

Norwegian University of Life Sciences
Faculty of Environmental Science and
Technology
Department of Environmental Sciences

Master Thesis 2016
30 ECTS

Validation of computer simulated air flow in forested areas

Peder Alexander Wiik Wessel



Contents

Contents	i
Abstract	iii
1 Introduction	1
1.1 Wind energy is renewable	1
1.2 Driving forces	2
1.3 Verification and Validation	3
1.4 Aim of study	3
2 Data	5
2.1 SSVAB-data	5
2.2 SMHI-data	10
2.3 Other data	17
3 Theory	19
3.1 CFD-code	19
3.2 Turbulence models	23
3.3 Forest model	24
4 Methodology	29
4.1 Software	29
4.2 Case selection	29
4.3 Validation	40
5 Results	41

5.1	Case selection	41
5.2	Case 1	44
5.3	Case 2	51
5.4	Case 5	58
5.5	Case 6	65
5.6	Atmospheric stability	73
6	Discussion	75
6.1	Uncertainty	76
6.2	Is the model validated?	76
7	Conclusion	79
A	Case selection module	81
	Bibliography	93
	List of Figures	97
	List of Tables	101



Abstract

In this thesis a method to select cases for validation are presented and the interaction between different turbulence models and a forest model is assessed. The terrain model generated in the simulation program is compared with a DEM produced from the same contour lines, but with a different interpolation technique. The forest model is assessed in light of data from land cover data to investigate the similarities among them as well.

Introduction

Wind modeling with *Computational Fluid Dynamics* (CFD) software is a method to predict the energy performance of a wind farm. Accurate simulation of air flow can make on-site measurements redundant. Thus wind energy resource assessment with CFD simulation can shorten the time span from the planning phase of a wind farm project to the operation phase (IEA 2009, p.700). Therefore wind simulation has the potential of being a considerable cost efficient method to assess the wind energy resource in a wind farm project (Church 2016),(Kato et al. 2016, p.316). In order to assess the accuracy of the simulations the flow models used in wind simulation should to be validated. This thesis encompasses a mathematical model regarding how a forest affects a wind field over complex terrain. In the forest model there are two source terms for friction and this thesis is an attempt to simulate a forest effect with one of two source terms. A method for selecting cases will also be developed which will be used for validation.

1.1 Wind energy is renewable

The share of renewable energy in the global energy mix of electricity is growing and will continue to grow in the foreseeable future. In 2010 9% of the global energy mix was generated from modern renewable energy (RE)¹ and is expected to increase to 30% by 2030 (*REmap 2030: A Renewable Energy Roadmap, June 2014* 2014). In order to reach the target the share of wind energy in the global energy mix of RE is expected to change from 2% in 2010 to 11% by 2030 (*REmap 2030: A Renewable*

¹hydroelectricity, wind, solar PV, modern bioenergy (waste and biomass power), geothermal power, concentrated solar power, biogas power

Energy Roadmap, June 2014 2014) with a proportional increase in the initiation of wind farm projects.

The earth is continuously exposed to solar energy on constant changing parts of the surface because of the earth's orbit around the sun and its constant rotation around the pole-to-pole axis. The energy from the rotating mass gives rise to numerous intriguing meteorological phenomena as a spatial distribution of temperature in the atmosphere, which can be divided into high and low pressure areas. Wind is a result of the displacement of air from an area of high pressure to another area of low pressure. The continuity of the rotating mass enables the replenishing of the wind energy whenever it is harvested, which gives electricity production from wind energy the property of being renewable (Sørensen 2004).

1.2 Driving forces

The main driving forces in promoting RE are i.e. national policies (i.e. "20-20-20"-target in EU, Clean Power Plan in USA and 12th five year plan in China and national energy security plans), increasing demand of energy and green house gas reductions (Right 2011). Worldwide there are 164 countries with a renewable energy target (Greenpeace International 2015). The increasing concern about the environmental impact related to the emissions related to electricity production the energy sector is an instrument in the national climate policy to achieve emission goals². Historically there has been a strong correlation with growth in energy consumption, economic growth and an increase in emissions (Saidi et al. 2015). The energy demand is expected to grow and the demand of RE is likely to proportionately.

A combination of widespread interest in RE as an advantage in a competitive trade market, RE as a mean to achieve national emission goals, technology learning in RE market and free fuel is contributing to increase the importance of planning wind energy projects. In the planning of wind energy projects simulation and thus validation of the wind models are essential parts.

By the end of 2014 the global installed capacity reached 51477MW and 1713MW was in the offshore wind energy sector (Council 2015b). In a policy scenario the Global Wind Energy Council (GWEC) have reported that the global cumulative installed capacity will be 712081 MW, where 10% will be at sea (Council 2015a). With a cumulative capacity offshore of 8771MW in 2014 the annual installed capacity will be 20GW in 2019³. GWEC projects that the global annual installed capacity will be

²Emission goals can be to reduce local air pollution, to mitigate climate change, reduce the detrimental effect on human health and agriculture and thus to reduce related socioeconomic costs.

³This number is found by interpolation with the rent formula.

66,5GW in 2019. It follows that in near future the largest share of installed wind energy technology will continue to be land based despite the forecasted accelerated growth in installed capacity in offshore wind energy.⁴

Before erecting a wind turbine cluster there has to be solid arguments for initiating the construction presented to and weighted by decision makers. The annual estimated production (AEP) is a tipping point for a wind park project and is a result of a wind park design. The wind park design is a result of a simulated location specific case with data derived from a land use planning area. Estimating the energy production of a wind farm with simulation is a cost effective method to either increase the AEP of an existing wind park or to provide credible arguments in the decision making process in the initiation phase of a wind power plant project — in an economic perspective it is to increase the project value or the bankability⁵. Since the electricity production in a wind farm is variable, because of the fluctuating energy carrier, it is of great interest for the producer, investors and system operator to reduce the uncertainty about the forecasted production. Wind simulation can reduce the uncertainty in estimated energy production.

1.3 Verification and Validation

A CFD model describes a fluid flow with sophisticated mathematics and computation and does not give accurate results, but insight in a flow problem (Versteeg et al. 2007). When assessing a model two essential steps are verification and validation, which can result in modification of the model or a finalization of a model (Roache 1997). One fundamental assumption is that the forest model, which is being investigated in this thesis, is verified by the model builder (A. R. Gravdahl et al. 2000).

1.4 Aim of study

The aim of study is to develop a method for selecting validation cases and test the model for canopy turbulence with different turbulence models; “standard k- ϵ ”, “RNG-k- ϵ ”, “YAP-modified k-epsilon” and “modified”.

⁴The projected amount of annual installed capacity describes a forecasted market interest in new wind projects where simulation will be likely to be used in this case.

⁵Bankability is often used as the opposite property of having low financial risk, that means if the bankability is high the financial risk is low.

Data

The data values needed to validate the forest model used in this thesis are geographical positions, topography, roughness height, humidity, temperature, the wind speed and wind direction. The norwegian wind engineering company called “Kjeller Vindteknikk” (KV) was the contractor of the measurement campaign of wind speed, wind direction and temperature and to logg them with respect to time. The orderer of the measurement campaign is the wind energy company “Statkraft SCA Vind” (SSVAB). The data for pressure and relative humidity is provided from the Swedish meteorological office (SMHI).

2.1 SSVAB-data

The main purpose of the SSVAB-datasets used in this thesis is to calculate a possible annual energy production (AEP). The datasets for wind speed and direction and temperature difference used in this thesis has been filtered for ice by the contractor. After filtering “Not A Number”-values, date and synchronizing the different datasets 40 % of the aforementioned data is used in this thesis. The measurements are performed in a forested area and therefore it possible to find a significant number of suitable cases to validate the forest model. The measured parameters that lays the ground for discussion of the validation are described in this chapter.

Geographical position

The geographical positions of the fourteen meteorological masts are retrieved from the datasets for wind speed and direction. The geographical positions are recorded

Table 2.1: The geographical positions of the meteorological masts which has been used for recording wind speed and temperature. The elevation is the height above sea to the bottom of the meteorological mast. The coordinates refers to the map datum WGS84 and is the UTM-zone 33N.

Station number	Elevation above sea level (m)	Latitude (UTM)	Longitude (UTM)
6603	480	7043583	540090
6604	480	7041796	547093
6605	536	7030718	552333
6606	481	7034280	556179
6610	479	7044947	539518
6611	477	7039682	539321
6612	443	7040574	541850
6613	423	7044869	546995
6614	526	7040645	548555
6615	522	7033958	549464
6616	483	7023898	555587
6617	503	7026941	552508
6618	571	7031142	557531
6619	478	7027207	559795

with a manual GPS around 6 times for each meteorological mast. The GPS positions are recorded in the map datum WGS 84. The listed coordinates in table 2.1 are the mean values of the coordinates that is recorded. The fourteen meteorological masts are visualized in figure 2.1. The map is produced with the geographical information system (GIS) tool QGIS (version 2.14.2). The layers in the map are provided by the swedish national mapping department (in swedish: Lantmäteriet) (*GSD-general map, vector - product description* 2016).

Topography

The elevation profile between two meteorological masts is acquired from contour line maps. Contour line maps are usually a result of measurement with laser scanning equipment done by air. It is not mentioned in the sources how the height data is captured. The contour line map in figure 2.2a is a merged map between two merged contour line maps with 5mx5m resolution of the nearest surroundings of the meteorological masts (at four project sites) and 10mx10m resolution elsewhere. The green map in figure 2.2a is made for the project sites Ögonfågeln eastward Bodhögarna. The brown map focuses on the project sites Björkhöjden which is eastward of Björkvattnet.

The digital elevation model (DEM) generated from the contour line maps is

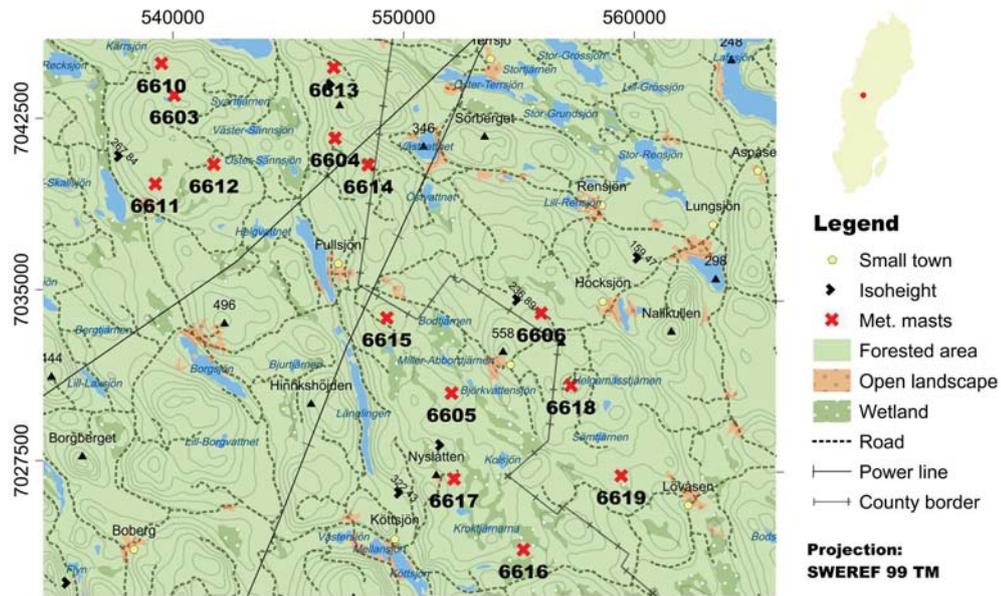


Figure 2.1: The location of the fourteen meteorological masts where wind speed and the associated wind direction and temperature have been measured from december 2007 and throughout 2011 in the wind farm areas “Björkhöjden”, “Ögonfågeln” and “Bodhögarna”.

visualized in figure 2.2b with the geoinformation system Geomatica (*Geomatica Core - Technical Specifications* 2016) with the algorithm “VDEMINT”. The “VDEMINT”-algorithm generates a raster digital elevation model from vector segments by interpolating values of points, contours and 3-D structure lines. The height values are shown in grayscale where the brightest parts represents the highest parts and the darkest the lowest part. The parts of the map with lower resolution is blurrier than the parts with higher resolution. The argument for merging two merged maps with is to get a contour line map with all the project sites with the largest resolution possible. By comparing the elevation values at 10 000 points on the contour line map and the DEM with a comparison function (ELVECRMS) in Geomatica it is found that the total RMS and average error is 0,025692m and $-0,00015$ m. The DEM is therefore an acceptable representation of the terrain. However the zero values in the south west and north east side of the merged contour line map gives rise to non zero height values in the DEM that are indeed invalid and erroneous. The lines of interest are between the four project sites. Furthermore the invalid height values the purpose of including the DEM in thesis is to get a rough representation of the terrain between two meteorological. The possible edge effects the invalid height on the interpolation of the points on the contour lines are not of great importance in this matter.

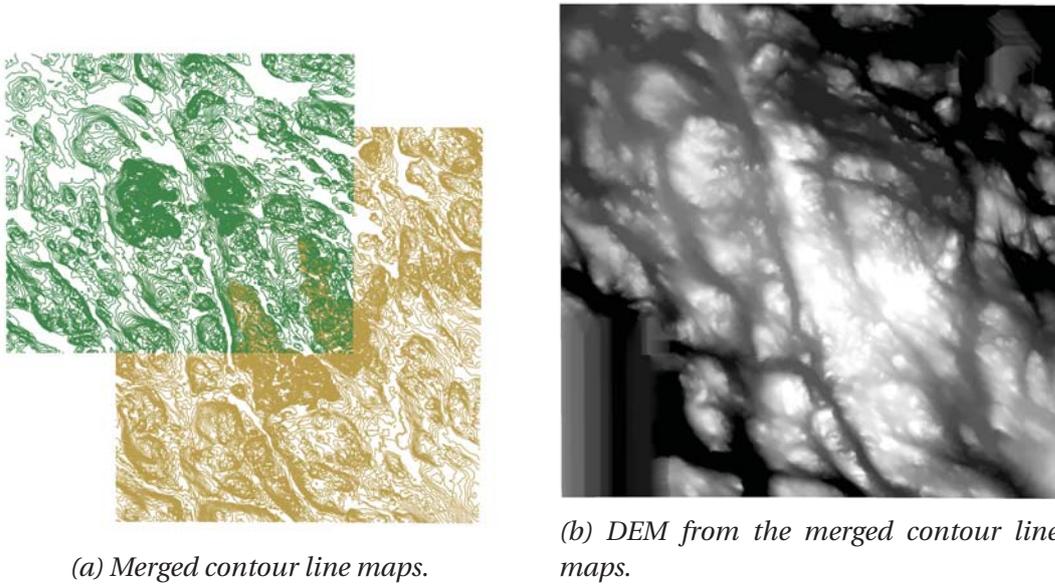


Figure 2.2: The maps used to retrieve topographical data used in the case selection.

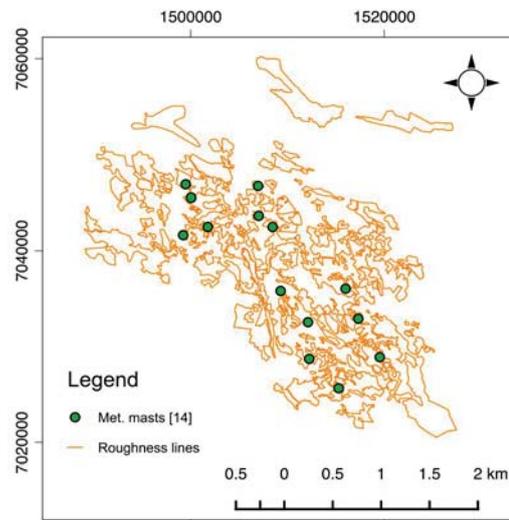


Figure 2.3: The roughness map used to as input data for the simulation.

Roughness height

KV has provided a roughness map which has been made manually based on background maps and aerial photos. The roughness chosen for the validation will be from retrieved from this map where the roughness is associated with a geographical coordinate.



(a) NRG # 40

(b) Thies Transmitter

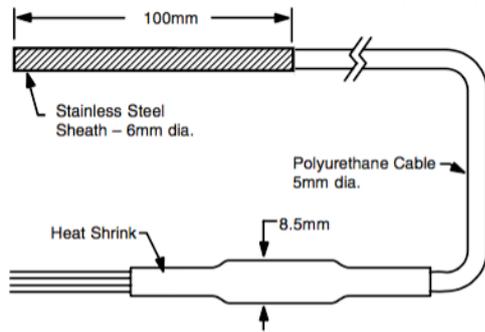
Figure 2.4: The anemometers used to measure the wind speed at the different locations (The world of weather data 2011; NRG #40C Anemometer datasheet 2010).

Wind speed

The wind speed is measured with cup anemometers of the type “NRG #40” and “Wind transmitter First Class” (The world of weather data 2011; NRG #40C Anemometer datasheet 2010). The correlation factor between the electronic AC frequency used to log the wind speed and the wind speed is 0,99995 of the type cup anemometer Thies. The accuracy of “NRG # 40”-type is within 0,1m/s for the range 5 m/s to 25m/s. The wind speed range recorded with the Thies cup anemometer is 0,23 – 18,74m/s and the values of wind speed interest are above 5m/s which implies that the accuracy probably will not be a significant source of uncertainty in the thesis.

Temperature

One type of temperature sensor has been used to measure the temperature near the top of meteorological masts and another type to measure the temperature difference (a “termodiff” element) between 2m above ground level and height level near the top. Data from the latter sensor type is used in this thesis. The “termodiff” element consists of two temperature probes of the type “Campbell thermistor 107 Probe”. The temperature range in the period between 23rd of february 2009 and 22nd of february 2010 at station 6604 at 58m is $-12\text{ }^{\circ}\text{C}$ – $-25,4^{\circ}\text{C}$ which implies that the measuring error is potentially $\pm 0,1^{\circ}\text{C}$ (107, 108 & 105T Temperature Probes & 41303-5 Radiation Shield 2005). That can lead to an aggregation error and thus an uncertainty in the calculation in the part of atmospheric stability in the case selection module which will be explained in section 4.2.



(a) Construction of 107 thermistor probe.



(b) A thermistor in a radiation shield.

Figure 2.5: The temperature sensors used to measure the temperature difference between 2m above ground level and the top of the met masts (107, 108 & 105T Temperature Probes & 41303-5 Radiation Shield 2005).

2.2 SMHI-data

The Swedish meteorological office “Sveriges Meteorologiska och Hydrologiska institut” (SMHI) have operated and maintained 135 automatic meteorological stations by 2013 across Sweden for decades. The SMHI weather stations “Hallhåxåsen A”, “Junsele A” and “Krångede A” are in the proximity and are oriented in a triangulating manner around the fourteen meteorological stations. Therefore there are solid arguments for assuming a correlation between the measured parameters at the SMHI and SSVAB stations. At the SMHI meteorological stations of interest in this thesis the parameters used will be relative humidity, air temperature and air pressure. In the following section the choice of datasets operated by to include is discussed.

