NTNU
Norwegian University of
Science and Technology

Urban wind: CFD analysis of Gløshaugen campus based on measured data

Stian Næve Burkeland

Master of Energy and Environmental Engineering
Submission date: June 2018
Supervisor: Tania Bracchi, EPT
Co-supervisor: Are Simonsen, SINTEF

Norwegian University of Science and Technology
Department of Energy and Process Engineering

NTNU
Norges teknisk-naturvitenskapelige universitet
Institutt for energi- og prosessteknikk

EPT-M-2018-14

MASTEROPPGAVE

for

Stian Næve Burkeland

Våren 2018

Urban vind: CFD analyse av Gløshaugen campus, basert på målt data

Urban wind: CFD analysis of Gløshaugen campus based on measured data

Bakgrunn og målsetting

”Urban wind” er et tema som blir stadig mer relevant, se for eks. EUs Horizon 2020 program [1]. To aspekter kan knyttes til ”urban wind”: city planning (smart og bærekraftige byer) og energi produksjon i nyskapende lavenergi bygninger.

Vind i bebyggelse er turbulent og ustabil, men den kan også være en ressurs. Bygninger og passasjer mellom bygg gir økt vindhastighet og ettersom energi potensialet øker med 3 potens av vindhastighet kan lokale effekter være svært gunstig med hensyn på lokal energiproduksjon fra vindturbiner. Ved oppføring av nybygg er det også mulig å utforme disse på en slik måte at vindturbinene kan integreres i bygningene, se e.g. [2]. For å utnytte vindenergien må man vite hvor på bygningene er det mest fornuftig å plassere en vindturbinene, eller eventuellt hvordan kan man utforme bygningen på best måte i forhold til å få økt vindhastighet. De komplekse vindforholdene i urbane strøk gjør det praktisk sett umulig å lage vindkart for gjennomsnittsvind slik man gjør i åpent terreng. Strømningsforholdene dannet av de omliggende byggene er rett og slett for komplekse for meterologiske "lokale" beregninger som typisk operere med en oppløsning av størrelseorden 100~1000m. Bruk av numeriske beregninger, eller "Computational Fluid Dynamics (CFD)", er en løsning for å beregne og hjelpe å forutsi vindforhold i byene. Med CFD kan man lage en detaljert modell med oppløsning på ~1m over utvalgte områder og foreta detaljerte analyser. Dette krever at man også har informasjon om typiske storskale vindforhold (vindretning, vind styrke, og turbulens) for området man ser på.

I denne oppgave er Gløshaugen campus ble valgt som ”test case” for CFD analyse av lokale vindforhold.

Oppgaven bygges opp på prosjektoppgaven ”Urban Wind:CFD analysis of Gløshaugen campus”. I denne oppgaven ble det:

- Tilpasset matematiske modeller til vindprofildata målt med LIDAR1 og definert hvordan disse kan inkorporeres i en 3D CFD modell.
- Utført et litteraturstudie på "Best Practice Guidelines" for CFD beregninger i URBANE strøk.
- Utført simuleringer med ANSYS Fluent på en 3D modell av Gløshaugen campus med et forenklet innløpsprofil

Målet med masteroppgaven er å bygge videre på prosjektoppgaven ved å:

- Utvide eksisterende campus geometri og forbedre eksisterende numerisk modell.
- Simulere med målt innløpsprofil
- Sammenligne simulering med målt data fra LIDAR2 (som representerer et referansepunkt på Gløshaugen campus)
- Se på effekten av:
    - Turbulensmodell k-ε, k-ω
    - LES
    - Grid oppløsning
    - Turbulens intensitet ved innløp
    - Turbulent lengdeskala ved innløp

Oppgaven bearbeides ut fra følgende punkter

1. Forbedre geometri av campus
2. Simulere med målt innløpsprofil med Ansys Fluent
3. Kjør en sensitiv analyse, med endring av forskjellige parameter (i.e. turbulensmodell, LES, gris oppløsning, turbulensegenskaper ved innløpsprofil)

Referanser:

[1] Union, Innovation. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions. Brussels, 2014

[2] Sintef (2013, February 13). Vindmøller på taket. Retrieved from http://www.sintef.no/sistenytt/vindmoller-pa-taket/

” - ”

Senest 14 dager etter utlevering av oppgaven skal kandidaten levere/sende instituttet en detaljert fremdrift- og eventuelt forsøksplan for oppgaven til evaluering og eventuelt diskusjon med faglig ansvarlig/veiledere. Detaljer ved eventuell utførelse av dataprogrammer skal avtales nærmere i samråd med faglig ansvarlig.

Besvarelsen redigeres mest mulig som en forskningsrapport med et sammendrag både på norsk og engelsk, konklusjon, litteraturliste, innholdsfortegnelse etc. Ved utarbeidelsen av teksten skal kandidaten legge vekt på å gjøre teksten oversiktlig og velskrevet. Med henblikk på lesning av besvarelsen er det viktig at de nødvendige henvisninger for korresponderende steder i tekst, tabeller og figurer anføres på begge steder. Ved bedømmelsen legges det stor vekt på at resultatene er grundig bearbeidet, at de oppstilles tabellarisk og/eller grafisk på en oversiktlig måte, og at de er diskutert utførlig.

Alle benyttede kilder, også muntlige opplysninger, skal oppgis på fullstendig måte. For tidsskrifter og bøker oppgis forfatter, tittel, årgang, sidetall og eventuelt figurnummer.

Det forutsettes at kandidaten tar initiativ til og holder nødvendig kontakt med faglærer og veileder(e). Kandidaten skal rette seg etter de reglementer og retningslinjer som gjelder ved alle (andre) fagmiljøer som kandidaten har kontakt med gjennom sin utførelse av oppgaven, samt etter eventuelle pålegg fra Institutt for energi- og prosessteknikk.

Risikovurdering av kandidatens arbeid skal gjennomføres i henhold til instituttets prosedyrer. Risikovurderingen skal dokumenteres og inngå som del av besvarelsen. Hendelser relatert til kandidatens arbeid med uheldig innvirkning på helse, miljø eller sikkerhet, skal dokumenteres og inngå som en del av besvarelsen. Hvis dokumentasjonen på risikovurderingen utgjør veldig mange sider, leveres den fulle versjonen elektronisk til veileder og et utdrag inkluderes i besvarelsen.

I henhold til "Utfyllende regler til studieforskriften for teknologistudiet/sivilingeniørstudiet" ved NTNU § 20, forbeholder instituttet seg retten til å benytte alle resultater og data til undervisnings- og forskningsformål, samt til fremtidige publikasjoner.

Besvarelsen leveres digitalt i DAIM. Et faglig sammendrag med oppgavens tittel, kandidatens navn, veileders navn, årstall, instituttnavn, og NTNUs logo og navn, leveres til instituttet som en separat pdf-fil. Etter avtale leveres besvarelse og evt. annet materiale til veileder i digitalt format.

☐ Arbeid i laboratorium (vannkraftlaboratoriet, strømningsteknisk, varmeteknisk)
☐ Feltarbeid

NTNU, Institutt for energi- og prosessteknikk, 15. januar 2018

Tania Bracchi (NTNU)
Faglærer/veileder

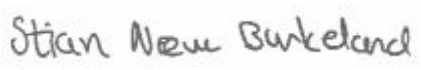
Are T. Simonsen
Are Simonsen (Sintef)
Medveileder

Preface

This Master's thesis is a part of a Master of Science in Engineering Fluid Mechanics. It is written during the Spring of 2018 at the Department of Energy and Process Engineering at the Norwegian University of Science and Technology (NTNU). It is carried out in collaboration with NTNU and SINTEF, where the work has focused on performing a computational fluid dynamics (CFD) simulation of the wind conditions at Gløshaugen campus, based on measured data.

I would like to thank my supervisor, Associate Professor Tania Bracchi, for guidance and support throughout the project. I highly appreciate the helpful feedback and her enthusiasm for the project. Further, I will like to thank Research Scientist Are Simonsen for valuable scientific support about the simulation tool ANSYS Fluent. In addition, he designed a suitable three-dimensional model of Gløshaugen Campus, that was of great use in the project. Lastly, I would give my sincere gratitude to Ph.D. candidate Felix Kelberlau for analyzing wind data measured by ZephIR LIght Detection And Ranging (LIDAR) and Windcube.

Trondheim, 2012-06-18



Stian Næve Burkeland

Abstract

In this study, a CFD analysis of the wind condition at the Norwegian University of Science and Technology based on local measurements is conducted of inlet wind profiles. The simulation is performed on Abel computer cluster, with ANSYS Fluent as simulation tool and $k-\epsilon$, $k-\omega$, and LES as turbulence models. The aim of the analysis is based on finding most promising sites for wind turbines and characterize differences in the turbulence models based on turbulent parameters at the inlet.

Five wind measurements have been carried out at the campus, three of them were located at the inlet of the computational domain, while the two others were located inside the campus. The inflow condition is estimated based on the three measurements at the inlet, while the two others have been used to validate the simulation results numerically. Both RANS models and LES predicted a higher mean velocity at the two measurement location inside the campus compared to the measurements. However, a hit rate value for the simulated results was larger than $66\ \%$, meaning that the simulation is applicable for predicting the flow through the university.

Furthermore, preliminary analysis of the most promising building for siting wind turbines, based on high-velocity and low-turbulence regions. It was concluded that the high-rise building at the center of the campus was the most suitable location for wind turbines. Based on this, a more thourogh analysis of the effect of different inlet parameters on flow conditions at this location was carried out.

The simulations are carried out with turbulence intensity at the inlet equal to 0.1, 5 and $10\ \%$ for the RANS models, and No Perturbation and $5\ \%$ for the LES model. The mean velocity calculated by both RANS models and LES model above the building was not affected by different turbulent inflow condition. While the turbulence intensity distribution for the RANS models was highly increased as the turbulent intensity at the inlet was increased. Additionally, the turbulent length scale at the inlet was analyzed based on two hydraulic diameters at the inlet, 50 and 100 meters respectively, for the $k-\epsilon$ simulation. It was shown that turbulent length scale at the inlet does not influence the velocity and turbulence intensity downwind of the domain. Based on the findings from the CFD simulation, some recommendation is performed for most promising sits for wind turbines.

Summary

I denne oppgaven er det utført en CFD analyse av vindforholdene på Norges Tekniske -Naturvitenskapelig Universitet basert på lokale vindmå linger. Simuleringene er utført på Abel Computer Cluster med ANSYS Fluent som simuleringsverktøy og turbulensmodellene $k-\epsilon$, $k-\omega$ og LES. Analysen er basert på å finne optimale lokasjon for installering av vindturbiner, samt karakterisere forskjeller i de forskjellige turbulensmodellene basert på ulike turbulente innløpsbetingelser.

Det er utført fem vindmålinger på universitetet hvor tre av de er lokalisert ved innløpet i beregningsdomenet. De to siste mer utført på campus. Hastighetprofilet ved innløpet er estimert basert på målingene som er utført ved innløpet, og de to siste målingene er brukt til å validere simuleringsresultatene. Begge RANS modellene og LES overestimerer vindhastighetene sammenlignet med de to vindmålingene som er utført på campus. Uansett gav de simulerete hastighetsprofilene en hit-rate verdi som er større enn $66\ \%$, noe som tilsier at simuleringen er et godt estimat for predikasjon av vindforholdene på universitet.

Videre er det utført en preliminr studie av optimale lokasjoner for installering av vindturbiner basert på regioner med høy hastighet og lav turbulens. Resultatene viser at the er det høyeste bygningen på universitetet som har de beste vindforholdene basert på de nevnte kriteriene. Dermed er det utført en grundigere analyse av vindforholdene over den nevnte bygningen basert på endringer i turbulens ved innløpet.

Simuleringene er utført med turbulens-intensitet ved innløpet lik 0.1, 5 og$10\ \%$ for RANS modellene, og No Perturbations og $5\ \%$ for LES modellen. For alle tre turbulens-modellene er hastigheten uavhengig av turbulensen ved innløpet. Derimot, turbulensfordelingen for RANS modellene øker kraftig med økt turbulens-intensitet ved innløpet. I tillegg er det blitt analysert hvordan endringer i turbulent lengdeskale påvirker vindstrømningen for the $k-\epsilon$ modellen. Den hydrauliske diameteren ved innløpet ble satt til å vre lik 50 og 100 meter. Resultatene viste at den turbulente lengedeskalaen øker med økt hydraulisk diameter, men at vindforholdene ikke endrer seg med hensyn til turbulent lengdeskala ved innløpet. Basert på funnene, er det avslutningsvis kommet med noen forslag på optimale lokasjoner for installering av vindturbiner.

Table of Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

BPG	Best Practice Guidelines
CFD	Computational Fluid Dynamics
CWE	Computational Wind Engineering
DNS	Direct Numerical Simulation
IEC	International Electrotechnical Commission
LES	Large Eddy Simulations
LIDAR	LIght Detection And Ranging
RANS	Reynolds Averaged Navier-Stokes
SST	Shear Stress Transport
TKE	Turbulent Kinetic Energy
WE	Wind Engineering

Latin symbols

g	gravity [m/s^2]
H_{max}	Height of the highest building [m]
I	Turbulence intensity [%]
k	Turbulent kinetic energy [m$^2/s^2$]
N	Correlation between measured and simulated data [-]
n	Number of data points [-]
O	Simulated data divided by reference value [-]
P	Measured data divided by reference value [-]
$\overline{P}$	Time-average pressure [N/m^2]
q	Hit rate [%]
RD	Relative standard deviation [-]
$\overline{S}$	Strain rate tensor [kg/m$^3 s$]
U_{ref}	Reference Velocity [m/s]
u	Instantaneous velocity [m/s]
$\overline{u}$	Time-average mean velocity [m/s]
u'	Fluctuating velocity [m/s]
x_i	Cartesian coordinates [m]
Z_{ref}	Reference height [m]

Greek symbols

α	Power Law exponent [-]
δ	Kronecker delta [-]
ϵ	Turbulent eddy dissipation [m$^2/s^3$]
μ	Dynamic viscosity [kg/ms]
μ_t	Turbulent viscosity [m$^2/s$]
ρ	Density of air [kg/m^3]
τ	Sub-grid scale stress [kg/ms^2]
ω	Turbulent specific dissipation [1/s]

1. Introduction

Deploying wind turbines in an urban environment has been adopted in recent years due to an increased awareness of the rising level of pollution. By installing building-integrated micro-wind turbines, one can achieve a reduction in the emission of greenhouse gases at potentially low costs (Ledo et al., 2011a). Besides, since the majority of the earth's population lives in urban areas, it is favorable to distribute power where it is needed. However, predicting wind flow in urban environments is complicated due to variation in topography and roughness, in addition to the creation of stagnation and recirculating flow when the wind interacts with buildings. These complexities lead to regions with low mean wind speed and high levels of turbulence, which reduces the performance of turbines (Wang et al., 2018a). Hence, to find suitable locations for installing wind turbines in an urban environment requires detailed wind resource assessment (Mohamed and Wood, 2016a).

Over the past few years, several experiments and numerical studies have been conducted within the field of wind comfort in urban areas, ventilation, wind loading, dispersion of air pollutants, and the corresponding wind flow predictions (Tutar and Oguz, 2002a). Murakami and Mochida (1988) conducted a wind tunnel experiment of how the wind flow interacted with a cubic model, while Meng and Hibi (1998) determined the flow field around a 2:1:1 (h:w:d) shaped square prism placed in a turbulent boundary layer (Yoshie et al., 2007a). Wiren (1975) made an experimental study of how the wind behaves in passages between building models, placed either in parallel or perpendicular to each other (Tutar and Oguz, 2002b). Another wind tunnel study has been carried out by Rafailidis (1997), which describes how the shape of the roof influence the flow field and the amount of turbulence intensity.

Although local measurements and wind tunnel experiments are effective ways to determine wind flow distribution and the corresponding characteristics, it is also both time-consuming and expensive (Wang et al., 2018a). By using improved computational processing and software, omputational fluid dynamics (CFD) techniques have been seen as an easier way to gain insight of the flow pattern in urban areas (Wang et al., 2018a). Increased progress in computational processing and development of software for numerical analysis has made it possible to predict wind flow around high-rise buildings based on CFD (Yoshie et al., 2007b). When using CFD in wind engineering - called computational wind engineering (CWE), a large number of choices needs to be made by the user. Therefore, several studies have been carried out to describe the best practice guidelines (BPG) for a CFD set-up. These guidelines consider the different equations that have to be solved in order to describe the turbulent wind flow such as Reynolds Average Navier-Stokes Simulation (RANS) and Large Eddy Simulation (LES) (Blocken et al., 2011). Also, they cover the level of detail in the geometrical representations of the urban environment, the size of the computational domain, the boundary conditions, the computational grid, the discretization schemes, the initialization data, the time-steps, and the iterative convergence criteria (Blocken et al., 2011).

Furthermore, several CFD simulations are carried out based on BPG, where there are implemented different turbulence models to describe how the urban environment influence the boundary layer (Tutar and Oguz, 2002a). Blocken et al. (2012) investigated the pedestrian wind comfort the wind safety at the Eindhoven University campus by solving the realisable $k-\epsilon$ model, and the results are compared with long-term and short-term measurements. Tabrizi et al. (2014a) performed an analysis of the optimal location for installing micro-wind turbines on the roof of a warehouse by solving the $SST\ k-\omega$ equations, where it was found that the flow field is sensitive to building height and shape, roof shape, wind direction and turbine in-

stallation height and location. Blocken and Persoon (2009) solved 3-D steady RANS equations to establish the changes in wind comfort around a football stadium before and after the rise of a high building. Yoshie et al. (2007c) calculated the entire flow field around two types of building complexes in actual urban areas. In both cases were the standard $k - \epsilon$ turbulence model used, three types of grid systems, and interpolated values of U and k from experiments to determine the inflow boundary condition. Further, the CFD results were validated by field measurements and wind tunnel experiments. Mohamed and Wood (2016b) conducted a CFD analysis of the wind flow over the University of Calgary campus to determine the most promising sites for installing wind turbines and photovoltaic modules, where the influence of the terrain is discussed. The paper covers simulation results performed by different turbulence models, such as standard $k - \epsilon$, SST $k - \omega$, baseline (BSL) RSM and the RNG $k - \epsilon$. The study concludes that the choice of turbulence model is not crucial if the purpose of the simulation is to identify regions for further exploration via wind measurements (Mohamed and Wood, 2016b). Wang et al. (2018b) conducted a CFD analysis of the wind flow over an actual building with the purpose to locate the optimal mounting site for wind turbines at the roof. The analysis is based on regions with low turbulence intensity and accelerating velocity. It was concluded that the best sites for wind turbines are at the upwind edge of the building. A numerical simulation of the wind flow between two buildings using the LES model was done by Tutar and Oguz (2002c). It was concluded that the simulated results were sufficient compared to the experimental results. A further study by Tutar and Oguz (2002c) summarizes other studies that have used different numerical methods to investigate the interaction between wind and buildings. The summary concludes that the LES seems to be more suitable for analyzing the flow of interest around buildings. On the other hand, it is well known that the usage demands a highly increased computational cost compared to RANS models.

This present research represents a CFD analysis of the wind conditions at the Norwegian University of Science and Technology (NTNU) to localize the most promising sites for installation of wind turbines. The CFD simulation is based on three turbulence models: standard $k - \epsilon$, $k - \omega$, and LES. There are performed wind measurements at two different location at the campus to validate the simulation results, and three measurements at the inlet to establish a suitable inflow condition.

2. Methodology

2.1 Generating the Campus Geometry

The university is located $2km$ south of the city center of Trondheim, Norway. The campus length is $840m$, width is $540m$, and the height of the tallest building (H_{max}) is $40m$. Generation of the campus geometry was done in a previous study of urban wind at NTNU, where Google Maps and Google Sketchup were used to build the campus geometry as well as the surrounding terrain (Simonsen, 2016). Further, the geometry was imported to ANSYS SpaceClaim for generation of the computational domain (Burkeland, 2017; Simonsen, 2016). The university is illustrated in Figure 2.1.

Figure 2.1: Norwegian University of Science and Technology.

2.2 Site Measurements

Local on-site measurements are conducted at the campus to estimate the wind profile at the inlet and to validate the CFD simulations. The measurements are carried out at five different locations at the university. Three of them were installed at the inlet of the campus, while the two others were installed inside campus (inside the associated the computational domain) as shown in Figure 2.2. Location no. 1, 2, 3 and 4 are measured at ground level by a $ZephIR$ 300 wind $LIght\ Detection\ And\ Ranging$ ($LIDAR$) and the last measurement was performed at the roof of the $Varmeteknisk$ building by a $WindCube$. Throughout this paper, the measurement spots are referred to as location no. 1, 2, 3, 4 and 5, as illustrated in Figure 2.2. The LIDAR measured wind velocity and direction at a reference height equal 38 meters above ground and a range from 65 to 110 meters above ground with an increment of five meters. The Windcube measured wind velocity and direction for heights equal 40 to 90 meters above the roof, with an increment of five meters, and a final height of 100 meters above the roof. There were performed two measurements by LIDAR at each location and one measurement by the Windcube. The date and time-interval for each measurement are given in Table 2.1.

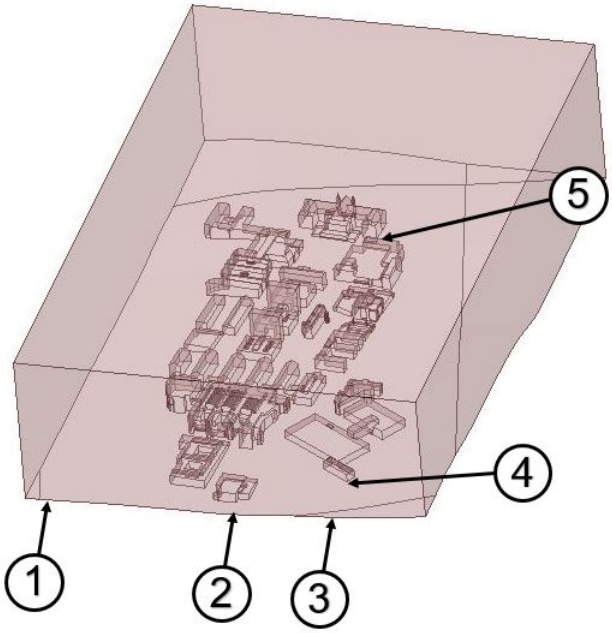


Figure 2.2: Locations at the campus where it is conducted wind measurements.

Table 2.1: Date and time for the LIDAR and Windcube measurements.

