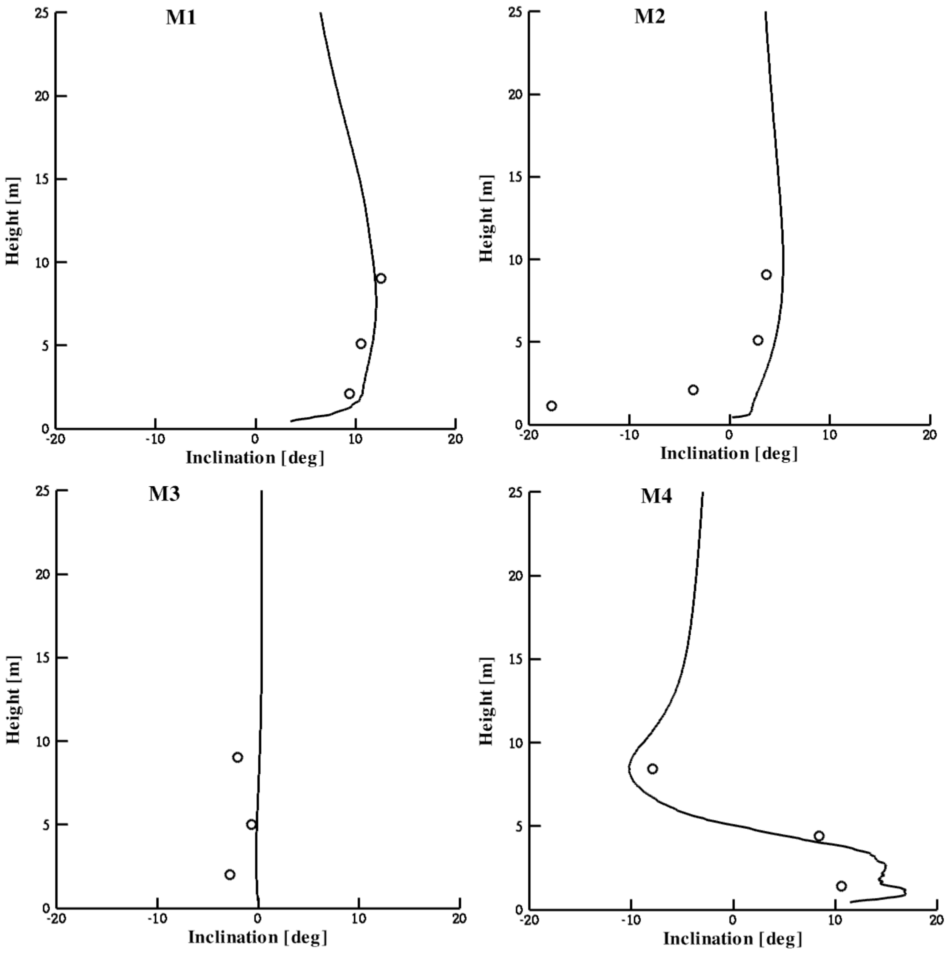

Figure 9.30: Prediction of inclination angle at four mast locations, M1, M2, M3 and M4 compared to measurement for wind direction 239^o

of separated flow and shows that the coordinate of the center of the separation bubble is not predicted correctly using RANS. At mast 3, the predictions and experiment show small variations in wind angles. At mast 4, the wind inclination is predicted well at all heights. The qualitative trend of the turning angle is also predicted well. However quantitatively, the turning angles are largely over estimated close to the wall.

All computations are done on a high-performance Linux cluster. The computational time per iteration per node is 2.11×10^{-6} seconds on four Opteron 8384 processors and approximately 20,000 iterations are required to reduce the residual to 10^{-6}. Iteration per node with the same set up omitting the immersed surface takes 2.07×10^{-6} seconds on the same platform. Thus the IBM carries an insignificant penalty (2%) in computational time. However it is well known that in IBM simulations, the grids are often not optimized in terms of distribution of grid points. Thus, approximately 30 – 50% of grid nodes are located below the immersed surface and do not carry meaningful information. These non-optimal grids may result in large computational time for the simulations. For the two-dimensional hill where the turbulent boundary layer needs to be resolved using IBM, the Cartesian grid is approx-

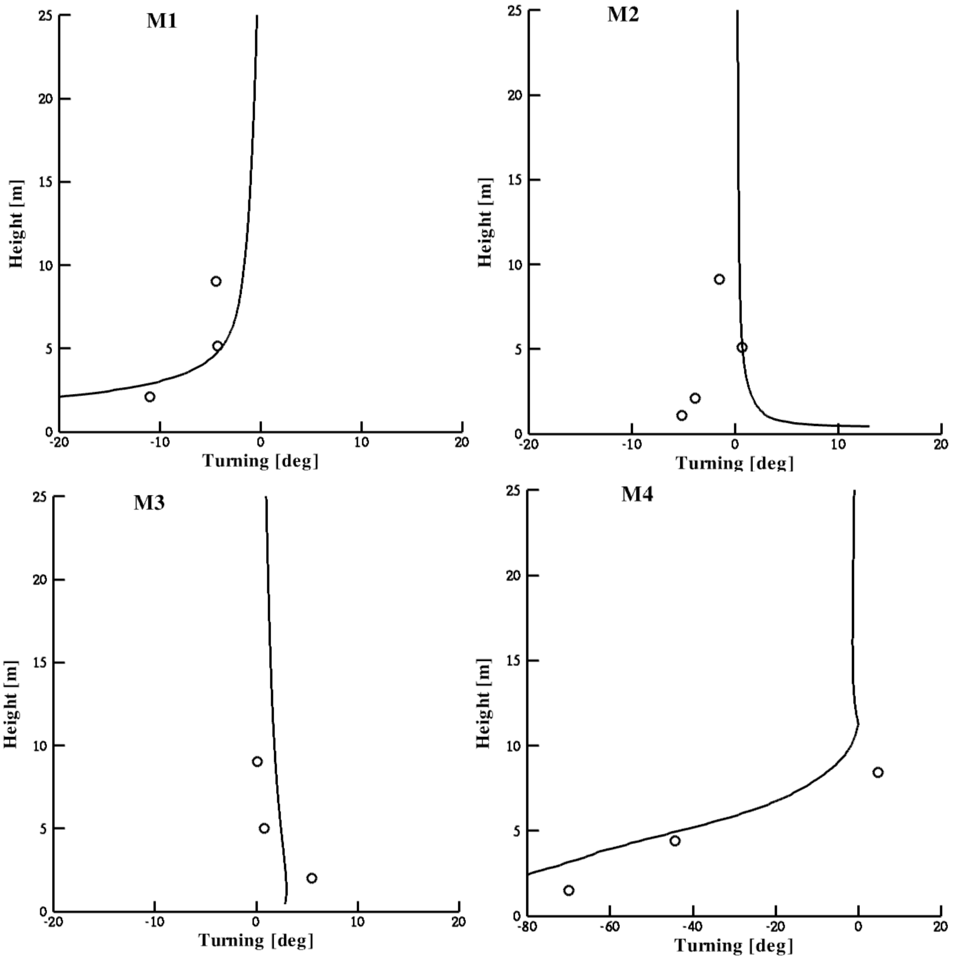

Figure 9.31: Prediction of turning angle at four mast locations, M1, M2, M3 and M4 compared to measurement for wind direction 239^o.

imately three times larger than the body-fitted grid. Therefore, the simulation time is correspondingly larger. The use of optimized Cartesian grids would substantially reduce the computational time. Nevertheless, results from the Bolund Hill simulations show that for atmospheric flow simulations, accurate predictions can be obtained using relatively small grid size. It must be emphasized that the overall assessment of the computational efficiency of the immersed boundary method for atmospheric flow must also consider the required time for grid generation.

9.4 Model Accuracy Close to Ground

As mentioned in previous sections, the implementation of immersed boundary method for high Reynolds number atmospheric flow is rather challenging. The implementation of a second order no-slip condition close to the ground using coarse grid typicall of wind simulations results in negative velocities at cells close to the ground which sometimes leads to instabilities in the solver. The first order boundary conditions is easy to implement and relatively robust. However, it results in inaccuracies close to ground depending on the location of the immersed boundary with respect to the ghost node.