Choice of SMHI-data sets

SMHI offers historic data with hourly temporal resolution of relative humidity and air pressure among other weather parameters at three measurement stations around the projected wind farm in a triangulating manner (the four project sites); “Hallhåxåsen A”, “Junsele A” and “Krångede A”. The location of the four stations operated by SMHI can be viewed in figure 2.6. For reasons of time efficiency, ie. to fasten the programming effort, the intention is to only include historic data from one weather station for calculations at all fourteen meteorological masts. The distance between the SMHI weather stations and the SSVAB meteorological stations and a correlation factor between the datasets are the criteria for choosing one of the datasets. The most northern and southern measurement station is 6610 and 6616

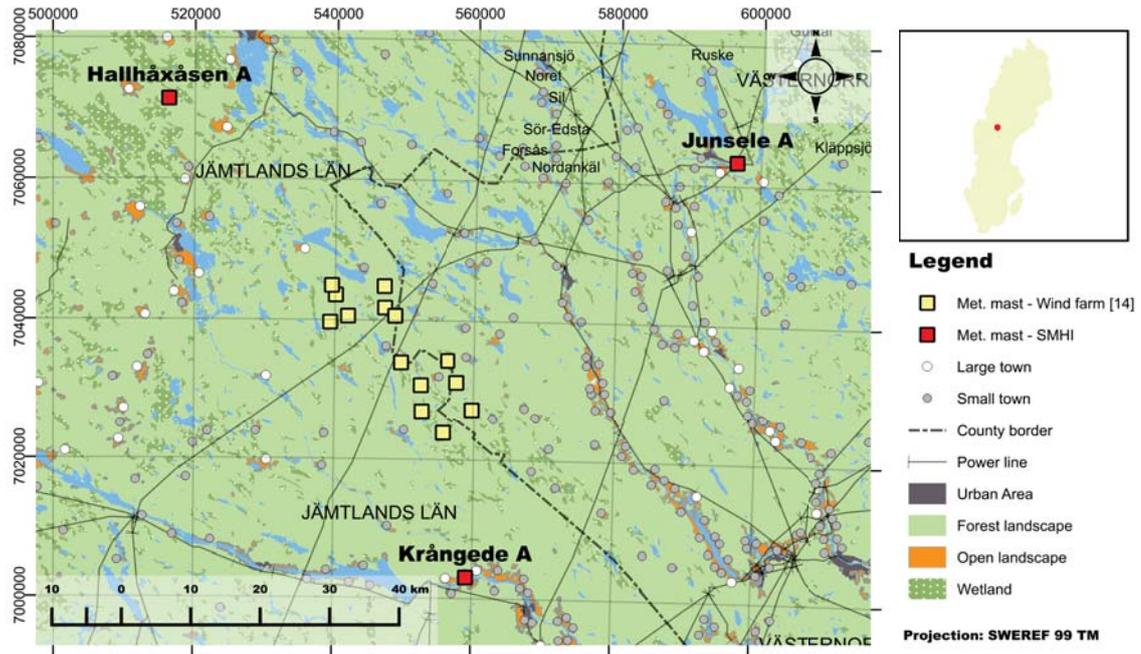


Figure 2.6: The location of the meteorological masts installed by SMHI and the wind measurement firm.

Table 2.2: The distances in kilometers between the most northern and southern meteorological stations and the SMHI automatic weather stations.

Station name	Krångede A	Hallhåxåsen A	Junsele A
6610	45	35	60
6616	20	61,5	57

(see figure 2.1) so the four project sites spans from north west to south east. The distances are described in table 2.2.

According to the distances in table 2.2 the closest SMHI weather station is “Krångede A”. Therefore “Krångede A” is the most geographically correlated weather stations to the projected wind farm. The subsequent feature to investigate is the correlation between the temperature 2m above ground level at Krångede A and the SSVAB weather stations 6610 and 6616. Temperature and pressure describes the state of a gas. The equation of state of an ideal gas is formulated as (Atkins et al. 2009):

$$pV = nRT \quad (2.1)$$

where p is the pressure, V is the volume, n is the molar fraction, R is the gas constant, T is the temperature of a gas. Thus the correlation between the temperature at the chosen SMHI weather stations and the SSVAB meteorological

stations is also a correlation between the pressure at the weather stations. To find the correlation between the datasets the Fischer correlation formulas is suitable (Fischer 1934, p.199). When comparing two data sets the following formula has been used:

$$r_{1,2} = \frac{1}{Ns^2} \sum_{n=1}^N (x_{n,1} - \bar{x})(x_{n,2} - \bar{x}) \quad (2.2)$$

where

$$\bar{x} = \frac{1}{2N} \sum_{n=1}^N (x_{n,1} + x_{n,2}) \quad (2.3)$$

\bar{x} is the average value, s is the standard deviation and N is the dimension of the two data sets.

$$s^2 = \frac{1}{2N} \left[\sum_{n=1}^N (x_{n,1} - \bar{x})^2 + \sum_{n=1}^N (x_{n,2} - \bar{x})^2 \right] \quad (2.4)$$

To verify the correlation coefficient the Python function `pearsonr` in the statistics module `scipy.stats` is used for two datasets. The implemented formula in the function is the sample correlation coefficient formula for interclass coefficients (Devore et al. 2012).

$$r_{1,2} = \frac{\sum_{n=1}^N (x_{n,1} - \bar{x}_1)(x_{n,2} - \bar{x}_2)}{\sqrt{\sum_{n=1}^N (x_{n,1} - \bar{x}_1)^2 \cdot \sum_{n=1}^N (x_{n,2} - \bar{x}_2)^2}} \quad (2.5)$$

and when comparing three data sets the Fischer intraclass correlation formula is expanded further to (Fischer 1934, p.201):

$$r_{1,2,3} = \frac{1}{3Ns^2} \sum_{n=1}^N [(x_{n,1} - \bar{x})(x_{n,2} - \bar{x}) + (x_{n,1} - \bar{x})(x_{n,3} - \bar{x}) + (x_{n,2} - \bar{x})(x_{n,3} - \bar{x})] \quad (2.6)$$

where

$$\bar{x} = \frac{1}{3N} \sum_{n=1}^N (x_{n,1} + x_{n,2} + x_{n,3}) \quad (2.7)$$

and

$$s^2 = \frac{1}{2N} \left[\sum_{n=1}^N (x_{n,1} - \bar{x})^2 + \sum_{n=1}^N (x_{n,2} - \bar{x})^2 + \sum_{n=1}^N (x_{n,3} - \bar{x})^2 \right] \quad (2.8)$$

The formulas for intraclass correlation (equations 2.2, 2.6) have the property of comparing the discrete values in the data sets with the average value of the assimilated datasets in the covariance (the numerator) and the standard deviation (the denominator). That results in a more precise value of the correlation between the data sets belonging to the same class than the correlation using the Pearson r formula

Table 2.3: Statistics of the correlation between the temperature at SSVAB meteorological stations and Krångede A.

Station number	6610	6616
Standard deviation	9,65	9,23
Average value	4,92	4,75
Sample number	6700	5206
Sample percentage	76 %	59 %
Fischer correlation coefficient (eq: 2.2)	0,94	0,94
Pearson correlation coefficient (eq: 2.5)	0,96	0,95

(equation 2.5, Fischer 1934, p.200). In table 2.3 it is clear that the Pearson interclass coefficient gives a stronger positive correlation than the Fischer correlation coefficient.

The term sample percentage (in tables 2.3, 2.5, 2.4) is the sample number related to the maximum possible sample numbers. The maximum possible sample numbers with measurements every hour is 8760 between the concurrent periods of measurements of all the meteorological masts at the project sites. The concurrent period of measurement is between 23rd of february 2009 and 22nd of february 2010. Thus each meteorological mast has different timeline of measurement. The positive correlation coefficients in table 2.3 and the SMHI weather stations of circa 95% shows that there is a strong linear relation between the temperature in the data sets at the meteorological stations 6610 and 6616. The low sample percentage of the synchronous measurements between station 6616 and Krångede A may result in a higher correlation coefficient. The similar correlation coefficients between stations 6610 and 6616 may indicate that a higher sample percentage would result in the same correlation coefficient thus the correlation coefficients are likely to be representative for the whole observation period.

The next feature to investigate is the correlation between the measurements of humidity and pressure at Krångede A, Hallhåxåsen A and Junsele A in order to identify the regional variation of the parameters. The hypothesis is if there is high variation among the synchronous discrete values there is a weak positive correlation and thus the use of one of the data seta in this thesis will result in high uncertainty. Humidity and pressure are not proportional unities and must therefore be considered separately.

The difference between the correlation coefficients regarding relative humidity and pressure in table 2.4 and 2.5 indicates that there is a higher local variation of

Table 2.4: Statistics of the correlation between the moisture content in the air, as relative humidity, recorded 2m above ground level at the three weather stations operated and maintained by SMHI which are in the proximity of planned the wind farm.

	Krångede A/ Junsele A	Krångede A/ Hallhåxåsen A	Junsele A/ Hallhåxåsen A	All
Standard deviation (%)	17,35	16,58	17,06	17,02
Average value (%) ¹	82,40	83,15	82,84	82,78
Sample number	8633	8754	8630	8629
Sample percentage(%)	99	100	99	99
Fischer coeff. (eq: (2.2,2.6))	0,91	0,85	0,87	0,87
Pearson coeff. (eq: 2.5)	0,91	0,85	0,88	

Table 2.5: Statistics of the correlation between the pressure at 2m above ground level at the three weather stations operated and maintained by SMHI and are located in the vicinity of the wind farm project site. The pressure is reduced to sea level pressure

	Krångede A/ Junsele A	Krångede A/ Hallhåxåsen A	Junsele A/ Hallhåxåsen A	All (2.6)
Standard deviation (hPa)	11,28	11,27	11,28	11,28
Average value (hPa)	1012,69	1012,99	1012,69	1012,82
Sample number	8553	8669	8553	8553
Sample percentage (%)	98	99	98	98
Fischer coeff. (eq: 2.2,2.6)	0,99	1	0,99	0,99
Pearson coeff. (eq: 2.5)	0,99	1	0,99	

moisture content in the air than atmospheric pressure. It should be noted that the recorded atmospheric pressures of the SMHI weather stations are projected to sea level in order to discern the variations of the air pressure if the pressures were at the same level. The validity of measurements relative humidity is confined to the planetary boundary layer at low heights. Therefore extrapolating data for relative humidity from Krångede A to the project site is a significant source of uncertainty as distinct from the discrete data values of pressure in the area which has a very strong positive correlation. A plausible cause for the lower correlation between the pairs with Hallhåxåsen A may be due to the fact that this stations is located more centered in a wetland area than Junsele A. The histogram of humidity at the different weather stations in figure 2.7 shows the difference in the distribution of relative at the sites. The correlation coefficients between the data sets of the SMHI weather stations

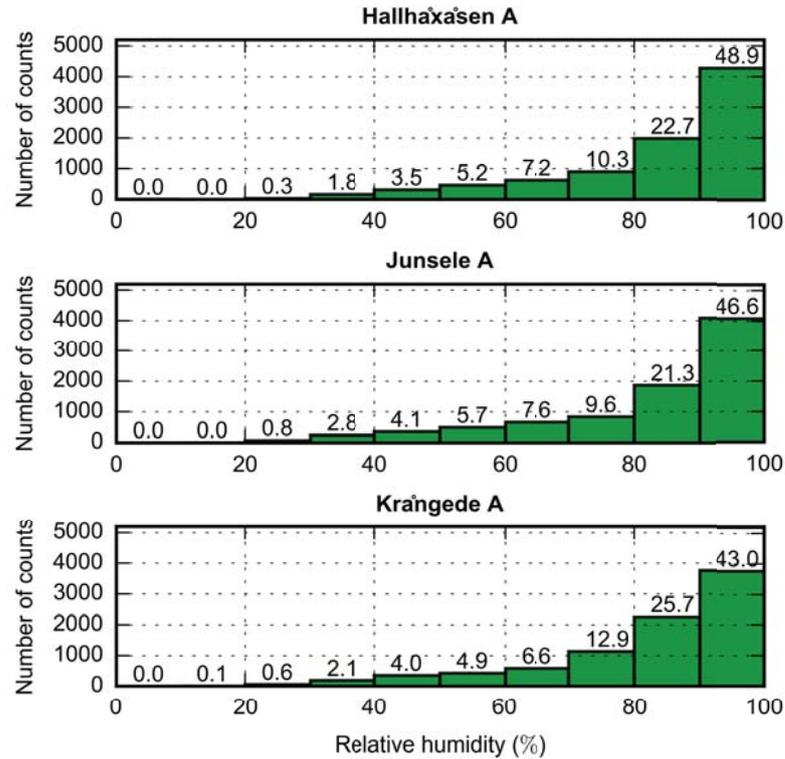


Figure 2.7: The distribution of relative humidity at the different meteorological stations around the four project sites. The number on top of the columns is sample number related to the potential dimension of the data set in percentage if it there were no data loss which is 8760.

is reflected in the histogram. In table 2.4 Hallhåxåsen A and Krångede are least correlated hence the largest difference between the binned values and Krångede A and Junsele A have the highest correlation hence the lowest difference in the binned values. The high sample percentage in the comparison between the SMHI-data sets states that the correlation coefficients have high reliability. The high correlation between the measurements at the project sites (SSVAB-data), the SMHI weather station and Krångede A states that the data sets for pressure produced at Krångede A can be extrapolated to the project sites. On the other hand the data set for humidity must be used with caution.

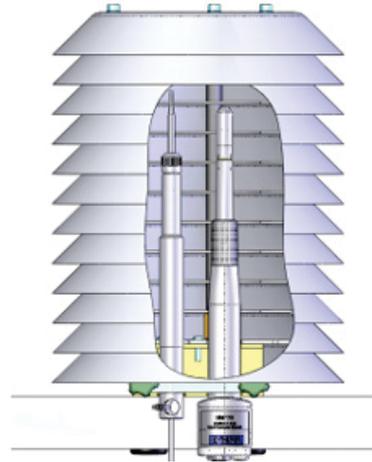
Humidity

The relative humidity is measured as an electronic pulse on a water sensitive polymer film at 2m above ground level. The humidity sensor on the standard weather station is the Vaisala HMP35D and is mounted with a temperature probe in a protective cage of the type Vaisala DTR13 (Persson 2000; *HMP35D Humidity and Temperature Probe*

- *Data sheet 1991; Installation av väderradar vid Leksand 2002*) which protects the sensors from precipitation and radiation. The relative humidity between 0% and 100% is proportional to a voltage output between 0V and 1V (*How do you measure air moisture?* 2016). The accuracy of the equipment is $\pm 2\%$ RH between 0% and 90% and $\pm 3\%$ RH between 90% and 100%RH.



(a) The Vaisala HMP35D in a radiation shield Vaisala DTR13



(b) Vaisala HMP155 with a temperature probe installed in DTR13.

Figure 2.8: The radiation shields protecting the temperature probe and the humidity sensor. The HMP35D is mounted in the same way as the HMP155 in the radiation shield (How do you measure air moisture? 2016).

Air pressure

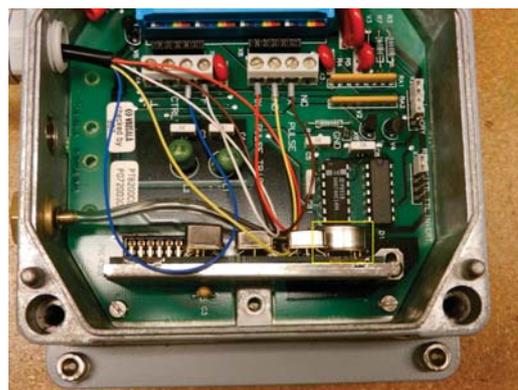


Figure 2.9: The Vaisala PTB201A-barometer. The pressure sensor is the unit within the yellow rectangle (How do you measure air pressure? 2015).

Air pressure is measured with an electronic sensor which is a silicon capacitive unit connected to a circuit board of the type Vaisala PTB201A and a data logging unit

2.9. The circuit board is mounted inside a box for protection against weathering. The total accuracy is ± 0.3 hPa. The logged air pressure in the data set is reduced to sea level pressure. The reduced air pressure implies that the air pressure at the station is projected to sea level and is related to latitude, elevation above sea level and ambient temperature from the following equation:

$$p_0 = p \cdot e^{\frac{H \cdot B}{T1}} \quad (2.9)$$

which is a formula derived from the barometric formula. The definitions of the symbols are,

- p_0 : The air pressure projected to sea level
- p : The air pressure measured at the weather station (station pressure)
- H : The altitude of the barometer above sea level
- B : $0,034163 \cdot (1 - 0,0026373 \cdot \cos(2L))$
- $T1$: $\begin{cases} 0,535 \cdot t + 275,6 & , -7^\circ\text{C} \leq t < 2^\circ\text{C} \\ 0,500 \cdot t + 275,0 & , t < -7^\circ\text{C} \\ 1,07 \cdot t + 274,5 & , t \geq 2^\circ\text{C} \end{cases}$
- L : The latitude of the station in decimal degrees (Datum WGS84)
- t : The air temperature in $^\circ\text{C}$

The air pressure data are thus projected to the elevation and the latitude of the meteorological stations in the project sites.

2.3 Other data

Forest height

In order to locate the forest in the area between two meteorological masts in the project sites forest height data produced by Swedish University of Agricultural Sciences (SLU) is used (*SLU Forest Map 2010*). The forest height data is produced with field data from the Swedish National Forest Inventory, satellite pictures from the Swedish satellite data base "SACCESS" and calibration data from the NASA sensor MODIS. The forest height data is in a raster file format with a resolution of 25mx25m and covers all of Sweden which implies that if the area of scope is less than a few hectares (100m^2) the data values should be used with caution.

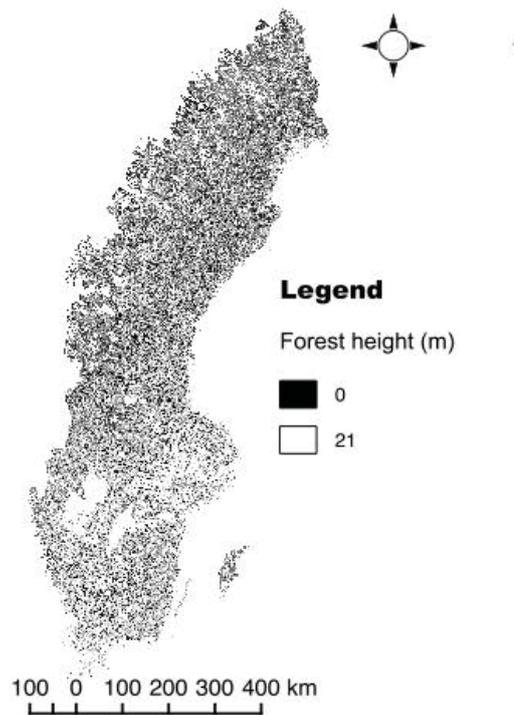


Figure 2.10: A visualization of the raster file of forest height data in Sweden provided by SLU.

Forest type

In order to evaluate the gridded forest generated in the simulation raster data of land cover distributed by the “European Environment Agency” will be used (*Corine Land Cover 2006 raster data 2012*). It has grid resolution of 100mx100m.

Map layers

The map layers used to make figures 2.1 and 2.6 that presents the locations and the terrain around the meteorological stations operated by SMHI and at the project site are provided by the Swedish map department “Lantmäteriet” (*GSD-general map, vector - product description 2016*). The map layers are vector layers in the map format “.shp” and are in the projection “SWEREF99TM” and the height data are in “RH2000” (*Enhetligt geodetiskt referenssystem 2009*) with a varying resolution depending on the layer property.

Theory

The forest model which is in focus in this thesis is a part of the simulation program WindSim. It is a user defined opportunity and is used if there is a forest near the area of interest and if it is likely to have a mesoscale effect i.e. if the forest is 1km or larger in diameter. The structure and governing equations in WindSim and the forest model will be described in this chapter.

3.1 CFD-code

WindSim is a commercial wind park optimizing program which is based on the source code PHOENICS of computational fluid dynamics (CFD) for general purpose. CFD codes are designed to solve fluid flow problems with numerical solutions. All CFD codes are composed of three main parts: i) a pre-processor, ii) a solver, iii) a post-processor (Versteeg et al. 2007).

Pre-processor

The pre-processor includes the definition of the geometry of region of interest, the grid structure, the choice of physical and chemical entities (an item and the fluid), the fluid properties (ie. viscosity, pressure, density) and boundary conditions.

Solver

There are three prevailing numerical solution techniques: finite difference, finite element and spectral methods. The finite volume method is a variant of the finite difference method and is the pivotal part of the PHOENICS CFD code. The finite

volume method solves flow problems by integrating the governing equations of fluid flow at the center of all cells in the grid, converting the integrated equations to algebraic notation (discretization) and solving the algebraic equations by an iterative method.