Location	Measurement tool	Measurement	Date	Start-time	End-time
1	LIDAR	1^{st} Measurement	29.09.2017	09.05	09.15
	LIDAR	2^{nd} Measurement	29.09.2017	10.17	10.27
2	LIDAR	1^{st} Measurement	29.09.2017	09.22	09.33
	LIDAR	2^{nd} Measurement	29.09.2017	10.36	10.43
3	LIDAR	1^{st} Measurement	29.09.2017	08.22	08.33
	LIDAR	2^{nd} Measurement	29.09.2017	09.41	09.51
4	LIDAR	1^{st} Measurement	29.09.2017	08.42	08.52
	LIDAR	2^{nd} Measurement	29.09.2017	10.03	10.08
5	WindCube	-	29.09.2017	08.30	10.40

2.2.1 Inlet Velocity Profile

The velocity profile at the inlet needs to be accurately estimated to provide valid results of the wind conditions in an urban environment (Tabrizi et al., 2014b). The inlet conditions in the present CFD study is based on local measurements performed at the location no. 1, 2 and 3, as previously mentioned. Moreover, a $Power\ Law$ exponent α, and a reference velocity U_{ref} were estimated at each location, based on the measurements. With the mentioned variables and measured data, one can approximate the velocity profile for the locations. Velocity profile for each measurement location at the inlet is shown in Appendix 5.1. Additionally, from the geometrical model of the campus, one can generate data points that describe the terrain at the inlet. By interpolation and extrapolation of α, U_{ref} and the reference height Z_{ref} between the

estimated velocity profiles with regard to the terrain, one can estimate the entire velocity profile at the inlet, as illustrated in Figure 2.3. The x-axis represents the ground along the inlet, the y-axis is the height above the ground, while the colorbar represents the magnitude of the velocity, the green marks represent the measurement locations and the white area is the terrain (below ground) along the inlet of campus. The interpolated values for α, U_{ref} and Z_{ref} along the inlet and how the terrain changes are illustrated in Appendix 5.1. Similarly is the simulated inlet velocity profile shown in Appendix 5.5.1.

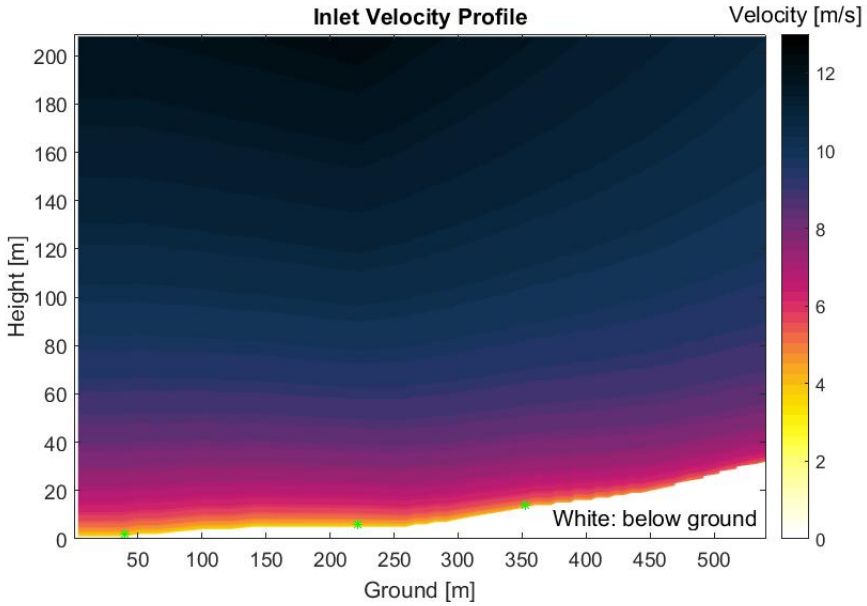


Figure 2.3: Estimated inlet velocity profile based on on-site measurements at three different locations along the inlet of the computational domain

2.3 Computational Fluid Dynamics Model

2.3.1 Navier-Stokes Equations and Turbulence Models

The CFD simulations were performed by ANSYS Fluent 18.2 and supported by Abel Computer Cluster service. When calculating turbulent flow, there are three approaches that can be used to solve the equations: $Direct$ NumericalSimulation (DNS), LES, and RANS. LES and RANS methods require turbulence models while DNS just requires a defined mesh to solve flow equations (ANSYS-Inc, 2017a). There are typically two criteria that need to be considered when choosing the most appropriate turbulence model. The first criteria require details of the flow, while the second one is dependent on computational cost (Tabrizi et al., 2014a). The DNS method is not available in ANSYS Fluent due to the computational cost (ANSYS-Inc, 2017a). Moreover, the computational resources needed for the LES approach are also still too large for most of the practical applications, thus the most widely used method for solving turbulent flow is the RANS model with various turbulent-viscosity models (ANSYS-Inc, 2017a). However, with available computational processing, both RANS and LES approaches are used in the present research.

RANS Approach

As previously stated, the generally used method for solving turbulent flow in CWE is the RANS approach, where different turbulent-viscosity models are applied (Franke et al., 2004). Reynolds decomposition is employed for the Navirer-Stokes (N-S) and continuity equations, where the instantaneous velocity is divided into time-average and fluctuating parts ($u = \overline{u} + u'$). The governing equations are expressed as follows (Tabrizi et al., 2014a).

$$\rho\left(\frac{\partial\overline{u_i}}{\partial t} + \overline{u}_j\frac{\partial\overline{u}_i}{\partial x_j}\right) = -\frac{\partial\overline{P}}{\partial x_j} + \rho\overline{g}_i + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i}\right)\right] - \frac{\partial}{\partial x_j}\rho\overline{u_i'u_j'} \tag{2.1}$$

$$\frac{\partial\overline{u}_k}{\partial x_k} = 0 \tag{2.2}$$

The term $\overline{u_i'u_j'}$ is called $Reynolds\ stresses$ and represents additional stresses due to fluctuation in the flow. When solving the RANS equations, it is necessary to add another term that can relate the mean velocity and the mentioned stresses. This is performed by the Boussinesq hypothesis (ANSYS-Inc, 2006a; Tabrizi et al., 2014a)

$$\overline{-pu_i'u_j'} = \mu_t\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i}\right) - \frac{2}{3}\rho k\delta_{ij} \tag{2.3}$$

where δ_{ij} is the $Kronecker\ delta$, μ_t represent the turbulent viscosity and $\overline{u_i}$ is the time-averaged mean velocity.

Following the hypothesis in equation 2.3, one can solve the RANS equations by specifying the turbulent viscosity through different two-equation models (Aliferis et al., 2017; ANSYS-Inc, 2017a). The chosen models for this study were the $Standard\ k-\epsilon\ model$ and the $Standard\ k-\omega\ model$. It is well known that the $k-\epsilon\ model$ provides a fair compromise between computational cost and accuracy (Tabrizi et al., 2014a), but as the flow gets more complex with strong separation, large streamline curvature, and pressure gradients, the performance is rather poor (ANSYS-Inc, 2006a). However, the $k-\omega\ model$ utilizes a different dissipation term with the scalar ω, which has the following relation with k and ϵ (Pope, 2000; Aliferis et al., 2017)

$$\omega \equiv \frac{\epsilon}{k} \tag{2.4}$$

The usages of ω make the model superior for complex boundary layer flow with large pressure gradients and inflow separation. On the contrary, the model produce less accurate results in the free stream region (ANSYS-Inc, 2017b; Aliferis et al., 2017).

The turbulent kinetic energy (TKE) is defined as follows in the turbulent-viscosity models (ANSYS-Inc, 2017c)

$$k = \frac{1}{2}\left(\overline{(u_{ii}')^2} + \overline{(u_{jj}')^2} + \overline{(u_{kk}')^2}\right) \tag{2.5}$$

Further, the transport equations for $k-\epsilon$ is given by (Ferziger, 2002)

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho\overline{u_j}k)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_t}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \epsilon \tag{2.6}$$

$$\frac{\partial(\rho\epsilon)}{\partial t} + \frac{\partial(\rho u_j\epsilon)}{\partial x_j} = C_{\epsilon 1}P_k\frac{\epsilon}{k} - \rho C_{\epsilon 2}\frac{\epsilon^2}{k} + \frac{\partial}{\partial x_j}\left(\frac{\mu_t}{\sigma_\epsilon}\frac{\partial\epsilon}{\partial x_j}\right) \tag{2.7}$$

while the transport equations for $k-\omega$ is given by (Ferziger, 2002)

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho\overline{u_j}k)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k^*}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \rho\beta^* k\omega \tag{2.8}$$

$$\frac{\partial(\rho\omega)}{\partial t} + \frac{\partial(\rho\overline{u_j}\omega)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\omega^*}\right)\frac{\partial\omega}{\partial x_j}\right]\alpha\frac{\omega}{k}P_k - \rho\beta\omega^2 \tag{2.9}$$

where P_k is the production of TKE, and $C_{\epsilon 1}$, $C_{\epsilon 2}$, σ_ϵ, σ_k, α, β, β^*, $^*_k$ and $^*_\omega$ are closure coefficients.

LES Approach

LES solves turbulent flows with the filtered N-S equation, where eddies smaller than the grid size are modeled, and larger eddies are directly solved numerically. The filtered governing equations is defined as (ANSYS-Inc, 2006b)

$$\frac{\partial\overline{u_i}}{\partial t} + \frac{\partial\overline{u}_i\overline{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial\overline{P}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\nu\frac{\partial\overline{u}_i}{\partial x_j}+\right) - \frac{\partial\tau_{ij}}{\partial x_j} \tag{2.10}$$

$$\frac{\partial\rho}{\partial t} + \frac{\partial\rho u_i}{\partial x_i} = 0 \tag{2.11}$$

where $\overline{u}_i$ and $\overline{P}$ are the filtered values of the velocity and pressure, and τ_{ij} is the sub-grid scale stresses. Furthermore, the sub-grid scale stress is modeled by a viscous analogy to close the filtered N-S equations (Tutar and Oguz, 2002d). The viscous analogy is expressed as follows

$$\tau_{ij} = -2\mu_t\overline{S}_{i,j} \tag{2.12}$$

where, μ_t is the sub-grid scale viscosity and $\overline{S}_{ij}$ is the strain rate tensor.

2.3.2 Computational Mesh and and Grid-Sensitivity Analysis

The domain has to be discretized by a computational mesh which defines the spatial resolution of the numerical solution (Franke et al., 2004; Burkeland, 2017). The mesh setup is created by a proximity and curvature sizing function with fine relevant center and $5\ 112\ 917$ tetrahedral cells. The highest grid resolution has been applied near the buildings, while lower resolution at the outskirt of the campus. Furthermore, for the purpose of reduce the convergence difficulties

is mesh metric set to be equal skewness (Ansys-Inc, 2017a; Burkeland, 2017). When importing the mesh to ANSYS Fluent the grid was changed from tetrahedral cells to polyhedral cells, due to reduction in the computational cost. Cells merges together and the number of cells was reduced to $1\ 299\ 049$ cells.

Grid-sensitivity analysis is performed for the present study by constructing an additional tetrahedral grid with $2\ 589\ 695$ cells (both mesh is illustrated in Figure 2.4). The mean wind speed at measurement locations no. 4 and 5 are compared for both grids by the $k - \epsilon$ model. Figure 2.5 shows that the differences in mean velocity for the basic and finer grid at both locations are small. This indicates that the basic polyhedral grid should be suitable for further research of the wind conditions at Gløshaugen campus.

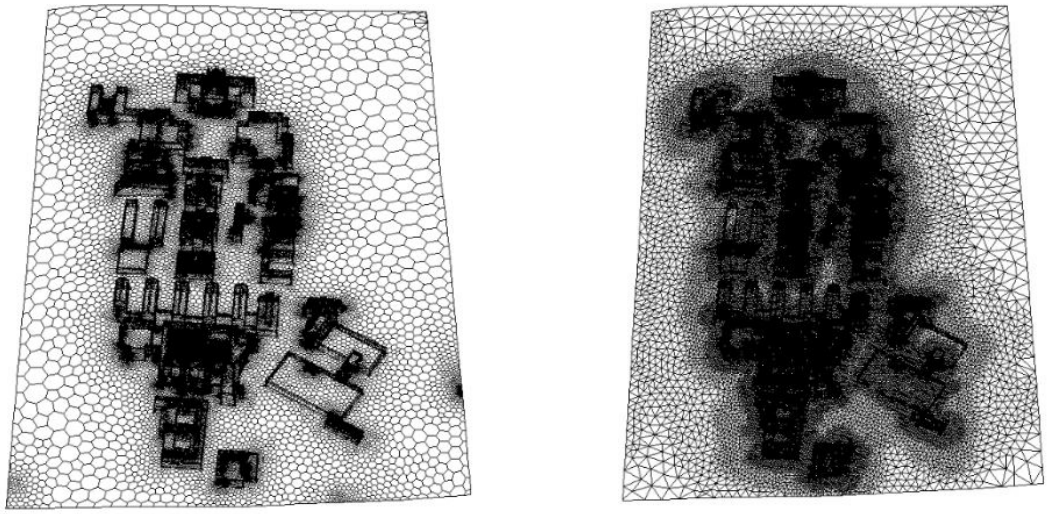


Figure 2.4: Polyhedral mesh (left) and tetrahedral mesh (right)

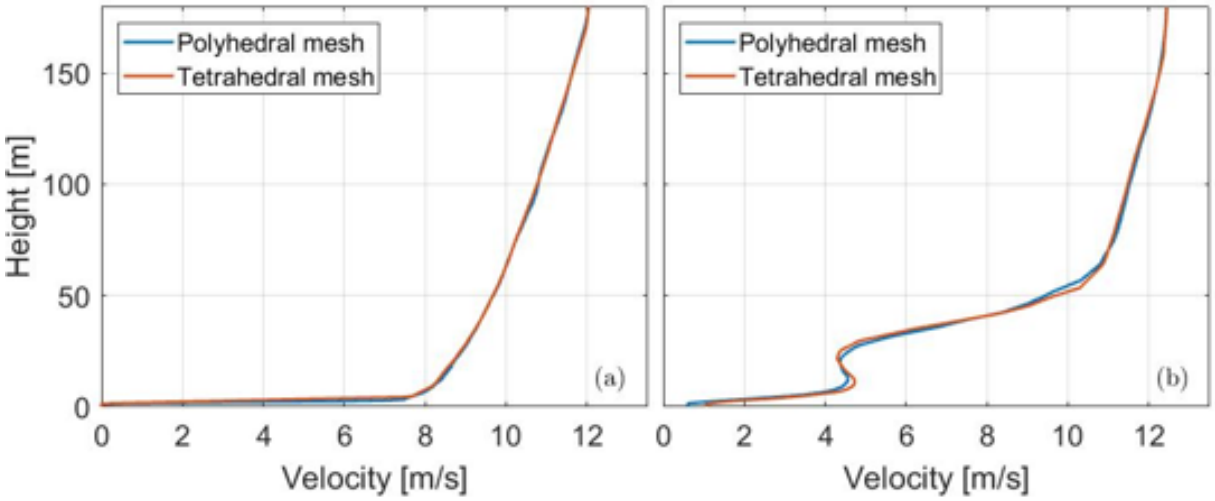


Figure 2.5: Results of grid-sensitivity analysis. Comparison of the mean wind speed at location no. 4 (a) and location no. 5 (b).

2.3.3 Numerical Scheme

There has been conducted a transient simulation of the wind flow distribution through the university. The procedure for statistics is based on that simulations have reached steady state before time statistic data have been collected, this is due to vacuum condition at the initial state in the computational domain.

Furthermore, the numerical scheme is based on the Finite Control Volume Discretization method, where the equations are calculated by a Pressure-Based solver. For the spatial discretization scheme, the gradients are solved by Least Square Cell-Based method. The pressure, momentum, TKE, ϵ, ω and transient formulations in the RANS models are solved with second-order schemes. While in the LES model is the momentum equation was solved with Bounded Central Differencing.

2.3.4 Boundary Domain and Boundary Conditions

The building structures were placed in an approximately rectangular domain of height equal 209 meters, some deviation due to differences in the terrain. The vertical height of the computational domain equals $5.1H_{max}$, which is consistent with the recommendations made by Franke et al. (2004), due to an artificial acceleration of the flow above the tallest building. Distances of $1.65H_{max}$ and $2.14H_{max}$ were used as the lateral left and right side of the domain. The inflow distance equals $0.64H_{max}$, and the outflow boundary equals a distance of $4.09H_{max}$. Franke et al. (2004) recommends that the outflow distance is $15H_{max}$ for achieving fully developed flow. However, due to surrounding urban environment at the outskirts of the campus and a rapid change in terrain, the chosen outlet length was considered equally adequate.

The inlet boundary condition was specified as User-Defined Function (UDF), which represents the estimated velocity profiles from the measurements (see Figure 2.3 and Appendix 5.1), and the turbulence intensity was equal to 0.1 % in the $k-\epsilon$ and $k-\omega$ simulations, while No Perturbations in the LES simulation. The outflow boundary was equal to a zero pressure gradient and the solid boundaries, such as terrain and buildings, were set to no-slip conditions with ANSYS Fluent Standard Wall-Functions to model the flow near the walls. The standard wall-functions take advantage of the fact that (for equilibrium turbulent boundary layers) a log-law correlation can supply the required boundary conditions (ANSYS-Inc, 2006c). The sides and the roof of the computational domain were defined as walls without shear stresses. Hence they do not introduce any boundary layer blockage into the domain. The boundary conditions are displayed in Figure 2.6 for illustrative purposes.

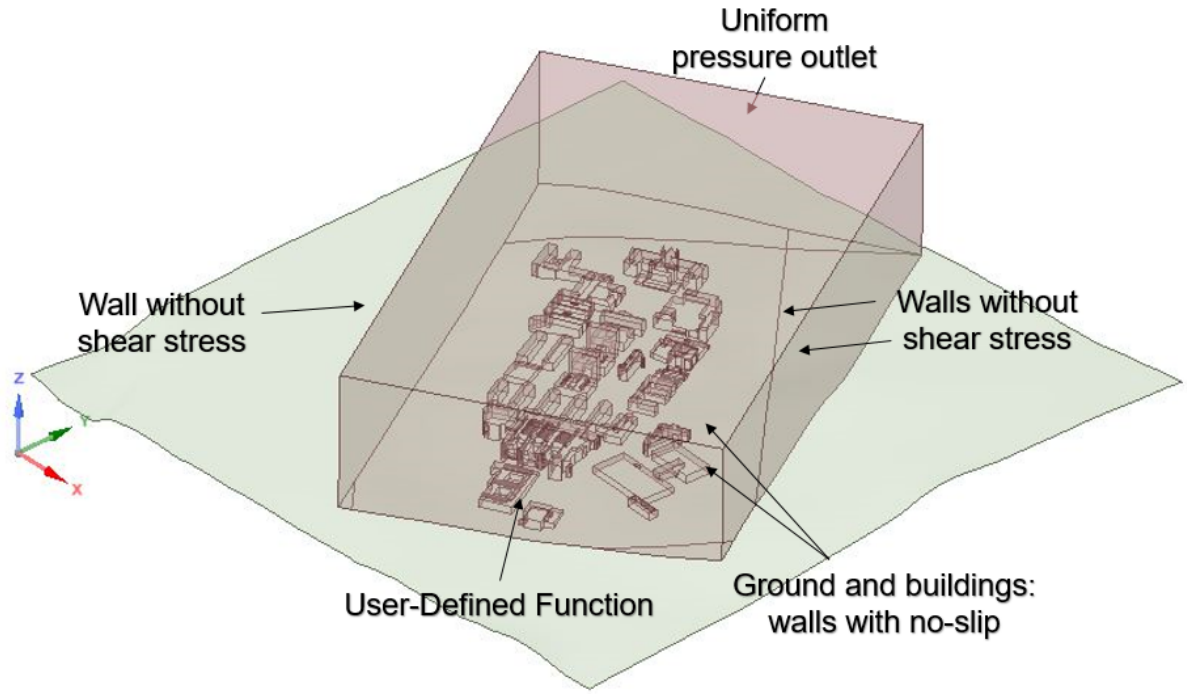


Figure 2.6: Boundary conditions

3. Numerical Validation

It is acknowledged that a model validation should be conducted due to uncertainties in the CFD code. The current simulations are validated by on-site measurements at five locations at the campus, as previously mentioned.

Figure 3.1 illustrates the simulated velocity profile versus the measured wind data and an estimated velocity profile based on the measurement. Where (a), (b), (c), (d), and (e) represent location no. 1, 2, 3, 4, and 5 respectively. In general, it can be seen that the majority of the predicted velocities computed by the CFD simulations fits the measured data, and the curves have the same shape as the estimated velocity profile. Especially at the inlet and close to the inlet where buildings do not influence the wind flow. On the other hand, the velocity profile at the location no. 5 (above the roof of the $Varmeteknisk$ building) deviates slightly with one standard deviation from the measurement observations (see Figure 3.1e). The noticeable differences between estimated and simulated profile below the measurement heights are due to separating and recirculating flow over the rooftop. The estimated profile was conducted by the *Power Law* which do not consider recirculating flow. Moreover, to verify that the simulation results are considered adequately compared to the measured data, they are compared by using the hit rate correlation by Schlnzen et al. (2004). The hit rate, q, is an indicator of how valid the simulation results are compared to measured data. It defines what percentage of the simulated results is within a range of one relative standard deviation from the measured data. They are considered acceptable for validation if $q > 66\ \%$. The correlation is described as follows (Schlnzen et al., 2004)

$$N_i = \begin{cases} 1, \left|\frac{P_i - O_i}{O_i}\right| \leq RD \\ 0, otherwise \end{cases} \tag{3.1}$$

$$q = \frac{1}{n}\left(\sum_{i=1}^{n} N_i\right) \times 100 \tag{3.2}$$

where n is the total number of data points, O_i is the simulated data, P_i is the measured data and RD is the relative standard deviation. The relative standard deviation is defined as the standard deviation divided by the mean value of the measured wind data. The correlation is based on normalized values, hence the measured and simulated velocity data are normalized by the reference velocity. The normalized data is expressed in the following Equations 3.3 and 3.4.

$$O_i = \frac{U_{simulated}}{U_{ref}} \tag{3.3}$$

$$P_i = \frac{U_{measured}}{U_{ref}} \tag{3.4}$$

Further, the hit rate values for all simulations are shown in table 3.1. It can be seen that the hit rate value at location no. 5 is within the acceptable range, even though the calculated velocity profiles differ from the measured data. This is due to the a lower magnitude of the standard deviation of the wind data. Moreover, the hit rate values meet the requirement $q > 66\%$, hence the numerical model in this study is confirmed applicable for simulating the wind flow through Gløshaugen campus.