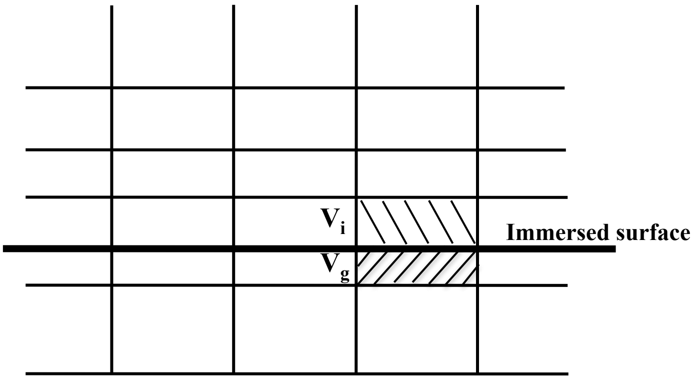

Figure 9.32: Immersed flat plate in a Cartesian grid.

The inaccuracy is further demonstrated using a case of immersed flat plate on a Cartesian grid (Figure 9.32). The thickness of the first node is chosen such that by moving the immersed surface between two nodes the y^+ value of the first node above virtual surface varies between 50 and 100. Figure 9.33 and 9.34 show that how the velocity and turbulent profiles deviate from the theoretical and fully resolved solution as the distance of the immersed surface from the ghost node increases.

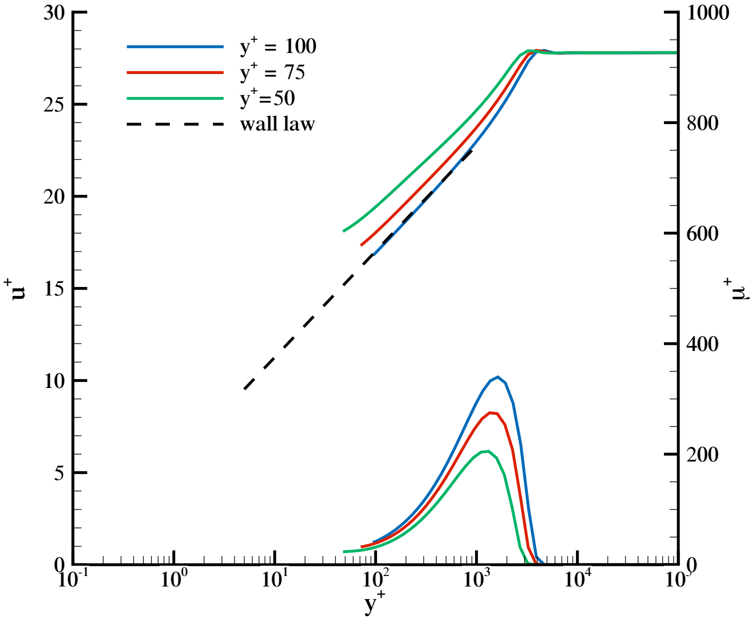

Figure 9.33: Poor performance of immersed boundary method in prediction of velocity and eddy viscosity in vicinity of the immersed wall as the distance between the virtual wall and ghost nodes increases.

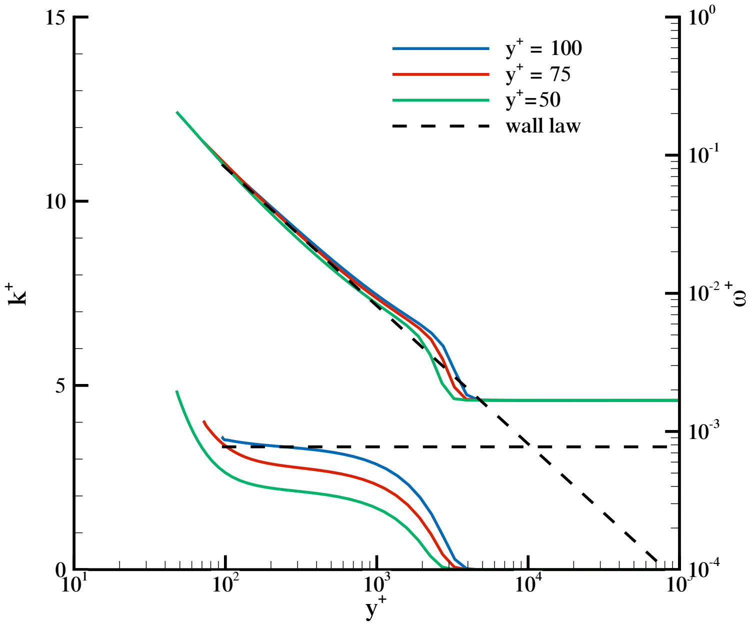

Figure 9.34: Prediction of k and ω over flat plate using immersed boundary method.

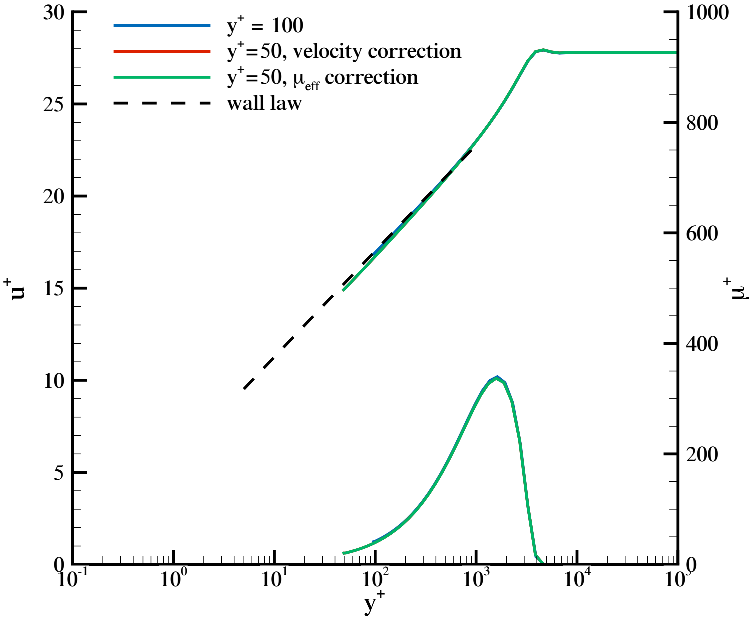

Figure 9.35: Prediction of velocity and eddy viscousity in boundary layer using the corrections suggested in Eq. 9.22 and 9.23 in connection with immersed boundary method

Improving the accuracy of immersed boundary method close to ground may not be of crucial importance for the wind energy application, however from a numerical perspective, it is desirable to overcome this problem and further improve the accuracy of the method. Since the implementation of higher order schemes poses some other challenges such as negative flow variables within interfacial cells, an alternative for improving the accuracy is

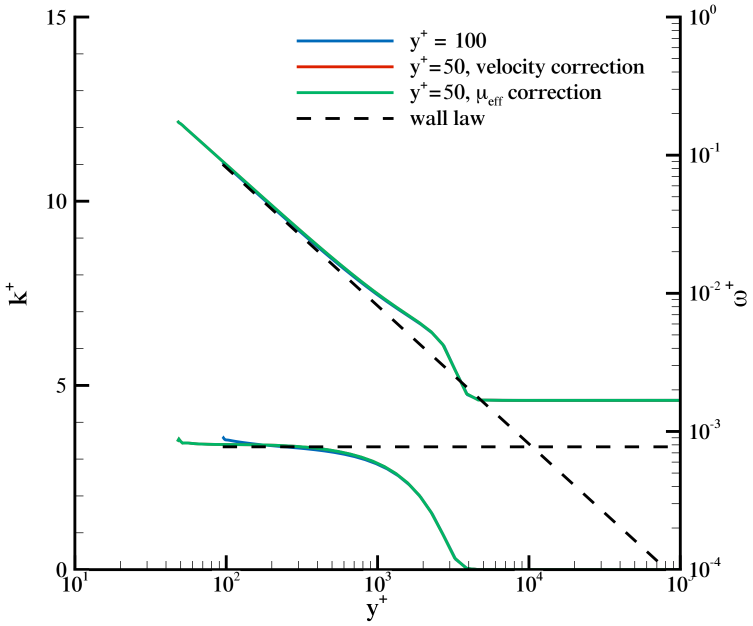

Figure 9.36: Prediction of k and ω in boundary layer using the corrections suggested in Eq. 9.22 and 9.23.

investigated in this work.

The new methodology first suggested by Ackerman et al. [1] is focused on correcting the velocity at the interfacial node rather than altering the velocity at ghost node. The use of wall functions for atmospheric flow simulations, further facilitates this approach, since an analytical solution for velocity in wall adjacent node is given (Eq. 9.22)

$$\frac{u_2}{u_1} = \frac{ln\left(\frac{z_2}{z_0}\right)}{ln\left(\frac{z_1}{z_0}\right)} \tag{9.22}$$

In Eq. 9.22 z_1 is the distance of the first node above the wall with velocity u_1 and z_2 is the height of the second node above the immersed surface with velocity u_2 and z_0 is the roughness height. This equation could be easily derived by applying the law of the wall (Eq. 6.1) to the two first nodes above the wall. As shown in Figure 9.35 and 9.36 using Eq. 9.22 at all distances above the immersed surface, excellent agreement is achieved. The extension of this approach for three-dimensional flows with pressure variations must be investigated further since the Eq. 9.22 is not valid anymore in presence of large streamwise and perpendicular pressure gradient.