Post Processor

The post processor converts the solved fluid flow problem which is a systemized array of numbers. The conversion can be visualization of the grid, vector plots, line, shaded 2D or 3D plots and so forth. Hence PHOENICS is the pre-processor and solver and WindSim is the post-processor in the CFD program and a graphical user interface (GUI) for the pre-processor.

Governing equations for fluid flow

PHOENICS solves the governing equations for fluid flow that is subject to the conservation laws of physics. The formulation for conservation of matter, momentum (Newton's second law) and energy (first law of thermodynamics).

Conservation of matter

The conservation of mass is confined to a closed system where rate of increase of mass equals to the net rate of flow of mass into a fluid element. By considering the

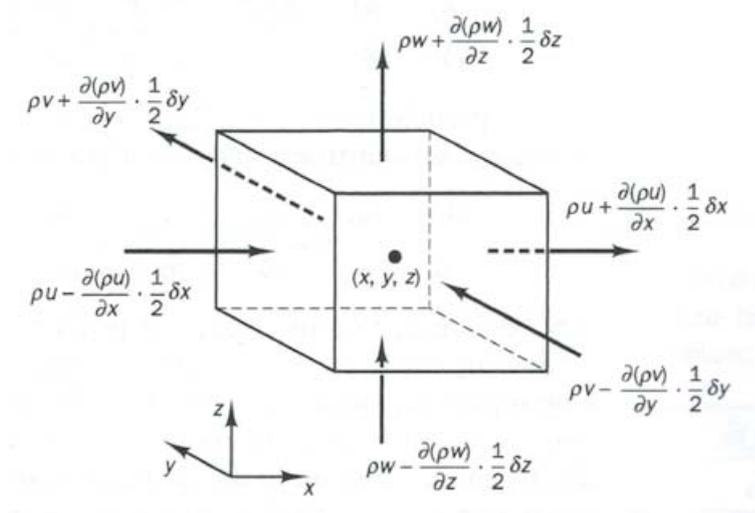


Figure 3.1: The flow of matter in and out of a fluid element (Versteeg et al. 2007).

mass flow in and out of a finite element described in figure 3.1 the mass balance

yields

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial t} + \frac{\partial \rho v}{\partial t} + \frac{\partial \rho w}{\partial t} = 0 \quad (3.1)$$

or

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \quad (3.2)$$

The conservation of matter is thus formulated in equation 3.2 and is the unsteady, three-dimensional continuity equation for a point in a compressible fluid. In this thesis only incompressible flows will be assessed, in which the density is constant, so equation 3.2 is formulated

$$\nabla \cdot \mathbf{u} = 0 \quad (3.3)$$

which states that the net flow of mass out of the element through its boundaries equals to zero.

Newton's second law

Newton's second law states that the rate of change of momentum equals to the sum of the forces on a fluid particle

$$\frac{1}{V} \sum \mathbf{F} = \rho \frac{d\mathbf{u}}{dt} \quad (3.4)$$

where V [m³] is the volume of a finite fluid element and \mathbf{u} is the translation vector. By adding the pressure, viscous and body forces acting of the surfaces of a finite fluid element described in figure 3.1 the x-component of the momentum equation is

$$\rho \frac{Du}{dt} = \frac{-\partial p + \tau_{xx}}{\partial x} + \frac{\tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + S_{Mx} \quad (3.5)$$

in which τ [N] is the viscous stress exerted by surrounding fluid on the fluid element's surfaces in a laminar flow and S_{Mx} is a source term of a body force and

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$$

The y- and z-components in the momentum equation are derived in a similar manner (for full derivation see Versteeg et al. 2007, chap.2).

However the gravity force is a ubiquitous force acting on the fluid elements in a mesh and is thus a source term in the momentum equation in PHOENICS.

The first law of thermodynamics

The grid structure that the user defines when initiating a simulation in WindSim is a closed and isolated system and therefore the first law of thermodynamics is put

into force and is to be implemented in the transport equations of fluid particles. The law states that the internal energy of an isolated system can not change (Atkins et al. 2009):

$$\Delta U = w + q \quad (3.6)$$

in which U [J] is the internal energy, w [J] and q [J] is the energy exchanged between the system and its surroundings. The first law of thermodynamics applied on a fluid particle in motion per unit volume can also be written as the rate of change of energy (Holton 2004, chap.2.7):

$$\rho \frac{D}{Dt} e = -p \nabla \cdot \mathbf{u} + \rho q \quad (3.7)$$

where q [J/kgs] is the rate of heating per unit mass due to radiation, conduction and latent heat release and e is thermal. This equation can be formulated in a general form as the energy equation for a finite fluid element enveloped in a continuous fluid (for full derivation see Versteeg et al. 2007, chap. 2):

$$\begin{aligned} \rho \frac{DE}{Dt} = & \left[-\nabla \cdot p \mathbf{u} + \frac{\partial(u\tau_{xx})}{\partial x} + \frac{\partial(u\tau_{yx})}{\partial y} + \frac{\partial(u\tau_{zx})}{\partial z} + \frac{\partial(v\tau_{xy})}{\partial x} + \frac{\partial(v\tau_{yy})}{\partial y} \right. \\ & \left. + \frac{\partial(v\tau_{zy})}{\partial z} + \frac{\partial(w\tau_{xz})}{\partial x} + \frac{\partial(w\tau_{yz})}{\partial y} + \frac{\partial(w\tau_{zz})}{\partial z} \right] + \nabla \cdot (k \nabla T) + S_E \end{aligned} \quad (3.8)$$

The terms within the brackets is the total rate of work done on the fluid particle by surface stresses and $\nabla \cdot k \nabla T$ is rate of heat addition to the fluid particle due to heat conduction across element boundaries which resembles the simpler form in equation 3.6. S_E is a source term for energy. The subscripts i and j in τ_{ij} describes that the viscous stress (viscous force per surface area) acts in the j -direction on an area perpendicular to the i -direction. From the form in equation 3.8 equations for the rate of change internal energy, kinetic energy and temperature can be derived. The viscous stress τ_{ij} is unknown and is for most cases the continuum in focus is a newtonian fluid. Viscosity terms for newtonian fluids are solved by using the Navier-Stokes equations.

Transport equation

The equations for conservation of matter, energy and momentum (called property) are differential equations and in order for a property ϕ to propagate from cell to cell in a grid, it has to be integrated. The transport equation for steady state flows with for incompressible fluid and is formulated as:

$$\int_A \mathbf{n} \cdot (\rho \phi \mathbf{u}) dA = \int_A \mathbf{n} (\Gamma \nabla \phi) dA + \int_{CV} S_\phi dV \quad (3.9)$$

The left side is the net rate of decrease of ϕ due to convection, the right side is the net rate of increase of ϕ due to diffusion and net rate of production of ϕ inside the boundaries of the fluid element.

Reynold Averaged Navier Stokes equations

Turbulence causes the instantaneous velocity u to fluctuate with time (Finnemore et al. 2009). Therefore the instantaneous velocity can be written as:

$$u(t) = \bar{u} + u' \quad (3.10)$$

in which \bar{u} [m/s] is the average velocity and u' is the difference between the instantaneous and the average velocity. WindSim uses time-averaged transport equations for incompressible flow (constant density) which means that the average difference between the instantaneous and average velocity is zero:

$$\overline{u'} = \overline{u(t) - \bar{u}} = 0$$

which is the idea behind the Reynold Averaged Navier Stokes equations (RANS). The RANS equations includes the time-averaged mathematical expressions for momentum, viscous shear stress (the aforementioned Navier Stokes equations) and transport equations which has been described in earlier in this section. This is the most widely used fundamental set of equations for turbulence modeling because the instantaneous value of a property (e.g. temperature, kinetic energy or heat) is rarely of interest as opposed to the average value (see Versteeg et al. 2007, chap. 3). The RANS equations for incompressible flow are integral parts of PHOENICS, however the effect turbulence has on the average velocities is yet to be described as part of turbulence models.

3.2 Turbulence models

The turbulence models solved with PHOENICS are transport equations added to the set of RANS equations. In turbulent flow the total shear stress consists of the laminar shear stress and the turbulent shear stress (Daugherty et al. 1989, chap.8):

$$\tau_{zy} = \mu \cdot \frac{du}{dy} + \eta \cdot \frac{du}{dy} [\text{Pa}] \quad (3.11)$$

in which μ (Pa·s) is viscosity and η is the eddy viscosity. The laminar shear stress is included in the energy equation (eq-3.8) and is solved by the Navier Stokes equations. Turbulent shear stress is the result of a fluid particle in motion which

changes translational speed rapidly both in magnitude and direction and interacts with adjacent fluid particles and results in small eddies. Subsequently the small eddies exerts a shear force on adjacent fluid and is called eddy viscosity. The eddy viscosity for a turbulent flow of a steady incompressible fluid k and ϵ is described as following in the $k - \epsilon$ modell:

$$\mu_t = C\rho\vartheta\ell = \rho C_\mu \frac{k^2}{\epsilon} \quad (3.12)$$

where k and ϵ are used to define the velocity scale ϑ and the length scale ℓ :

$$\vartheta = k^{\frac{1}{2}} \quad \ell = \frac{k^{\frac{3}{2}}}{\epsilon} \quad (3.13)$$

in which ϵ is the rate of dissipation of kinetic energy and k is the kinetic energy. The turbulent length scale ℓ describes the size of the small large eddies. In the Wilcox “ $k - \omega$ ”-model proposed the eddy viscosity is described

$$\mu_t = \rho \frac{k}{\omega} [Pa \cdot s] \quad (3.14)$$

in which $\omega [s^{-1}]$ is the turbulence frequency:

$$\omega = \frac{\epsilon}{k}$$

The length scale ℓ and ω are thus related:

$$\ell = \sqrt{\frac{k}{\omega}}$$

Subsequently the scalars “ $k - \epsilon$ ” and “ $k - \omega$ ” interact with the RANS equations in the “ $k - \epsilon$ ” and “ $k - \omega$ ”-models. In this thesis four different turbulence models will be used in addition to the forest model; standard “ $k - \epsilon$ ”, “ $k - \epsilon$ ” with YAP correction, Modified and RNG “ $k - \epsilon$ ” (“Turbulence models in Phoenics” 2008).

3.3 Forest model

The canopy model that is to be assessed in this thesis is described by the canopy height (h_c), roughness length (z_0), porosity (β) and the drag coefficients C_1 and C_2 which will be described in this section.

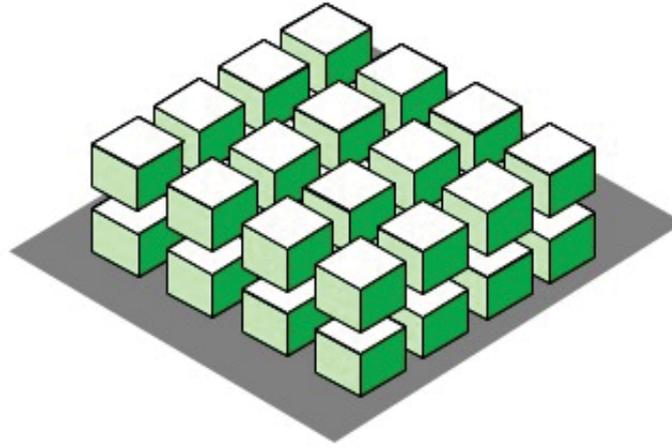


Figure 3.2: The structure of the porosity of a canopy model (Crasto 2005).

Darcy's law

The porosity of a volume is the volume of a fluid occupying a solid phase and is formulated mathematically:

$$\beta = \frac{\text{Volume of fluid phase}}{\text{Volume of solid phase}} \quad (3.15)$$

In this context the volume of the fluid phase is uniformly distributed in the canopy. The initial inflow in the grid is horizontal and thus the porosity in two dimensions should also be considered:

$$\beta_s = \frac{\text{area of holes}}{\text{total area}} \quad (3.16)$$

The structure in figure 3.2 gives a relation between β and β_s (Crasto 2005):

$$\beta_s = 1 - (1 - \beta)^{\frac{2}{3}} \quad (3.17)$$

By using the Darcy velocity that is valid for saturated, laminar porous media flow where inertial forces (Dingman 2002, chap.2), the fluid velocity can be deduced:

$$U_d = \beta \cdot U \quad (3.18)$$

where U_d [m/s] is the Darcy velocity and U [m/s] is the fluid velocity by not using the canopy model. A source term for the momentum equation (see eq-3.5) inside the canopy can now be introduced:

$$S_U = C_v U \quad \left[\frac{\text{N}}{\text{m}^3} \right] \quad (3.19)$$

and

$$C_v = \frac{\mu}{\kappa} = \frac{\nabla P}{U_d}$$

in which $\nabla P[\text{Pa/m}]$ is the pressure gradient (Crasto 2005), κ is the permeability tensor and is formulated as:

$$\kappa = \frac{nu}{C_1} = C \frac{\beta^2}{1 - \beta^2} \quad (3.20)$$

in which $C = 0,00450799[\text{m}^2]$ and C_1 is the drag coefficient in the term of viscous force.

Roughness length

The roughness length $z_0[\text{m}]$ is defined as the height corresponding when the mean velocity equals to zero in a neutral boundary layer and momentum equals to zero. The roughness length is a measure of surface roughness and varies according to the physical property of the surface (Arya 1988, chap.10) and is described in the logarithmic velocity profile law:

$$\frac{\bar{u}}{u_*} = \frac{1}{k} \cdot \ln \frac{z}{z_0} \quad (3.21)$$

where $\bar{u} [\text{m/s}]$ is the mean velocity, k is the von Karman constant (0,435), z is the height above ground level and

$$u_*^2 = \left| \overline{u'w'} \right| \quad \left[\frac{\text{m}^2}{\text{s}^2} \right]$$

in which $u_* [\text{m/s}]$ is the friction velocity and $\left| \overline{u'w'} \right|$ is the kinematic turbulent momentum flux (Holton 2004).

Canopy friction

The canopy friction is the drag force assigned to the top layer of the cells representing the forest and reduces the mean velocity on the surface. Hence the canopy friction is a source term in the momentum equation (see eq-??):

$$S_u = \underbrace{-\rho C_1 \vec{U}}_{\text{viscous force}} - \underbrace{\rho C_2 |\vec{U}| \vec{U}}_{\text{pressure force}} \quad \left[\frac{\text{N}}{\text{m}^3} \right] \quad (3.22)$$

in which $\rho[\frac{\text{kg}}{\text{m}^3}]$ is density of the continuum, $C_1[\frac{1}{\text{s}}]$ and C_2 are the drag coefficients. For high Reynold numbers the viscous force is negligible.

Canopy turbulence source

The source terms for canopy turbulence added to the transport equation for kinetic energy k (eq-3.13):

$$S_k = C_2(\beta_p |U|^3 - \beta_D |U| k) \quad (3.23)$$

and the canopy turbulence added to the transport for rate of dissipation of kinetic energy ϵ (eq-3.13):

$$S_\epsilon = C_2(C_{\epsilon 4} \beta_p \cdot \frac{\epsilon}{k} \cdot |U|^3 - C_{\epsilon 5} \beta_D |U| \epsilon) \quad (3.24)$$

where $\beta_p = 1.0$, $\beta_D = 6,51$, $C_{\epsilon 4} = C_{\epsilon 5} = 1,24$. The source terms are inspired by the canopy model in the articles written by Katul et al. (Katul et al. 2004) and Sanz (Sanz 2003). The constants has been adjusted according to the default set of constants of the standard “k- ϵ ” turbulence model.

Methodology

The process of validating a forest model is a two-step procedure; firstly select a case and secondly reproduce the case with the use of the forest model. Therefore the case selection and the validation will follow two separate methodologies and will be described in this chapter.

4.1 Software

The integral software to complete this thesis is Python (version 2.7.10), WindSim (version 7.0.0), Geomatica 2016 and QGIS (version 2.14.2 for mac). Python is used to develop the case selection module (section 4.2, Langtangen 2011). The forest model that is to be validated is a part of WindSim (section 3). Geomatica is used to make a digital elevation model (DEM) from contour lines (section 2.1). QGIS is essential to interpret geographical information in different datums and to make maps for instance (section 2.2 and 2.1).

4.2 Case selection

The ideal case

A suitable case for validation of the forest model is retrieved from an air flow between two meteorological masts. The validation case needs to be unaffected by vertical forcing which implies an air flow in a neutral planetary boundary layer. To reduce the terrain effects on the air flow, the terrain between the masts should be as flat as possible. Air flow over a hill top induces speed up and requires additional

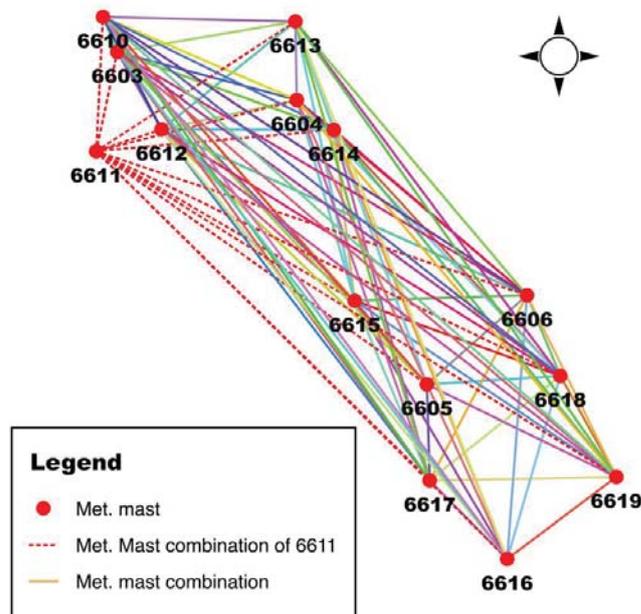


Figure 4.1: Web of combinations of 14 meteorological masts and 2 possibilities.

microscale refinement and deviates significantly from measurements and is a source of uncertainty for forest modeling. If the terrain induces a multitude of speed up the deviation can magnify (??). To enhance the forest effect one meteorological mast should be in open landscape and the other should be nearby a forest (A. Gravdahl 2016).

The case selection module

In order to find the ideal case among 1,3 million observations a search method needs to be developed which is the motivation to develop a case selection module (see appendix A).

At the project sites there are fourteen meteorological masts (the data is described in section 2.1) with two combinations with no withdrawal gives

$${}_{14}C_2 = \frac{14!}{2! \cdot (14-2)!} = 91 \quad (4.1)$$

The 91 combinations of lines are visualized in figure 4.1 where the combinations of station 6611 have been highlighted. The list of combinations is an essential part of the case selection module. In table 4.1 an exempt of the list of combinations is viewed. From the UTM-coordinates the orientation of the line between the meteorological masts is found. The method used to find the orientation is the cosine

Table 4.1: An exempt of the table of the total combinations used in the case selection module with the focus on station 6611.

	Point 1			Point 2			Orientation°	
	Station 1	Longitude	Latitude	Station 2	Longitude	Latitude		
1	6611	539321	7039682	6603	540090	7043583	11	191
2	6611	539321	7039682	6604	547093	7041796	75	255
3	6611	539321	7039682	6605	552333	7030718	125	305
4	6611	539321	7039682	6606	556179	7034280	108	288
5	6611	539321	7039682	6610	539518	7044947	2	182
6	6611	539321	7039682	6614	548555	7040645	84	264
7	6611	539321	7039682	6615	549464	7033958	119	299
8	6611	539321	7039682	6616	555587	7023898	134	314
9	6611	539321	7039682	6617	552508	7026941	134	314
10	6611	539321	7039682	6618	557531	7031142	115	295
11	6611	539321	7039682	6619	559795	7027207	121	301
12	6612	541850	7040574	6611	539321	7039682	71	251
13	6613	546995	7044869	6611	539321	7039682	236	56

rule:

$$\theta = \cos^{-1} \left(\frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}|} \right) \quad (4.2)$$

where \mathbf{u} is the vector spanning the coordinates in the combinations and \mathbf{v} is the north vector $[0, 1]$.