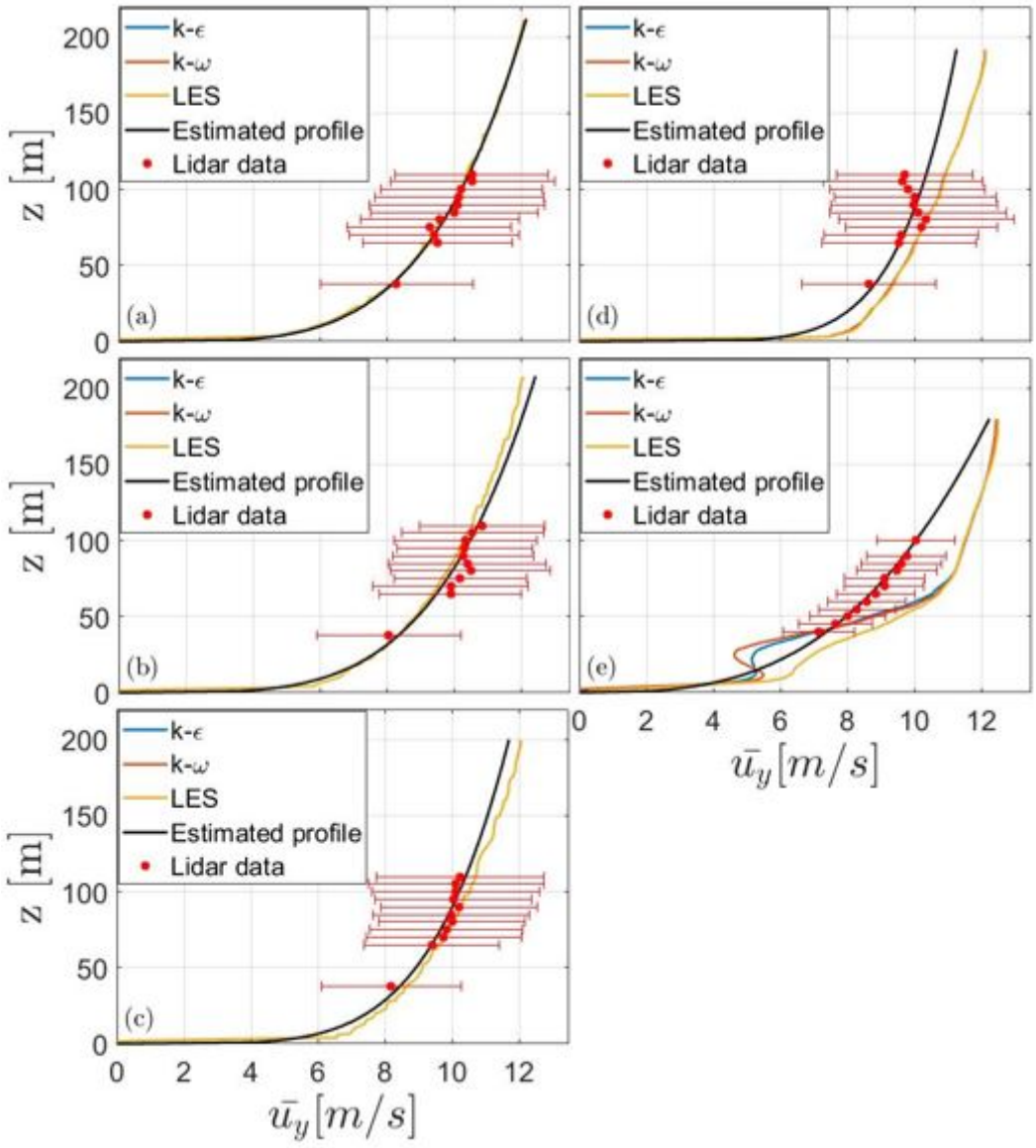


Figure 3.1: Comparison of simulated data versus measured data at location no.1 (a), location no. 2 (b), location no. 3 (c), location no. 4 (d) and location no. 5 (e). The height scale (a), (b), (c) and (d) is above the ground, while (e) is above the roof of the $Varmeteknisk$ building.

Table 3.1: Hit rate values for the simulation results at all location with different turbulence models.

Location	Turbulence model	Hit rate [%]
1	$k-\epsilon$	100
	$k-\omega$	100
	LES	100
2	$k-\epsilon$	100
	$k-\omega$	100
	LES	100
3	$k-\epsilon$	100
	$k-\omega$	100
	LES	100
4	$k-\epsilon$	90.91
	$k-\omega$	90.91
	LES	90.91
5	$k-\epsilon$	91.67
	$k-\omega$	91.67
	LES	91.67

4. Results and Discussion

Urban environment has a significant influence on the flow field. When the wind interacts with buildings, a separation of the wind flow occurs with an inner and outer layer. The outer layer experiences an acceleration, while the wind velocity decreases and the turbulence intensity increases within the inner layer downwind due to wake production and recirculating flow. Therefore, it is essential to analyze the wind flow characteristics in an urban environment in advance of installing wind turbines (Mertens, 2006).

In the coming sections, some of the figures are displayed as contour plots with relative height, z, above the campus surface. Relative height means that the height is specific for each building and position on the ground. For illustrative purpose, there are shown a plane with relative height 130 meters above the campus in Figure 4.1. Additionally, all the coming figures correspond to the same coordinate system as defined in the right corner of the mentioned figure, where Y is the stream-wise-direction, X is cross-stream-wise-direction, and Z is vertical-direction.

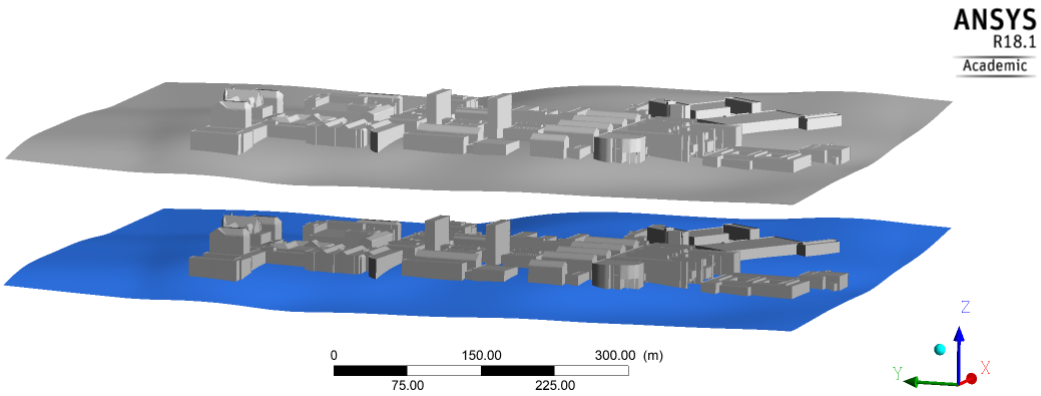


Figure 4.1: Plane for contour plots.

4.1 Wind Velocity

Contour plots of the $\overline{u}_y$ from $k-\epsilon$, $k-\omega$ and LES models are shown in Figures 4.2, 4.3 and 4.4. From the figures, it can be observed that the velocity increases with height above buildings and ground. It especially increases at the high-rise building at the center of the university and the buildings closer to the inlet. However, in-between buildings and upwind- and downwind of buildings, the velocity variation is highly noticeable. This is due to stagnation velocity upwind, while wake production and low-velocity recirculating flow occur downwind when the flow interacts with obstacles.

With that being said, Figure 4.5 illustrates the $\overline{u}_y$ through the middle of the campus simulated by the LES model (the same figure simulated by $k-\epsilon$ and $k-\omega$ is shown in Appendix 5.5.1). It can clearly be seen that the simulation is able to predict several of the physical phenomenon previously mentioned. If one takes a closer look at the high-rise building in the middle of the campus, one can observe flow stagnation at the upwind face and flow separation at the upwind edge. The outer layer accelerates, while the inner layer velocity decreases by the recirculating flow. As stated, the high-velocity region is at the upwind edge, however, as the height increases above the inner layer, the high-velocity area enlarges.

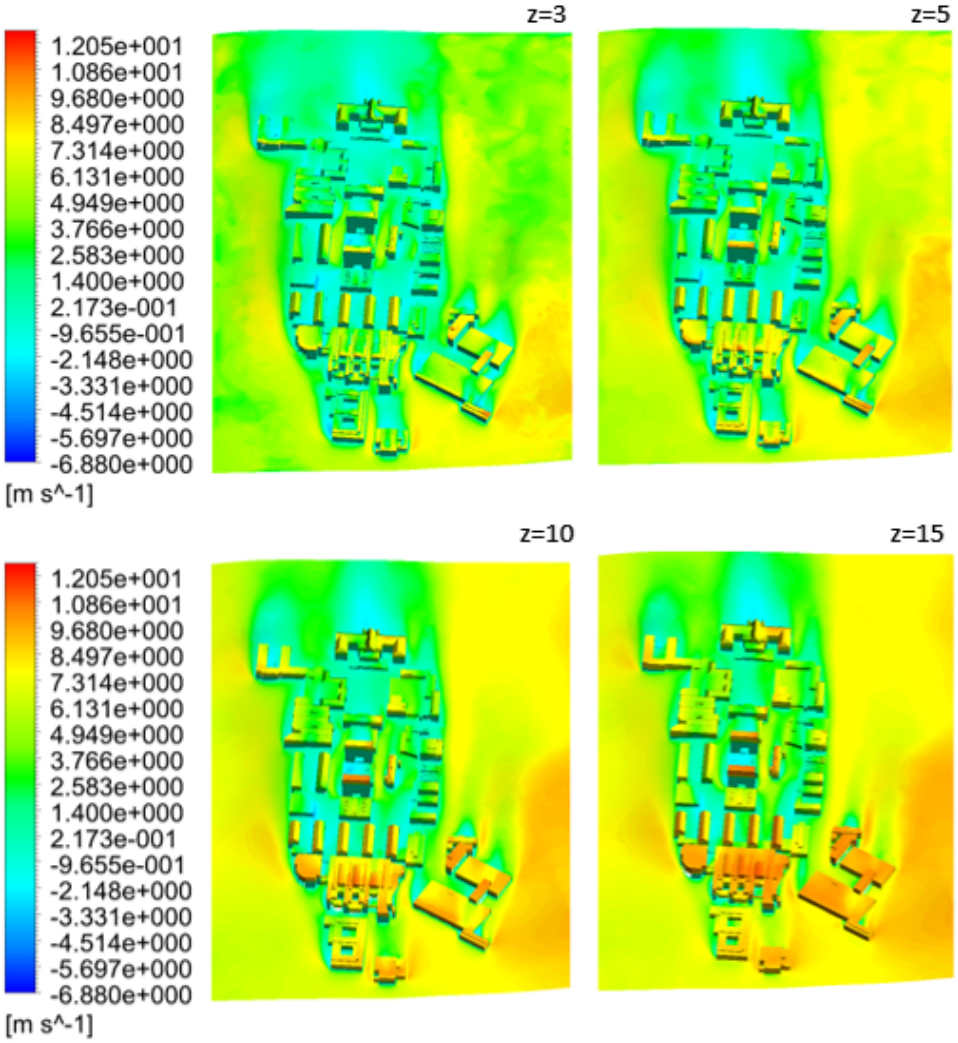


Figure 4.2: $\overline{u_y}$ at z=3 to z=15 meters above ground and buildings by $k - \epsilon$ model.

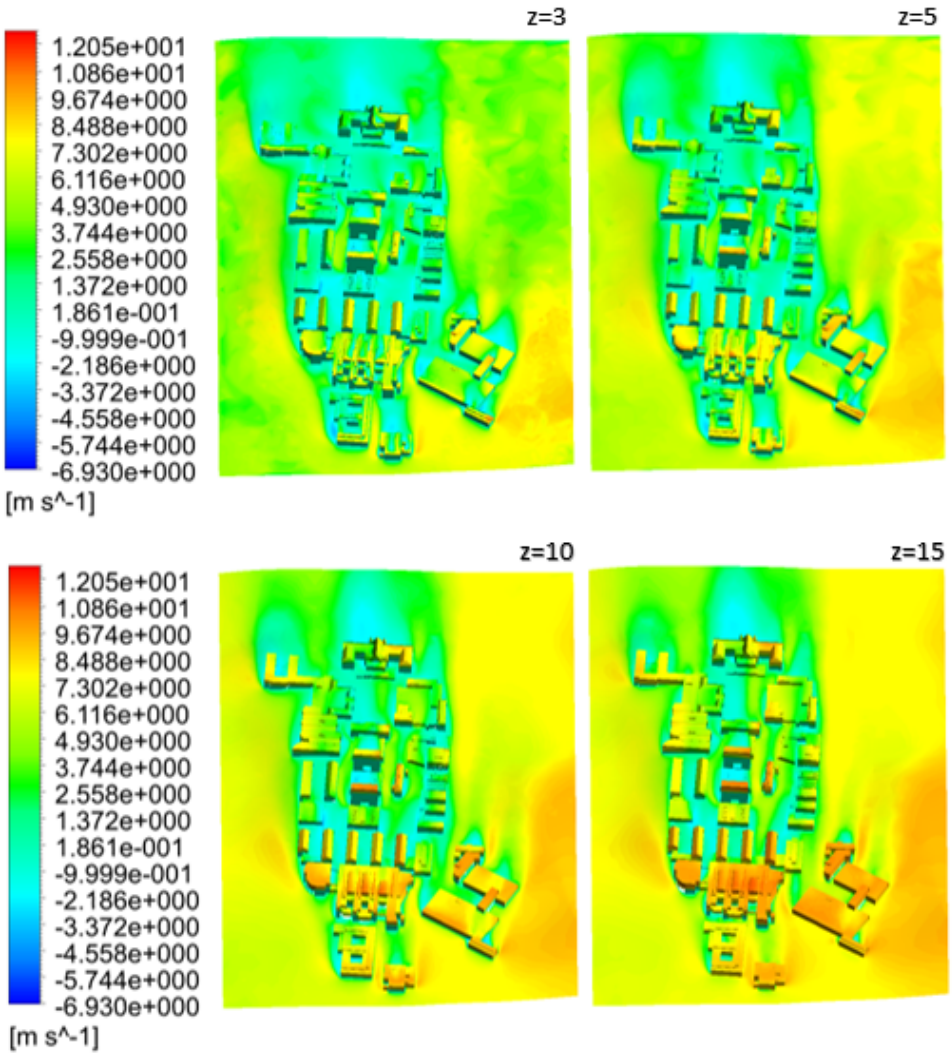


Figure 4.3: $\overline{u_y}$ at z=3 to z=15 meters above ground and buildings by $k-\omega$ model.

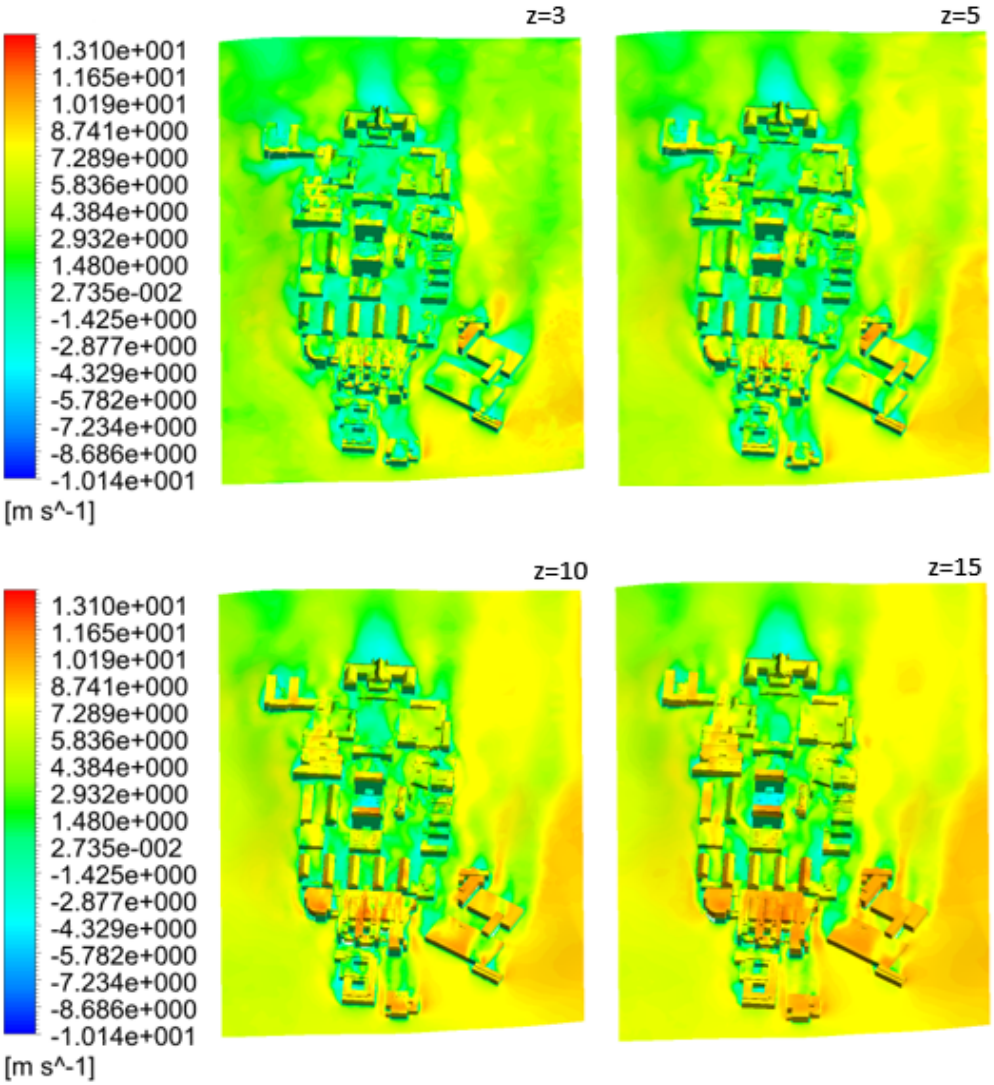


Figure 4.4: $\overline{u_y}$ at z=3 to z=15 meters above ground and buildings by LES model.

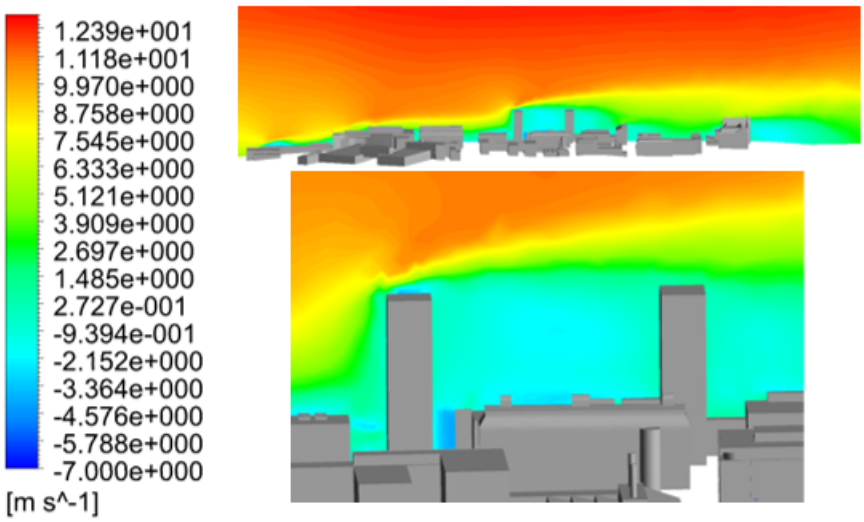


Figure 4.5: $\overline{u_y}$ at a center line along the campus by LES.

4.2 Turbulence

4.2.1 Turbulence Intensity

This section describes the turbulence intensity, I, distribution of the wind flow through campus. The turbulence intensity is defined as follows

$$I = \frac{1}{u}\sqrt{\frac{2k}{3}} \tag{4.1}$$

where k is defined by Equation 2.5 and u is the mean wind velocity defined as follows

$$u = \sqrt{u_x^2 + u_y^2 + u_z^2} \tag{4.2}$$

where u_i is the mean velocity component. The turbulence intensity affects performance and lifetime of wind turbines (Wang et al., 2018c). According to the International Electrotechnical Commission (IEC) Standard 61400-2, turbines should not be exposed to wind that contains more than $25\ \%$ turbulence (Wang et al., 2018c). Thus, it is important to identify the locations at the university where the turbulence intensity level is below the mentioned criteria. Furthermore, there is used Intensity and Hydraulic Diameter as Specification Method for inlet boundary condition in the CFD set-up for $k-\epsilon$ and $k-\omega$ models, where the turbulence intensity is set to be equal $0.1\ \%$. While in the LES set-up, the Fluctuating Velocity Algorithm is set to No Perturbations, which means that the stochastic velocity components of the flow are neglected, and the instantaneous velocity is equal to the mean velocity. Hence, there is no creation of fluctuation velocity components at the inlet. Thus, the turbulence intensity is negligible (Ansys-Inc, 2017b).

The turbulence intensity level at $z = 25$ meters above the campus for $k-\epsilon$, $k-\omega$ and LES are shown in the Figure 4.6, where the scale is global for easier comparison. In general, the RANS models calculate a larger magnitude of turbulence intensity over a wider area compared to LES. However, one can see similarities of the turbulence intensity distribution through the campus for the different turbulence models. The center building and the buildings near the

inlet experience a smaller magnitude of turbulence compared to the other buildings. While in-between buildings and at the downwind face of buildings, the wind are more affected by turbulent flow. This can be compared with the velocity contour plot from the Figures 4.2, 4.3, and 4.4. One can observe regions with recirculating flow between and behind buildings, which increases the amount of turbulence. Moreover, in all three cases, the building at the center of the university has the lowest magnitude of turbulence intensity. Hence, the flow characteristics around that building, called $Sentralbygget$, will be further analyzed to determine the optimal location for installing wind turbines.