The second approach suggested by [109] is to account for the varying distance between the first node and the immersed surface by scaling the effective viscosity introduced by the wall function implementation using the ratio of cell's volume to the non-immersed volume. The modified effective viscosity defined at the interracial cell-center becomes:

$$\mu_{eff_{IBM}} = \frac{1}{8} \sum_{node=1}^{8} \frac{V_i + V_g}{V_g} \mu_{eff} \tag{9.23}$$

The scaling of the effective viscosity is justified considering the fact that only part of the cell volume that must be considered in defining the wall shear must be the sector bounded between the immersed body and the first vertices connecting the interracial nodes. Similar to the previous approach, the scaling of the effective viscosity proves to be effective in correcting the results close to the ground over the flat plate (Figure 9.35 and 9.36) but the application in 3D was not reported to be successful.

The above-mentioned alternative approaches could be further followed and investigated in order to improve the accuracy of the immersed boundary method close to the wall boundary once applied to high Reynolds number atmospheric flow which are often used with coarse grids.

9.5 Summary

An immersed boundary method has been developed for high Reynolds number flows. The method works in connection with zero-equation Baldwin-Lomax and two-equation $k-\omega$ turbulence model. Using the immersed boundary method, a Cartesian grid is used to simulate the flow over any arbitrary topography. Hence a single grid can be used to cover the whole wind rose. The wall functions are integrated to the method in order to decrease the grid resolution requirements close to the surface as well as to allow the surface roughness to be accounted for. The implementation of the model into laminar solver is validated for the laminar flow over cylinder. The prediction of separation point is compared with measurements. The capability of the model to simulate high Reynolds number flows is examined for the flow over a two-dimensional hill. The results were then compared to the simulations obtained using a body-fitted grid to isolate the effect of boundary conditions. Good agreement with the experiment as well as with the results obtained with a body-fitted grid demonstrates the capability of the immersed boundary method. To show the capability of the model in handling full-scale scenarios, a three-dimensional flow was simulated over Askervein Hill. In addition, the three-dimensional flow over complex terrain test case, Bolund Hill was simulated. The general agreement of the predicted speed-up over the hill was good at 5 m above the ground. The fact that the wind speed predictions are satisfactory at 5 m above ground level for the 12 m height Bolund Hill gives evidence that the method can also provide efficient predictions at hub height of a wind turbine in complex terrain. Furthermore, results obtained with single grid for various wind direction also were compared with experiment. This also demonstrates the relevance of the method to wind energy application, since the same Cartesian grid, which is easily generated, can be used to produce results for different wind directions of a wind rose. Furthermore, the immersed boundary method enables the solver to handle highly complex geometries with large gradients and sharp corners with higher stability and accuracy.

CHAPTER

10

Summary and Conclusion

In the context of this thesis, an explicit multistage time integration method primarily suitable for compressible flow simulations was developed. In order to have an all purpose solver suitable for low Mach number flows as well as the high Mach number compressible flows, the Weiss-Smith preconditioning algorithm was implemented into the hybrid Runge-Kutta scheme, and its application was demonstrated for low Mach number high Reynolds number atmospheric flow. The required modifications and improvements for successful use of preconditioning for microscale wind simulations are also reported. To further accelerate the convergence and reduce the computational cost, a multigrid approach was implemented into the preconditioned solver. The wall function formulation and implementation which is necessary for wind flow simulations are also described in this thesis.
Next, the simulations are performed over several test cases including Kettles Hill, Askervein Hill and Bolund Hill to evaluate the performance of the RANS solver in prediction of wind flow over moderately and highly complex topography and address the shortcomings of the method. The simulations were performed using $k-\omega$ turbulence model with modified coefficients tuned for atmospheric flow simulations. The predictions of wind flow speed-up are in general good agreement with the measurement for all cases both at hilltop and downstream. Overall the average error in predicted speed-up at all points is 5%, 40% and 8.5% for Kettles, Askervein and Bolund Hills respectively. Even though the overall uncertainty of RANS modeling are expected to increase as the terrain complexity increases, but predictions show smaller errors over Bolund Hill compared to Askervein Hill. This is due to the relatively large under estimation of speed-up upstream of the hill along line A-A. Neglecting those points, the errors reduces to 14.5% over Askervein Hill. This could also be partially due to more ideal and well-defined inflow and terrain conditions as well as the lower uncertainties in the measurements. The turbulent kinetic energy is also compared to measurements over Askervein and Bolund Hill. The average error at 10 m above ground for Askervein Hill is 36% and over Bolund Hill is 25% for a wind direction of 270^o using the preconditioned solver.

The flow predictions and the convergence behavior of the preconditioned solver and preconditioned multigrid is discussed and compared to that of unpreconditioned scheme. The results overall demonstrates the superior convergence and robustness of the preconditioned solver. It was shown that preconditioning could result in 4-13x speed-up when used for simulation of microscale wind flow over topography. The preconditioned solver also shows a superior accuracy compared to unpreconditioned scheme except for the case of Bolund Hill at 2 m above ground level. In this case preconditioning seems to suppress the separation bubble at leading edge of the hill. This could be partially due to the coupling of the error of turbulence modeling with the excessive dissipation that is present in the unpreconditioned solver. However the exact reason is unclear.

In the next step, a novel model is developed for simulating the wake flow behind a wind turbine. In the developed model, called immersed wind turbine model, the flow field close to the rotor and near wake region are modeled. The resultant flow field is mapped at the end of the near wake region and the remaining far wake is resolved on the computational grid. The model is proposed as an alternative to the commonly used actuator disc model in order to bring the gird requirements of the microscale wind simulations and wake flows within wind farms closer to each other. Using such an approach, the simultaneous simulations of wind and wake flow becomes feasible. The velocity deficit at the end of near wake is introduced using a Gaussian profile. The value for the deficit is obtained using the mass conservation within the body which shape represents the energy extraction and the blockage introduced by the rotor. The boundary conditions of the model is implemented on the Cartesian grid using the immersed boundary method. The formula for the added turbulent kinetic energy by the rotor and the equation for ω is derived based on empirically driven relations available in literature.

The performance of the model is first demonstrated for a case of single wake of a sub-scale wind turbine operating in a wind tunnel and the single and double wakes measurements of Sexbierum wind farm. The comparison of results with wind tunnel measurements at different operating conditions shows a good agreement in prediction of velocity at 5 and 7 diameter downstream of the turbine. The value of turbulent intensity is over predicted by the model at all operating points. The excessive turbulence results in higher mixing and therefore a reduced velocity deficit at 2.5D downstream compared to measurements. The qualitative shape of the profile and evolution of turbulence intensity however are captured. The comparison of the model prediction for single wake simulations in Sexbierum wind farm overall shows a good agreement in terms of both prediction of velocity deficit and turbulent intensity at 2.5D and 8D distance.

The predictions of the model for double wake interactions are also examined. The simulations show 10% power loss from second to third turbine for the fully merged condition (0^o wind direction). The difference reduces once the turbines are partially merged in the upstream wake. The sensitivity of power loss to wind direction was also investigated for Lillgrund wind farm. The simulations were performed for 11 different wind directions. In two of the directions the power loss was greatly over-estimated by the model. However the average relative error for remaining 9 directions was 9%.

Last the performance of two wind farms located in complex terrain in Spain and Switzerland were studied and estimations of array loss was compared with SCADA data. The simulations are performed for 3 and 6 directions over two farms respectively. The first wind farm located in Spain consists of 43 turbines located on moderately complex terrain. The simulations overall show RMS error of 0.038, 0.215, 0.293, and 0.16 in four consecutive rows. The results show 150 %, and 34 %, 27% and 38% improvements in rows 1 to 4 compared to results obtained using WAsP reported in [116] which employs linear models. It must be also noted that the terrain slope is at the limits of the WAsP accuracy. Additional more complex test cases need to be investigated. The simulations are also performed for farm with similar layout and boundary conditions located on flat terrain to investigate the effect of wake-terrain interaction on the estimations. The results show up to 20% faster recovery and up to 40% increase in predicted power due to effects related to topography. Overall larger sensitivity to wind direction is observed for the farm located in moderately complex topography compared to the case with flat terrain. For the Mont Crosin wind farm located in Switzerland the relative power of two turbines in the farm is studied for southerly-westerly wind directions when the underlying topography has steep slopes and both turbines go in the wake shadow of upstream turbines. The average error between measurements and simulations for all 6 directions is 14% . The error is the largest for wind direction of 230^o when the power deficit of downstream turbine is maximum. The maximum power deficit is predicted as 88% compared to measured value of 71%.