On average there are measured 33357 measurements at every meteorological mast in the observation period with the following possibilities:

$$91 \cdot 33357 \cdot (33357 - 1) = 101\,251\,704\,372 \approx 101 \cdot 10^9 \quad (4.3)$$

The suitable cases for validation of the forest model are among $101 \cdot 10^9$ possibilities where every criteria reduces the amount of possibilities. The reduction of the large number of possibilities — the filtration — is therefore suitable for computation that is used a method in this thesis. When validation of the forest model the wind speed measured at one meteorological mast is used for convergence criteria and another for calibration. The ideal case is when the orientation of the air flow at both meteorological stations is in the same direction with a speed is above 5m/s in a neutral boundary layer at the same time.

Neutral boundary layer

The neutral atmospheric stability criteria is essential because of the logarithmic profile law (equation 3.21). The logarithmic profile law describes a vertical wind

profile in a neutral planetary boundary layer and is implemented in the transport equations solved by PHOENICS. Neutral atmospheric stability is a weather condition when there is no vertical acceleration of a parcel of air and is one of three stability conditions Arya 1988:

1. Unstable, $s < 0$, $\frac{\partial \Theta_v}{\partial z} < 0$, or $\frac{\partial T_v}{\partial z} < -\Gamma$
2. Neutral, $s = 0$, $\frac{\partial \Theta_v}{\partial z} = 0$, or $\frac{\partial T_v}{\partial z} = -\Gamma$
3. Stable, $s > 0$, $\frac{\partial \Theta_v}{\partial z} > 0$, or $\frac{\partial T_v}{\partial z} > -\Gamma$

where

- s : static stability parameter
- Θ_v : virtual potential temperature
- T_v : virtual temperature
- z : height above ground level
- Γ : adiabatic lapse rate

The criteria for the different conditions of atmospheric stability refers to the term buoyant acceleration:

$$a_b \simeq -\frac{g}{T_v} \left(\frac{\partial T_v}{\partial z} + \Gamma \right) \Delta z = -\frac{g}{T_v} \frac{\partial \theta_v}{\partial z} \Delta z \quad (4.4)$$

The static stability parameter is the term $-\frac{g}{T_v} \frac{\partial \theta_v}{\partial z}$ and Δz is the displacement of an air parcel from the equilibrium. The criteria used for determining the atmospheric stability of a measurement in this thesis is:

$$a_b = 0, \quad \text{when } \frac{\partial T_v}{\partial z} = -\Gamma \quad \left[\frac{m}{s^2} \right] \quad (4.5)$$

hence determining the virtual temperature (T_v) and the lapse rate (Γ) are integral part when ascertaining the atmospheric stability of the measurements in this thesis.

Adiabatic lapse rate

The adiabatic lapse rate is the rate the temperature of an air parcel changes with respect to height and depicts an adiabatic process. In an adiabatic process an air parcel does not exchange heat with the surrounding air. When considering firstly a vertical upward movement of an air parcel the parcel expands because of a reduction of pressure —*adiabatic expansion*. Secondly the parcel compresses and therefore moves towards ground level without changing the temperature —*isothermal compression*. Finally the parcel heats at the same measure of pressure at ground level —*isobaric heating*. The movement of the air parcel constitutes a thermodynamic cycle.

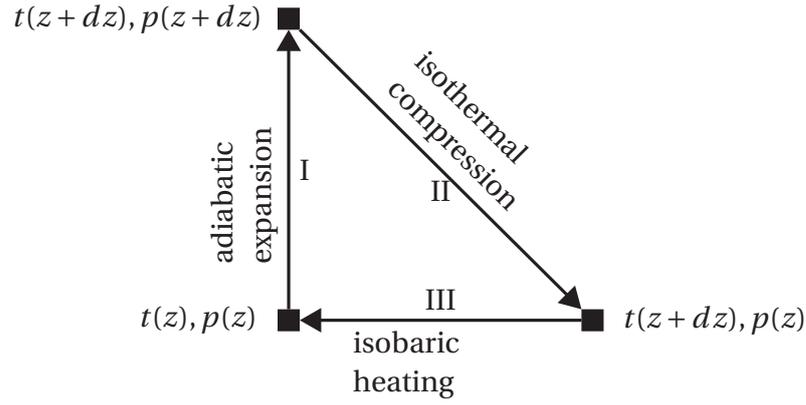


Figure 4.2: The thermodynamic cycle of an air parcel (Jacob 1999). The roman letter refers to each part of the cycle. t is temperature and p is pressure.

The adiabatic lapse rate is therefore quickly derived by considering the enthalpy (H) —the energy content of the air parcel. The thermodynamic cycle implies that,

$$\Delta H_I + \Delta H_{II} + \Delta H_{III} = 0 \quad (4.6)$$

which states that there is no net change of enthalpy in the thermodynamic cycle. Subsequently,

$$V dp = m C_p dT \quad (4.7)$$

where V [m^3] is the volume, m [kg] is the mass, C_p [J/K] is the specific heat capacity at constant pressure of an air parcel (for full derivation see Jacob 1999, p.56-57). The vertical pressure gradient is defined as:

$$\frac{dp}{dz} = -\rho_a g \quad (4.8)$$

where ρ_a is the density of an air parcel and g is the gravitational acceleration constant. By inserting the equation for the pressure gradient in equation eq: lapse rate and the similarity $\rho V = m$,

$$V(-\rho_a g dz) = \rho V C_p dT \quad (4.9)$$

consequently the adiabatic lapse rate is defined (for full derivation see Jacob 1999, p.56-57):

$$\Gamma = \frac{g}{C_p} = -\frac{dT}{dz} \quad \left[\frac{\text{K}}{\text{m}} \right] \quad (4.10)$$

In this thesis the gravitational acceleration constant $9,80665 \frac{m}{s^2}$ is used (Finnemore et al. 2009, appendix), but the specific heat capacity at constant pressure is yet to investigate.

Specific heat capacity — C_p

The specific heat capacity at constant pressure is a property of a substance related to the temperature difference resulting from the absorption of heat energy Atkins et al. 2009:

$$C_p = \left(\frac{\partial h}{\partial T} \right)_p \quad (4.11)$$

where h is the specific enthalpy.

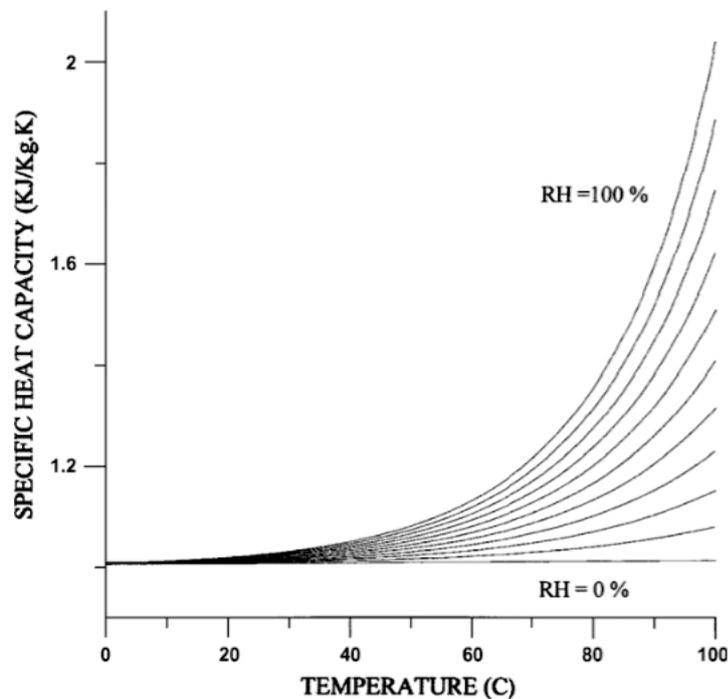


Figure 4.3: The specific heat capacity as a function of relative temperature, relative humidity plotted (Tsilingiris 2008).

Figure 4.3 visualizes the equation for specific heat capacity formulated by Tsilingiris. The equation derives from a polynomial fit of tabulated values for heat capacity for dry air and air saturated by water as relative humidity related to temperature and of temperature. In figure 4.3 the equation for C_p is nearly horizontal in the temperature range of interest in this thesis ($-12 \text{ }^\circ\text{C} - 25,4 \text{ }^\circ\text{C}$ at station 6604) as representative case) regarding both dry ($\text{RH}=0\%$) and saturated ($\text{RH}=100\%$) air which implies convergence towards a constant value.

Table 4.2: The values of C_p derived from linear interpolation between two values of the specific heat enthalpy of dry air as ideal gas property at 1 bar pressure retrieved from (Wyllen et al. 1994).

n	h (J/kg)	t (°C)	$C_{pn+1} = \frac{h_{n+1} - h_n}{T_{n+1} - T_n}$ (J/kgK)
1	2400267	240	—
2	260323	260	1002,8
3	280390	280	1003,5
4	290430	290	1004
5	298615	298,15	1004,3
6	300473	300	1004,3
7	320576	320	1005,2

Table 4.3: Results of the regression analysis of the enthalpy related to temperature with the python function `scipy.stats.linregress`.

Slope	Intercept	r-value	p-value	Standard error
1003.78597374	-657.58338349	0.999999875317	1.05430995384e-17	0.224168227702

By performing a linear regression of the values of enthalpy for dry air in the temperature range of interest the following equation is generated:

$$h(T) = 1003,78597374 \cdot T - 657,58338349 \quad (4.12)$$

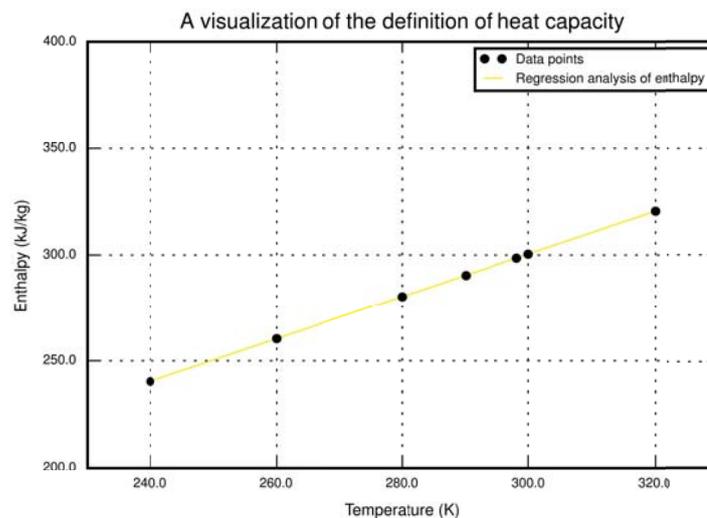


Figure 4.4: A visualization of the regression analysis of tabulated data points for specific enthalpy related to temperature with a validity from $-33,15^{\circ}\text{C}$ to $+46,15^{\circ}\text{C}$.

which is valid between $-33,15^{\circ}\text{C}$ and $+46,85^{\circ}\text{C}$ and is visualized in figure 4.4. An r -value of 1 implies that every point on the regression line represents the real value with 100%. The converging curves a constant value in the temperatures of interest implies a linear relation between enthalpy and temperature for both dry and saturated air. By derivative of function 4.12 gives the specific heat capacity:

$$C_p = \left(\frac{\partial h}{\partial T} \right)_p = 1003,78597374 \quad (4.13)$$

that is the value of C_p used in this thesis.

Virtual temperature

The virtual temperature is the temperature a moist air of parcel would if it was dry with the same pressure and density and is defined as:

$$T_v = T \left[1 + \left(\frac{m_d}{m_w} - 1 \right) Q \right] = T(1 + 0,61Q) \quad (4.14)$$

in which m_d and m_w are the mean molecular masses of water and dry air and Q is the specific humidity.

Specific humidity

The specific humidity is the mixing ration between water content and moist air and is defined as:

$$q \equiv \frac{M_w}{M_w + M_d} = \frac{\rho_v}{\rho_a} \simeq 0,622 \frac{e}{P} \quad \left[\frac{\text{kg}}{\text{kg}} \right] \quad (4.15)$$

where M_w is the molecular mass of the water in the air and $(M_w + M_d)$ is the mass of moist air. ρ_v and ρ_a are the densities of vapor and total air density, e is the partial pressure of water vapor and P is the ambient pressure. The latter approximate definition will be used in this thesis.

The vapor pressure of water in moist air is related to the relative humidity:

$$e = RH \cdot e^* \quad [\text{Pa}] \quad (4.16)$$

The ambient pressure is retrieved from the data sets provided by SMHI at 2 meters above ground level. The pressure in the planetary boundary layer reduced proportional to the hydrostatic formula, related to the pressure gradient (equation 4.8, Arya 1988):

$$dp = -\rho g dz \quad (4.17)$$

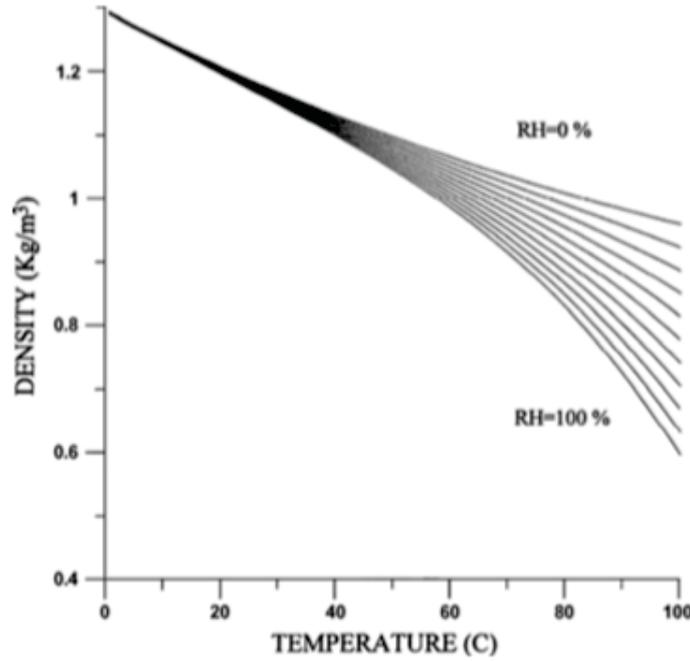


Figure 4.5: The visualization of the density formula (equation 4.19).

so that the density at the top of the masts will be:

$$\rho_{\text{top}} = \rho_{2m} - \rho g \Delta z \quad (4.18)$$

where Δz is the height difference between the top of the mast and $2m$ above ground level.

Density of air

The density of air (ρ) varies considerably with the temperature. Between 0°C and 100°C air saturated with water vapor (RH=100%) can be expressed by the formula (Tsilingiris 2008):

$$\rho_m = 1,293393662 - 5,538444326 \cdot 10^{-3} \cdot t + 3,860201577 \cdot 10^{-5} \cdot t^2 - 5,2536065 \cdot 10^{-7} \cdot t^3 \quad (4.19)$$

In figure 4.5 the mathematical expressions for density of air of dry and saturated air nearly converges to a linear shape for temperatures below 35 degrees which is in the upper boundary of the temperature of interest. A further assumption is that the linear trend of density as a function of temperature continues for subzero temperatures. By including the observations $1,395 \frac{\text{kg}}{\text{m}^3}$ at -20°C and $1,515 \frac{\text{kg}}{\text{m}^3}$ at -40°C and 20 observations with equation 4.19 for temperatures below 35°C a linear regression can be performed.

Table 4.4: Results of the regression analysis of the density related to temperature with the python function `scipy.stats.linregress`. The *r*- and *p*-values and the standard error are truncated.

Slope	Intercept	r-value	p-value	Standard error
-0.0051504754610588429	1.296159304053393	-0.9987	3.88e-27	6.012e-05

The results from the linear regression in table 4.4 indicates that the function of density can be expressed with high certainty:

$$\rho(t) = 1,296159304053393 \cdot t - 0,0051504754610588429 \quad (4.20)$$

The values for slope and intercept are not rounded off because the r-value of -1 applies for the density equation without changing the number of decimals.

Saturation vapor pressure

The final feature to investigate in order to find the atmospheric stability and the specific humidity (equation 4.15) at each observation is the saturation vapor pressure —often denoted as e^* . Vapor pressure is the partial pressure a constituent in a gas mixture. The saturation vapor pressure of water in air can be regarded as the maximum concentration of water vapor that is thermodynamically stable. The saturation vapor pressure of water is proportional to the kinetic energy the water molecule (H_2O) have in the transition between the liquid or solid state to the gaseous state and is described in the well-known Clausius-Clapeyron equation (Atkins et al. 2009):

$$\ln e' = \ln e + \frac{\Delta_{vap}H}{R} \left(\frac{1}{T} - \frac{1}{T'} \right) \quad (4.21)$$

in which $\Delta_{vap}H$ is the vaporization enthalpy, the mark ref. The Clausius-Clapeyron equation lays the ground for several function saturation vapor pressure of water for both solid and liquid state, for example:

The Sonntag equation in the liquid state (Murphy et al. 2005):

$$\begin{aligned} \ln e_l^* = & 16,635764 - \frac{6096,9385}{T} - 2,711193 \cdot 10^{-2} \cdot T \\ & + 1,673952 \cdot 10^{-5} \cdot T^2 + 2,433502 \ln(T) \end{aligned} \quad (4.22)$$

The Buck equation for both solid and liquid state (Buck 1981):

$$e_{s,l}^* = 6,1121 \cdot \exp \left(\left(18,678 - \frac{t}{234,5} \right) \cdot \left(\frac{t}{257,14 + t} \right) \right) \quad (4.23)$$

in which the vapor pressure is in hPa and t is in °C.

The simplified Goff-Gratch equation (Dingman 2002, appendix):

$$e_{s,1}^* = 0,611 \cdot \exp\left(\frac{17,3 \cdot t}{t + 237,3}\right) \quad (4.24)$$

where t is in °C. The mean squared error is a method to compare the three formulas that are expected to give results with small differences (Devore et al. 2012):

$$\text{mean squared error} = \frac{1}{n} \sum_{i=1}^n (y_{1,i} - y_{2,i})^2 \quad (4.25)$$

in which n is the sample number and y is a class of samples. By using 1000 points between -30°C and 30°C the mean squared errors are 0,34 between the Sonntag equation and the Buck equation, 6,71 between Buck and the simplified equation and 4,12 between the Sonntag equation and the simplified equation. The comparisons with the simplified equation produces the most error and therefore the Buck and Sonntag are most reliable. In this thesis the Sonntag equation is used.

Governing equations

The governing equations in the case selection module to find the atmospheric stability in function `Static_para_array` of the observations are:

$$Q_{2m} = 0,622 \cdot RH \cdot \frac{e^*}{p_{2m}} \quad (4.26)$$

$$\rho(t_{\text{top}}) = -0.005150475461058842 \cdot t_{\text{top}} + 1.296159304053393$$

$$p_{\text{top}} = p_{\text{top}} - \rho \cdot g \cdot z_{\text{top}}$$

$$Q_{\text{top}} = 0,622 \cdot RH \cdot \frac{e^*}{p_{\text{top}}} \quad (4.27)$$

So the stability parameter can be computed:

$$\frac{\partial T_v}{\partial z} = \frac{t_{2m} \cdot (1 + 0,61 \cdot Q_{2m}) - (t_{\text{top}} \cdot (1 + 0,61 \cdot Q_{\text{top}}))}{z_{2m} - z_{\text{top}}} \quad (4.28)$$

There are two assumptions in the computation of equation 4.28; firstly the relative humidity is uniformly distributed in the planetary boundary layer between the four project sites and Krångede A and secondly the relative humidity does not change within the hour. The data set of relative humidity and pressure are therefore synchronized with the SSVAB data sets.