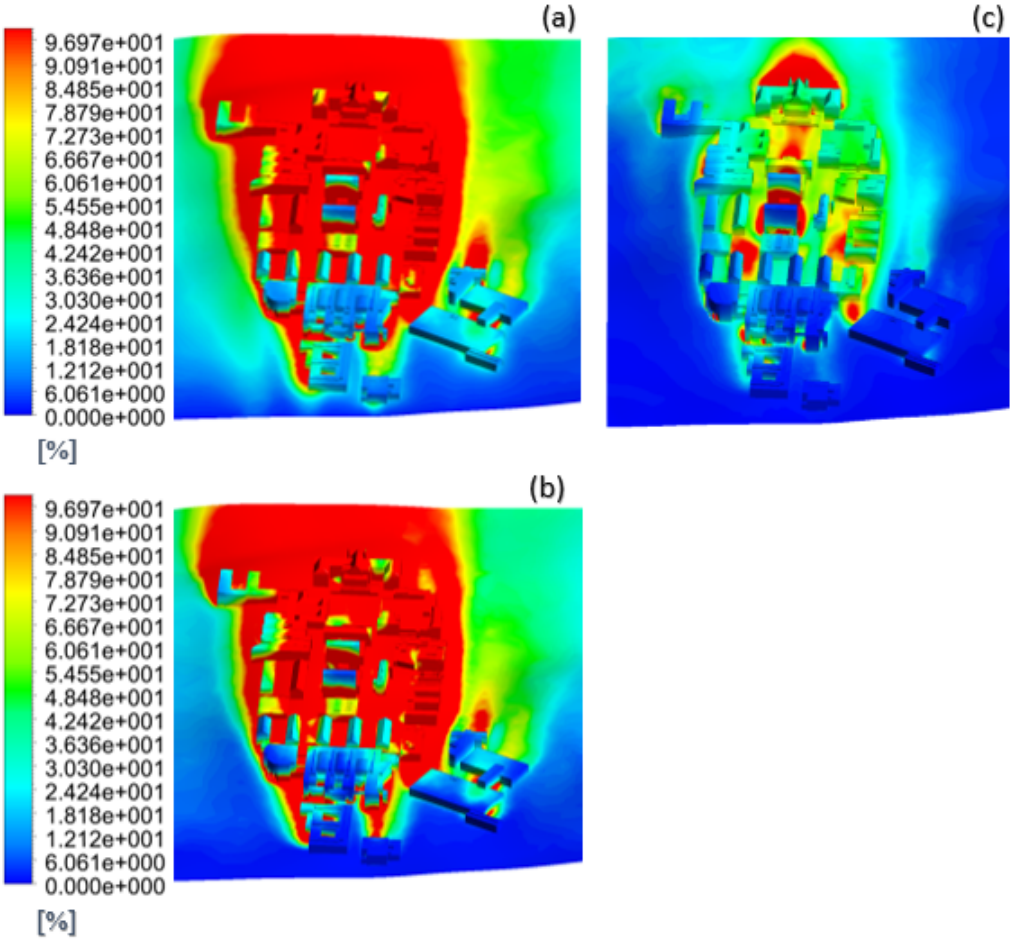


Figure 4.6: Turbulence intensity simulated by $k-\epsilon$ (a), $k-\omega$ (b), and LES (c) at z=25 meters.

For further analysis of the turbulence intensity above $Sentralbygget$, the roof is separated into fifteen regions - three rows and five columns, as shown in Figure 4.7. For each region, there is calculated how the turbulence intensity develops towards the upper boundary of the computational domain. The turbulence intensity at column no. 1 at the roof is shown in Figure 4.8 while the other regions are shown in Appendix 5.5.2.

One can see that the turbulence intensity increases towards a peak close to the wall before it gradually decreases towards the inflow condition. This is a characteristic growth of turbulence intensity due to the turbulent boundary layer (the turbulent boundary layer is shown in figure 4.10). The turbulent boundary layer is divided into an inner and outer layer, where the inner layer is separated into three regions. The axes are described by non-denationalized variables, where the x-axis is distance to the wall, and the y-axis is the mean velocity (these variables

C1	C2	C3	C4	C5
B1	B2	B3	B4	B5
A1	A2	A3	A4	A5

Figure 4.7: The roof area separated into fifteen regions.

and a further explanation of near-wall treatment are further explained in Appendix 5.4). In the innermost layer, the $Viscous\ Sublayer$, viscous forces dominants the flow field. Due to large mean velocity gradients, which cause high turbulence production the fluctuating velocity components are at its largest close to the wall (Versteeg and Malalasekera, 2007a). The mean velocity components experience a maximum value far away from the wall and sharply decreases close to the wall due to the no-slip condition (Versteeg and Malalasekera, 2007a). With low mean velocity and high fluctuating components will the turbulence intensity increase in the near-wall region. Far away from the wall, viscous effects are negligible, and inertial forces dominate the flow. Which means that the mean velocity increases and the eddy motions - and the associated velocity fluctuations, have less impact on the flow. The turbulence intensity will then decrease towards the free stream region (Versteeg and Malalasekera, 2007a).

The turbulence intensity stabilizes towards the inflow condition at approximately 40 meters above the roof in all three cases. For instance, by following the turbulence intensity curve $A1$ in Figure 4.8a, the turbulence intensity is below $25\ \%$ when z reaches 13.3 meters above the roof. Hence, a wind turbine can be installed above that height by following the IEC criteria. The remaining values for the turbulence intensity $\geq 25\ \%$ dependent on height for every region on the roof is displayed in Table 4.1.

Even though the shapes of the turbulence intensity seems legit in a physical manner, the results form $k-\epsilon$ and $k-\omega$ differ from the turbulence intensity calculated by LES. The turbulence intensity magnitude tends to increase from the frontier position at the roof to the downwind edge for lower heights (see Figure 4.8c). By comparing that with the velocity profile illustrated in Figure 4.5, it can be observed that the velocity decreases below zero towards the downwind edges. That indicates an inner layer with higher turbulence due to recirculating flow. The opposite occurs for the two RANS models where the turbulence intensity is at its largest at the upwind edge. The turbulence intensity distribution across a center-line through campus for the turbulence models are shown in Figure 4.9. $k-\epsilon$ and $k-\omega$ models calculate in general a higher turbulence intensity both in front- and above the roof compared to the LES simulation. As mentioned earlier, several $k-\epsilon$ simulations have been performed and validated by measurements to predict wind distribution. Driver and Seegmiller (1985) simulated the reattaching share layer in a rearward-facing step-flow, while Wright and Easom (2003) simulated flow over a building at full scale. Based on the measurements, $k-\epsilon$ overestimate the eddy viscosity, thus an overestimation of the turbulence intensity. While Jiang et al. (2003) conducted a LES simulation of air flow through an opening inside a "building-like" model and validated the results with wind tunnel measurements. Both the mean- and fluctuating velocity components were correctly predicted, which indicates that the turbulence intensity is calculated correctly. Furthermore, the highest wind speed was $12m/s$ and when the wind flows around building with high speed, most of the energy is contained in the larger eddies. Hence, when the larger eddies are more important will the LES model provide accurate flow results when the model can directly solve the

large-eddy motions Jiang et al. (2003). With that being said, the largest velocity in the current simulation and measurements is $12.7m/s$, which indicates that the flow prediction is accurate. Additionally, Tutar and Oguz (2002e) performed a CFD analysis of wind flow above a single building, where it was found that the $k-\epsilon$ model predicts a significantly higher value of TKE in-front of a building, compared to the prediction made by the LES model, which is consistent with the present simulation. On the other hand, the current geometry is more complicated, and the mesh is different. That will influence how the equations are solved and the final flow results. Another reason for the differences in turbulence intensity is the inflow condition. As mentioned, the turbulence intensity is equal to 0.1 % for $k-\epsilon$ and $k-\omega$, while no perturbations for the LES simulation. Even though 0.1 % is defined as a low value of turbulence intensity, it is still larger than if the turbulence is neglected at the inlet. The vortices created at the inlet will spread an can be the reason for higher turbulence for the RANS simulations.

However, the differences in turbulence intensity are not easy to locate precisely without more field measurements. As there were no experimental data available, the turbulence intensity was assumed in present work. To gain knowledge about the turbulence effects, the inlet level was increased to 5 and 10 % and compared with the 0.1 % case.

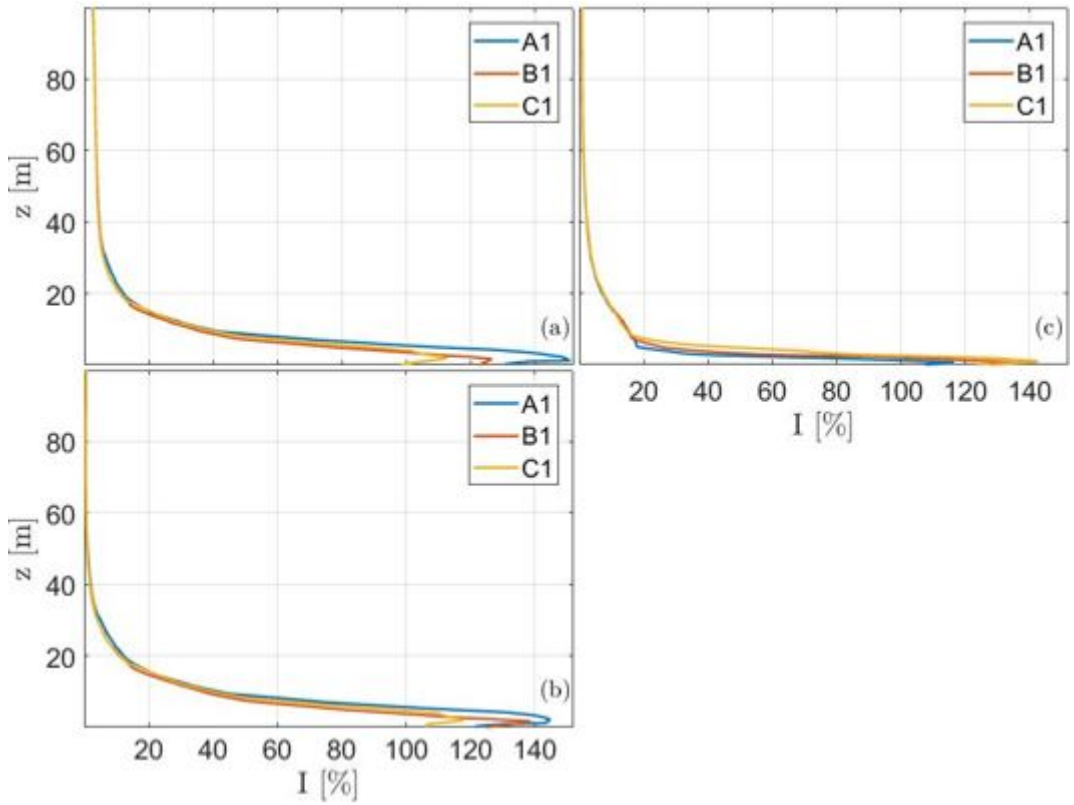


Figure 4.8: Turbulence intensity simulated by $k-\epsilon$ (a), $k-\omega$ (b), and LES (c) at column no. 1 at the roof.

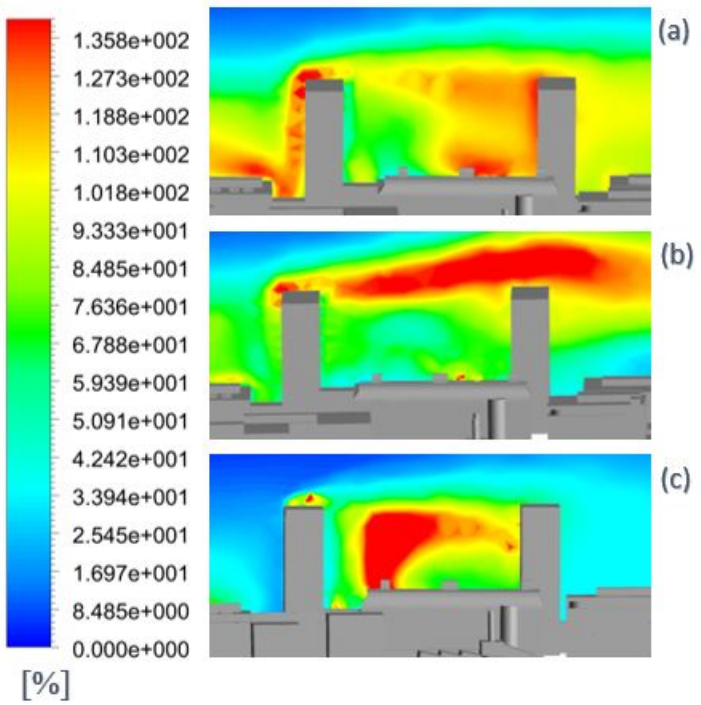


Figure 4.9: Turbulence intensity simulated by $k-\epsilon$ (a), $k-\omega$ (b), and LES (c) across a center-line above the roof of $Sentralbygget$.

Table 4.1: Height levels above the roof with turbulence intensity $\geq 25\%$ simulated by $k-\epsilon$, $k-\omega$ and LES.

Model \ Region	A	z [m] ($I \geq 25\%$)	B	z [m] ($I \geq 25\%$)	C	z [m] ($I \geq 25\%$)
$k-\epsilon$	A1	z $\leq$ 13.1	B1	z $\leq$ 12.4	C1	z $\leq$ 13.2
	A2	z $\leq$ 9.40	B2	z $\leq$ 11.7	C2	z $\leq$ 13.3
	A3	z $\leq$ 10.8	B3	z $\leq$ 11.9	C3	z $\leq$ 13.4
	A4	z $\leq$ 8.50	B4	z $\leq$ 11.5	C4	z $\leq$ 12.3
	A5	z $\leq$ 6.40	B5	z $\leq$ 10.0	C5	z $\leq$ 10.0
$k-\omega$	A1	z $\leq$ 13.9	B1	z $\leq$ 13.2	C1	z $\leq$ 13.9
	A2	z $\leq$ 9.40	B2	z $\leq$ 11.7	C2	z $\leq$ 13.2
	A3	z $\leq$ 10.8	B3	z $\leq$ 10.3	C3	z $\leq$ 13.6
	A4	z $\leq$ 9.10	B4	z $\leq$ 12.1	C4	z $\leq$ 13.2
	A5	z $\leq$ 6.60	B5	z $\leq$ 8.90	C5	z $\leq$ 10.3
LES	A1	z $\leq$ 3.00	B1	z $\leq$ 3.50	C1	z $\leq$ 6.80
	A2	z $\leq$ 3.60	B2	z $\leq$ 5.30	C2	z $\leq$ 6.20
	A3	z $\leq$ 2.90	B3	z $\leq$ 5.10	C3	z $\leq$ 9.20
	A4	z $\leq$ 2.00	B4	z $\leq$ 4.20	C4	z $\leq$ 6.60
	A5	z $\leq$ 1.00	B5	z $\leq$ 4.30	C5	z $\leq$ 4.90

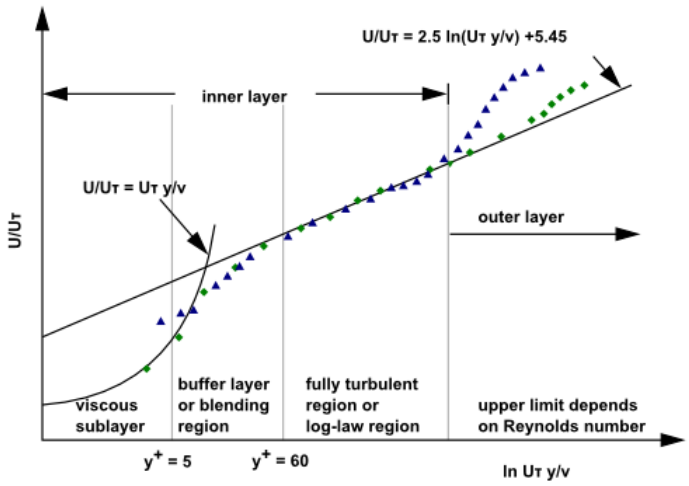


Figure 4.10: Turbulent boundary layer regions (ANSYS-Inc, 2017a).

4.3 Change of Turbulent Parameters at the Inlet Condition in the CFD set-up

In the earlier described simulations with $k-\epsilon$, $k-\omega$ and LES models is the turbulence intensity at the inlet set to be equal to 0.1 % and no perturbations. Ansys-Inc (2017d) indicates that a turbulence intensity level less or equal to 1 % is considered low, while greater or equal to 10% is considered as high. Therefore are there conducted simulations where the turbulence intensity is equal 5 % and 10 % which will be presented in what follows. For simplicity, only values for the first column at the roof of $Sentralbygget$ are shown.

4.3.1 Velocity

The time-average streamwise velocity for column no. 1 at the roof is displayed in Figure 4.11. The velocity is simulated when the turbulence intensity at the inlet is equal to 0.1 %, 5 %, and 10 % for $k-\epsilon$ and $k-\omega$ and no perturbations and 5 % by LES. As shown, the time-average velocity hardly changes dependent on the turbulence intensity settings at the inlet. Thus, a further analysis of the turbulence intensity at the roof will be presented in the next section.

4.3.2 Turbulence Intensity

From the turbulence intensity definition given in Equation 4.1 will the fluctuating components increase as the mean velocity increases in the boundary layer, when the turbulence intensity is equal to 5 % and 10 %. The turbulence intensity distribution is illustrated in Figure 4.12. By comparing the turbulence intensity described in Section 4.2.1 and when the turbulence intensity is increased, one can see that the inlet value of turbulence intensity has a significant influence on the turbulence distribution throughout the campus for the RANS models.

First of all, it can be observed that the shape of the turbulence intensity above the roof for all cases is sufficient, due to turbulent boundary layer theory (Versteeg and Malalasekera, 2007a). The turbulence intensity increases at lower height levels due to low mean velocity and vortex creation near the wall. As the height increases, the mean velocity components increases more

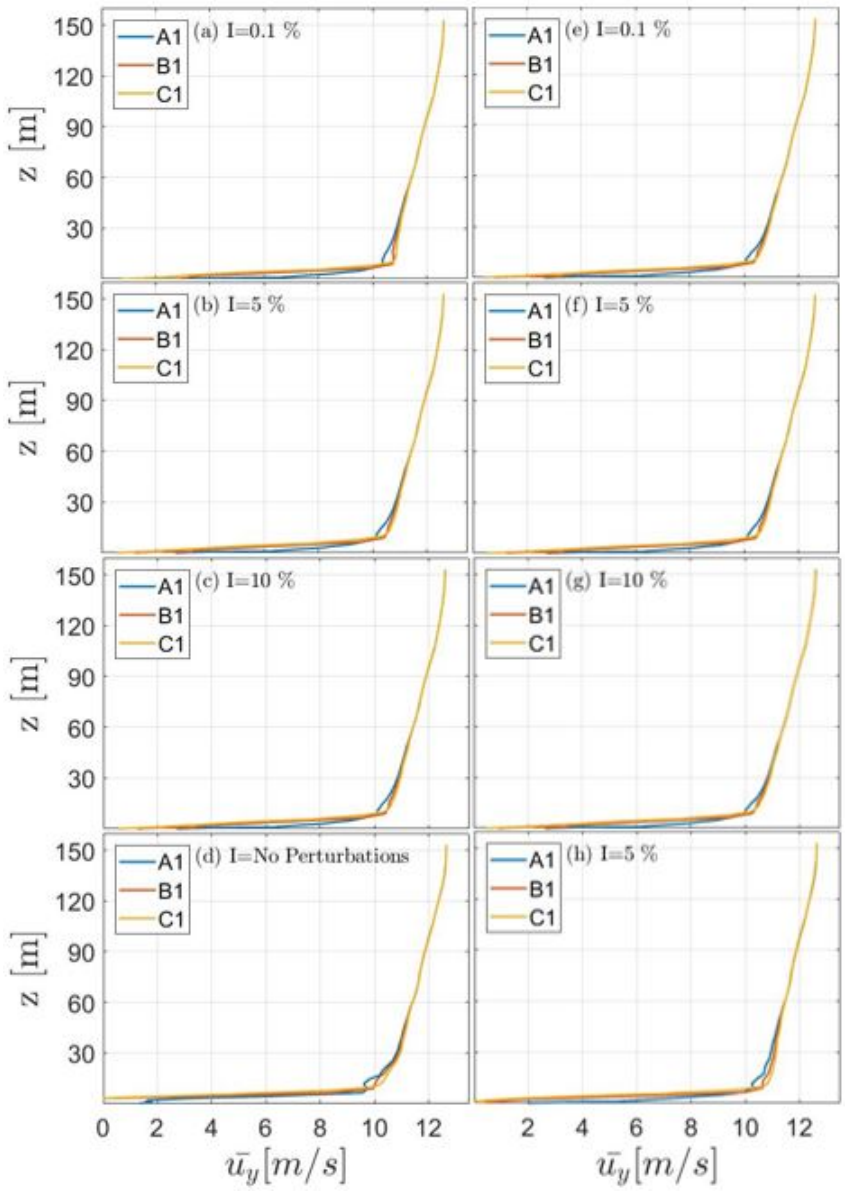


Figure 4.11: $\overline{u}_y$ simulated by $k-\epsilon$ (a,b,c), $k-\omega$ (e,f,g), and LES (d,h) at column no. 1 at the roof.

rapidly, and the turbulence intensity decreases. Secondly, the turbulence intensity stabilizes at a constant value throughout the boundary layer. The LES simulation is highly affected by the turbulence intensity at the inlet. The turbulence is just slightly increased close to the roof. The turbulence intensity calculated by $k-\omega$ at 4.12g differs slightly from the others. The turbulence vortices at the inlet spreads further downwind in the domain compared to $k-\epsilon$, hence the turbulence intensity increases for some heights in the boundary layer (this is shown in Appendix 5.12). Moreover, the turbulence intensity level calculated by the RANS models never falls below the IEC criteria when $I \geq 5,10\ \%$. That means, in a purely physical manner, that the fluctuating velocity components still influence the mean velocity profile throughout the boundary layer. This is questionable due to common turbulence theory that states that shear- and viscous forces decreases along the boundary layer towards the free stream region (Ferziger, 2002). With that being said, when the turbulent condition at the inlet is equal to $5\ \%$ and $10\ \%$, turbulent vortices are created. These vortices will develop throughout the campus and increase

the level of intensity.

Comparing the turbulence intensity behavior as previously described with the turbulence intensity calculated at location no. 2 at the inlet (shown in Figure 4.13, for simplicity, there are only shown values at location no. 2). One can see that the turbulence intensity at the inlet for LES is lower than the other simulations. With lower differences at the inlet will there also be fewer differences further downstream. Furthermore, at the inlet in the RANS simulations, the turbulence intensity increases in the entire boundary layer. As mentioned earlier, turbulence is created by viscous forces in the inner layer of the turbulent boundary layer. Further away from the wall, towards the free stream region the flow is only depended on inertia forces. Thus the turbulence intensity should decrease towards zero (Versteeg and Malalasekera, 2007a). With that being said, one can see that the inlet condition for turbulence intensity influences the entire flow field. Thus, for further research, there is essential to estimate the amount of turbulence at the inlet to increase the accuracy of the turbulence intensity above each building at the campus.