In the final part of this thesis, the problem of grid generation for the wind flow simulations was addressed. Since the immersed wind turbine model allows for the simulation of different turbine arrangement using a single grid, the feasibility of simulation of multiple wind directions using the same grid could further facilitate the use of Computational Fluid Dynamics for site optimization purposes. The popular approach of immersed boundary method for wall modeling used for simulation of low and medium Reynolds number flow, was applied to very high Reynolds number microscale wind simulations. The method was implemented in connection with Ni's Lax-Wendroff finite volume scheme and $k-\omega$ turbulence model. The performance and shortcomings of the model were then assessed for flow over Askervein and Bolund Hill. Overall, even though the performance of the method was satisfactory for prediction of wind speed-up at 10 m above ground level over Askervein Hill and 5 m above ground level over Bolund Hill, The performance of the method proved to be quite poor in closer distances to the ground. The capability of the method for flow simulation of multiple wind direction using single Cartesian grid was demonstrated for the case of Bolund Hill where wind directions of 239^o, 270^o and 301^o were performed resulting in satisfactory predictions at 5 m above ground level. While the accuracy of the method at heights relevant to wind energy application were demonstrated, further suggestions are provided for improvement of the accuracy of the method close to the ground. The extension of the methodologies to three-dimensional flow requires further investigations.

10.1 Future Work

The application of preconditioned multistage solver for atmospheric and wake flow simulations were demonstrated in this work, however additional investigation in some aspects could further reduce the computational time and improve the accuracy of the solver. In terms of preconditioning formulations, even though the Weiss-Smith preconditioning proved to be quite effective, different preconditioning formulations could be implemented to assess the performance and robustness of different schemes for this particular application. In addition, different possibly more effective multigrid cycles such as full-miltigrid cycle could be used to further improve the convergence rate.

GPU acceleration currently underway could also provide considerable reduction in computational time compared to CPU parallelization. This could reduce the computational time by a factor of 10-1000x making the RANS solver a competitive design tool. With the current computational costs in the CPU version, it is still difficult to obtain grid independent results over large wind farms such as Lillgrund or Horns Rev using RANS solver. Additional accelerations could further facilitate the path for use of CFD for array loss estimations and micrositing.

In regards to the immersed wind turbine model, further improvements could be accomplished by generalizing the ω boundary condition at the end of the near wake. New empirical formulas could be developed to account for wake meandering as well as the effect of averaging over different bands of wind direction in measurements. The accuracy of the model must be further improved in order to justify the additional computational cost and compete with already established and rather cheap engineering models used in the wind industry. The utility of the model could be greatly increased by integrating the controller algorithm to adjust the operating conditions as well as the nacelle direction of the turbine or so called yaw-alignment. This could remove some of the uncertainties in power prediction of wind farms due to local wind variations that result in the situations where some turbines operate above rated power. The yaw adjustments, also ensures that the immersed body is aligned with the local flow direction which could be different that the global wind direction at inlet plane over a highly complex terrain.

In terms of predictions of wind farm performance, effect of topography and array loss, steady models such as actuator disk model and the presented immersed wind turbine model could be effective tools. They provide acceptable accuracy with reasonable computational time for the lifetime of the turbine. As demonstrated in this thesis, they could also be used in connection with mesoscale data via an "offline" coupling. However, it is also important to understand and predict the unsteady dynamics of the flow due to measoscale wind variations or unsteady interaction of wind flow with topography. As a complementary engineering tool it is desirable to develop and utilize unsteady models (such as actuator line model) in connection with a microscale simulation tool which operates with time-varying wind flow and direction as boundary conditions provided by possibly an online coupling with mesoscale model. This second class of coupled solvers may not be efficient in wind farm performance prediction over the lifetime, but they can provide valuable insight to unsteady interaction of atmospheric boundary layer with wind turbine and wake flow as well as the unsteady loading on wind turbine blades which could be useful in structural design and optimization of wind turbines.

Bibliography

[1] A. Ackerman, I. Senocak, N. Mansour, and D. Stevens, “Topography Modeling in Atmospheric Flows Using the Immersed Boundary Method,” *Center for Turbulence Research, in Annual Research Briefs*, pp. 331–341, 2004.

[2] U. E. I. Administration, “International Energy Outlook 2013,” November 2013. [Online]. Available: http://www.eia.gov/forecasts/ieo/

[3] J. Ainslie, “Calculating the Flowfield in the Wake of Wind Turbines,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 27, pp. 213–224, 1988.

[4] P. Alfredson, J. Dahlberg, and F. Bark, “Some Properties of the Wake Behind Horizontal Axis Wind Turbines,” *Proceedings 3rd International Symposium on Wind Energy Systems*, pp. 469–484, 1980.

[5] T. Allen and A. Brown, “Large-Eddy Simulation of Turbulent Separated Flow over Rough Hills,” *Boundary-Layer Meteorology*, vol. 102, pp. 177–198, 2002.

[6] T. Ansorge, M. Fallen, P. Gunther, C. Ruh, and T. Wolfanger, “Numerical Simulation of Wake-Effects in Complex Terrain and Application of a Reynolds-Stress Turbulence Model,” *Proceedings of European Wind Energy Conference*, pp. 448–453, 1994.

[7] A. Arnone and R. Swanson, “A Navier Stokes Solver for Cascade Flows,” *Technical Report*, vol. NASA CR-181682, 1988.

[8] E. W. E. Association, “Wind in Power: 2012 European Statistics,” November 2013. [Online]. Available: http://www.ewea.org/statistics/

[9] ——, “ Wakes: Do We Need Different Models for Onshore and oOfshore Wind Farms?” January 2014. [Online]. Available: http://www.ewea.org/annual2014/conference/programme/info.php?id=48

[10] W. W. E. Association, “Worldwide Wind Capacity,” November 2013. [Online]. Available: http://www.wwindea.org/home/index.php?option=com_content&task=blogcategory&id=21&Itemid=43

[11] J. Badger, A. Hahmann, X. Larsen, A. Diaz, E. Batchvarova, S.-E. Gryning, R. Floors, and H. Jorgensen, “Comprehensive Utilization of Mesoscale Modelling for Wind Energy Applications,” *Proceedings of European Wind Energy Conference*, 2011.

[12] E. Balaras, “Modeling Complex Boundaries Using an External Force Field on a Fixed Cartesian Grid in Large-Eddy Simulations,” *Computer and Fluids*, vol. 33, pp. 375–404, 2004.

[13] B. S. Baldwin and H. Lomax, “Thin-Layer Approximation and Algebric Model for Seperated Turbulent Flows,” *AIAA Paper*, no. 78-257, 1978.

[14] R. Barthelmie, S. Frandsen, O. Rathmann, K. Hansen, E. Politis, J. Prospathopoulos, D. Cabezon, K. Rados, S. van der Pijl, J. Schepers, W. Schlez, J. Phillips, and A. Neubert, “Flow and Wakes in Large Wind Farms in Complex Terrain and Offshore,” *Proceedings of European Wind Energy Conference*, 2008.

[15] A. Basol, “Turbine Design Optimizations Using High Fidelity CFD,” PhD Thesis, ETH Zurich, 2013.

[16] A. Bechmann, N. Sorensen, J. Berg, J. Mann, and P. Rethore, “The Bolund Experiment, Part II: Blind Comparison of Microscale Flow Models,” *Boundary-Layer Meteorology*, vol. DOI: 10.1007/s10546-011-9637-x, 2011.

[17] A. Beljaars, J. Walmsley, and P. Taylor, “A Mixed Spectral, Finite Difference Model for Neutrally Stratified Boundar-Layer Flow over Roughness Changes and Topography,” *Boundary-Layer Meteorology*, vol. 38, no. 3, pp. 273–303, 1987.

[18] J. Berg, J. Mann, A. Bechmann, M. Courtney, and H. Jorgensen, “The Bolund Experiment, Part I: Flow Over a Steep, Three-Dimensional Hill,” *Boundary-Layer Meteorology*, vol. DOI: 10.1007/s10546-011-9636-y, 2011.

[19] J. Blazek, *Computational Fluid Dynamics: Principles and Applications.* Oxford, UK: ELSEVIER, 2005.