Case selection module dismantled

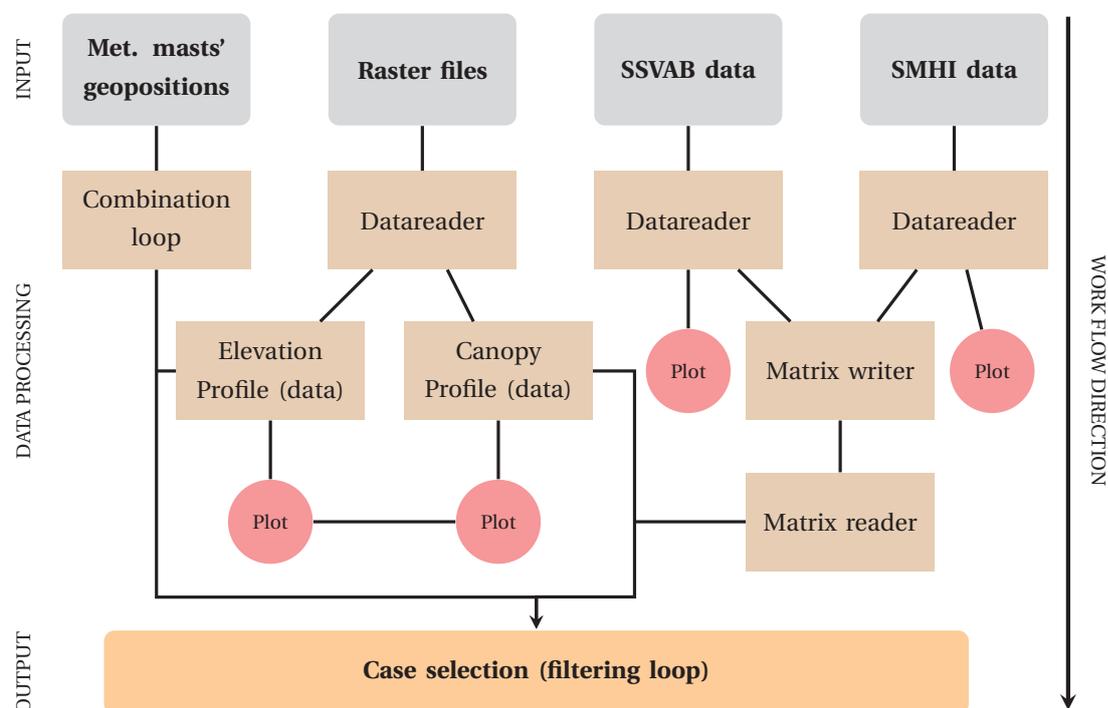


Figure 4.6: The structure of the case selection module that is used to find a case to perform the validation of the forest model (see A for the essential source code).

4.3 Validation

The suitable observations that matches the criteria set in the case selection module will be compared with simulated values by evaluating the profiles of vertical distribution of wind speed. The canopy turbulence will be tested against different turbulence models. The turbulence models are “standard $k-\epsilon$ ”, “RNG- $k-\epsilon$ ”, “YAP-modified $k-\epsilon$ ” and a modified turbulence model. In order to see the effect of the canopy model wind profiles with incremental “C2”-values will be compared with a zero baseline —where the model is disregarded. The gridded forest will be assessed in light of landcover data from Corine and canopy profile with data from the Swedish University of Agricultural Sciences (SLU). The digital elevation model generated in the simulation will be evaluated with the digital elevation model generated in relation to this thesis (see 2.2b).

Results

In this chapter the results from the case selection and the simulated values compared with the measurements are presented.

5.1 Case selection

When considering wind speed from the same direction on two masts and the distance between them case 1,2,5 and 6 are the most suitable in table 5.1 and 5.2. However the longer the distance is between the two meteorological masts the higher the probability is that other effects can affect the wind flow between the masts.

When disregarding the neutral atmospheric stability and choosing among the pairs of meteorological masts with the shortest distance which are listed in table 5.3.

Table 5.1: The 4 of 9 results from the case selection loop, sorted by distance. The number in the parenthesis refers to the mast number. The letter “z” and “s” symbols forest height and stability parameter.

Case	1	2	3	4
Tolerance (s)	0,1	0,15	0,1	0,15
Tolerance (wd)	0	0	10	0
Mast 1	6605	6618	6605	6610
Mast 2	6615	6619	6618	6611
Date (1)	05-05-09 21:30	28-04-09 20:20	20-11-09 14:10	29-04-09 04:20
Date (2)	05-05-09 21:30	28-04-09 20:20	20-11-09 14:10	29-04-09 04:20
Static stability parameter (1)	-0,00896	-0,00842	-0,00896	-0,00842
Static stability parameter (2)	-0,00948	-0,00843	-0,00947	-0,00948
Wind direction (tol1)	310	342	181	338
Wind direction (tol2)	318	330	255	2
Wind direction (1)	138	150	85	2
Wind direction (2)	318	330	265	182
Wind speed (1)	5,02	8,05	4,81	5,77
Wind speed (2)	5,23	7,94	8,02	8,65
Forest height (1)	0	8	0	15
Forest height (2)	12	6	8	0
Standard deviation (z)	6,59	4,35	5,42	7,32
Distance (km)	4,33	4,54	5,22	5,27

Table 5.2: The last of 9 results from the case selection loop, sorted by distance. The number in the parenthesis refers to the mast number. The letter “z” symbols forest height and stability parameter.

Case	5	6	7	8	9
Tolerance (s)	0,1	0,1	0,2	0,15	0,15
Tolerance (wd)	10	2,5	2,5	7,5	7,5
Mast 1	6605	6615	6611	6611	6611
Mast 2	6619	6618	6613	6617	6619
Date (1)	16-03-09 20:00	01-05-09 05:30	28-02-09 01:20	03-03-09 16:20	25-02-09 21:40
Date (2)	16-03-09 20:00	01-05-09 05:30	28-02-09 01:20	03-03-09 16:20	25-02-09 21:40
Static stability parameter (1)	-0,01074	-0,0095	-0,01051	-0,01052	-0,00946
Static stability parameter (2)	-0,00947	-0,01055	-0,01156	-0,00841	-0,00841
Wind direction (tol1)	269	277	52	171	129
Wind direction (tol2)	285	290	58	131	128
Wind direction (1)	295	109	236	134	121
Wind direction (2)	115	289	56	314	301
Wind speed (1)	7,22	6,56	8,24	3,01	6,78
Wind speed (2)	9,76	6,86	5,02	2,06	6,06
Forest height (1)	0	12	0	0	0
Forest height (2)	6	8	15	16	6
Standard deviation (z)	5,66	6,63	6,77	6,41	6,16
Distance (km)	8,25	8,55	9,27	18,34	23,98

Table 5.3: The seven closest pairs of meteorological masts.

Combination	Met Mast 1	Met Mast 2	Distance between Mast 1&2 (km)	Elevation difference (1-2)(m)
1	6610	6603	1,5	-1
2	6604	6614	1,9	-46
3	6612	6611	2,7	-34
4	6613	6604	3,1	-57
5	6606	6618	3,4	-90
6	6612	6603	3,5	-37
7	6617	6605	3,8	-33

5.2 Case 1

Velocity data

Table 5.4: The speed data in case 1.

6605				6615			
z (m)	Speed(m/s)	Std(m/s)	Dir (°)	z (m)	Speed(m/s)	Std(m/s)	Dir (°)
59,0	5,02	0,26	-	100,7	5,23	0,33	-
59,0	5,06	0,26	-	100,7	5,25	0,33	-
57,5	-	-	310	96,4	5,17	0,32	-
57,0	4,92	0,27	-	97,7	-	-	324
55,3	-	-	330	97,6	-	-	318
44,5	4,47	0,34	-	96,4	5,17	0,32	-
31,50	3,61	0,4	-	80,7	4,89	0,53	-
				57,8	5,08	0,35	-

(a) (b)

The velocity data for case 1 are described in tables 5.4b and ???. “z”, “Std” and “Dir” is height above ground level, standard deviation and speed direction.

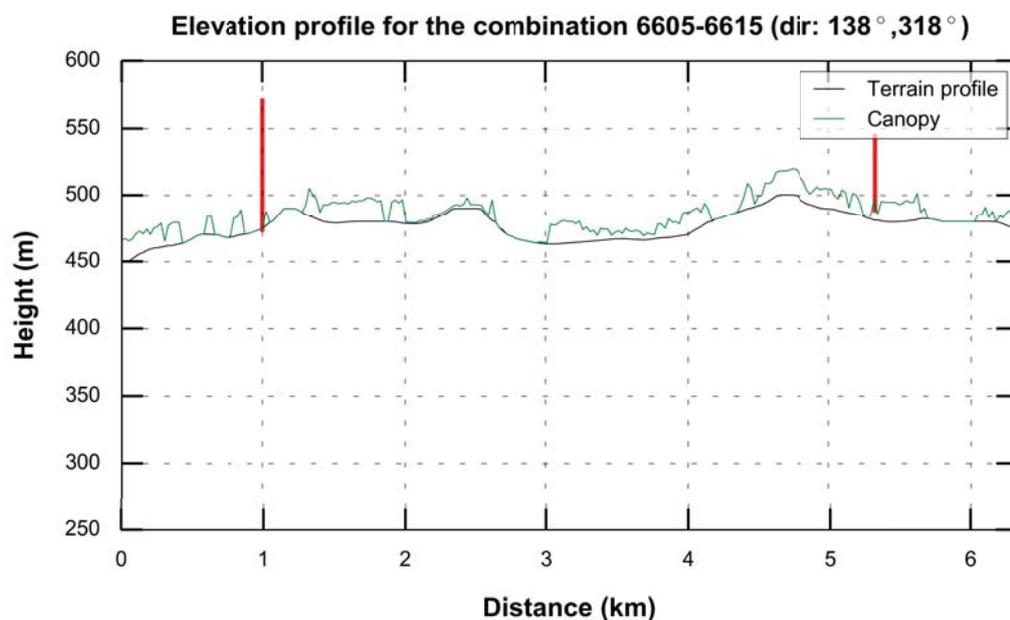
Terrain

Flow model

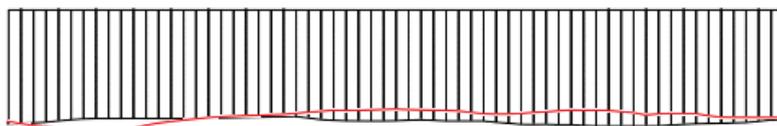
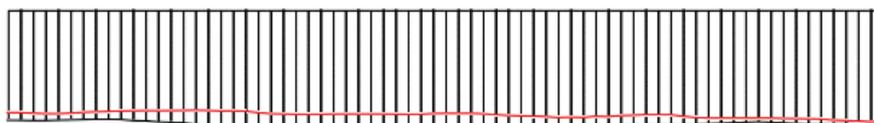
Grid

Table 5.5: The grid data for the for simulation in the area surrounding met. masts 6605 and 6615.

	x	y	z	total
Grid spacing (m)	100	100	-	
Number of cells	62	70	20	86800



(a) Terrain and canopy profile.



(b) Digital Elevation Model of the simulation.

Figure 5.1: The top figure shows elevation and canopy profile of the terrain between the masts 6615 and 6605 with data from SLU (SLU Forest Map 2010). The red pole is the meteorological masts with the corresponding height. The bottom figure shows the profile of the grid.

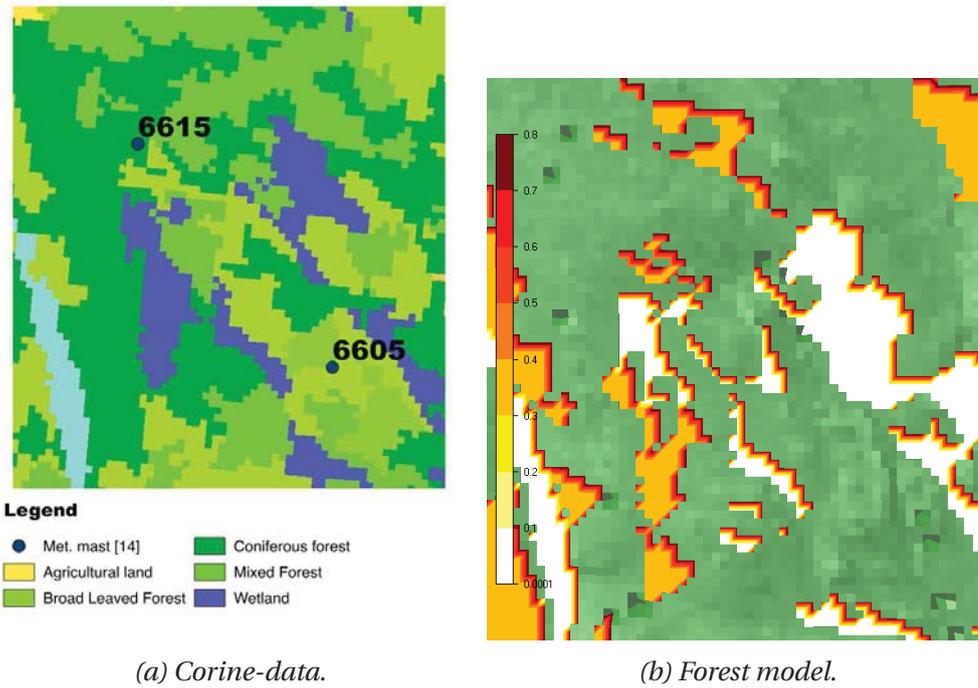


Figure 5.2: A comparison of corresponding two forest models.

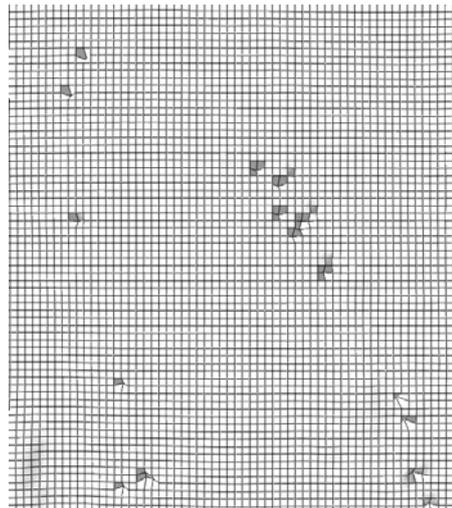


Figure 5.3: The grid of the flow model.

Vertical speed distribution

standard “k- ϵ ”

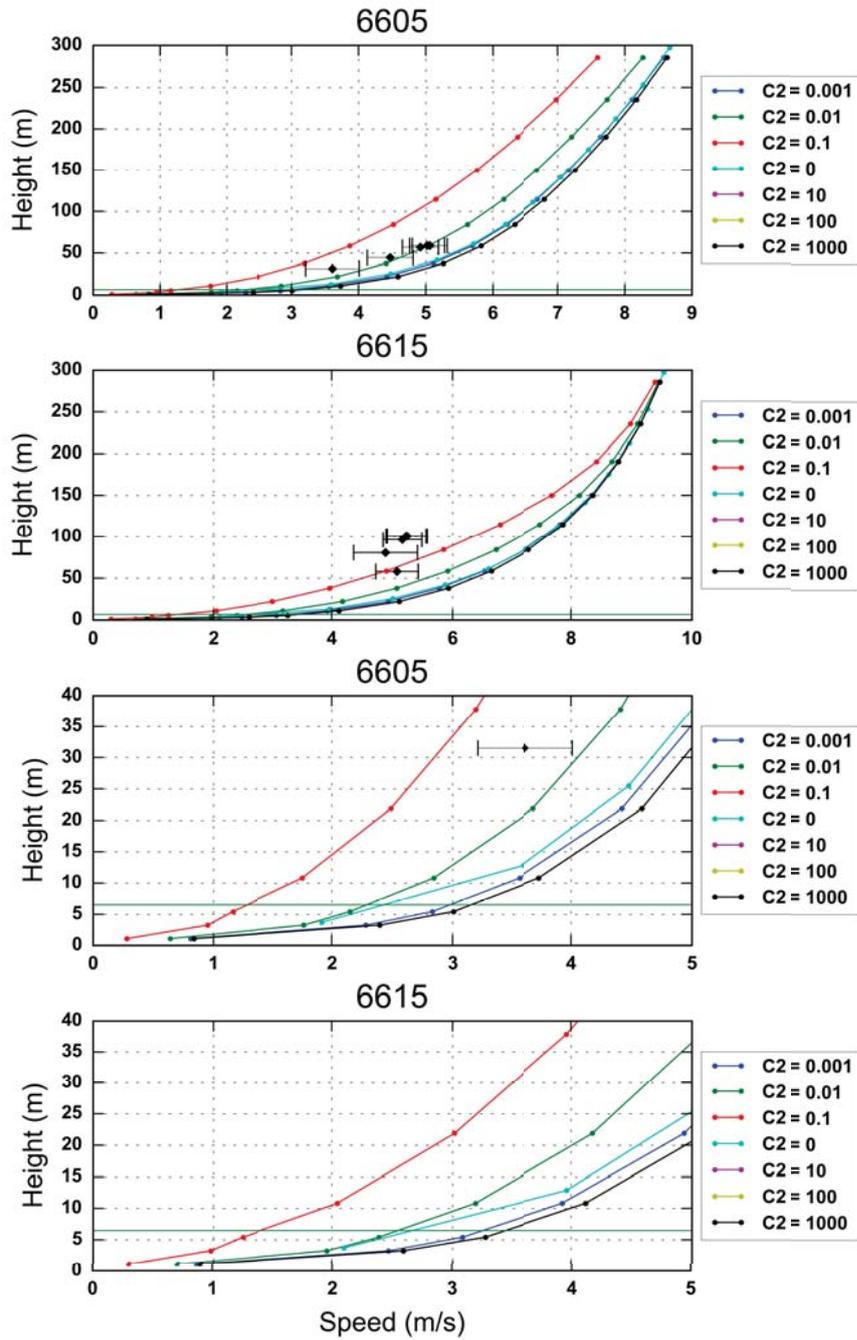


Figure 5.4: Comparison between measured and simulated data between the met. stations 6605 and 6615. The turbulence model standard “k- ϵ ”-model has been used. The green horizontal line is the canopy height.

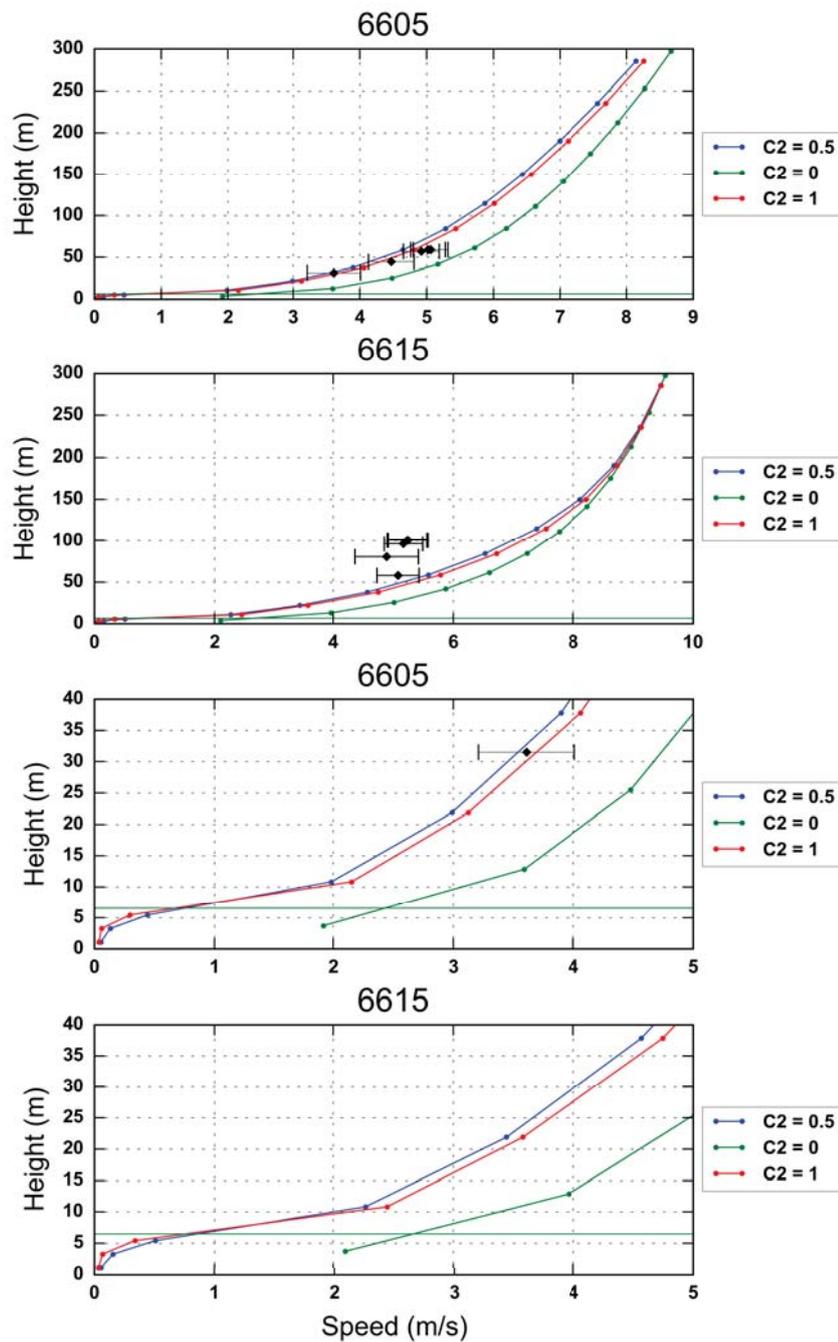
YAP-modified “k- ϵ ”

Figure 5.5: Comparison between measured and simulated data between the met. stations 6605 and 6615. The YAP-modified “k- ϵ ”-model has been used. The green horizontal line is the canopy height.