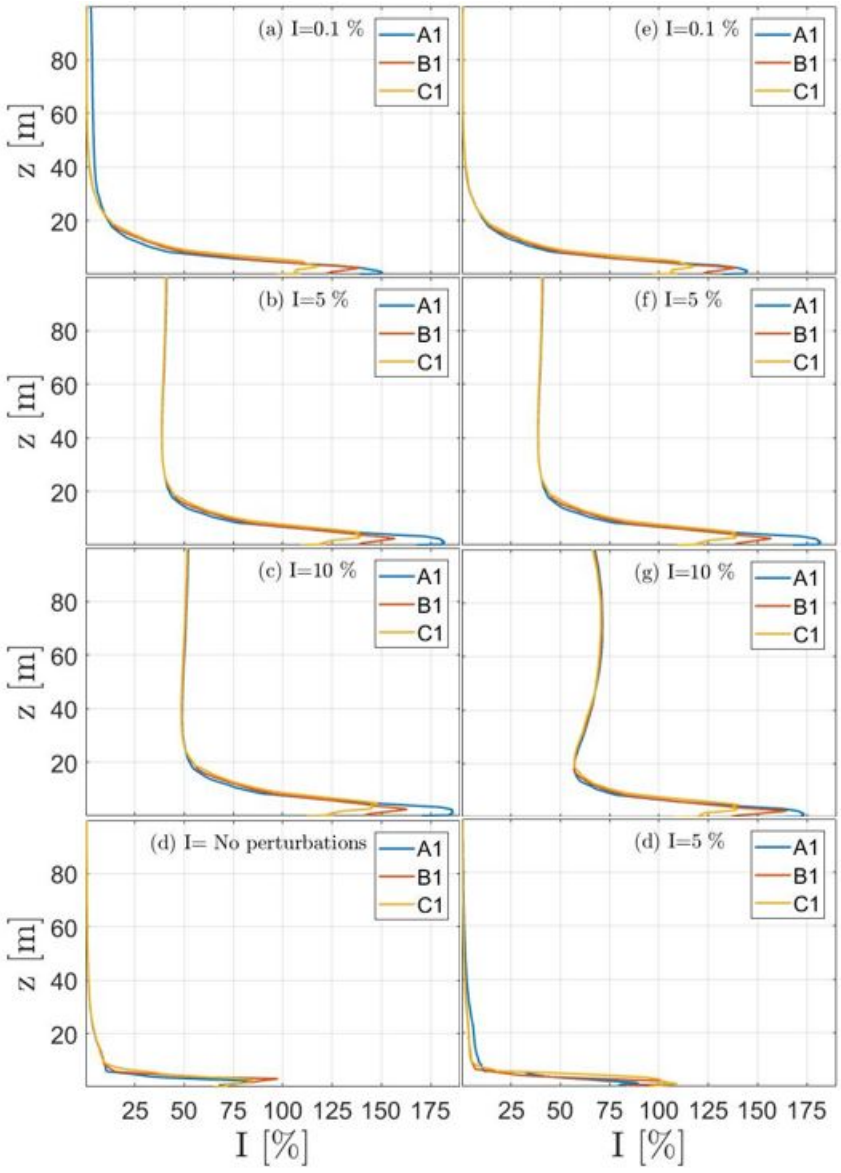


Figure 4.12: Turbulence intensity simulated by $k-\epsilon$ (a,b,c), $k-\omega$ (e,f,g), and LES (d,h) at column no. 1 at the roof.

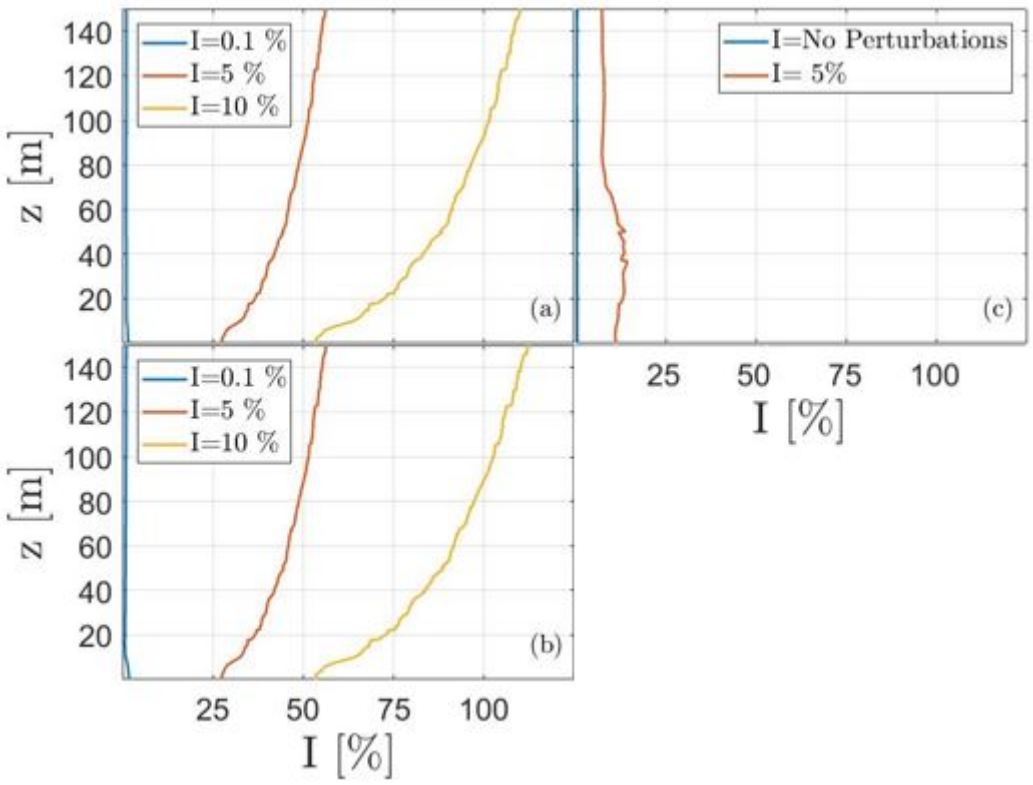


Figure 4.13: Turbulence intensity at the inlet simulated by $k-\epsilon$ (a), $k-\omega$ (b), and LES (c).

4.3.3 Turbulent Length Scale at the Inlet

Turbulent length scale l is related to the size of the largest eddies that contains energy, and it is defined as (Versteeg and Malalasekera, 2007b).

$$l = \frac{k^{\frac{3}{2}}}{\epsilon} \tag{4.3}$$

Figure 4.14 illustrates the turbulent length scaled at the inlet for two simulations performed by the $k-\epsilon$ model (the scale is global for easier comparison). In both simulations, the turbulence intensity is equal to $5\ \%$, while the hydraulic diameter is equal to 50 and 100 meters respectively. One can see that the hydraulic diameter influences the size of the eddies. However, the velocity profiles and turbulence intensity downstream in the domain are not affected by the change of turbulent length scale at the inlet (see Figures 4.15 and 4.16).

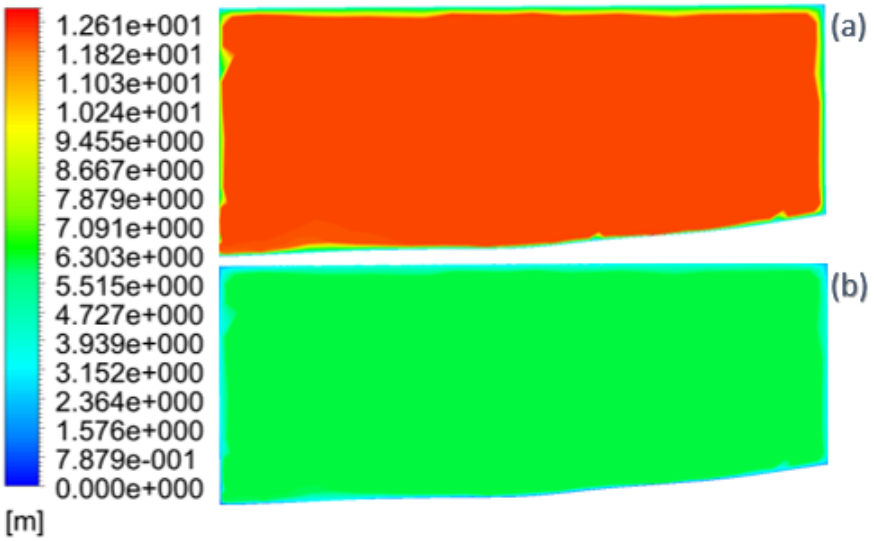


Figure 4.14: Turbulent length scale at the inlet. Simulated by $k - \epsilon$ with hydraulic diameter equal 100m (a) and 50m (b).

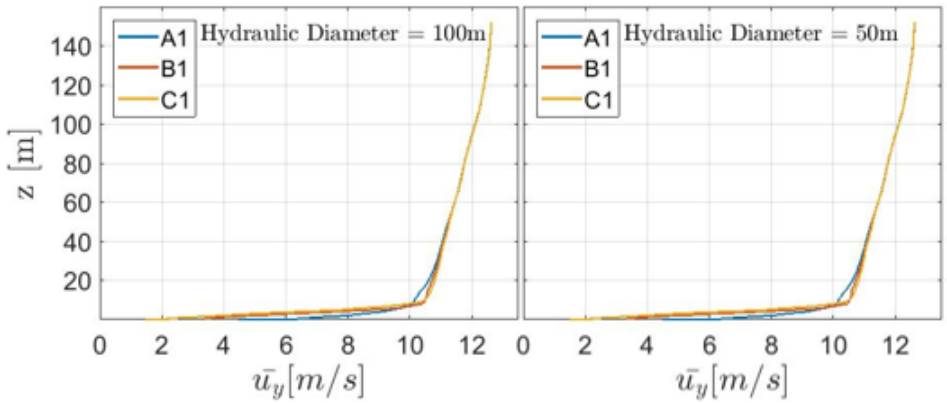


Figure 4.15: Velocity profile at region no. 1 at the roof calculated for inlet hydraulic diamter equal to 50 and 100 meters.

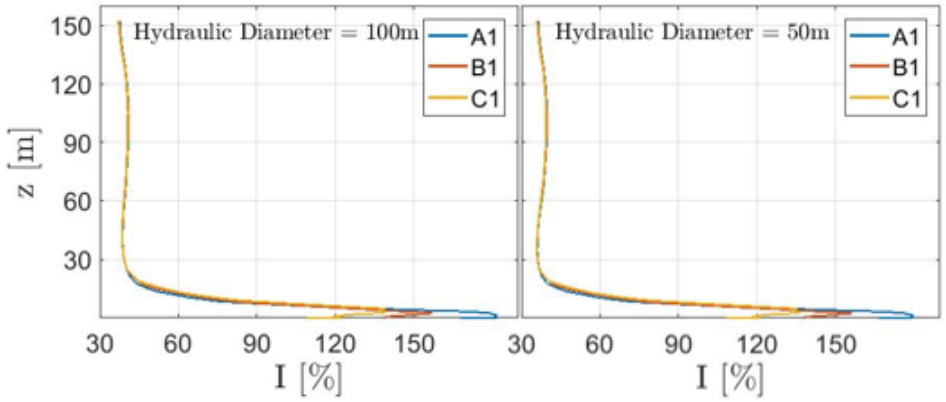


Figure 4.16: Velocity profile at region no. 1 at the roof calculated for inlet hydraulic diamter equal to 50 and 100 meters..

4.4 Power Density

In the previous sections, there have been discussing possible locations for mounting wind turbines based on turbulence intensity. In this section, it is presented results for the power density above the roof. Due to too high turbulence intensity by $k-\epsilon$ and $k-\omega$ when $I \geq 5,10\ \%$ at the inlet will the following section only discuss power density when the turbulence intensity is equal to $0.1\ \%$ and no perturbations. The power density is defined as follows (Ledo et al., 2011b)

$$P_d = \frac{1}{2}\rho u^3 \tag{4.4}$$

where ρ is the density of the air, and u is the mean wind speed defined by Equation 4.2. The development of the power density at column no. 1 at the roof for the three models are displayed in Figure 4.17, while the power density at the remaining parts of the roof are illustrated in Appendix 5.6. First of all, the highest calculated value of z dependent on $I \geq 25\ \%$ is 13.9

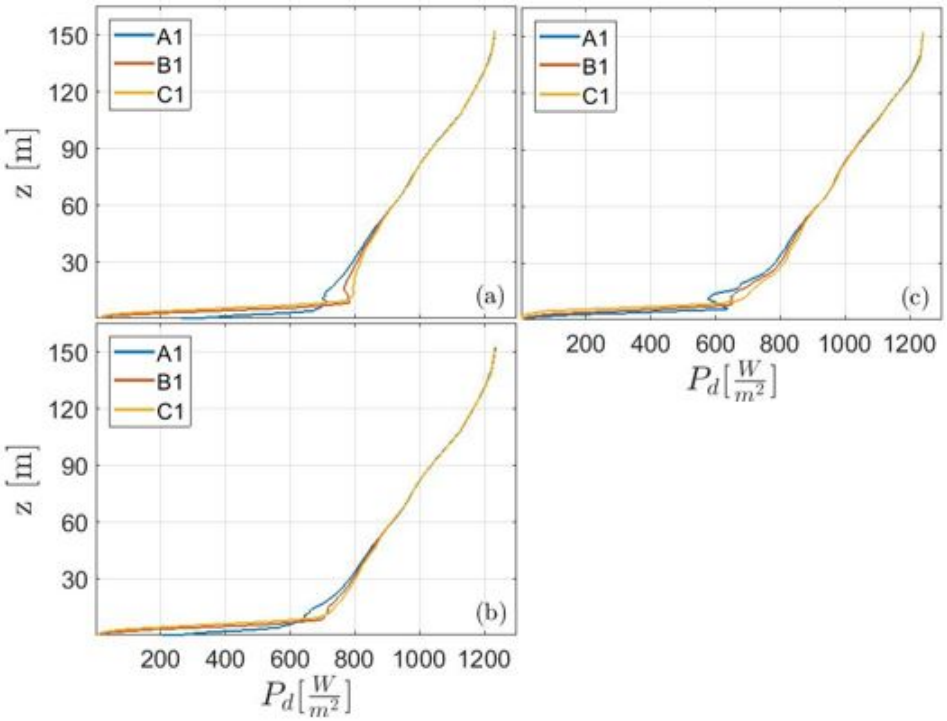


Figure 4.17: Power density at column no. 1 calculated by $k-\epsilon$ (a), $k-\omega$ (b) and LES (c).

meters above the roof (see Table 4.1). Which means that every position above that height is a suitable location for wind turbines, based on the IEC criteria. Finally, above that height, the power density is within the range of $580 W/m^2$ to 1230 W/m^2 at the upper boundary of the computational domain.

5. Conclusion

In this study, a CFD analysis and local on-site measurements have been carried out as a wind resource assessment for estimating the wind flow at the university. There are used three turbulence models, $k-\epsilon$, $k-\omega$ and LES respectively, and different turbulence parameters to locate favorable sites for siting wind turbines at the campus.

Local on-site measurements have been performed to estimate the inlet velocity profile as well as to validate the CFD results. The simulated velocities overestimate the measured velocities; however, the simulation was confirmed applicable for predicting the wind flow through campus based on the hit rate correlation between the measured and simulated velocities.

The CFD simulation was performed with different turbulence intensity at the inlet. It was found that the velocity distribution through campus is independent of the turbulent inflow conditions. Furthermore, the models were in good agreement about the separation and recirculating flow, and that the high-velocity region is above the high-rise building in the middle of the campus.

Accordingly to the velocity, the low-turbulence region is above the high-rise building. Even though it is found similarities of the turbulence distribution throughout the campus for the different turbulence models, the RANS models predict, in general, a more significant magnitude of turbulence intensity in-front, in-between and downwind of buildings compared to the LES model. Similar to Tutar and Oguz (2002e), results show that $k-\epsilon$ models predict a higher turbulence in-front of buildings, while LES predict the highest level of turbulence at the downwind edge.

A detailed study of how the flow field dependent on turbulent inlet conditions was carried out. It showed that the RANS models are significantly affected by increasing the turbulence intensity at the inlet, compared to the LES model. When the turbulence intensity is equal to 5 and $10\ \%$ at the inlet, the fluctuating velocity components are increased, and created vortices at the inlet influence the entire flow field throughout the domain. On the other hand, it was found that the velocity and turbulence distribution is independent of the turbulent length scale at the inlet.

Based on the thorough analysis of the turbulence distribution of the campus, it is concluded that the high-rise building in the middle of the campus is the most suitable location for installing wind turbines.

Bibliography

Aliferis, A. D., Burkeland, S. N., Jessen, M. S., Meek, C., 2017. Pipe flow simulations for re up to 10^6, 1 to 3.

ANSYS-Inc, 2006a. Ansys: Modeling turbulent flows, introductory fluent training, 8 to 14.

ANSYS-Inc, 2006b. Ansys: Modeling turbulent flows, introductory fluent training, 8 to 14.

ANSYS-Inc, 2006c. Ansys: Modeling turbulent flows, introductory fluent training, 15 to 16.

Ansys-Inc, 2017a. Ansys fluent theory guide, 680.

Ansys-Inc, 2017b. Ansys fluent theory guide, 109.

Ansys-Inc, 2017c. Ansys fluent theory guide, 114 to 118.

Ansys-Inc, 2017d. Ansys fluent user guide, 288.

ANSYS-Inc, 2017a. Lecture from ansys module 07: Turbulence, introduction to ansys fluent, 5 to 14.

ANSYS-Inc, 2017b. Lecture from ansys module 07: Turbulence, introduction to ansys fluent, 23 to 24.

ANSYS-Inc, 2017c. Lecture from ansys module 07: Turbulence, introduction to ansys fluent, 50.

Blocken, B., Janssen, W., van Hooff, T., 2012. Cfd simulation for pedestrian wind comfort and wind safety in urban areas: General decision framework and case study for the eindhoven university campus. Environmental Modelling Software 30 (Supplement C), 1–20.

Blocken, B., Persoon, J., 2009. Pedestrian wind comfort around a large football stadium in an urban environment: Cfd simulation, validation and application of the new dutch wind nuisance standard. Journal of Wind Engineering and Industrial Aerodynamics 97 (5), 1 – 16.

Blocken, B., Stathopoulos, T., Carmeliet, J., Hensen, J., 2011. Application of computational fluid dynamics in building performance simulation for the outdoor environment: An overview. Journal of Building Performance Simulation (4), 5–6.

Burkeland, S. N., 2017. Urban wind: Cfd analysis of gløshaugen campus 1 (1), 20.

Driver, D. M., Seegmiller, H. L., 1985. Features of a reattaching turbulent shear layer in divergent channelflow. AIAA Journal 23, 1 – 9.

Ferziger, J. H., 2002. Computational methods for fluid dynamics.

Franke, J., Hirsch, C., Jensen, A., Krüs, H., Schatzmann, M., Westbury, P., Miles, S., Wisse, J., Wright, N., 2004. Recommendations on the use of cfd in wind engineering. In: Cost action C. Vol. 14. pp. 5–10.

Jiang, Y., Alexander, D., Jenkins, H., Arthur, R., 2003. Natural ventilation in buildings: measurement in a wind tunnel and numerical simulation with large-eddy simulation. Journal of Wind Engineering and Industrial Aerodynamics 91 (3), 331 – 353.

Ledo, L., Kosasih, P., Cooper, P., 2011a. Roof mounting site analysis for micro-wind turbines. Renewable Energy 36 (5), 1 – 3.

Ledo, L., Kosasih, P., Cooper, P., 2011b. Roof mounting site analysis for micro-wind turbines. Renewable Energy 36 (5), 10 – 12.

Meng, Y., Hibi, K., 1998. Turbulent measurements of the flow field around a high-rise buildings (76), 55–64.

Mertens, S., 09 2006. Wind energy in the built environment: Concentrator effects of buildings, 23 to 35.

Mohamed, M., Wood, D., 2016a. Computational modeling of wind flow over the university of calgary campus. Wind Engineering 40 (3), 1–2.

Mohamed, M., Wood, D., 2016b. Computational modeling of wind flow over the university of calgary campus. Wind Engineering 40 (3), 1–22.

Murakami, S., Mochida, A., 1988. 3-d numerical simulation of airflow around a cubic model by means of the k- model. Journal of Wind Engineering and Industrial Aerodynamics (2), 1–3.

Pope, S., 2000. Turbulent Flows, 2nd Edition. Cambridge University Press.

Rafailidis, S., Nov 1997. Influence of building areal density and roof shape on the wind characteristics above a town. Boundary-Layer Meteorology 85 (2), 3–11.

Schlnzen, K., Bchlin, W., Brnger, H., Eickhorn, J., Grawe, D., Schenk, R., Winkler, C., 2004. An evaluation guideline for prognostic microscale wind field models. 9th International Conference on Harmonization within Atmo-spheric Dispersion Modelling for Regulatory Purposes, 1 to 4.

Simonsen, A., 2016. Urban wind at the gløshaugen cammpus, 1 to 43.

Tabrizi, A. B., Whale, J., Lyons, T., Urmee, T., 2014a. Performance and safety of rooftop wind turbines: Use of cfd to gain insight into inflow conditions. Renewable Energy 67, 1–2.

Tabrizi, A. B., Whale, J., Lyons, T., Urmee, T., 2014b. Performance and safety of rooftop wind turbines: Use of cfd to gain insight into inflow conditions. Renewable Energy 67, 5.

Tutar, M., Oguz, G., 2002a. Large eddy simulation of wind flow around parallel buildings with varying configurations. Fluid Dynamics Research 31 (5), 1–2.

Tutar, M., Oguz, G., 2002b. Large eddy simulation of wind flow around parallel buildings with varying configurations. Fluid Dynamics Research 31 (5), 8–9.

Tutar, M., Oguz, G., 2002c. Large eddy simulation of wind flow around parallel buildings with varying configurations. Fluid Dynamics Research 31 (5), 1–27.

Tutar, M., Oguz, G., 2002d. Large eddy simulation of wind flow around parallel buildings with varying configurations. Fluid Dynamics Research 31 (5), 4–6.

Tutar, M., Oguz, G., 2002e. Large eddy simulation of wind flow around parallel buildings with varying configurations. Fluid Dynamics Research 31 (5), 14–20.

Versteeg, H. K., Malalasekera, W., 2007a. An Introduction to Computational Fluid Dynamics: The Finite Volume Method, 2nd Edition. Pearson Education.