[20] A. Bowen and N. Mortensen, “Exploring the Limits of WasP the Wind Atlas Analysis and Application Program,” *Proceedings of European Wind Energy Conference*, 1996.

[21] A. Brand, J. Peinke, and J. Mann, “Turbulence and Wind Turbines,” *Journal of Physics: Conference Series*, vol. 318, no. 7, 2011.

[22] P. Brodeur and C. Masson, “Numerical Site Calibration over Complex Terrain,” *Journal of Solar Energy Engineering*, vol. 130, p. 031020 (12 pages), 2008.

[23] M. Brower, J. Zack, B. Baily, M. Schwartz, and D. Elliot, “Mesoscale Modeling as a Tool for Wind Resource Assessment and Mapping,” *14th Conference on Applied Climatology*, 2004.

[24] A. Burdet, “A Computationally Efficient Feature-Based Jet Model for Prediction of Film-Cooling Flows,” PhD Thesis, ETH Zurich, 2005.

[25] A. Burdet and R. Abhari, “Three-Dimensional Flow Prediction and Improvement of Holes Arrangement of a Film-Cooled Turbine Blade Using a Feature-Based Jet Model,” *Journal of Turbomachinery*, vol. 129, pp. 58–268, 2007.

[26] ——, “Three-Dimensional Flow Prediction and Improvement of Holes Arrangement of a Film-Cooled Turbine Blade Using a Feature-Based Jet Model,” *Journal of Turbomachinery*, vol. 129, no. 2, pp. 258–268, 2007.

[27] ——, “On the Prediction of Film Cooling with Compound Angle Injection Using a Three-Dimensional Feature-Based Jet Model,” *Progress in Computational Fluid Dynamics*, vol. 8, no. 6, pp. 309–319, 2008.

[28] ——, “Influence of Near Hole Pressure Fluctuation on the Thermal Protection of a Film-Cooled Flat Plate,” *Journal of Heat Transfer*, vol. 131, p. 022202, 2009.

[29] F. Castro, J. Palma, and A. Silvia-Lopes, “Simulation of the Askervein Flow. Part1: Reynolds Averaged Navier-Stokes Equations ($k-\epsilon$ Turbulence Model),” *Boundary-Layer Meteorology*, vol. 107, pp. 501–530, 2003.

[30] P. Chaviaropoulos and D. Douvikas, “Mean Wind Field Prediction over Complex Terrain in the Presence of Wind Turbines,” *Proceedings of European Wind Energy Conference*, pp. 1208–1211, 1999.

[31] T. Chisholm, “Multigird Acceleration of an Approximately-Factored Algorithm for steady Aerodynamic Flow,” MS Thesis, University of Toronto, Toronto Canada, January 1997.

[32] M. Churchfield, S. Lee, P. Moriarty, L. Martinez, S. Leonardi, G. Vijayakumar, and J. Brasseur, “A Large-Eddy Simulation of Wind-Plant Aerodynamics,” *50th AIAA Aerospace Sciences Meeting*, 2012.

[33] J. Cleijne, “Results of the Sexbierum Wind Farm: Single Wake Measurements,” *TNO Report*, pp. 92–388, 1992.

[34] ——, “Results of the Sexbierum Wind Farm: Double Wake Measurements,” *TNO Report*, p. C19.3, 1993.

[35] P. Coppin, E. Bradley, and J. Finnigan, "Measurements of Flow over and Elongated Ridge and its Thermal Stability Dependence: the Mean Flow," *Boundary-Layer Meteorology*, vol. 69, pp. 173–199, 1994.

[36] G. P. Corten, "Flow Separation on Wind Turbines Blades," PhD Thesis, Natuur-en Sterrenkunde, Utrecht University, 2001.

[37] J. Counihan, "Adiabatic Atmospheric Boundary Layer: a Review and Analysis of Data from the Period 1880-1972," *Atmospheric Environment*, vol. 9, pp. 871–905, 1975.

[38] M. Countanceau and R. Bouard, "Experimental Determination of the Main Flow Features of the Viscous Flow in the Wake of Circular Cylinder in Uniform Translation. Part 1: Steady Flow," *Journal Fluid Mechanics*, vol. 79, pp. 231–256, 1977.

[39] A. Crespo and J. Hernandez, "A Numerical Model of Wind Turbine Wakes and Wind Farms," *Proceedings of European Wind Energy Conference*, vol. 2, pp. 111–115, 1986.

[40] ——, "Analytical Correlations for Turbulence Characteristics in the Wakes of Wind Turbines," *Proceedings of European Community Wind Energy Conference*, pp. 436–439, 1993.

[41] ——, "Turbulence Characteristics in Wind Turbine Wakes," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 61, pp. 71–85, 1996.

[42] A. Crespo, J. Hernandez, E. Fraga, and C. Andreu, "Experimental Validation of the UPM Computer Code to Calculate Wind Turbine Wakes and Comparison with other Models," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 27, pp. 77–88, 1988.

[43] A. Crespo, J. Hernandez, and S. Frandsen, "Survey of Modeling Methods for Wind Turbine Wakes and Wind Farms," *Wind Energy*, vol. 2, pp. 1–24, 1999.

[44] J. Dahlberg, "Assessment of the Lillgrund Wind Farm: Power Performance Wake Effects," 2009.

[45] D. Darmofal and B. von Leer, "Local Preconditioning of the Euler Equations: A Characteristic Interpretation," *Von Karman Institute of Fluid Dynamics, Lecture Series*, 1999.

[46] D. Darmofal and B. van Leer, "Local Preconditioning: Manipulating Mother Nature to Fool Father Time. Computing the Future II: Computational Fluid Dynamics and Transonic Flow," 1998.

[47] E. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations," *Journal Computational Physics*, vol. 161, pp. 35–60, 2000.

[48] T. Faxen, “Wake Interaction in Any Array of Wind-Mills. Theory and Preliminary Results,” *Proceedings 2nd International Symposium on Wind Energy Systems*, pp. 59–72, 1978.

[49] A. Ferreira, A. Lopes, D. Viegas, and A. Sousa, “Experimental and Numerical Simulation of Wind Flow Around Two-Dimensional Hills,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 54/55, pp. 173–181, 1995.

[50] S. Frandsen, L. Chacon, A. Crespo, P. Enevoldsen, R. Gomez-Elvira, J. Hernandez, J. Hojstrup, F. Manuel, K. Thomsen, and P. Sorensen, “Measurements on and Modelling of Onshore Wind Farms,” *Final Report EU Contract JOU2-CT93-0350*, vol. Report Riso-R-903(EN), 1996.

[51] R. Frehlich, Y. Meiller, and M. Jensen, “Measurements of Boundary Layer Profiles with In Situ Sensors and Doppler Lidar,” *Atmospheric and Oceanic Technology*, vol. 25, pp. 1328–1340, 2007.

[52] P. Frenzen and C. Vogel, “On the Magnitude and Apparent Range of Variations of the von Karman constant in the Atmospheric Surface Layer ,” *Boundary-Layer Meteorology*, vol. 72, pp. 371–392, 1995.

[53] J. Garrat, “Review of Drag Coefficients over Oceans and Continents,” *Monthly Weather Review*, vol. 105, pp. 915–929, 1977.

[54] D. Goldstein, R. Handler, and L. Sirovich, “Modeling a No-Slip Flow Boundary with an External Force,” *Journal of Computational Physics*, vol. 105, pp. 354–366, 1993.

[55] A. Grove, F. Shair, E. Peterson, and M. Andreas, “An Experimental Investigation of the Steady Separated Flow Past a Circular Cylinder,” *Journal Fluid Mechanics*, vol. 19, pp. 60–80, 1964.

[56] A. N. Hahmann, D. Rostkier-Edelstein, T. T. Wamer, Y. Liu, F. Vandenberg, R. Babarsky, and S. P. Swerdlin, “A reanalysis system for the generation of mesoscale climatographies,” *J. Appl. Meteor. Climatol.*, vol. 49, pp. 954–972, 2010.

[57] M. Hall, “Cell-Vertex Multigird Scheme for Solution of the Euler Equations,” *Proceedings of International Conference on Numerical Methods in Fluid Dynamics*, 1985.

[58] U. Hassan, “A Wind Tunnel Investigation of The Wake Structure within Small Wind Turbine Farms,” *ETSU WN 5113*, 1992.

[59] E. Hau, *Wind Turbines: Fundamentals, Technologies, Application, Economics.* Berlin: Springer, 2005.

[60] K. Hejranfar and R. Kamali-Moghadam, “Preconditioned Characteristic Boundary Conditions for the Solution of the Preconditioned Euler Equations at Low Mach Number Flows,” *Journal of Computational Physics*, vol. 231, pp. 4384–4402, 2012.