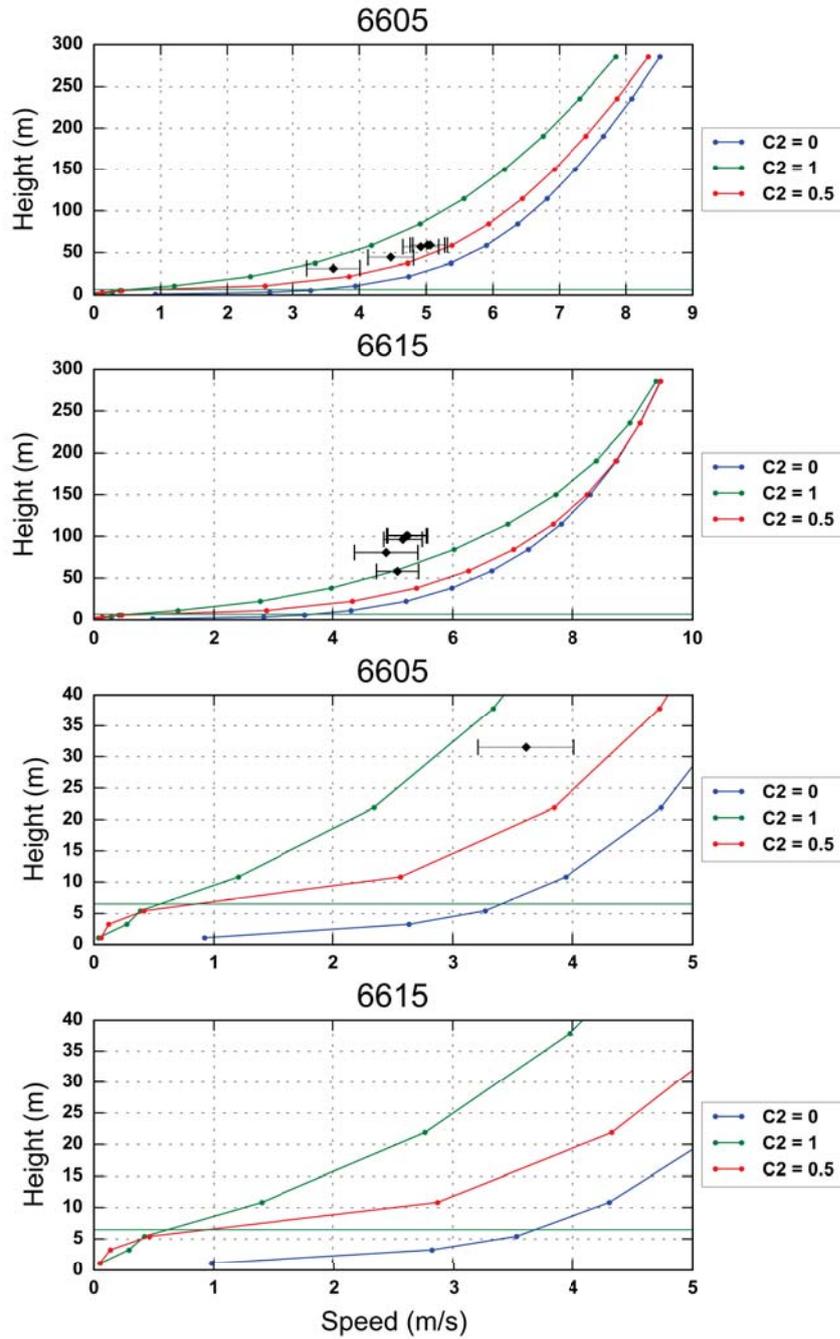
RNG- $k-\epsilon$ 

Figure 5.6: Comparison between measured and simulated data between the met. stations 6605 and 6615. The RNG“ $k-\epsilon$ ”-model has been used. The green horizontal line is the canopy height.

Modified

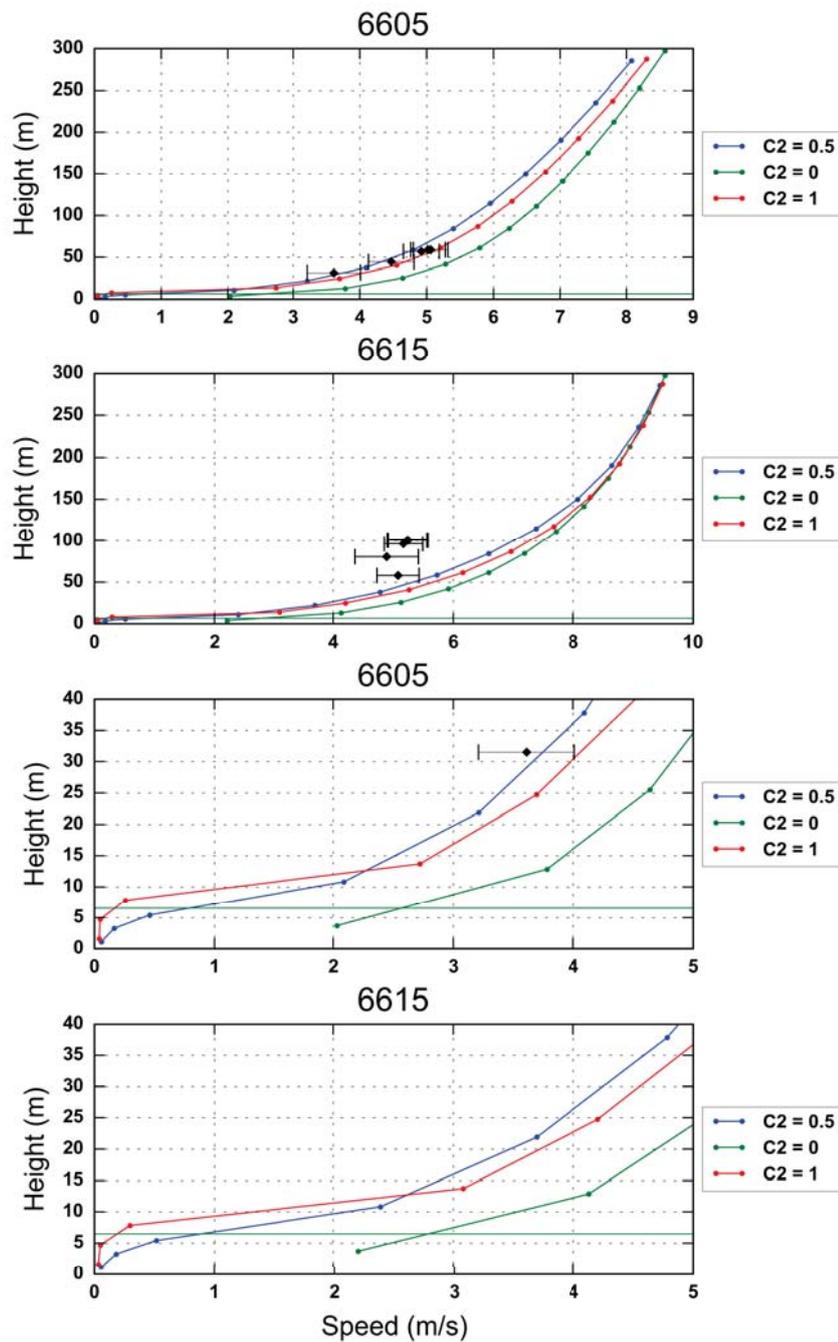


Figure 5.7: Comparison between measured and simulated data between the met. stations 6605 and 6615. The modified “ $k-\epsilon$ ”-model has been used. The green horizontal line is the canopy height.

5.3 Case 2

Velocity data

Table 5.6: The speed data in case 2.

6618				6619			
z (m)	Speed(m/s)	Std(m/s)	Dir (°)	z (m)	Speed(m/s)	Std(m/s)	Dir (°)
100,8	8,05	0,21	-	100,8	7,94	0,47	-
100,8	7,96	0,2	-	100,8	8,13	0,53	-
96,7	-	-	342	97,7	-	-	330
97,6	-	-	342	97,6	-	-	321
96,4	7,95	0,25	-	96,4	7,9	0,68	-
80,7	0,73	0,38	-	80,9	7,1	0,59	-
57,6	6,86	0,3	-	57,8	6,58	0,53	-

(a) (b)

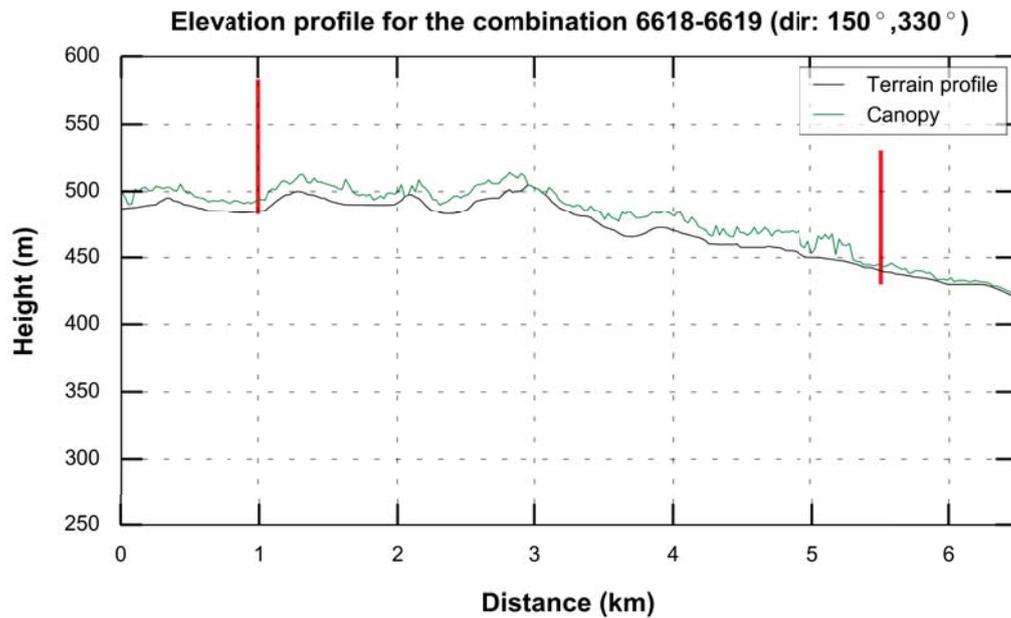
Terrain

Flow model

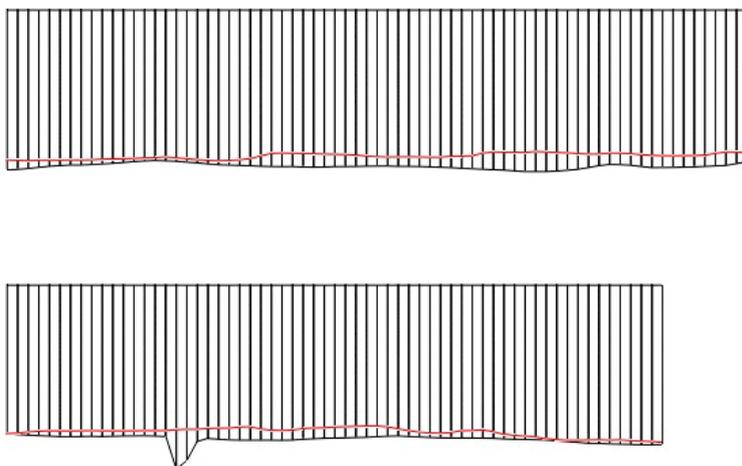
Grid

Table 5.7: The grid data for the for simulation in the area surrounding met. masts 6618 and 6619.

	x	y	z	total
Grid spacing (m)	96	96	-	
Number of cells	62	70	20	86800



(a) Terrain and canopy profile.



(b) Digital Elevation Model of the simulation.

Figure 5.8: The top figure shows elevation and canopy profile of the terrain between the masts 6618 and 6609 with data from SLU (SLU Forest Map 2010). The red pole is the meteorological masts with the corresponding height. The bottom figure shows the profile of the grid.

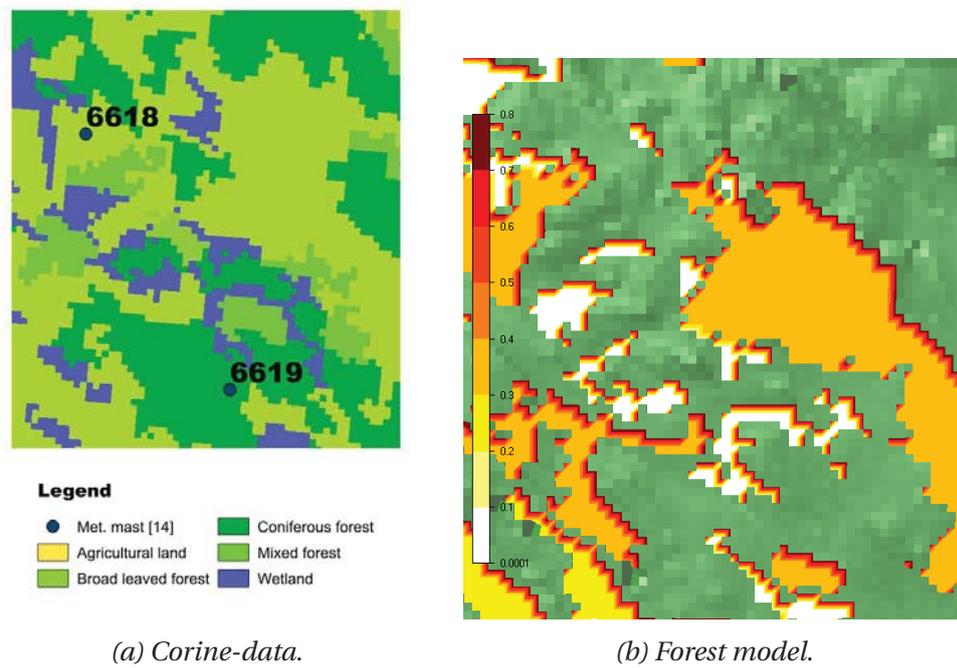


Figure 5.9: A comparison of two corresponding forest models.

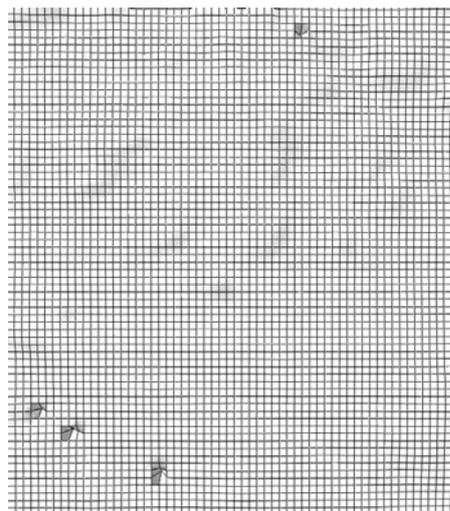


Figure 5.10: The grid of the flow model.

Vertical speed distribution

standard “k- ϵ ”

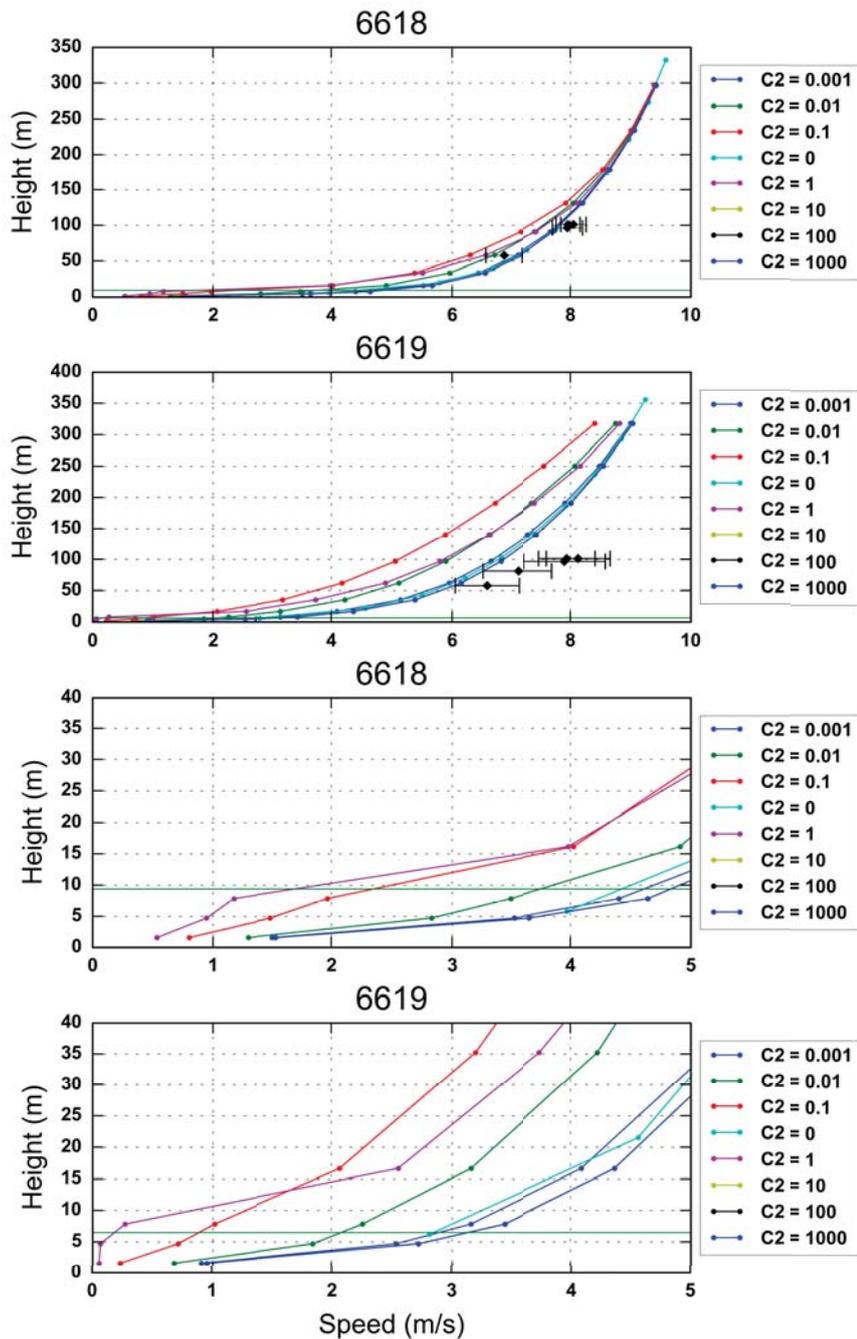


Figure 5.11: Comparison between measured and simulated data between the met. stations 6618 and 6619. The turbulence model standard “k- ϵ ”-model has been used.