Versteeg, H. K., Malalasekera, W., 2007b. An Introduction to Computational Fluid Dynamics: The Finite Volume Method, 2nd Edition. Pearson Education.

Wang, Q., Wang, J., Hou, Y., Yuan, R., Luo, K., Fan, J., 2018a. Micrositing of roof mounting wind turbine in urban environment: Cfd simulations and lidar measurements. Renewable Energy 115, 1 – 2.

Wang, Q., Wang, J., Hou, Y., Yuan, R., Luo, K., Fan, J., 2018b. Micrositing of roof mounting wind turbine in urban environment: Cfd simulations and lidar measurements. Renewable Energy 115, 1 – 16.

Wang, Q., Wang, J., Hou, Y., Yuan, R., Luo, K., Fan, J., 2018c. Micrositing of roof mounting wind turbine in urban environment: Cfd simulations and lidar measurements. Renewable Energy 115, 8 – 9.

Wiren, B., 1975. A wind tunnel study of wind velocities in passages between and through buildings. In: Proceedings of the 4th International Conference on Wind Effects on Buildings and Structures,(Heathrow 1975). pp. 465–475.

Wright, N. G., Easom, G. J., 2003. Non-linear k turbulence model results for flow over a building at full-scale. Applied Mathematical Modelling 27 (12), 1 – 22.

Yoshie, R., Mochida, A., Tominaga, Y., Kataoka, H., Harimoto, K., Nozu, T., Shirasawa, T., 2007a. Cooperative project for cfd prediction of pedestrian wind environment in the architectural institute of japan. Journal of Wind Engineering and Industrial Aerodynamics 95 (9), 4–5.

Yoshie, R., Mochida, A., Tominaga, Y., Kataoka, H., Harimoto, K., Nozu, T., Shirasawa, T., 2007b. Cooperative project for cfd prediction of pedestrian wind environment in the architectural institute of japan. Journal of Wind Engineering and Industrial Aerodynamics 95 (9), 2.

Yoshie, R., Mochida, A., Tominaga, Y., Kataoka, H., Harimoto, K., Nozu, T., Shirasawa, T., 2007c. Cooperative project for cfd prediction of pedestrian wind environment in the architectural institute of japan. Journal of Wind Engineering and Industrial Aerodynamics 95 (9), 1–28.

Appendix

5.1 Inlet Conditions

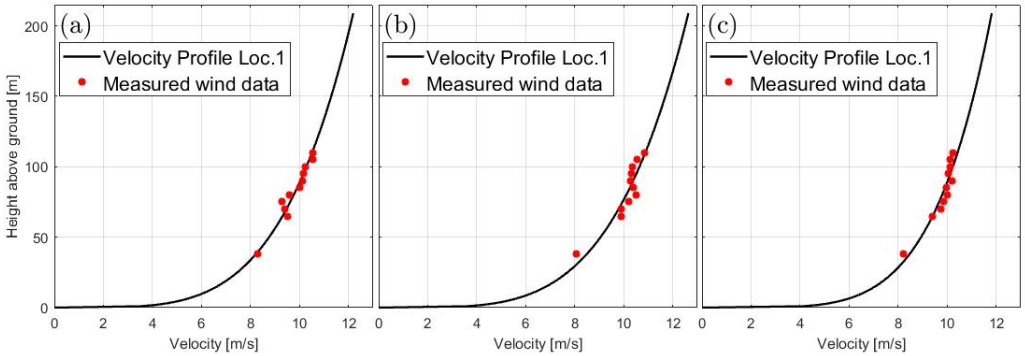


Figure 5.1: Estimated velocity profiles based on measured wind data

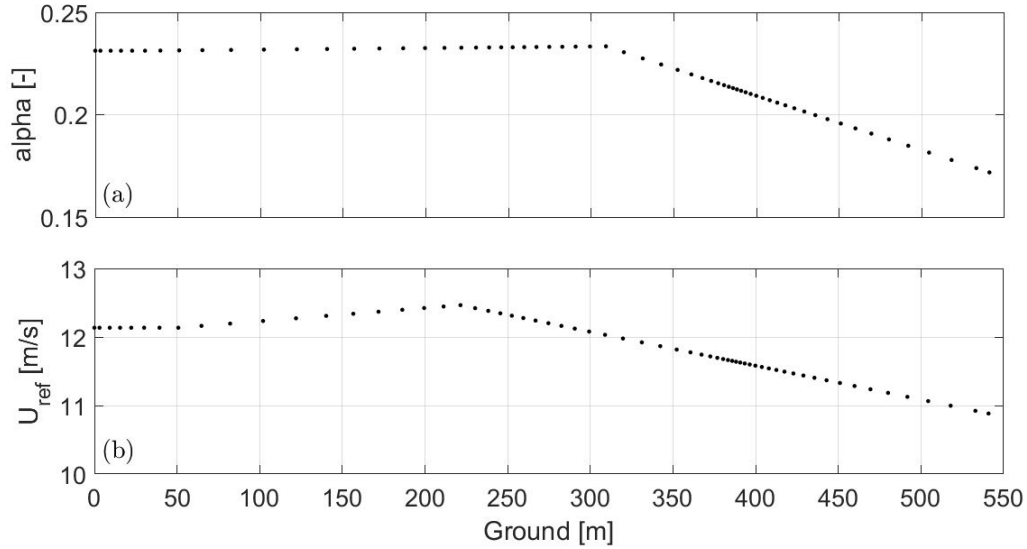


Figure 5.2: Variation of α (a) and U_{ref} (b) at the inlet

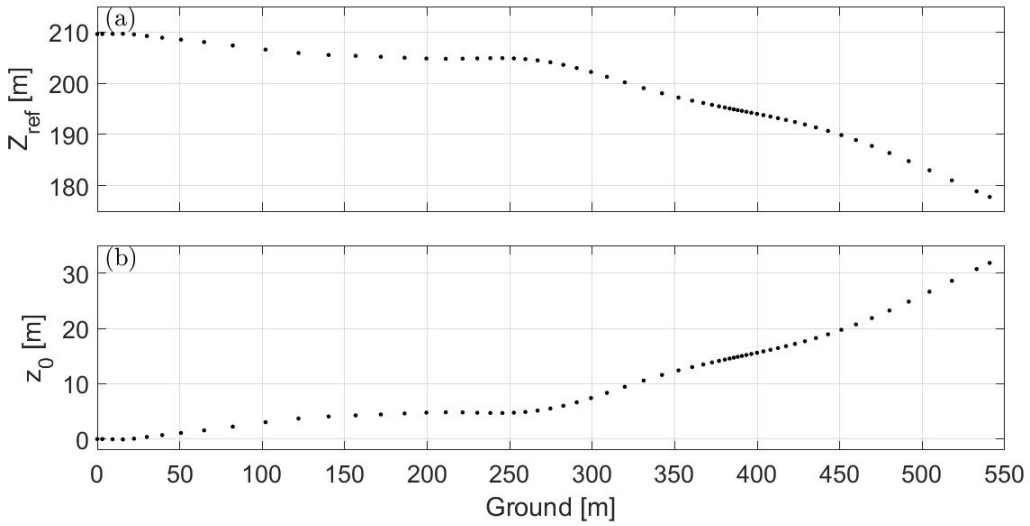


Figure 5.3: Variation of Z_{ref} (a) and z_0 (b) at the inlet

5.2 ZephIR LIDAR Measurements

5.2.1 Wind Velocity

Heights	110	105	100	95	90	85	80	75	70	65	38	MEAN
Location.1												
1st Measurement	10,3	10,34	10,53	10,5	10,29	9,9	9,85	10,12	10,03	9,78	8,1	9,98
2nd Measurement	10,18	9,92	9,7	9,59	10,12	10,06	10,15	9,56	9,59	9,02	8,3	9,64
Mean	10,24	10,13	10,12	10,05	10,21	9,98	10,00	9,84	9,76	9,40	8,20	9,81
Location.2												
1st Measurement	10,17	10,53	10,32	10,65	10,54	10,46	10,17	9,91	9,56	9,33	8,73	10,03
2nd Measurement	9,26	8,81	9,28	9,41	9,45	9,79	10,56	10,54	9,68	9,78	8,57	9,56
Mean	9,71	9,67	9,80	10,03	10,00	10,12	10,36	10,23	9,62	9,55	8,65	9,79
Location.3												
1st Measurement	10,26	10,05	9,87	9,69	9,79	9,65	9,42	9,02	9,45	9,53	7,94	9,52
2nd Measurement	29495	11,07	10,57	10,62	10,43	10,38	9,76	9,51	9,36	9,49	8,65	10,06
Mean	10,53	10,56	10,22	10,16	10,11	10,01	9,59	9,27	9,41	9,51	8,29	9,79
Location.4												
1st Measurement	11,18	11,01	10,98	11,16	11,28	11,52	11,6	11,83	11,51	10,24	8,58	10,81
2nd Measurement	10,51	10,1	9,7	9,48	9,28	9,3	9,42	9,56	9,31	9,57	7,55	9,44
Mean	10,84	10,56	10,34	10,32	10,28	10,40	10,51	10,20	9,91	9,91	8,07	10,12

Figure 5.4: Measured wind velocity at location no. 1 to 4

5.3 WindCube Measurements

5.3.1 Wind Velocity

Heights	40	45	50	55	60	65	70	75	80	85	90	100
Velocity	6,32	6,66	6,95	7,25	7,55	7,76	7,94	8,07	8,21	8,35	8,5	8,75
	6,66	7,2	7,64	8,04	8,43	8,71	8,97	9,17	9,34	9,5	9,67	9,97
	7,5	7,92	8,21	8,48	8,82	9,17	9,52	9,75	9,94	10,08	10,25	10,54
	6,72	7,11	7,39	7,65	7,99	8,35	8,72	8,98	9,21	9,4	9,62	10,04
	6,79	7,25	7,59	7,87	8,16	8,44	8,74	8,96	9,13	9,27	9,44	9,77
	7,96	8,54	8,95	9,28	9,56	9,81	10,06	10,26	10,46	10,64	10,82	11,09
	7,18	7,7	8,08	8,4	8,73	9,05	9,43	9,73	9,99	10,18	10,39	10,71
	7,26	7,79	8,18	8,49	8,78	9,03	9,3	9,5	9,66	9,77	9,86	10,05
	8,49	9,11	9,57	9,91	10,16	10,36	10,61	10,81	10,97	11,07	11,18	11,41
	7,35	7,96	8,37	8,68	8,96	9,21	9,48	9,68	9,82	9,93	10,05	10,27
	4,65	5,1	5,43	5,66	5,83	5,97	6,11	6,19	6,3	6,42	6,59	6,87
	7,41	7,83	8,16	8,43	8,71	8,97	9,24	9,45	9,66	9,83	9,99	10,28
	9,15	9,7	10,08	10,36	10,58	10,76	10,94	11,05	11,1	11,12	11,14	11,28
	6,59	7,1	7,43	7,69	7,91	8,12	8,42	8,67	8,88	9,05	9,24	9,68
Mean	7,15	7,64	8	8,3	8,58	8,84	9,11	9,31	9,48	9,62	9,77	10,5

Figure 5.5: Measured wind velocity above Varmeteknisk building

5.4 Near-Wall Treatment

The theory about near-wall treatment is based on previous research of the wind condition at Gløshaugen (Burkeland, 2017)

Turbulent flows are significantly affected by the presence of the walls by the *no-slip* condition, and large gradients near the wall due to high Reynolds numbers Ansys-Inc (2017c). Regions near the wall are divided into different layers as illustrated in Figure 4.10. The shapes are called *The law of the wall* and describe how the mean velocity can be approximated either linearly or logarithmically, depending on the distance to the wall. The non-denationalized variables that defines the x- and y-axis are defined as follows

$$y^+ \equiv \frac{\rho u_\tau y}{\mu} \tag{5.1}$$

and

$$u^+ = \frac{U}{u_\tau} \tag{5.2}$$

where u_τ is the friction velocity defined as

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{5.3}$$

Following the Figure 4.10, the inner layer is separated in different layers and the shape of the mean velocity changes from one layer to another. In the innermost layer, the *viscous layer*, the flow is almost laminar and viscous forces are dominant Ansys-Inc (2017c). Therefore,

the velocity profile is fully dependent on the viscous scale, thus the viscous sublayer can be expressed as

$$u^+ = y^+ \tag{5.4}$$

Moving further away from the wall towards the outer layer, turbulence plays a major role Ansys-Inc (2017c). As seen in Figure 4.10, when $y^+ \geq 60$, the flow is dominated by turbulence and the pattern of mean velocity is called *log-law of the wall*, and is expressed as

$$u^+ = \frac{1}{\kappa} \ln y^+ + C \tag{5.5}$$

The middle region, the region between $5 < y^+ < 60$, is called the buffer layer or the blending region. This region is both affected by viscous and turbulent forces, which makes it hard to calculate the mean velocity profile. Additionally, the viscous sublayer needs to be thoroughly solved in order to solve the buffer layer. The viscous sublayer can either be entirely solved or solved by embedded wall function in ANSYS Fluent. On the one hand, when solving for the viscous sublayer without wall functions, it is according to ANSYS Fluent necessary to use a finer mesh close to the wall, where the first grid cell should give a value of $y^+ \approx 1$ ANSYS-Inc (2017a). On the other hand, when wall functions are applied, the first grid point should have a value of y^+ in the logarithmic layer within the interval [60, 300]. Furthermore, one assumes that the behavior closer to the wall follows the characteristics of the *the law of the wall* ANSYS-Inc (2017a).

5.5 CFD-Post

5.5.1 Velocity

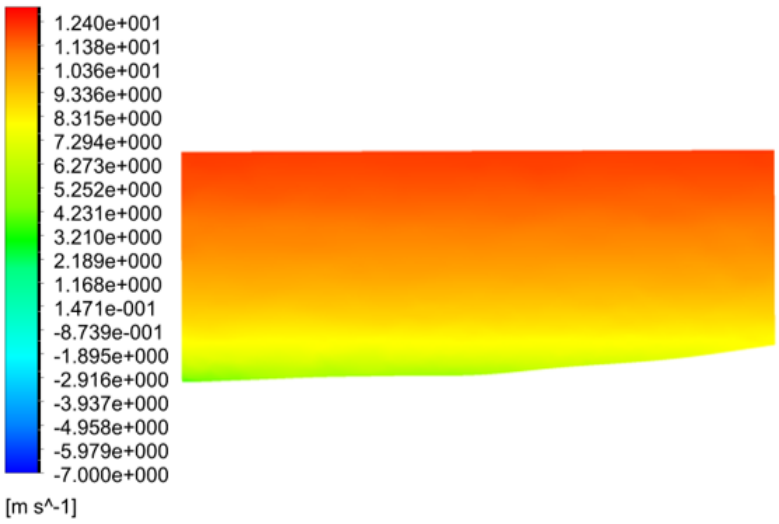


Figure 5.6: Inlet velocity profile for the CFD simulation.

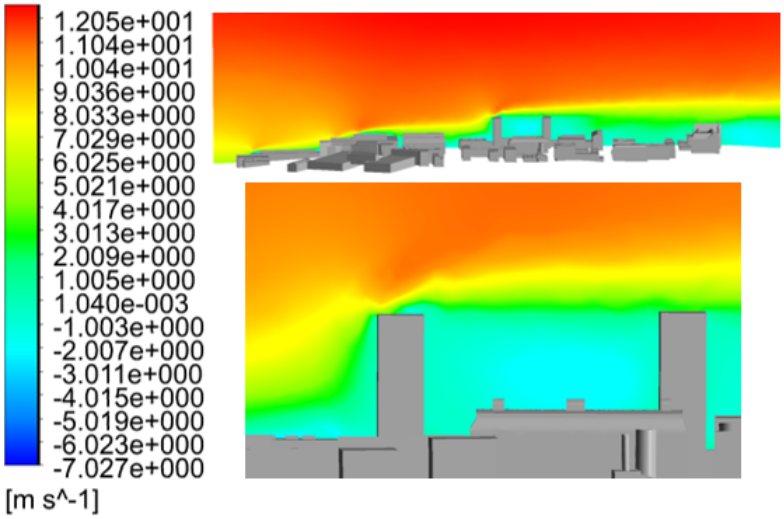


Figure 5.7: $\overline{u}_y$ at a center line along the campus by $k - \epsilon$.

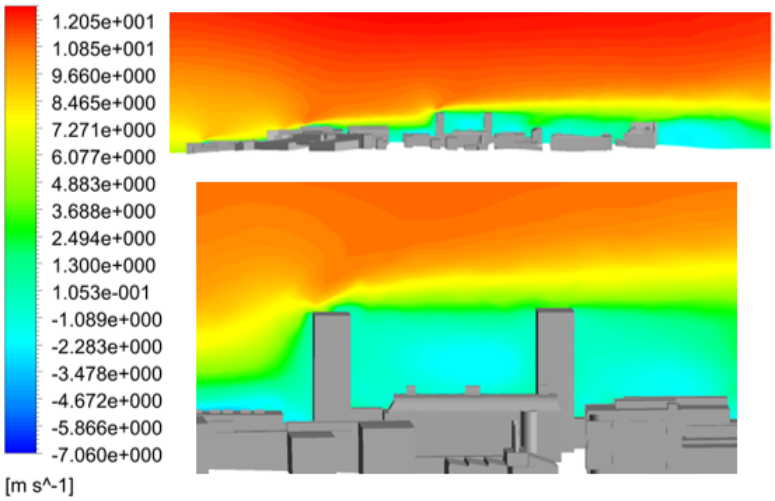


Figure 5.8: $\overline{u}_y$ at a center line along the campus by $k - \omega$.

5.5.2 Turbulence Intensity

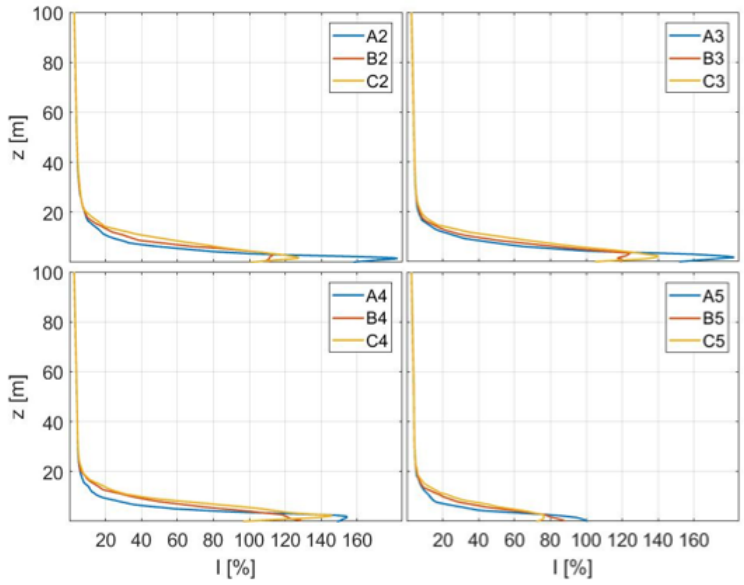


Figure 5.9: Turbulence intensity simulated by $k - \epsilon$ at column no. 2, 3, 4 and 5 at the roof.

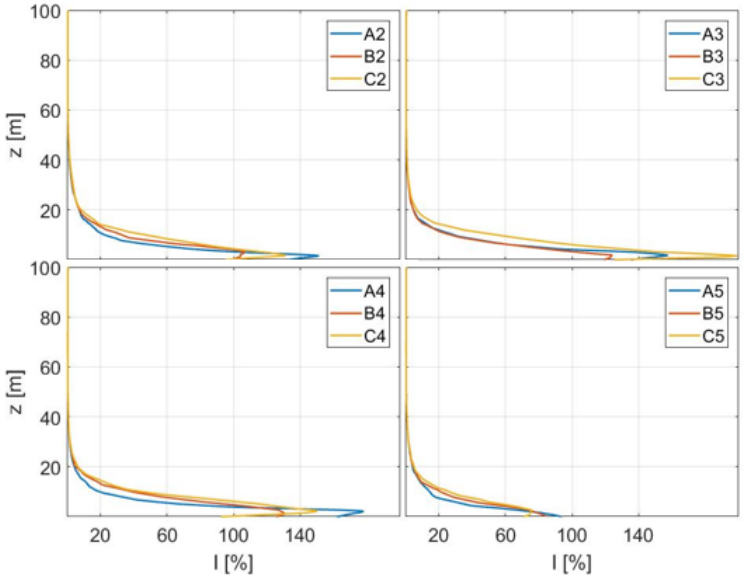


Figure 5.10: Turbulence intensity simulated by $k - \omega$ at column no. 2, 3, 4 and 5 at the roof.

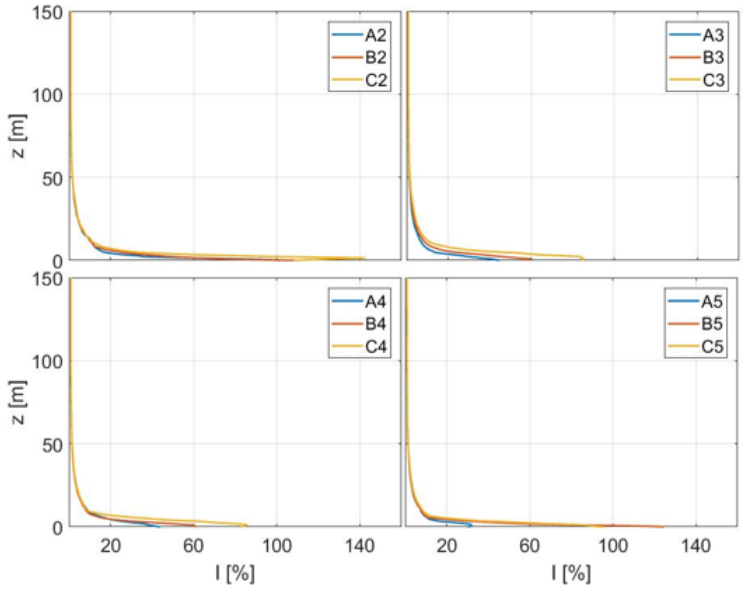


Figure 5.11: Turbulence intensity simulated by LES at column no. 2, 3, 4 and 5 at the roof.