[61] C. Helmis, J. Whale, K. Papadopoulos, C. Andersson, D. Asimakopoulos, and D. Skyner, “A Comparative Laboratory and Full-Scale Study of the Near Wake Structure of a Wind Turbine,” *Proceedings of European Wind Energy Conference*, pp. 465–469, 1994.

[62] C. Hirsch, *Numerical Computation of Internal and External Flows.* New York: Ed. John Wiley and Sons, 1990.

[63] U. Hogstrom, D. Asimakopoulos, H. Kambezidis, C. Helmis, and A. Smedman, “A Field Study of the Wake Behind a 2MW Wind Turbine,” *Atmospheric Environment*, vol. 22, pp. 803–820, 1988.

[64] J. Hojstrup and M. Courtney, “Turbulence in Wind Farms,” *Proceedings of European Wind Energy Conference*, vol. 383-386, 1993.

[65] K. Hosseini and J. Alonso, “Practical Implementation and Improvement of Preconditioning Methods for Explicit Multistage Flow Solvers,” *Proceedings of 42nd Aerospace Sciences Meeting and Exhibition*, 2004.

[66] S. Hulshoff, “An Euler Solution Algorithm for Steady Helicopter-Rotor Flows,” PhD Thesis, University of Toronto, 1994.

[67] P. Jackson and J. Hunt, “Turbulent Flow over a Low Hill,” *Quarterly Journal of the Royal Meteorological Society*, vol. 101, pp. 929–955, 1975.

[68] M. Jacobson, *Fundamentals of Atmospheric Modeling.* London: Cambridge University Press, 2005.

[69] S. Jafari, T. Sommer, N. Chokani, and R. Abhari, “Wind Resource Assessment Using a Mesoscale Model: The Effect of Horizontal Resolution,” *Proceedings of ASME Expo Turbo*, no. 69712, 2012.

[70] A. Jameson, “Solution of the Euler Equations by a Multigrid Method,” *Applied Mathematics and Computations*, vol. 13, pp. 327–356, 1983.

[71] ——, “A Vertex-Based Multigrid Algorithm for Three Dimensional Compressible Flow Calculations,” *ASME Symposium on Numerical Methods in Compressible Flow*, 1986.

[72] A. Jameson, R. Schmidt, and E. Turkel, “Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping,” *AIAA Paper*, no. 81-1259, 1981.

[73] L. Jensen, C. Morch, P. Sorensen, and K. Svendsen, “Wake Measurements from the Horns Rev Wind Farm,” *Proceedings of European Wind Energy Conference*, 2004.

[74] G. Kalitzin and G. Iaccariono, “Turbulence Modeling in an Immersed-Boundary RANS Method,” *Center for Turbulence Research, in Annual Research Briefs*, pp. 415–426, 2002.

[75] G. Kalitzin, G. Medic, G. Iaccarino, and P. Durbin, “Near-Wall Behavior of RANS Turbulence Models and Implications for Wall Functions,” *Journal of Computational Physics*, vol. 204, pp. 265–291, 2005.

[76] S. Kang, G. Iaccarino, and F. Ham, “DNS of a Buoyancy-Dominated Turbulent Flow Using an Immersed Boundary Method,” *Journal Computational Physics*, vol. 228, pp. 3189–3208, 2009.

[77] I. Katic, J. Hojstrup, and N. Jensen, “A Simple Model for Cluster Efficiency,” *Proceedings of European Wind Energy Conference*, pp. 407–410, 1986.

[78] M. Kato and B. Launder, “The Modeling of Turbulent Flow Around Stationary and Vibrating Square Cylinder,” *Proceedings of 9th Symposium on Turbulent Shear Flows*, pp. 10.4.1–10.4.6, 1993.

[79] H. Kim and C. Lee, “Pollutant Dispersion Over Two-Dimensional Hilly Terrain,” *KSME International Journal*, vol. 12, pp. 96–111, 1998.

[80] H. Kim, V. Patel, and C. Lee, “Numerical Simulation of Wind Flow over Hilly Terrain,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 87, pp. 45–60, 2000.

[81] C. Kress, N. Chokani, and R. Abhari, “Aerodynamic Effectiveness and Blade Loading of a Multi-Megawatt Wind Turbine,” *European Wind Energy Conference*, 2013.

[82] G. Larsen, J. Hojstrup, and H. Madsen, “Wind Fields in Wakes,” *Proceedings of European Wind Energy Conference*, pp. 764–768, 1996.

[83] D. Lee, “Local Preconditioning of the Euler and Navier-Stokes Equations,” PhD Thesis, University of Michigan, 1996.

[84] W. Lee, “Local Preconditioning of the Euler Equations,” PhD Thesis, University of Michigan, 1991.

[85] J. Leishman, “Challenges in Modeling the Unsteady Aerodynamics of Wind Turbines,” *AIAA Paper*, no. 2002-0037, 2002.

[86] P. Ligrani and R. Moffat, “Structure of Transitionally Rough and Fully Rough Flat Plate,” *Journal of Fluid Mechanics*, vol. 162, pp. 69–98, 1986.

[87] P. Lissaman, "Energy Effectiveness of Arbitrary Arrays of Wind Turbines," *AIAA Paper*, no. 79-0114, 1979.

[88] E. Luken, A. Talmon, and P. Vermeulen, "Evaluation of Two Mathematical Wind Turbine Wake Models in Various Types of Flows," *MT-TNO Report*, vol. 86-07, 1986.

[89] K. Lundquist, K. Chow, and J. Lundquist, "An Immersed Boundary Method for the Weather Research and Forecasting Model," *Monthly Weather Review*, vol. 138, pp. 796–817, 2010.

[90] J. Lynn, "Multigrid Solution of the Euler Equations with Local Preconditioning," PhD Thesis, Department of Aerospace Engineering and Scientific Computing, University of Michigan, 1995.

[91] E. Machefaux, N. Troldborg, G. Larsen, J. Mann, and H. Madsen, "Experimental and Numerical study of Wake to Wake Interaction in Wind Farms," *Proceedings of European Wind Energy Conference*, 2012.

[92] J. Mackenzie, "An Investigation of Artificial Dissipation for the Cell-Vertex Finite Volume Method," *International Journal of Numerical Methods in Fluids*, vol. 21, pp. 499–522, 1995.

[93] M. Magnusson, "A New Approach for Evaluating Measured Wake Data," *Proceedings of European Wind Energy Conference*, pp. 813–816, 1996.

[94] M. Magnusson and A. Smedman, "Infuence of Atmospheric Stability on Wind Turbine Wakes," *Wind Energy*, vol. 18, pp. 139–152, 1994.

[95] ——, "Air Fow Behind Wind Turbines," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 80, pp. 169–189, 1999.

[96] S. Majumdar, G. Iaccarino, and P. Durbin, "RANS Solver with Adaptive Structured Boundary Non-Conforming Grids," *Center for Turbulence Research, in Annual Research Briefs*, pp. 353–366, 2001.

[97] C. Maksymiuk, R. Swanson, and T. Pulliam, "A Comparison of Two Central Difference Schemes for Solving the Navier-Stokes Equations," *Technical Report*, vol. NASA TM-102815, 1990.

[98] M. Mansour, A. Caglar, N. Chokani, and R. Abhari, "Time-Resolved Measurement of the Near Wake of a 2 MW Wind Turbine," *Proceedings of ASME Expo Turbo*, 2013.

[99] M. Mansour, G. Kocer, C. Lenherr, N. Chokani, and R. Abhari, "Seven-Sensor Fast-Response Probe for Full-Scale Wind Turbine Flowfield Measurements," *J. Engineering for Gas Turbines and Power*, vol. 133, no. 8, pp. 081 601–081 601–8, 2011.

[100] L. Martinelli, "Calculation of Viscous Flows with a Multigrid Method," PhD Thesis, Department of Mechanical Engineering, Princeton University, 1987.

[101] J. Mason and R. Sykes, "Flow over an Isolated Hill of Moderate Slope," *Quarterly Journal of the Royal Meteorological Society*, vol. 105, pp. 383–395, 1979.

[102] C. Mass, D. Ovens, K. Westrick, and B. Colle, "Does Increasing Horizontal Resolution Produce More Skillful Forecasts?" *American meteorological Society*, pp. 407–430, March 2002.

[103] A. Maurizi, J. Palma, and F. Castro, "Numerical Simulation of the Atmospheric Flow in a Mountainous Region of the North of Portugal," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 74, pp. 219–228, 1998.

[104] D. Mavriplis and A. Jameson, "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes," *AIAA Journal*, vol. 28, pp. 1415–1425, 1990.