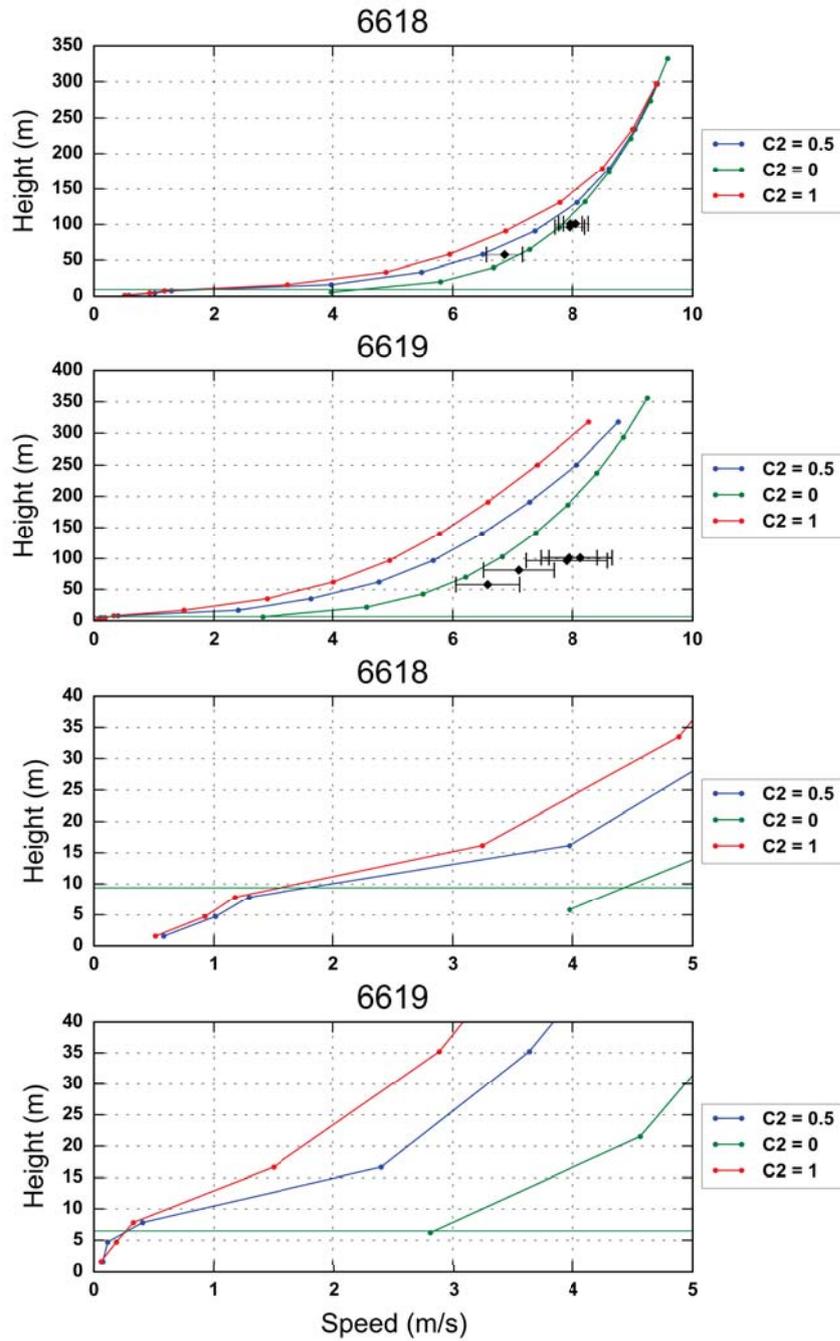
YAP-modified “k- ϵ ”

Figure 5.12: Comparison between measured and simulated data between the met. stations 6605 and 6615. The YAP-modified “k- ϵ ”-model has been used.

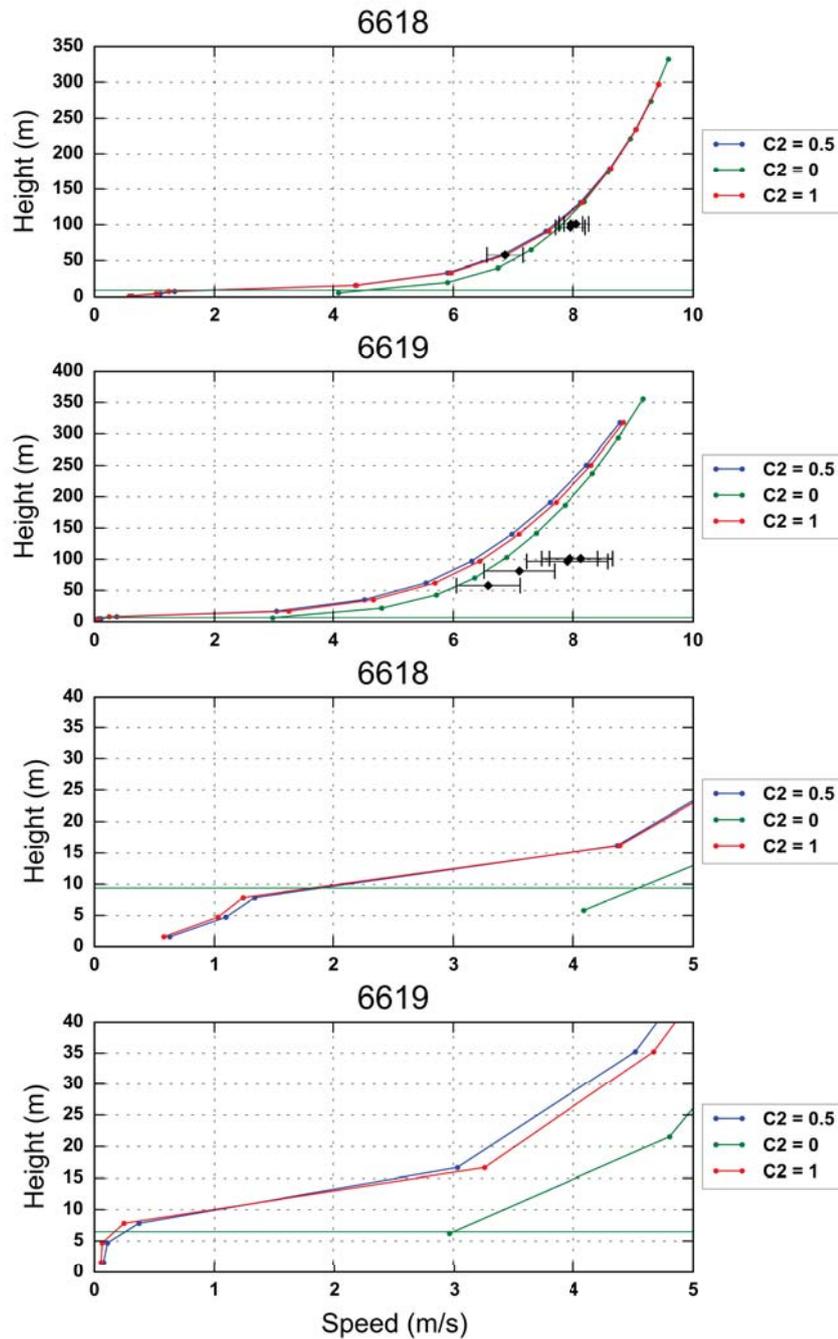
RNG- $k-\epsilon$ 

Figure 5.13: Comparison between measured and simulated data between the met. stations 6618 and 6619. The RNG “ $k-\epsilon$ ”-model has been used.

Modified

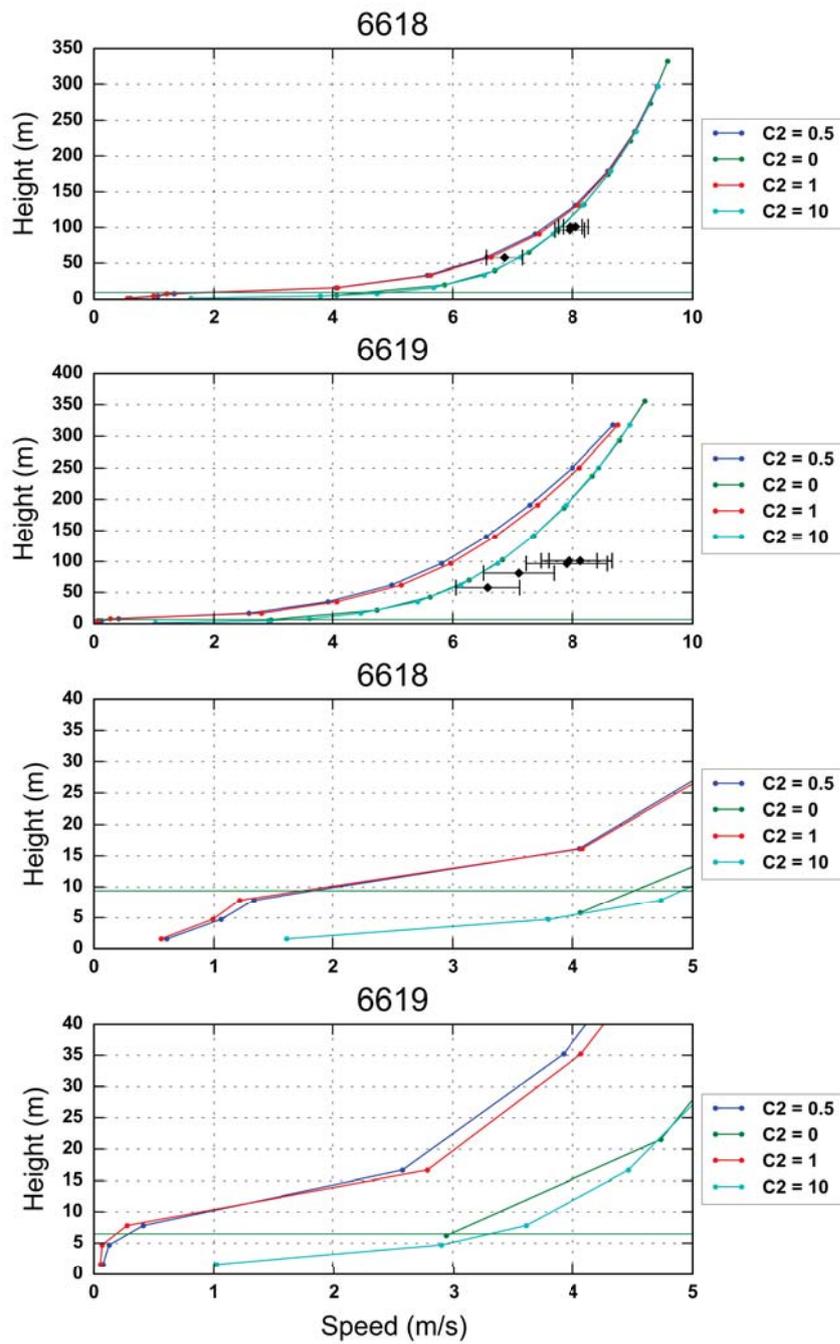


Figure 5.14: Comparison between measured and simulated data between the met. stations 6618 and 6619. The modified “k- ϵ ”-model has been used.

5.4 Case 5

Velocity data

Table 5.8: The speed data in case 5.

6605				6619			
z (m)	Speed(m/s)	Std(m/s)	Dir (°)	z (m)	Speed(m/s)	Std(m/s)	Dir (°)
59	7,22	0,77	-	100,8	9,76	0,4	-
59	7,34	0,78	-	100,8	9,66	0,39	-
57,5	-	-	269	97,7	-	-	285
57	6,98	0,71	-	97,6	-	-	276
55,3	-	-	289	96,4	9,33	0,42	-
44,5	6,39	0,74	-	80,9	8,89	0,43	-
31,50	5,95	0,78	-	57,8	7,83	0,49	-

(a)

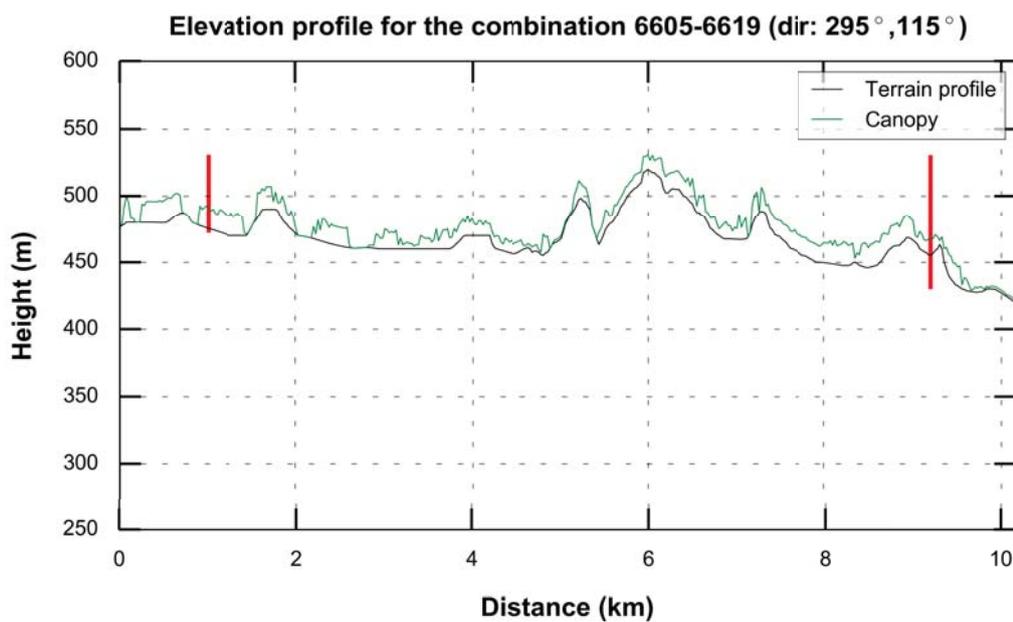
Terrain

Flow model

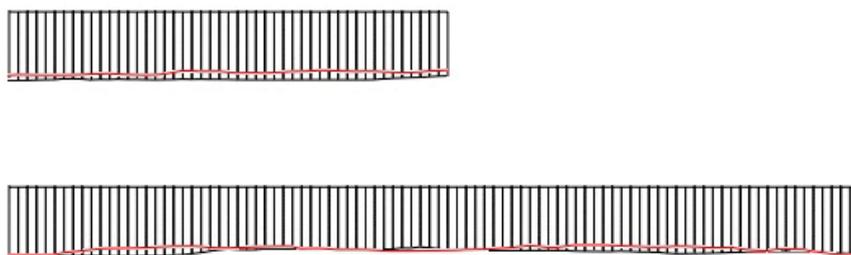
Grid

Table 5.9: The grid data for the for simulation in the area surrounding met. masts 6605 and 6619.

	x	y	z	total
Grid spacing (m)	152	152	-	
Number of cells	93	48	20	89280

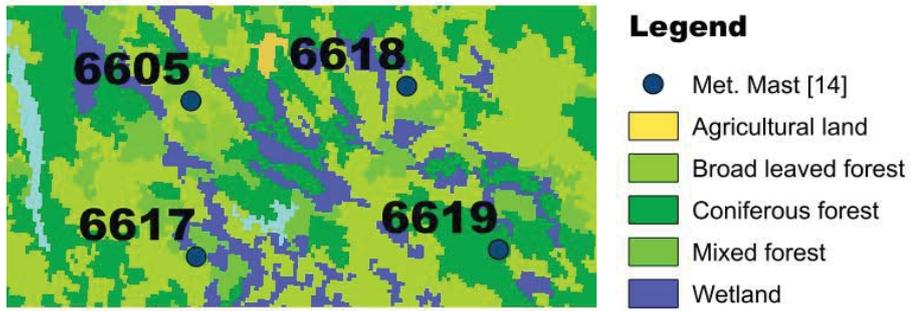


(a) Terrain and canopy profile.

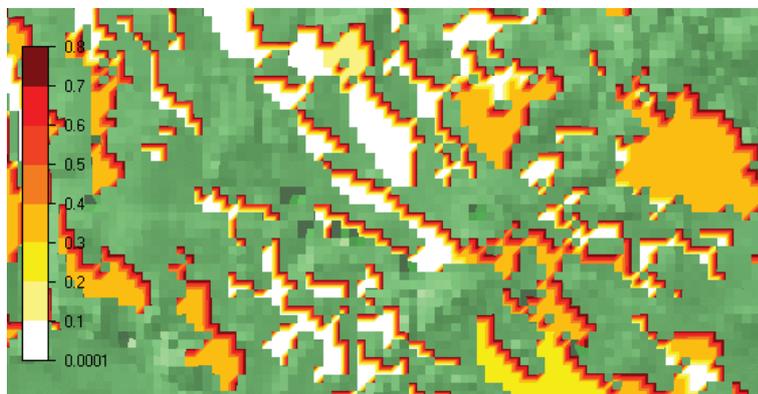


(b) Digital Elevation Model of the simulation.

Figure 5.15: The top figure shows elevation and canopy profile of the terrain between the masts 6605 and 6619 with data from SLU (SLU Forest Map 2010). The red pole is the meteorological masts with the corresponding height. The bottom figure shows the profile of the grid.



(a) Corine-data.



(b) Forest model.

Figure 5.16: A comparison of two corresponding forest models.

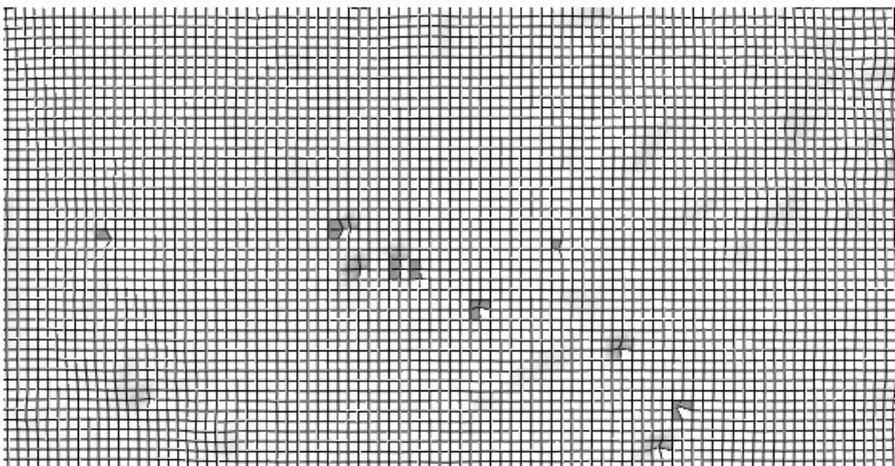


Figure 5.17: The grid of the flow model.