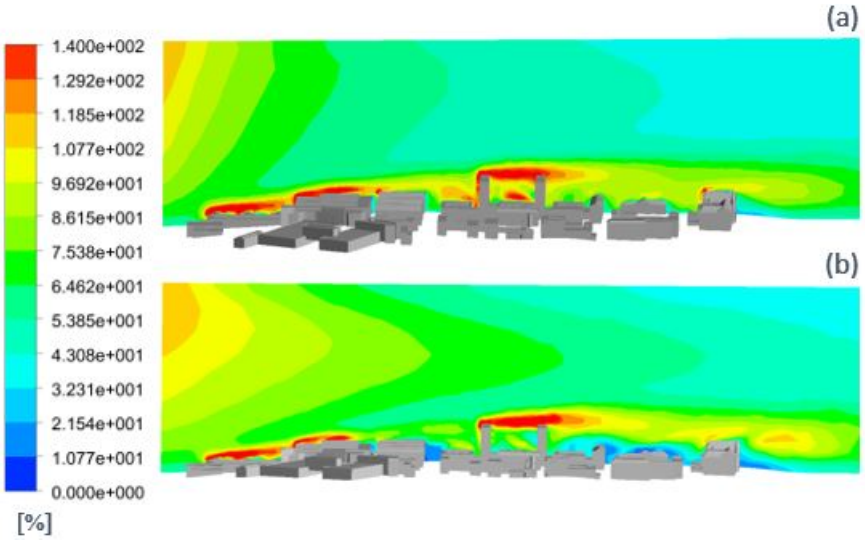


Figure 5.12: Turbulence intensity at a center-line through campus by $k - \epsilon$ (a) and $k - \omega$ (b).

5.6 Power Density

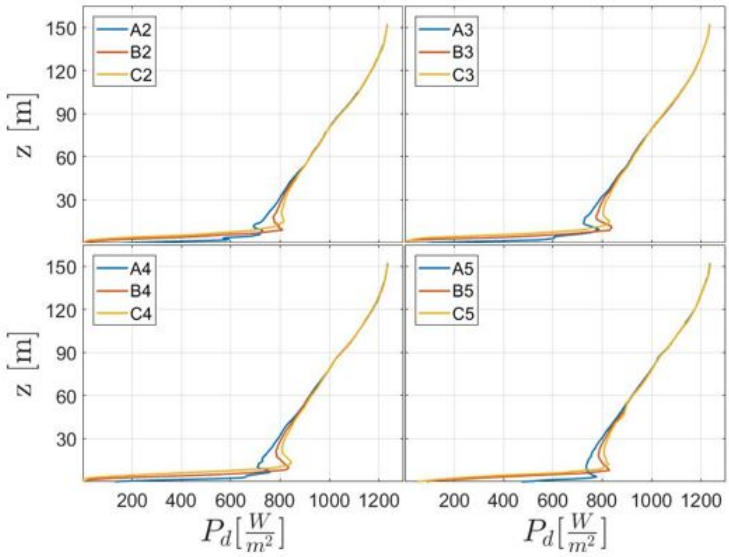


Figure 5.13: Power density calculated by $k-\epsilon$ at column no. 2, 3, 4 and 5 at the roof of $Sentralbygget$.

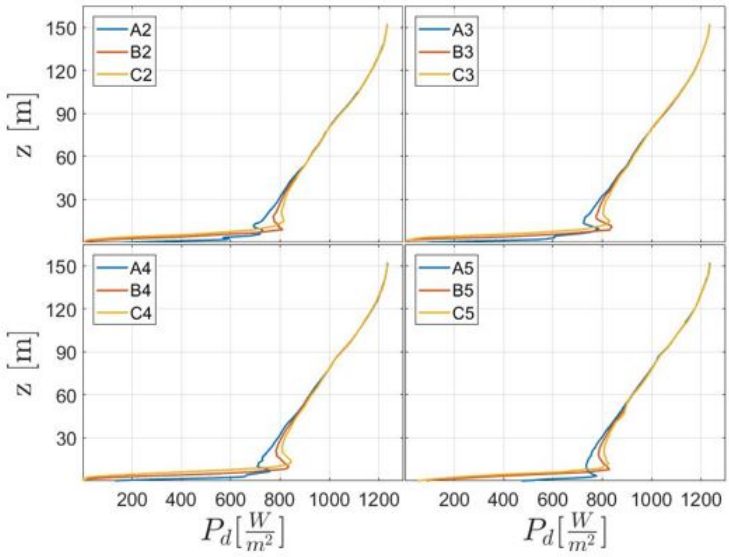


Figure 5.14: Power density calculated by $k-\omega$ at column no. 2, 3, 4 and 5 at the roof of $Sentralbygget$.

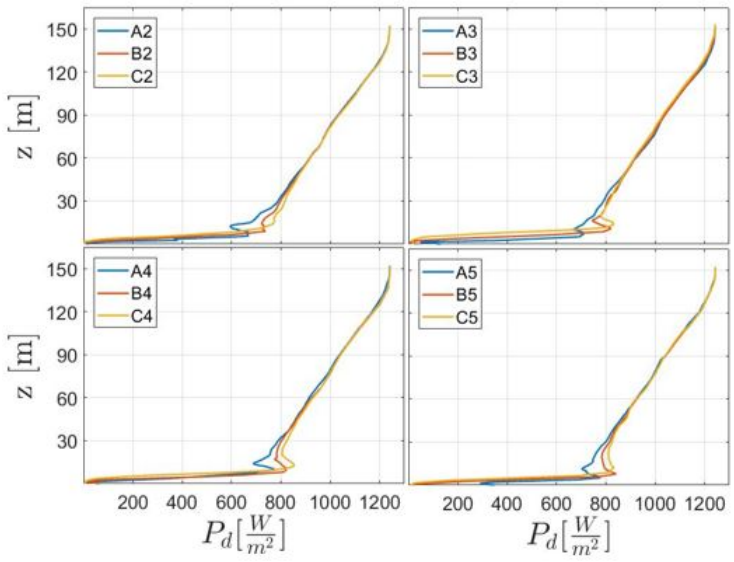


Figure 5.15: Power density calculated by LES at column no. 2, 3, 4 and 5 at the roof of $Sentralbygget$.

5.7 MATLAB Scripts

Table 5.1: Code Overview

Function	Usage
$InletVelocityProfile$	Main script for calculating the inlet condition of the CFD simulation.
$PowerLawExponent$	Function for calculating α based on measured data and known $U(z_{ref})$.
$NumericalValidationinlet$	Script for calculating the hit rate value for location no. 1, 2 and 3.
$NumericalValidationloc.4$	Script for calculating the hit rate value for location no. 4.
$NumericalValidationloc5$	Script for calculating the hit rate value for location no. 5.

5.7.1 Inlet Velocity Profile

```
clear all
close all
clc

%% Load velocity values from LIDAR measurements

load 'ZepIR_lidar' % Load data, mean velocities
load 'LIDAR1'      % Load data, standard deviation
U_mat = [meanLoc1; meanLoc3; meanLoc4]; % Collect data

%% Read terrain data

ground = xlsread('ground_matlab.xlsx', 'A1:B44150');
inlet = xlsread('inlet_matlab.xlsx', 'A1:B974');
[C, ia, ib] = intersect(inlet, ground, 'rows'); % Intersect rows for inlet and ground
ground_y = C(:,1); ground_z=C(:,2) - C(1,2);    % First coordinate is 0 for simplicity

loc_y = [7 19 34]; % Position along y-axis for the locations with measured data
Z = (0:209.61515); % Meters above ground

%% Preallocation of vectors and matrices

```

```
Zref = zeros(1,length(ground_y)); zref_tab = zeros(1,length(U_mat(:,1)));
Uref_tab = zeros(1,length(U_mat(:,1)));
U_terrain = zeros(length(U_mat(:,1)), length(Z));
Zero_vec = zeros(1,length(ground_y));
length2 = zeros(1,length(ground_y));
alpha_tab  = zeros(1);

%% Calculate U_ref and alpha for the three locations with measured data

for i = 1:length(ground_z)
    Zref(i) = 209.61515-ground_z(i);  % Reference height
end % end for, i

Ust = [meanLoc1(1,end);meanLoc3(1,end);meanLoc4(1,end)]; % Initial guess (velocity)
alpha_st = 1/7;                                          % Initial guess (alpha)

for i = 1:length(Ust)
    U = U_mat(i,:);      % Measured wind data for location no. i
    Zr = Zref(loc_y(i)); % Reference height for location no. i

    varpar = fminsearch(@(varpar) err_powerlaw2(varpar,Zr,...
        Zmeasured,U),[alpha_st Ust(i)],[]); % Find alpha and U for loc. i
    alpha_tab(i) = varpar(1);
    zref_tab(i)  = Zr;
    Uref_tab(i)  = varpar(2);

    %Print Uref and alpha for all locations
    if i == 1
        fprintf('Uref = %0.3f and %s = %0.4f, at loc. %d \n',...
            Uref_tab(i) , char(945), alpha_tab(i),i);
    else
        fprintf('Uref = %0.3f and %s = %0.4f, at loc. %d\n',...
            Uref_tab(i) , char(945), alpha_tab(i), i+1);
    end % end if
end % end for, i

%% Calculate velocity profiles for each location

% z0 = 0 for all y. No difference in terrain
for i = 1:length(alpha_tab)                                    % Iteration for alpha
    for j = 1:length(Z)                                        % Meters above ground
        U(i,j) = Uref_tab(i)*(Z(j)/Zref(loc_y(i)))^alpha_tab(i);  % Vertical wind speed
    end % end for, j
end % end for, i

% Taking the terrain into account
for i = 1:length(alpha_tab)
    count = 0;
    for j = 1:length(Z)
        if Z(j) < ground_z(loc_y(i))
            U_terrain(i,j) = NaN; % Don't plot velocity below ground
            %count = count + 1;    % Count meters above origo and subtract (Z(j) - count)
        else
            U_terrain(i,j) = Uref_tab(i)*((Z(j)-ground_z(loc_y(i)))/Zref(loc_y(i)))^alpha_tab(
                i); % Vertical wind speed
        end % end if
    end % end for, j
end % end for, i

%% Linear interpolation and extrapolation for alpha
a = alpha_tab;
L3 = loc_y(1); L4 = loc_y(2); L1 = loc_y(3); % Boundaries for interpolation and extrapolation

% Preallocating
a_0_3 = zeros(1,L3-1); Uref_0_3 = zeros(1,L3-1);
a_1_0 = zeros(1,length(ground_y)-L1); Uref_1_0 = zeros(1,length(ground_y)-L1);
a_3_4 = zeros(1,L4-L3); Uref_3_4 = zeros(1,L4-L3);
a_4_1 = zeros(1,L1-L4); Uref_4_1 = zeros(1,L1-L4);


%Interpolate between location no.3-4 and location no4-1
```

```
for i = L3+1:length(ground_y)
    if i >= L3+1 && i <= L4-1    % Between loc. 3-4
        a_3_4(i-7) = a(2)+(a(3)-a(2))*((ground_y(i)-ground_y(L3+1))/(ground_y(L4)-ground_y(L3
            +1)));
    elseif i == L4               % Loc. 4
        a_3_4(i-L3) = a(3);
    elseif i >= L4+1 && i <=L1-1 % Between loc. 4-1
        a_4_1(i-L4) = a(3)+(a(1)-a(3))*((ground_y(i)-ground_y(L4))/(ground_y(L1)-ground_y(L4))
            );
    elseif i == L1               % Loc. 1
        a_4_1(i-L4) = a(1);
    else
    end % end, if
end % end, for

%Extrapolate between location no.1 and wall (right side)
for i = L1+1:length(ground_y)
    if i == L1+1
        j = 1;
        a_1_0(j) = a_4_1(end-1)+((ground_y(i)-ground_y(i-2))/(ground_y(i-1)-ground_y(i-2)))*(a
            (j)-a_4_1(end-1));
    elseif i == L1+2
        j = 2;
        a_1_0(j) = a(1)+((ground_y(i)-ground_y(i-2))/(ground_y(i-1)-ground_y(i-2)))*(a_1_0(j
            -1)-a(1));
    elseif i >= L1+3 && i <=length(ground_y)
        j = j+1;
        a_1_0(j) = a_1_0(j-2)+((ground_y(i)-ground_y(i-2))/(ground_y(i-1)-ground_y(i-2)))*(
            a_1_0(j-1)-a_1_0(j-2));
    end % end if
end % end for, i

%extrapolate between wall (left side) and location no.3
for i = 1:L3-1
    j = L3-i;
    if j == L3-1
        a_0_3(j) = a_3_4(i)+((ground_y(j)-ground_y(j+2))/(ground_y(j+1)-ground_y(j+2)))*(a(i
            +1)-a_3_4(i));
    elseif j == L3-2
        a_0_3(j) = a(i)+((ground_y(j)-ground_y(j+2))/(ground_y(j+1)-ground_y(j+2)))*(a_0_3(j
            +1)-a(i));
    else
        a_0_3(j) = a_0_3(j+2)+((ground_y(j)-ground_y(j+2))/(ground_y(j+1)-ground_y(j+2)))*(
            a_0_3(j+1)-a_0_3(j+2));
    end % end if
end % end for, i

% Final vector for alpha
alpha_f2(1:L3-1) = a_0_3;
alpha_f2(L3) = a(2);
alpha_f2(L3+1:L4) = a_3_4;
alpha_f2(L4+1:L1) = a_4_1;
alpha_f2(L1+1:length(ground_y)) = a_1_0;


%% Linear interpolation and extrapolation for Uref
Uref = Uref_tab; % Boundaries for interpolation and extrapolation

% Interpolate between location no.3-4 and location no4-1
for i = L3+1:length(ground_y)
    if i >= L3+1 && i <= L4-1    % Between loc. 3-4
        Uref_3_4(i-L3) = Uref(2)+(Uref(3)-Uref(2))*((ground_y(i)-ground_y(L3+1))/(ground_y(L4)
            -ground_y(L3+1)));
    elseif i == L4               % Loc. 4
        Uref_3_4(i-L3) = Uref(3);
    elseif i >= L4+1 && i <=L1-1 % Between loc. 4-1
        Uref_4_1(i-L4) = Uref(3)+(Uref(1)-Uref(3))*((ground_y(i)-ground_y(L4))/(ground_y(L1)-
            ground_y(L4)));
    elseif i == L1               % Loc. 1
        Uref_4_1(i-L4) = Uref(1);
    else
```

```
    end % end, if
end % end for, i

% Extrapolate between location no.1 and wall (right side)
for i = L1+1:length(ground_y)
    if i == L1+1
        j = 1;
        Uref_1_0(j) = Uref_4_1(end-1)+((ground_y(i)-ground_y(i-2))/(ground_y(i-1)-ground_y(i
            -2)))*(Uref(j)-Uref_4_1(end-1));
    elseif i == L1+2
        j = 2;
        Uref_1_0(j) = Uref(1)+((ground_y(i)-ground_y(i-2))/(ground_y(i-1)-ground_y(i-2)))*(
            Uref_1_0(j-1)-Uref(1));
    elseif i >= L1+3 && i <=length(ground_y)
        j = j+1;
        Uref_1_0(j) = Uref_1_0(j-2)+((ground_y(i)-ground_y(i-2))/(ground_y(i-1)-ground_y(i-2))
            )*(Uref_1_0(j-1)-Uref_1_0(j-2));
    end % end, if
end % end for, i

% Extrapolate between wall (left side) and location no.3
for i = 1:L3-1
    j = L3-i;
    if j == L3-1
        Uref_0_3(j) = Uref_3_4(i)+((ground_y(j)-ground_y(j+2))/(ground_y(j+1)-ground_y(j+2)))
            *(Uref(i+1)-Uref_3_4(i));
    elseif j == L3-2
        Uref_0_3(j) = Uref(i)+((ground_y(j)-ground_y(j+2))/(ground_y(j+1)-ground_y(j+2)))*(
            Uref_0_3(j+1)-Uref(i));
    else
        Uref_0_3(j) = Uref_0_3(j+2)+((ground_y(j)-ground_y(j+2))/(ground_y(j+1)-ground_y(j+2))
            )*(Uref_0_3(j+1)-Uref_0_3(j+2));
    end % end, if
end % end for, i

% Final vector for Uref
Uref_f(1:L3-1) = Uref_0_3;
Uref_f(L3) = Uref(2);
Uref_f(L3+1:L4) = Uref_3_4;
Uref_f(L4+1:L1) = Uref_4_1;
Uref_f(L1+1:length(ground_y)) = Uref_1_0;

%% Calculate velocity profile along y-axis
%U_ref = 12.1272;
U_all = zeros(length(ground_y), length(Z));
for i = 1:length(ground_y)
   if ground_z(i) == 0      % Height of terrain equals 0
       for j = 1:length(Z)
           U_all(i,j) = Uref_f(i)*((Z(j)/Zref(i))^(alpha_f2(i)));
       end % end for, j
   elseif ground_z(i) < 0   % Height of terrain is less than z0_ref (=0)
       for j = 1:length(Z)+ground_z(i)
           U_all(i,j) = Uref_f(i)*((Z(j))/Zref(i))^(alpha_f2(i));
       end % end for, j
   elseif ground_z(i) > 0   % Height of terrain is higher than z0_ref (=0)
       for j = 1:length(Z)-ground_z(i)
           U_all(i,j) = Uref_f(i)*((Z(j))/Zref(i))^(alpha_f2(i));
       end % end for, j
   end % end, if
end % end for, i

% Zeros infront of matrix
U2_all = zeros(length(ground_y), length(Z));
for i = 1:length(ground_y)
    for j = 2:length(Z)
        if U_all(i,j) == 0
            Zero_vec(i) = j;
            break
        end % end, if
    end % end for, j
end % end for, i
```

```
219
220 for k = 1:length(ground_y)
221     for l = 1:length(Z)
222         length2(k) = length(Z)-Zero_vec(k);%+1;
223     end % end for, l
224 end % end for, k
225
226 for m = 1:length(ground_y)
227     for n = 1:length(Z)-length2(m)
228         U2_all(m,n+length2(m)) = U_all(m,n);
229     end % end for, n
230 end % end for, m
231
232 %% Plot
233
234 figure();
235 height = 0:209;
236 ax6 = axes('Position',[0.1 0.1 0.8 0.8]);
237 contourf(ground_y, height, U2_all',50,'EdgeColor','none');
238 colormap(ax6, flipud(morgenstemning(50-1)))
239 caxis([0 13])
240 hold on
241 plot(ground_y(loc_y(1)), height(3), 'g*');
242 plot(ground_y(loc_y(2)), height(7), 'g*');
243 plot(ground_y(loc_y(3)), height(15), 'g*');
244 set(gca,'fontsize',14)
245 title('Inlet Velocity Profile');
246 xlabel('Ground [m]'); ylabel('Height [m]');
247 colorbar;
248 title(colorbar, 'Velocity [m/s]');
249 str = 'White: below ground';
250 text(1.5, 6.5, str);
```

5.7.2 Estimate Power Law Exponent

```
1 function err = err_powerlaw(varpar,zref,z,Uref,U)
2
3 alpha = varpar(1);
4
5 alpha2 = log10(U./Uref)/log10(z./zref);
6
7 err = norm(alpha2-alpha);
8 end
```