[105] J. T. McQueen, R. R. Draxler, and G. D. Rolph, "Influence of grid size and terrain resolution on wind field prediction from an operational mesoscale model," *J. Appl. Meteor.*, vol. 34, pp. 2166–2188, 1995.

[106] F. Menter, "Zonal Two Equation k-w Turbulence Model Predictions," *AIAA Paper*, no. 93-2906, 1993.

[107] R. Mickle, J. Salmon, and P. Taylor, "Kettles Hill 84: Velocity ProïňĄle Measurements over a Low Hill," *Boundary Layer Research Division, Air Quality and Inter-Environmental Research Branch, Atmospheric Environment Service, Downsview*, vol. M3H 5T4, 1984.

[108] A. Mills and H. Hang, "On the Skin Friction Coefficient for a Fully Rough Flat Plate," *Journal of Fluid Engineering*, vol. 105, pp. 364–365, 1983.

[109] C. Mortier, "Generalized Wall Function Implementation in MULTI3 for Wind and Turbomachinery Applications," MS Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, April 2012.

[110] R. Ni, "A Multiple Grid Scheme for Solving the Euler Equations," *AIAA Journal*, vol. 20, pp. 1565–1571, 1981.

[111] P. Palma, M. Tullio, G. Pascazio, and M. Napolitano, "An Immersed-Boundary Method for Compressible Viscous Flows," *Computers and Fluids*, vol. 161, pp. 693–702, 2006.

[112] H. Panofsky and J. Dutton, *Atmospheric Turbulence*. New York: Wiley Interscience, 1984.

[113] T. Pedersen, S. Petersen, K. T. P. Madsen, and J. Hojstrup, “Loads for Wind Turbines in Inhomogeneous Terrain Measurements,” *Report Riso-M-2922*, 1991.

[114] C. Peskin, “Flow Patterns around Heart Valves: a Numerical Method,” *Journal of Computational Physics*, vol. 10, pp. 252–271, 1972.

[115] E. Politis and P. Chaviaropoulos, “Micrositing and Classification of Wind Turbines in Complex Terrain,” *Proceedings of European Wind Energy Conference*, pp. 126–130, 2008.

[116] E. Politis, J. Prospathopoulos, D. C. K. Hansen, P. Chaviaropoulos, and R. Barthelmie, “Modelling Wake Effects in Large Wind Farms in Complex Terrain: the Problem, the Methods and the Issues,” *Wind Energy*, vol. 15, pp. 161–182, 2012.

[117] F. Porte-Agel, C. Meneveau, and M. Parlange, “A Scale-Dependent Dynamic Model for Large-Eddy Simulation: Application to Neutral Atmospheric Boundary Layer,” *Journal Fluid Mechanics*, vol. 415, pp. 261–284, 2000.

[118] M. Prospathopoulos, E. Politis, and P. Chaviaropoulos, “Modeling Wind Turbines in Complex Terrain,” *Proceedings of European Wind Energy Conference*, 2008.

[119] G. Raithby, G. Stubley, and P. Taylor, “The Askervein Hill Project: A Finite Control Volume Prediction on Three-Dimensional Flows over the Hill,” *Boundary-Layer Meteorology*, vol. 39, pp. 107–132, 1987.

[120] J. Rodrigo, D. Cabezon, and B. Hevia, “A Systematic Validation Procedure for Wind Farm Models in Neutral Atmospheric Conditions,” *Proceedings of 13th International Conference on Wind Engineering*, pp. 147–155, 2011.

[121] E. Saiki and S. Biringen, “Numerical Simulation of a Cylinder in Uniform Flow: Application of a Virtual Boundary Method,” *Journal Computational Physics*, vol. 123, pp. 450–465, 1996.

[122] J. Salmon, H. Teunissen, R. Mickle, and P. Taylor, “The Kettles Hill Project: Field Observation, Wind-Tunnel Simulations and Numerical Model Predictions for Flow over a Low Hill,” *Boundary-Layer Meteorology*, vol. 43, pp. 309–343, 1988.

[123] R. Salvador, J. Calbo, and M. M. Milan, “Horizontal grid size selection and its influence on mesoscale model simulation,” *J. Appl. Meteor.*, vol. 38, pp. 1311–1329, 1999.

[124] B. Sanderse, “Aerodynamics of Wind Turbine Wakes,” *Technical Report, ECN-E-09-016*, 2009.

[125] A. Saxer, “A Numerical Analysis of 3D Inviscid Stator Rotor Interactions Using Non-Reflecting Boundary Conditions,” PhD Thesis, Massachusetts Institute of Technology, 1992.

[126] J. Schepers, “ENDOW: Validation and Improvement of ECN’s Wake Model,” *Technical Report, ECN-C-03-034*, 2003.

[127] H. Schlichting and K. Gersten, *Boundary-Layer Theory.* La Canada: Springer, 2000.

[128] M. Schwartz, “Wind Resource Estimation and Mapping at the National Renewable Energy Laboratory,” *NREL/CP-500-26245*.

[129] P. Sforza, P. Sheering, and M. Smorto, “Three-Dimensional Wakes of Simulated Wind Turbines,” *AIAA Journal*, vol. 19, pp. 1101–1107, 1981.

[130] A. Silvia-Lopes, J. Palma, and F. Castro, “Simulation of the Askervein Flow. Part 2: Large-Eddy Simulations,” *Boundary-Layer Meteorology*, vol. 125, pp. 85–108, 2007.

[131] R. Simon, D. Matson, and J. Fuchs, “Wake Effects in a Fayette 95-IIS Wind Turbine Array,” *SERI/STR-217-3186A*, 1988.

[132] R. B. Smith, “The influence of mountains on the atmosphere,” *Advances In Geographics*, pp. 87–230, 1981.

[133] A. Sogachev, “A Note on Two-Equation Closure Modeling of Canopy,” *Boundary-Layer Meteorology*, vol. 130, no. 3, pp. 423–435, 2009.

[134] J. Sorensen and A. Myken, “Unsteady Actuator Disc Model for Horizontal Axis Wind Turbines,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 39, pp. 139–149, 1992.

[135] R. Stull, *An Introduction to Boundary Layer Meteorology.* Dordrecht: Kluwer Academic, 1988.

[136] A. Talmon, “A Wind Tunnel Investigation into the Effects of Tower and Naccelle on Wind Turbine Wake Flow,” *Technical Report*, no. 84-08479, 1984.

[137] ——, “The Wake of a Horizontal Axis Wind Turbine Model, Measurements in Uniform Approach Flow and a Simulated Boundary Layer,” *Technical Report*, vol. 85-01021, 1985.

[138] P. Taylor, “Some Numerical Studies of Surface Boundary-Layer Flow above Gentle Topography,” *Boundary-Layer Meteorology*, vol. 12, pp. 439–465, 1977.

[139] ——, “On Wake Decay and Row Spacing for WECS Farms,” *Proceedings 3rd International Symposium on Wind Energy Systems*, pp. 451–468, 1980.

[140] P. Taylor, P. Mason, and E. Bradley, “Boundary Layer Flow over Low Hills,” *Boundary-Layer Meteorology*, vol. 39, pp. 107–132, 1987.

[141] P. Taylor, R. Mickle, and R. Salmon, “The Kettles Hill Experiment - Site Description and Mean Flow Results,” *Boundary Layer Research Division, Air Quality and Inter-Environmental Research Branch, Atmospheric Environment Service, Downsview*, vol. M3H 5T4, 1983.

[142] P. Taylor, J. Walmsley, and J. Salmon, “ Simple Model of Neutrally Stratified Boundary-Layer Flow Over Real Terrain Incorporating Wavenumber-Dependent Scaling,” *Boundary-Layer Meteorology*, vol. 26, pp. 169–189, 1983a.

[143] H. Tinaztepe, “A Finite Volume Method for Compressible Viscous Flows,” PhD Thesis, Middle East Technical University, 1997.

[144] A. Tindal, “Dynamic Loads in Wind Farms,” *Final Report*, vol. CEC Project JOUR-0084-C, 1993.

[145] R. Tiwari, “Unsteady Reynolds-Averaged Navier-Stokes Simulations of Atmospheric Boundary Layer over Complex Terrains,” Semester Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, June 2011.

[146] N. Troldborg, G. Larsen, H. Madsen, K. Hansen, J. SÃÿrensen, and R. Mikkelsen, “Numerical Simulations of Wake Interaction Between Two Wind Turbines at Various Inflow Conditions,” *Wind Energy*, vol. 14, no. 7, pp. 859–876, 2010.