5.7.3 Hit Rate Value at the Inlet

```
1 clear all
2 close all
3 clc
4
5 %% Load Measured data
6 load 'ZepIR_lidar';
7 U_mat = [meanLoc1; meanLoc3; meanLoc4]; % Collect data
8
9 load 'StandardDeviationValues'; % Standard deviation loc no. 3
10
11 %% Load data from Ansys
12 load 'locVelocity';
13 Loc1(:,2) = Loc1(:,2)-Loc1(1,2); % z(1) = 0;
14 Loc2(:,2) = Loc2(:,2)-Loc2(1,2); % z(1) = 0;
15 Loc3(:,2) = Loc3(:,2)-Loc3(1,2); % z(1) = 0;
16
17 ke_loc1 = csvread('ke_loc1_NV.csv',7,0);
18 ke_loc2 = csvread('ke_loc2_NV.csv',7,0);
19 ke_loc3 = csvread('ke_loc3_NV.csv',7,0);
```

```
kw_loc1 = csvread('kw_loc1_NV.csv',7,0);
kw_loc2 = csvread('kw_loc2_NV.csv',7,0);
kw_loc3 = csvread('kw_loc3_NV.csv',7,0);
LES_loc1 = csvread('LES_loc1_NV.csv',7,0);
LES_loc2 = csvread('LES_loc2_NV.csv',7,0);
LES_loc3 = csvread('LES_loc3_NV.csv',7,0);

%% Initialize height vector

ke_loc1(:,2) = ke_loc1(:,2)-ke_loc1(1,2); %ke_a1(:,1) = ke_a1(:,1)*100;
ke_loc2(:,2) = ke_loc2(:,2)-ke_loc2(1,2); %ke_a2(:,1) = ke_a2(:,1)*100;
ke_loc3(:,2) = ke_loc3(:,2)-ke_loc3(1,2); %ke_a3(:,1) = ke_a3(:,1)*100;

kw_loc1(:,2) = kw_loc1(:,2)-kw_loc1(1,2); %ke_a4(:,1) = ke_a4(:,1)*100;
kw_loc2(:,2) = kw_loc2(:,2)-kw_loc2(1,2); %ke_a5(:,1) = ke_a5(:,1)*100;
kw_loc3(:,2) = kw_loc3(:,2)-kw_loc3(1,2); %ke_b1(:,1) = ke_b1(:,1)*100;

LES_loc1(:,2) = LES_loc1(:,2)-LES_loc1(1,2); %ke_b2(:,1) = ke_b2(:,1)*100;
LES_loc2(:,2) = LES_loc2(:,2)-LES_loc2(1,2); %ke_b3(:,1) = ke_b3(:,1)*100;
LES_loc3(:,2) = LES_loc3(:,2)-LES_loc3(1,2); %ke_b4(:,1) = ke_b4(:,1)*100;


%% Load ground and inlet data
ground = xlsread('ground_matlab.xlsx', 'A1:B44150');
inlet = xlsread('inlet_matlab.xlsx', 'A1:B974');
[C, ia, ib] = intersect(inlet, ground, 'rows'); % Intersect rows for inlet and ground
ground_y = C(:,1); ground_z=C(:,2) - C(1,2);    % First coordinate is 0 for simplicity

loc_y = [7 19 34]; % Position along y-axis for the locations with measured data
Z = (0:209.61515); % Meters above ground

%% Calculate U_ref and alpha for the three locations with measured data

for i = 1:length(ground_z)
    Zref(i) = 209.61515-ground_z(i);  % Reference height
end % end for, i

Ust = [meanLoc1(1,end);meanLoc3(1,end);meanLoc4(1,end)]; % Initial guess (velocity)
alpha_st = 1/7;                                          % Initial guess (alpha)

for i = 1:length(Ust)
    U = U_mat(i,:);      % Measured wind data for location no. i
    Zr = Zref(loc_y(i)); % Reference height for location no. i

    varpar = fminsearch(@(varpar) err_powerlaw2(varpar,Zr,...
        Zmeasured,U),[alpha_st Ust(i)],[]); % Find alpha and U for loc. i
    alpha_tab(i) = varpar(1);
    zref_tab(i)  = Zr;
    Uref_tab(i)  = varpar(2);

    %Print Uref and alpha for all locations
    if i == 1
        fprintf('Uref = %0.3f and %s = %0.4f, at loc. %d \n',...
            Uref_tab(i) , char(945), alpha_tab(i),i);
    else
        fprintf('Uref = %0.3f and %s = %0.4f, at loc. %d\n',...
            Uref_tab(i) , char(945), alpha_tab(i), i+1);
    end % end if
end % end for, i

Z1 = (0:212);
for i = 1:length(Z1)                                        % Meters above ground
     U1(i) = Uref_tab(2)*(Z1(i)/Zref(2))^alpha_tab(2);  % Vertical wind speed
end % end for, j

Z2 = (0:208);
for i = 1:length(Z2)                                        % Meters above ground
     U2(i) = Uref_tab(3)*(Z2(i)/Zref(3))^alpha_tab(3);  % Vertical wind speed
end % end for, i

Z3 = (0:200);
```

```
for i = 1:length(Z3)                                    % Meters above ground
     U3(i) = Uref_tab(1)*(Z3(i)/Zref(1))^alpha_tab(1);  % Vertical wind speed
end % end for, i

%% Calculate hit rate (q)

%Normalize simulated data, Loc1
Loc1_norm(:,1) = Loc1(:,1)/Loc1(end,1); % Normalized simulated velocity at loc.1
Loc1_norm_Lidarheights = [Loc1_norm(276,1), Loc1_norm(263,1), Loc1_norm(251,1), Loc1_norm
    (239,1), Loc1_norm(226,1), Loc1_norm(214,1), Loc1_norm(201,1), Loc1_norm(189,1), Loc1_norm
    (176,1), Loc1_norm(164,1), Loc1_norm(96,1)];
%Normalize simulated data, Loc2
Loc2_norm(:,1) = Loc2(:,1)/Loc2(end,1); % Normalized simulated velocity at loc.2
Loc2_norm_Lidarheights = [Loc2_norm(276,1), Loc2_norm(263,1), Loc2_norm(251,1), Loc2_norm
    (239,1), Loc2_norm(226,1), Loc2_norm(214,1), Loc2_norm(201,1), Loc2_norm(189,1), Loc2_norm
    (176,1), Loc2_norm(164,1), Loc2_norm(96,1)];
%Normalize simulated data, Loc3
Loc3_norm(:,1) = Loc3(:,1)/Loc3(end,1); % Normalized simulated velocity at loc.3
Loc3_norm_Lidarheights = [Loc3_norm(276,1), Loc3_norm(263,1), Loc3_norm(251,1), Loc3_norm
    (239,1), Loc3_norm(226,1), Loc3_norm(214,1), Loc3_norm(201,1), Loc3_norm(189,1), Loc3_norm
    (176,1), Loc3_norm(164,1), Loc3_norm(96,1)];

meanLoc3_norm = meanLoc3(:)/meanLoc3(1); % Normalized measured velocity at loc.1
meanLoc4_norm = meanLoc4(:)/meanLoc4(1); % Normalized measured velocity at loc.2
meanLoc1_norm = meanLoc1(:)/meanLoc1(1); % Normalized measured velocity at loc.3

U_std_r1 = U_std_loc1./meanLoc3(1,:);    % Relative standard deviation at loc.1
U_std_r2 = U_std_loc2./meanLoc4(1,:);    % Relative standard deviation at loc.2
U_std_r3 = U_std_loc3./meanLoc1(1,:);    % Relative standard deviation at loc.3

% Hit rate correlation for each location
for i = 1:length(U_std_r1)
    if abs((meanLoc3_norm(i) - Loc1_norm_Lidarheights(i))/Loc1_norm_Lidarheights(i)) <=
        U_std_r1(i)
        N_Loc1(i) = 1;
    else
        N_Loc1(i) = 0;
    end % end if
    if abs((meanLoc4_norm(i) - Loc2_norm_Lidarheights(i))/Loc2_norm_Lidarheights(i)) <=
        U_std_r2(i)
        N_Loc2(i) = 1;
    else
        N_Loc2(i) = 0;
    end % end if
    if abs((meanLoc1_norm(i) - Loc3_norm_Lidarheights(i))/Loc3_norm_Lidarheights(i)) <=
        U_std_r3(i)
        N_Loc3(i) = 1;
    else
        N_loc3(i) = 0;
    end % end if
end

% Hit rate
q_Loc1 = 1/length(N_Loc1)*sum(N_Loc1); % hit rate at loc.1
q_Loc2 = 1/length(N_Loc2)*sum(N_Loc2); % hit rate at loc.2
q_Loc3 = 1/length(N_Loc3)*sum(N_Loc3); % hit rate at loc.3
```

5.7.4 Hit Rate Value at Location. no. 4

```
clear all
close all
clc

%Load data from Measurements
%% Load data from LIDAR2 measurements
load ('LIDAR2_40.mat');load ('LIDAR2_45.mat');load ('LIDAR2_50.mat');
load ('LIDAR2_55.mat');load ('LIDAR2_60.mat');load ('LIDAR2_65.mat');
load ('LIDAR2_70.mat');load ('LIDAR2_75.mat');load ('LIDAR2_80.mat');
```

```
load ('LIDAR2_85.mat');load ('LIDAR2_90.mat');load ('LIDAR2_100.mat');
load ('LIDAR2_AVG.mat');

%% Load data from Ansys
load 'ZepIR_lidar';
load 'StandardDeviationValues'

% Lidar1ke(:,2) = Lidar1ke(:,2)-Lidar1ke(1,2);    % z(1) = 0;
% Lidar1kw(:,2) = Lidar1kw(:,2)-Lidar1kw(1,2);    % z(1) = 0;
% Lidar1LES(:,2) = Lidar1LES(:,2)-Lidar1LES(1,2); % z(1) = 0;

Lidar2_ke = csvread('ke_ref_NV.csv',7,0);
Lidar2_kw = csvread('kw_ref_NV.csv',7,0);
Lidar2_LES = csvread('LES_ref_NV.csv',7,0);

Lidar2_ke(:,2) = Lidar2_ke(:,2)-Lidar2_ke(1,2);    % z(1) = 0;
Lidar2_kw(:,2) = Lidar2_kw(:,2)-Lidar2_kw(1,2);    % z(1) = 0;
Lidar2_LES(:,2) = Lidar2_LES(:,2)-Lidar2_LES(1,2); % z(1) = 0;


%% Calculate U_ref and alpha for all locations

U_refpunkt = meanLoc2;% Collect data
Z = (0:192); % Meters above ground
Zref = 110;  % Reference height

Uref_tab = max(U_refpunkt) %U_refpunkt(1); % Uref at z=110m
alpha_st = 1/7;           % Initial guess
alpha_tab  = zeros(1)';   % Preallocating

 for i = 1:length(Uref_tab)
    varpar = fminsearch(@(varpar) err_powerlaw(varpar,Zref,...
        Zmeasured(1,:), Uref_tab(i,1),U_refpunkt(i,:)),[alpha_st],[]); % Find alpha for loc. i
    alpha_tab(i) = varpar(1);
end % end for, i


for i = 1:length(alpha_tab)                          % Iteration for alpha
for j = 1:length(Z)                                  % Meters above ground
U(i,j) = Uref_tab(i)*(Z(j)/Zref)^alpha_tab(i);       % Vertical wind speed
end % end for, j
end % end for, i

%% Calculate hit rate (q)

%Normalize simulated data, ke
Lidar2ke_norm(:,1) = Lidar2_ke(:,1)/Lidar2_ke(end,1); %Uref;
Lidar2ke_norm_Lidarheights = [Lidar2ke_norm(276,1), Lidar2ke_norm(263,1), Lidar2ke_norm(251,1)
    , Lidar2ke_norm(239,1), Lidar2ke_norm(226,1), Lidar2ke_norm(214,1), Lidar2ke_norm(201,1),
    Lidar2ke_norm(189,1), Lidar2ke_norm(176,1), Lidar2ke_norm(164,1), Lidar2ke_norm(96,1)];
%Normalize simulated data, kw
Lidar2kw_norm(:,1) = Lidar2_kw(:,1)/Lidar2_kw(end,1); %Uref;
Lidar2kw_norm_Lidarheights = [Lidar2kw_norm(276,1), Lidar2kw_norm(263,1), Lidar2kw_norm(251,1)
    , Lidar2kw_norm(239,1), Lidar2kw_norm(226,1), Lidar2kw_norm(214,1), Lidar2kw_norm(201,1),
    Lidar2kw_norm(189,1), Lidar2kw_norm(176,1), Lidar2kw_norm(164,1), Lidar2kw_norm(96,1)];
%Normalize simulated data, LES
Lidar2LES_norm(:,1) = Lidar2_LES(:,1)/Lidar2_LES(end,1); %Uref;
Lidar2LES_norm_Lidarheights = [Lidar2LES_norm(276,1), Lidar2LES_norm(263,1), Lidar2LES_norm
    (251,1), Lidar2LES_norm(239,1), Lidar2LES_norm(226,1), Lidar2LES_norm(214,1),
    Lidar2LES_norm(201,1), Lidar2LES_norm(189,1), Lidar2LES_norm(176,1), Lidar2LES_norm(164,1)
    , Lidar2LES_norm(96,1)];

%Measured normalized data
meanLoc2_norm = meanLoc2(:)/meanLoc2(1); % Normalized measured velocity at refLoc

U_std_r = U_std_refloc./meanLoc2(1,:);    % Relative standard deviation at loc.1

% Hit rate correlation
for i = 1:length(U_std_r)
    if abs((meanLoc2_norm(i) - Lidar2ke_norm_Lidarheights(i))/Lidar2ke_norm_Lidarheights(i))
        <= U_std_r(i)
```

```
        N_ke(i) = 1;
    else
        N_ke(i) = 0;
    end %end, if
    if abs((meanLoc2_norm(i) - Lidar2kw_norm_Lidarheights(i))/Lidar2kw_norm_Lidarheights(i))
        <= U_std_r(i)
        N_kw(i) = 1;
    else
        N_kw(i) = 0;
    end %end, if
    if abs((meanLoc2_norm(i) - Lidar2LES_norm_Lidarheights(i))/Lidar2LES_norm_Lidarheights(i))
         <= U_std_r(i)
        N_LES(i) = 1;
    else
        N_LES(i) = 0;
    end % end, if
end % for, i

% Hit rate
q_ke = 1/length(N_ke)*sum(N_ke);
q_kw = 1/length(N_ke)*sum(N_kw);
q_LES = 1/length(N_LES)*sum(N_LES);
```

5.7.5 Hit Rate Value at Location. no. 5

```
clear all
close all
clc

%Load data from Measurements
%% Load data from LIDAR2 measurements
load ('LIDAR2_40.mat');load ('LIDAR2_45.mat');load ('LIDAR2_50.mat');
load ('LIDAR2_55.mat');load ('LIDAR2_60.mat');load ('LIDAR2_65.mat');
load ('LIDAR2_70.mat');load ('LIDAR2_75.mat');load ('LIDAR2_80.mat');
load ('LIDAR2_85.mat');load ('LIDAR2_90.mat');load ('LIDAR2_100.mat');
load ('LIDAR2_AVG.mat');

%% Load data from Ansys

Lidar1ke = csvread('ke_Varme_NV.csv',7,0);
Lidar1kw = csvread('kw_Varme_NV.csv',7,0);
Lidar1LES = csvread('Lidar1LES.csv',7,0);

Lidar1ke(:,2) = Lidar1ke(:,2)-Lidar1ke(1,2);    % z(1) = 0;
Lidar1kw(:,2) = Lidar1kw(:,2)-Lidar1kw(1,2);    % z(1) = 0;
Lidar1LES(:,2) = Lidar1LES(:,2)-Lidar1LES(1,2); % z(1) = 0;


%% Calculate Velocity profile for Lidar2 based on measured data

Z = (0:180); % Meters above ground
Zref = 100;  % Reference height
Zmeasured = LIDAR2_AVG (1,1:12);%+26; % Measured height levels

Uref = LIDAR2_AVG(2,12); % Reference Velocity
U_measured = LIDAR2_AVG(2,1:12);   % Mean velocities over all height levels
alpha_st = 1/7;            % Initial guess
alpha = 0;                 % Preallocation

varpar = fminsearch(@(varpar) err_powerlaw(varpar,Zref,...
        Zmeasured(1,:), Uref(1,1),U_measured(1,:)),[alpha_st],[]); % Find alpha
alpha(1) = varpar(1);

for i = 1:length(Z)
    U_profile(i) = Uref*(Z(i)/Zref)^alpha(1); % Vertical wind speed
end % for

%% Calculate standard deviation for all height levels
```

```
U_mat = [LIDAR2_40(1:14,1),LIDAR2_45(1:14,1),LIDAR2_50(1:14,1),LIDAR2_55(1:14,1),LIDAR2_60
    (1:14,1),LIDAR2_65(1:14,1),LIDAR2_70(1:14,1),LIDAR2_75(1:14,1),LIDAR2_80(1:14,1),LIDAR2_85
    (1:14,1),LIDAR2_90(1:14,1),LIDAR2_100(1:14,1)];
U_avg = [LIDAR2_40(15,1),LIDAR2_45(15,1),LIDAR2_50(15,1),LIDAR2_55(15,1),LIDAR2_60(15,1),
    LIDAR2_65(15,1),LIDAR2_70(15,1),LIDAR2_75(15,1),LIDAR2_80(15,1),LIDAR2_85(15,1),LIDAR2_90
    (15,1),LIDAR2_100(15,1)];

U_sd = std(U_mat,0,1); % Standard deviation

%% Calculate hit rate (q)
%Normalize simulated data, ke
Lidar1ke_norm(:,1) = Lidar1ke(:,1)/Lidar1ke(end,1); %Uref;
Lidar1ke_norm_Lidarheights = [Lidar1ke_norm(101,1), Lidar1ke_norm(114,1), Lidar1ke_norm(126,1)
    , Lidar1ke_norm(139,1), Lidar1ke_norm(151,1), Lidar1ke_norm(164,1), Lidar1ke_norm(176,1),
    Lidar1ke_norm(189,1), Lidar1ke_norm(201,1), Lidar1ke_norm(214,1), Lidar1ke_norm(226,1),
    Lidar1ke_norm(251,1)];
%Normalize simulated data, kw
Lidar1kw_norm(:,1) = Lidar1kw(:,1)/Lidar1kw(end,1); %Uref;
Lidar1kw_norm_Lidarheights = [Lidar1kw_norm(101,1), Lidar1kw_norm(114,1), Lidar1kw_norm(126,1)
    , Lidar1kw_norm(139,1), Lidar1kw_norm(151,1), Lidar1kw_norm(164,1), Lidar1kw_norm(176,1),
    Lidar1kw_norm(189,1), Lidar1kw_norm(201,1), Lidar1kw_norm(214,1), Lidar1kw_norm(226,1),
    Lidar1kw_norm(251,1)];
%Normalize simulated data, LES
Lidar1LES_norm(:,1) = Lidar1LES(:,1)/Lidar1LES(end,1); %Uref;
Lidar1LES_norm_Lidarheights = [Lidar1LES_norm(101,1), Lidar1LES_norm(114,1), Lidar1LES_norm
    (126,1), Lidar1LES_norm(139,1), Lidar1LES_norm(151,1), Lidar1LES_norm(164,1),
    Lidar1LES_norm(176,1), Lidar1LES_norm(189,1), Lidar1LES_norm(201,1), Lidar1LES_norm(214,1)
    , Lidar1LES_norm(226,1), Lidar1LES_norm(251,1)];

%Measured normalized data
LIDAR2_AVG_norm=LIDAR2_AVG(2,1:12)/Uref;

% Relative standard deviation
U_std_r = U_sd./LIDAR2_AVG(2,1:12);

% Hit rate correlation
for i = 1:length(U_std_r)
    if abs((LIDAR2_AVG_norm(i) - Lidar1ke_norm_Lidarheights(i))/Lidar1ke_norm_Lidarheights(i))
        <= U_std_r(i)
        N_ke(i) = 1;
    else
        N_ke(i) = 0;
    end % end, if
    if abs((LIDAR2_AVG_norm(i) - Lidar1kw_norm_Lidarheights(i))/Lidar1kw_norm_Lidarheights(i))
        <= U_std_r(i)
        N_kw(i) = 1;
    else
        N_kw(i) = 0;
    end % end, if
    if abs((LIDAR2_AVG_norm(i) - Lidar1LES_norm_Lidarheights(i))/Lidar1LES_norm_Lidarheights(i
        )) <= U_std_r(i)
        N_LES(i) = 1;
    else
        N_LES(i) = 0;
    end % end, if
end % for, i

% Hit rate
q_ke = 1/length(N_ke)*sum(N_ke);
q_kw = 1/length(N_ke)*sum(N_kw);
q_LES = 1/length(N_LES)*sum(N_LES);
```

5.8 C Script

5.8.1 ANSYS Fluent UDF Code

```
#include "udf.h"
#include "math.h"

DEFINE_PROFILE(inlet_x_velocity, thread, index)
{
real x[ND_ND];
real y, z;
face_t f;

/* Polynomial coefficients */

real alpha_C[9] = {2.06736114811560e-21, -5.06582956945179e-18, 4.97440187203092e-15,
    -2.49223399262436e-12, 6.72072167380863e-10, -9.55126083525284e-08, 6.49483790022443e-06,
    -0.000159100877755149, 0.231892828611586};

real z0_C[9] = {-2.09698223370584e-18, 4.32523691267378e-15, -3.54481372442861e-12,
    1.45903580308598e-09, -3.12240507079986e-07, 3.22937343791619e-05, -0.00127806776714647,
    0.0345660118548688, 152.496604304848};

real Ur_C[13] = {-1.04239654354713e-28, 3.57539317616947e-25, -5.37169418421412e-22,
    4.63781029724729e-19, -2.53699074359185e-16, 9.14161052387869e-14, -2.18647089578171e-11,
    3.41759403918153e-09, -3.35824032350335e-07, 1.92737517630026e-05, -0.000550645100048728,
    0.00586055875301183, 12.1261076929833};

real Zr = 209+152;      /* Reference height */
real z0 = 0.; real alpha = 0.; real Ur = 0.;

begin_f_loop(f, thread)
   {
     F_CENTROID(x, f, thread);
     y = x[1];
     z = x[2];

     /* Calculate z0 */
     z0 = z0_C[0]*pow(y,8)+ z0_C[1]*pow(y,7) + z0_C[2]*pow(y,6) + z0_C[3]*pow(y,5) + z0_C[4]*
    pow(y,4) + z0_C[5]*pow(y,3)+ z0_C[6]*pow(y,2) + z0_C[7]*y + z0_C[8];

     /* Calculate alpha */
     alpha = alpha_C[0]*pow(y,8)+ alpha_C[1]*pow(y,7) + alpha_C[2]*pow(y,6) + alpha_C[3]*pow(y
    ,5) + alpha_C[4]*pow(y,4) + alpha_C[5]*pow(y,3) + alpha_C[6]*pow(y,2) + alpha_C[7]*y +
    alpha_C[8];

     /* Calculate U_{ref} */
     Ur = Ur_C[0]*pow(y,12)+Ur_C[1]*pow(y,11)+Ur_C[2]*pow(y,10)+Ur_C[3]*pow(y,9)+Ur_C[4]*pow(y
    ,8)+Ur_C[5]*pow(y,7)+Ur_C[6]*pow(y,6)+Ur_C[7]*pow(y,5)+Ur_C[8]*pow(y,4)+Ur_C[9]*pow(y,3)+
    Ur_C[10]*pow(y,2)+Ur_C[11]*y+Ur_C[12];

     /* Power Law */
     F_PROFILE(f, thread, index) = Ur*pow((z-z0)/(Zr-z0),alpha);
   }
end_f_loop(f, thread)
}
```

5.9 HSE Report

NTNU HMS	Kartlegging av risikofylt aktivitet	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2601	22.03.2011
		Godkjent av		Erstatter
		Rektor		01.12.2006

Enhet: Department of Energy and Process Engineering **Dato: 22.01.2018**

Linjeleder: Terese Løvås

Deltakere ved kartleggingen (m/ funksjon): **Tania Bracchi, Stian Næve Burkeland, Are Simonsen**
(Ansv. veileder, student, evt. medveiledere, evt. andre m. kompetanse)

Kort beskrivelse av hovedaktivitet/hovedprosess: Masteroppgave student Stian Næve Burkeland. Tittel på oppgaven. Urban Wind: CFD analysis of Gløshaugen campus base don measured data

Er oppgaven rent teoretisk? (JA/NEI): **JA** *«JA» betyr at veileder innestår for at oppgaven ikke inneholder noen aktiviteter som krever risikovurdering. Dersom «JA»: Beskriv kort aktiviteteten i kartleggingskjemaet under. Risikovurdering trenger ikke å fylles ut.*

Signaturer: *Ansvarlig veileder:* *Student:* Stian N. Burkeland

ID nr.	Aktivitet/prosess	Ansvarlig	Eksisterende dokumentasjon	Eksisterende sikringstiltak	Lov, forskrift o.l.	Kommentar
1	CFD analysis of the wind conditions at Gløshaugen campus based on measured data	Stian Næve Burkeland, Tania Bracchi				

NTNU HMS	Kartlegging av risikofylt aktivitet	Utarbeidet av	Nummer	Dato	
		HMS-avd.	HMSRV2601	22.03.2011	
		Godkjent av		Erstatter	
		Rektor		01.12.2006	

NTNU HMS	Risikovurdering	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2603	22.03.2011
		Godkjent av		Erstatter
		Rektor		01.12.2006

Enhet: **Dato:**

Linjeleder:

Deltakere ved kartleggingen (m/ funksjon):
(Ansv. Veileder, student, evt. medveiledere, evt. andre m. kompetanse)

Risikovurderingen gjelder hovedaktivitet: Masteroppgave student xx. Tittel på oppgaven.

Signaturer: *Ansvarlig veileder:* *Student:*

ID nr	Aktivitet fra kartleggings-skjemaet	Mulig uønsket hendelse/ belastning	Vurdering av sannsyn-lighet	Vurdering av konsekvens:				Risiko-Verdi (menn-eske)	Kommentarer/status Forslag til tiltak
			(1-5)	Menneske (A-E)	Ytre miljø (A-E)	Øk/ materiell (A-E)	Om-dømme (A-E)		