[147] M. Tullio, P. Palma, G. Pascazio, and M. Napolitano, “Wall Stress Models for an Immersed Boundary Method,” *Academy Colloquium, Immersed Boundary Methods: Current Status and Future Research Directions*, 2009.

[148] E. Turkel, “Review of Preconditioning Methods in Fluid Dynamics,” *Applied Numerical Mathematics*, vol. 72, pp. 277–298, 1987.

[149] E. Turkel, V. Vasta, and R. Radspiel, “Preconditioning Methods for Low Speed Flow,” *AIAA Paper*, no. 96-2460, 1996.

[150] O. Undheim, “Microscale Flow Modeling Possibilities and Limitations,” *Proceedings of European Wind Energy Conference*, 2006.

[151] O. Undheim, H. Andersson, and E. Berge, “Non-Linear, Microscale Modelling of the Flow over Askervein Hill,” *Boundary-Layer Meteorology*, vol. 120, pp. 477–495, 2006.

[152] S. Venkateswaran and L. Merkle, “Analysis of Preconditioning Methods for the Euler and Navier-Stokes Equations,” *von Karman Institute of Fluid Dynamics, Lecture Series*, 1999.

[153] L. Vermeer, J. Sorensen, and A. Crespo, “Wind Turbine Wake Aerodynamics,” *Progress in Aerospace Science*, vol. 39, pp. 467–510, 2003.

[154] P. Vermeulen and P. Builtjes, “Practical Applications of Mathematical Wake Interaction Model,” *Proceedings 4th International Symposium on Wind Energy Systems*, pp. 437–448, 1982.

[155] ——, “Turbulence Measurements in Simulated Wind-turbine Clusters,” *TNO Report*, vol. 89-160, 1982.

[156] G. Volpe, “Performance of Compressible Flow Codes at Low Mach Number,” *AIAA Journal*, vol. 31, pp. 49–56, 1993.

[157] S. Voutsinas, J. Glekas, and A. Zervos, “Investigation of the Effect of the Initial Velocity Profile on the Wake Development of a Wind Turbine,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 39, pp. 293–301, 1992.

[158] J. Walmsley, J. Salmon, and P. Taylor, “ On the Application of a Model of Boundary-Layer Flow over Low Hills to Real Terrain,” *Boundary-Layer Meteorology*, vol. 23, pp. 17–46, 1982.

[159] J. Walmsley and P. Taylor, “Boundary-Layer Flow over Topography: Impacts on the Askervein Study,” *Boundary-Layer Meteorology*, vol. 78, pp. 291–320, 1996.

[160] J. Weiss and W. Smith, “Preconditioning Applied to Variable and Constant Density Flows,” *AIAA Journal*, vol. 33, pp. 207–223, 1995.

[161] J. Wieringa, “Updating the Davenport Roughness Classification,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 41, no. 44, pp. 357–368, 1992.

[162] D. Wilcox, “Reassessment of the Scale-Determining Equation for Advanced Turbulence Model,” *AIAA Journal*, vol. 26, no. 11, pp. 1299–1310, 1988.

[163] ——, *Turbulence Modeling for CFD.* La Canada: DCW Industries, 1993.

[164] N. Wilson and R. Shaw, “A High Order Closure Model for Canopy Flow,” *Journal of Applied Meteorology*, vol. 16, pp. 1197–1205, 1977.

[165] Y. Wu and F. Porte-Agel, “Simulation of Turbulent Flow Inside and Above Wind Farms: Model Validation and Layout Effects,” *Boundary-Layer Meteorology*, vol. 146, no. 2, pp. 181–205, 2013.

Curriculum Vitae

Personal data

- **Name:** Samira Jafari
- **Birth:** September 1^{st}, 1984, Zanjan, Iran

Education

- Ph.D., Laboratory for Energy Conversion, Institut f*ü*r Energietechnik, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland
  December 2008 – March 2014
- M.Sc., Aerospace Engineering (Aerodynamics and Propulsion), Sharif University of Technology, Tehran, Iran
  September 2006 – December 2008
- B.Sc., Aerospace Engineering Engineering, Sharif University of Technology, Tehran, Iran
  September 2002 – September 2006

Professional experience

- Review chair of 8th European PhD seminar on wind energy held in ETH Zurich, September 2012.
- Teaching assistant for Aerospace Propulsion course, ETH Zurich, 2009-2013.
- Supervision of master and semester student projects, ETH Zurich, 2009-2013.

List of Publications

Journal publications

- **S. Jafari**, N. Chokani and R.S. Abhari, *Simulations of Wake Interactions in Wind Farms Using an Immersed Wind Turbine Model*, Journal of Turbomachinery, doi:10.1115/1.4025762, Nov 2013.

- C. Tsalicoglou, **S. Jafari**, N. Chokani and R.S. Abhari, *RANS Computations of Wind Turbine Near-Wake Aerodynamics in Uniform and Yawed Inflow*, Journal of Engineering for Gas Turbine and Power, 136(1), 011202, doi:10.1115/1.4025362, Oct 2013.

- **S. Jafari**, N. Chokani and R.S. Abhari, *An Immersed Boundary Method for Simulation of Wind Flow over Complex Terrain*, Journal of Solar Energy Engineering, 134, 011006, 2012.

- S. Barber, Y. Wang, **S. Jafari**, N. Chokani and R.S. Abhari, *The Impact of Ice Formation on Wind Turbine Performance and Aerodynamics*, Journal of Solar Energy Engineering, 133(1), 011007, 2011.

Recent submissions

- **S. Jafari**,N. Chokani and R.S. Abhari, *Numerical prediction of Array Loss and Terrain-Wake Interaction in Wind Farms*, submitted, Feb 2014.

- **S. Jafari**, A. Basol, N. Chokani and R.S. Abhari, *Simulations of Atmospheric Flow and Wind Turbine Wakes Using a Preconditioned Multigrid*, submitted, Feb 2014.

- A. Basol, **S. Jafari**, N. Chokani and R.S. Abhari, *A New Anisotropic Scaling for the Artificial Dissipation for the Cell Vertex Finite Volume Method*, submitted, Feb 2014.

- **S. Jafari**, T. Sommer, N. Chokani and R.S. Abhari, *Effect of Horizontal Grid Spacing on Accuracy of Wind Resource assessment Using a Mesoscale Model*, submitted, Jan 2014.

Peer reviewed conferences

- **S. Jafari**, N. Chokani and R.S. Abhari, *Estimation of Array Loss in Onshore and Offshore Wind Farms Using RANS*, to be presented in European Wind Energy Conference, Barcelona, Spain, 2014.

- **S. Jafari**, N. Chokani and R.S. Abhari, *Simulations of Wake Interactions in Wind Farms Using an Immersed Wind Turbine Model*, ASME Expo Turbo, San Antonio, Texas, USA, 2014.

- **S. Jafari**, N. Chokani and R.S. Abhari, *Simulations of Wake Interactions in Wind Farms Using an Immersed Wind Turbine Model*, ASME Expo Turbo, San Antonio, Texas, USA, 2013.

- C. Tsalicoglou, **S. Jafari**, N. Chokani and R.S. Abhari, *RANS Computations of Wind Turbine Near-Wake Aerodynamics in Uniform and Yawed Inflow*, ASME Expo Turbo, San Antonio, Texas, USA, 2013.

- **S. Jafari**, N. Chokani and R.S. Abhari, *Wind Farms in Complex Terrain: Numerical Simulations of Wind and Wakes for Optimised Micrositing*, European Wind Energy Conference, Vienna, Austria, 2013.

- C. Tsalicoglou, **S. Jafari**, N. Chokani and R.S. Abhari, *Full RANS Simulations of MEXICO Rotor: Predictions for a Novel Wind Turbine Model*, European Wind Energy Conference, Vienna, Austria, 2013.

- **S. Jafari**, T. Sommer, N. Chokani and R.S. Abhari, *Wind Resource Assessment Using a Mesoscale Model: The Effect of Horizontal Resolution*, ASME Expo Turbo, Copenhagen, Denmark, 2012.

- **S. Jafari**, N. Chokani and R.S. Abhari, *Terrain Effects on Wind Flow: Simulations with an Immersed Boundary Method*, ASME Expo Turbo, Vancouver, Canada, 2011.

- **S. Jafari**, N. Chokani and R.S. Abhari, *An Immersed Boundary Method for Efficient Simulation of Wind Flow over Complex Terrain*, European Wind Energy Conference, Brussels, Belgium, 2011.

- E. Kochman **S. Jafari**, N. Chokani and R.S. Abhari, *Impact of Shear, Swirl and Yaw on Wind Turbine Performance*, European Wind Energy Conference, Brussels, Belgium, 2011.