Vertical speed distribution

standard “k- ϵ ”

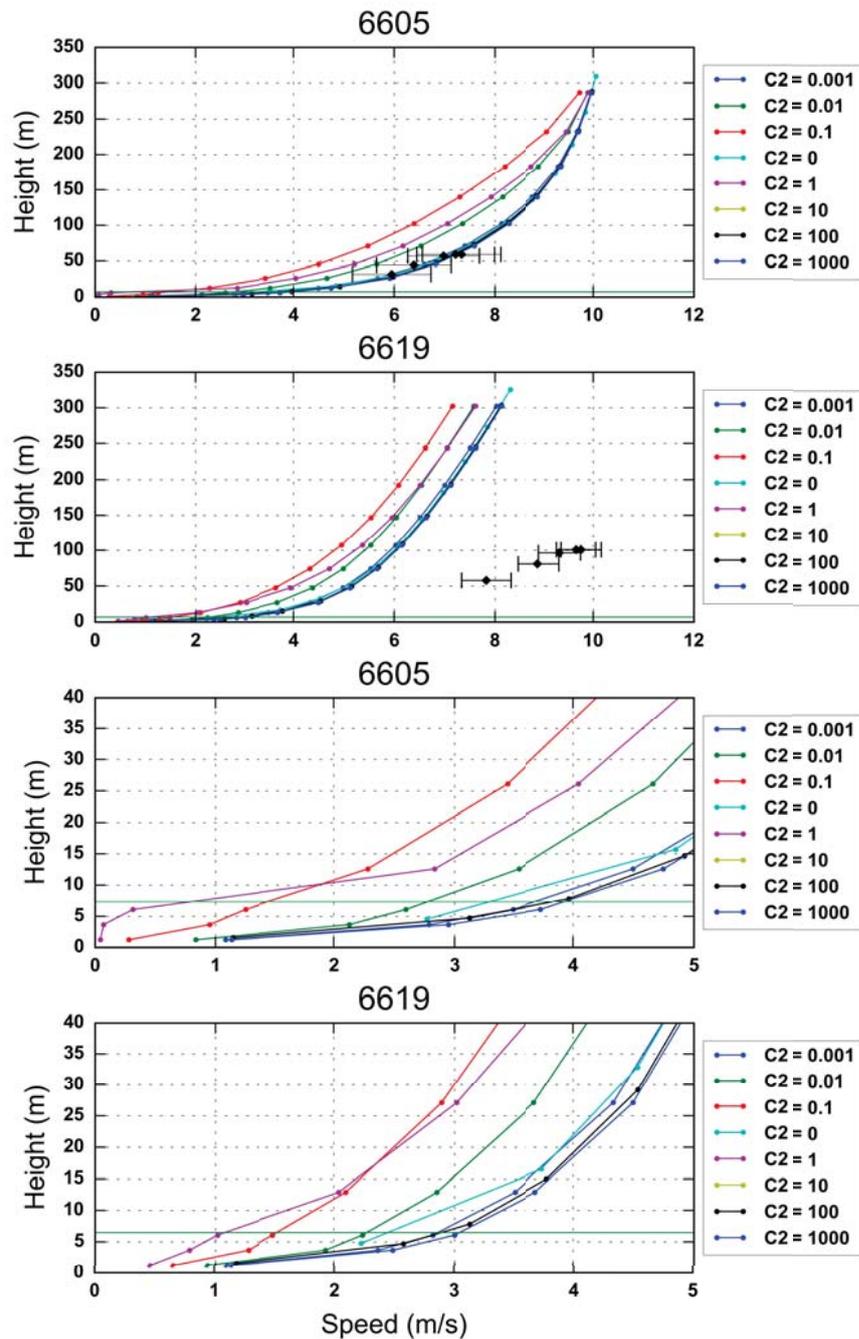


Figure 5.18: Comparison between measured and simulated data between the met. stations 6605 and 6619. The turbulence model standard “k- ϵ ”-model has been used.

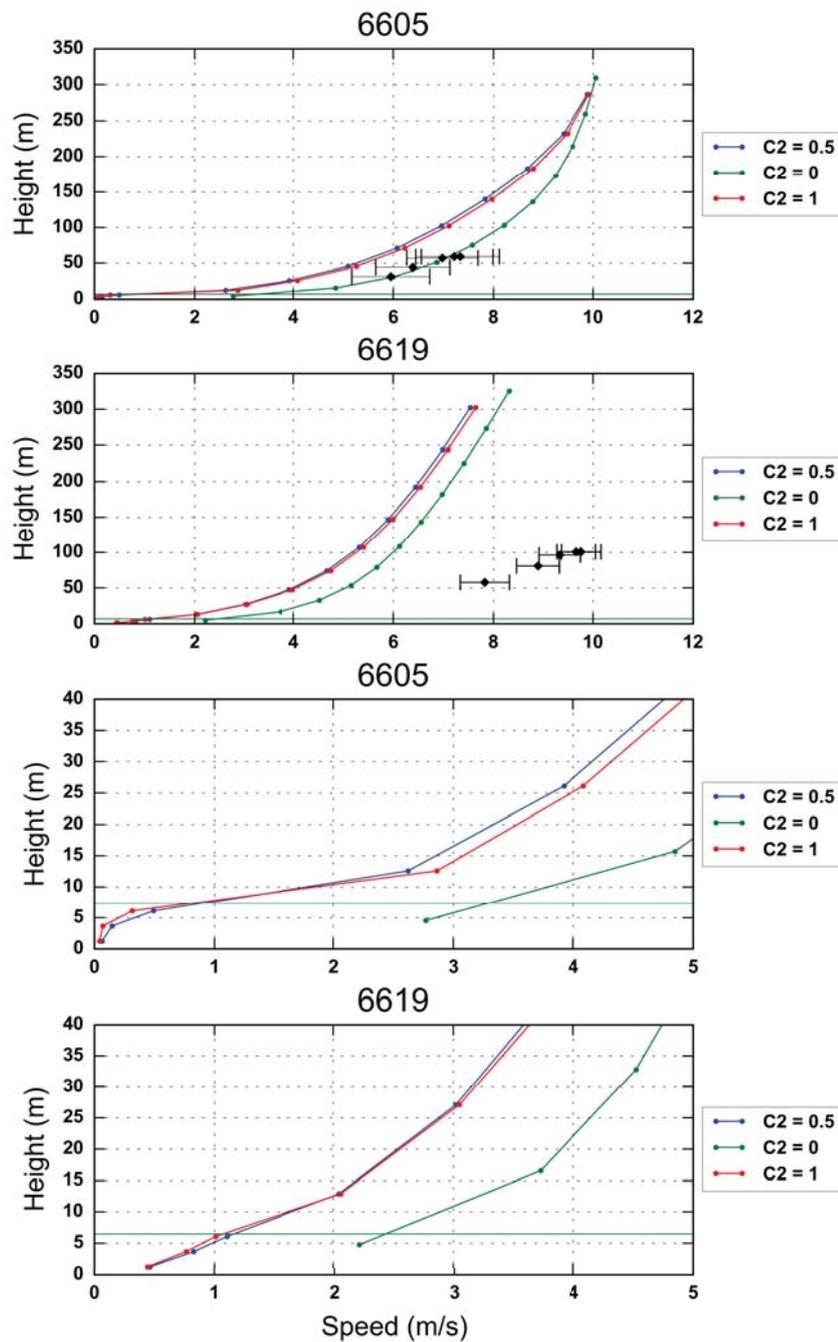
YAP-modified “k- ϵ ”

Figure 5.19: Comparison between measured and simulated data between the met. stations 6605 and 6619. The YAP-modified “k- ϵ ”-model has been used.

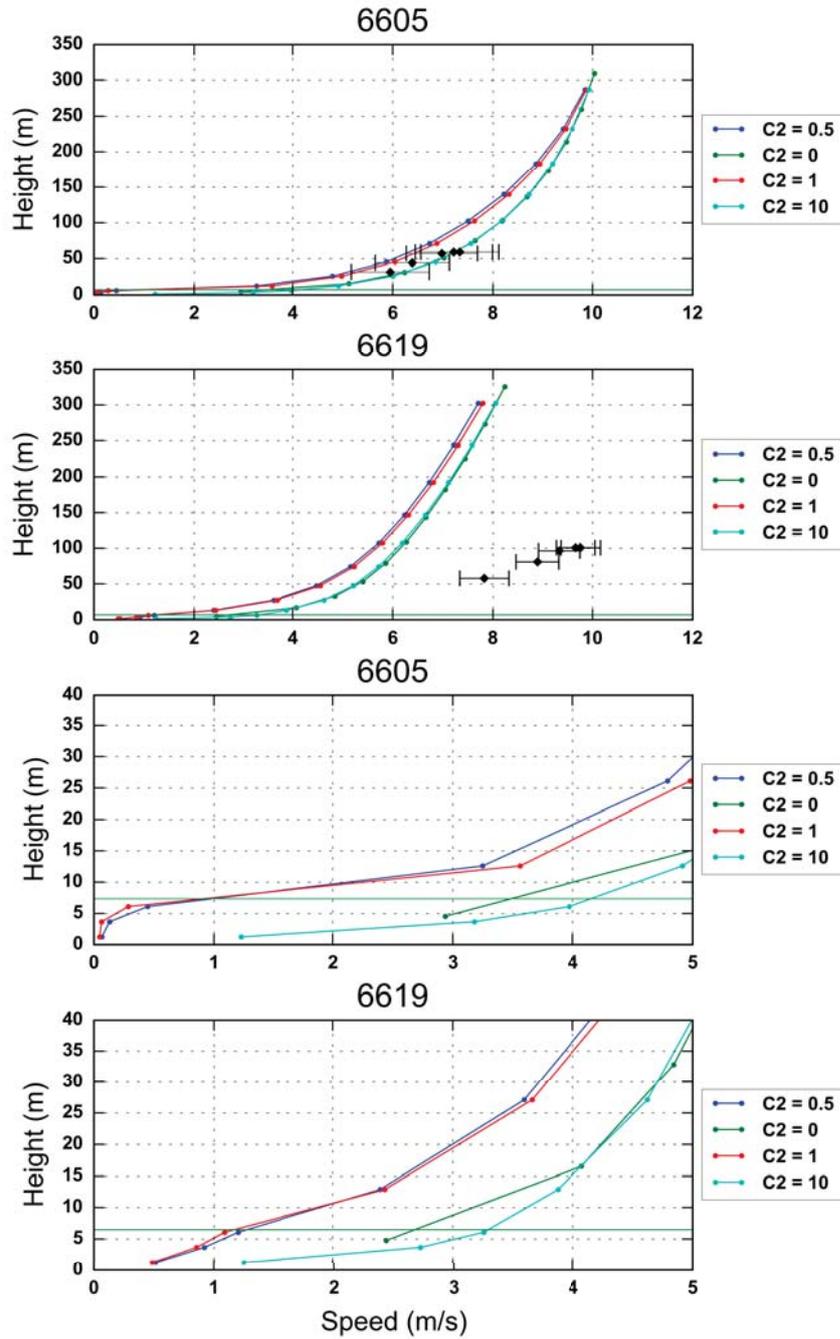
RNG- $k-\epsilon$ 

Figure 5.20: Comparison between measured and simulated data between the met. stations 6605 and 6619. The RNG“ $k-\epsilon$ ”-model has been used.

Modified

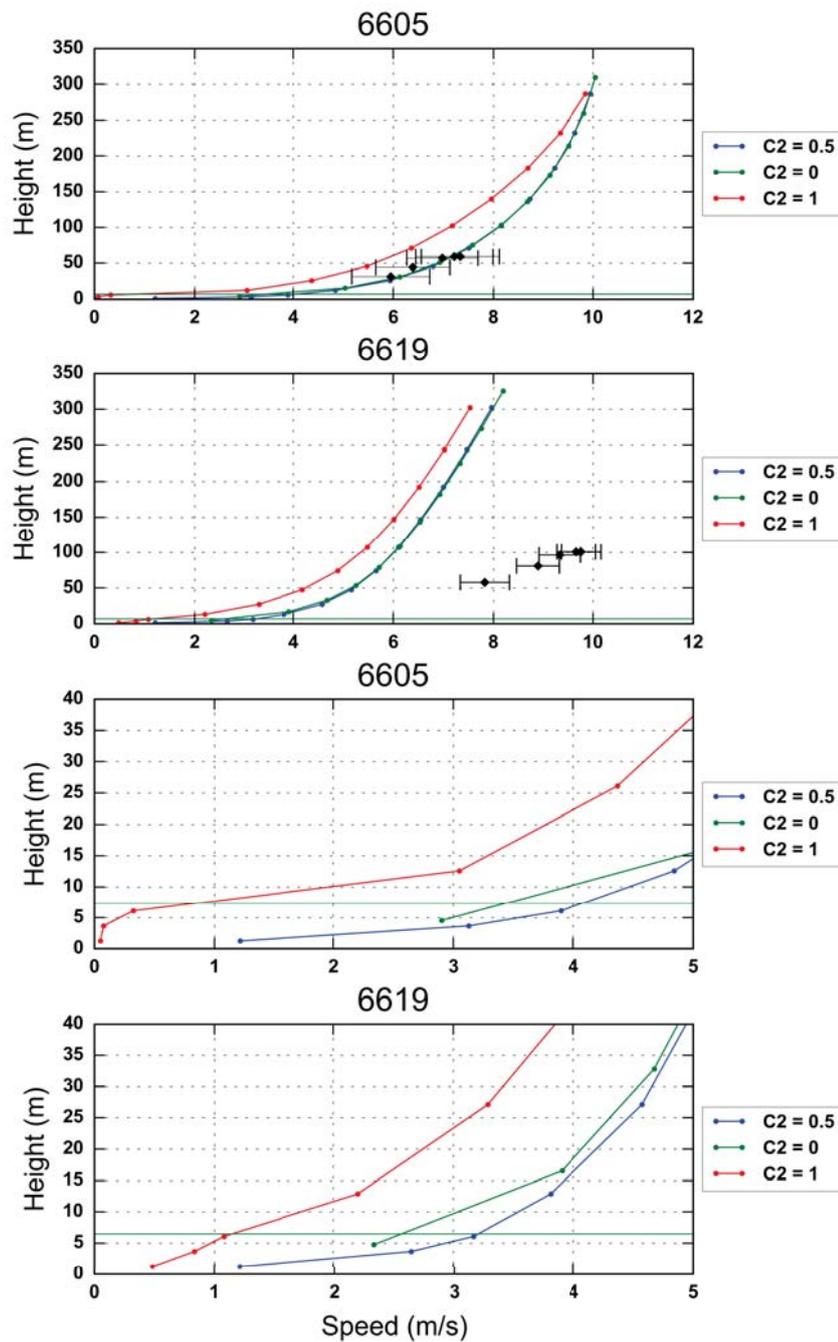


Figure 5.21: Comparison between measured and simulated data between the met. stations 6605 and 6619. The modified “k- ϵ ”-model has been used.

5.5 Case 6

Velocity data

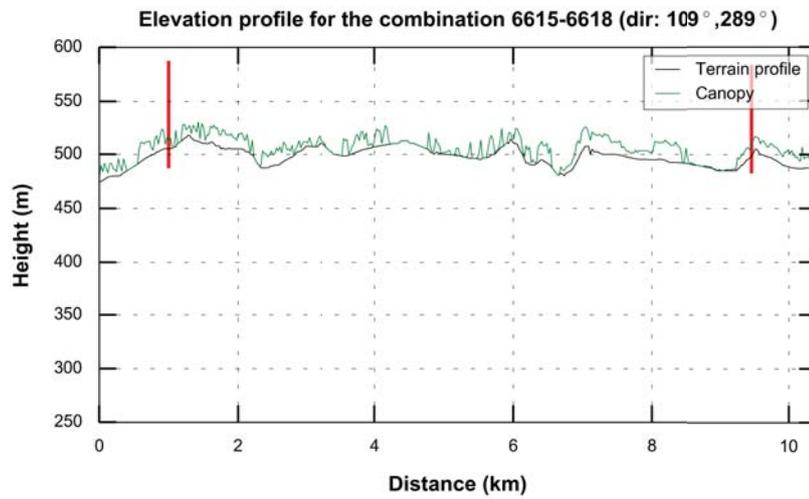
Table 5.10: The speed data in case 6.

6615				6618			
z (m)	Speed(m/s)	Std(m/s)	Dir (°)	z (m)	Speed(m/s)	Std(m/s)	Dir (°)
59	7,22	0,77	-	100,8	6,86	0,19	-
100,7	6,56	0,17	-	100,8	6,85	0,19	-
100,7	6,57	0,17	-	97,7	-	-	290
97,7	-	-	281	97,7	-	-	289
97,6	-	-	277	96,4	6,49	0,22	-
96,4	6,12	0,23	-	80,7	6,73	0,26	-
80,7	6,25	0,25	-	57,6	6,48	0,26	-
57,8	4,16	0,35	-				

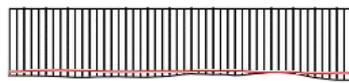
(a)

The velocity data for case 6 are described in tables ?? and ??.

Terrain

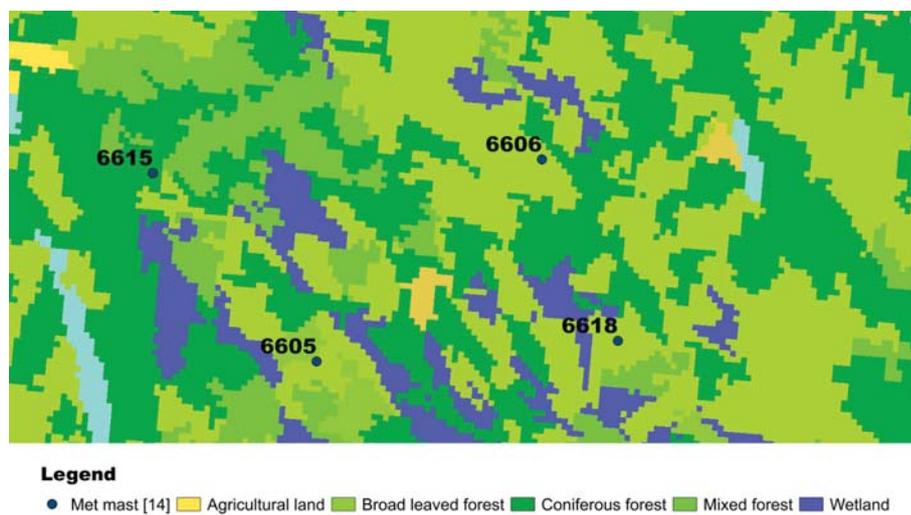


(a) Terrain and canopy profile.

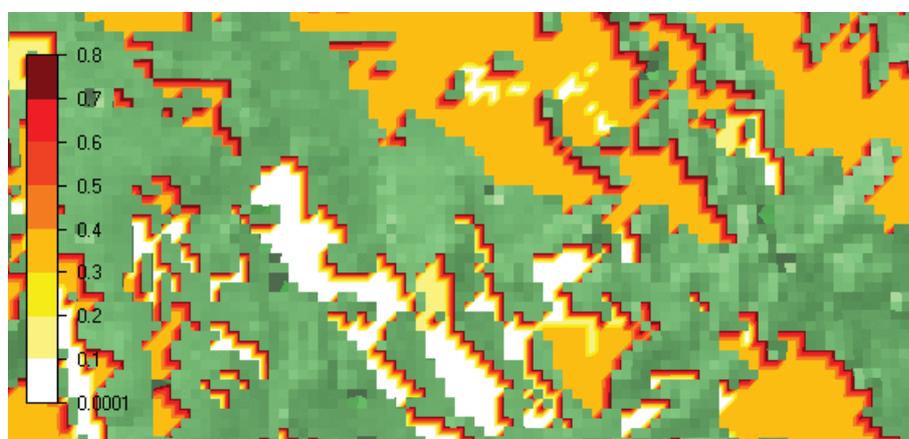


(b) Digital Elevation Model of the simulation.

Figure 5.22: The top figure shows elevation and canopy profile of the terrain between the masts 6615 and 6618 with data from SLU (SLU Forest Map 2010). The red pole is the meteorological masts with the corresponding height. The bottom figure shows the profile of the grid.



(a) Corine-data.



(b) Forest model.

Figure 5.23: A comparison of two corresponding forest models.

Flow model

Grid

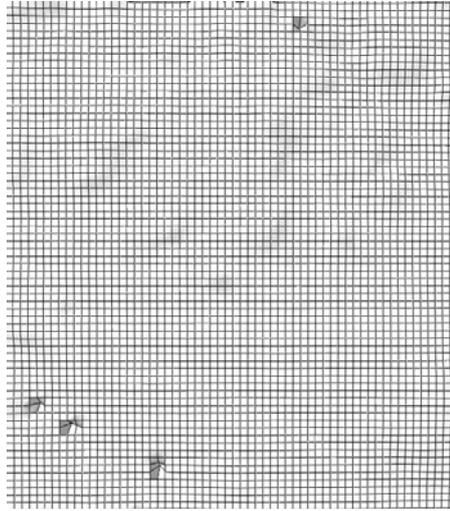


Figure 5.24: The grid of the flow model.

	x	y	z	total
Grid spacing (m)	160	160	-	
Number of cells	98	47	20	92120

Figure 5.25: The grid data for the simulation in the area surrounding met. masts 6615 and 6618.

Vertical speed distribution

standard “k- ϵ ”

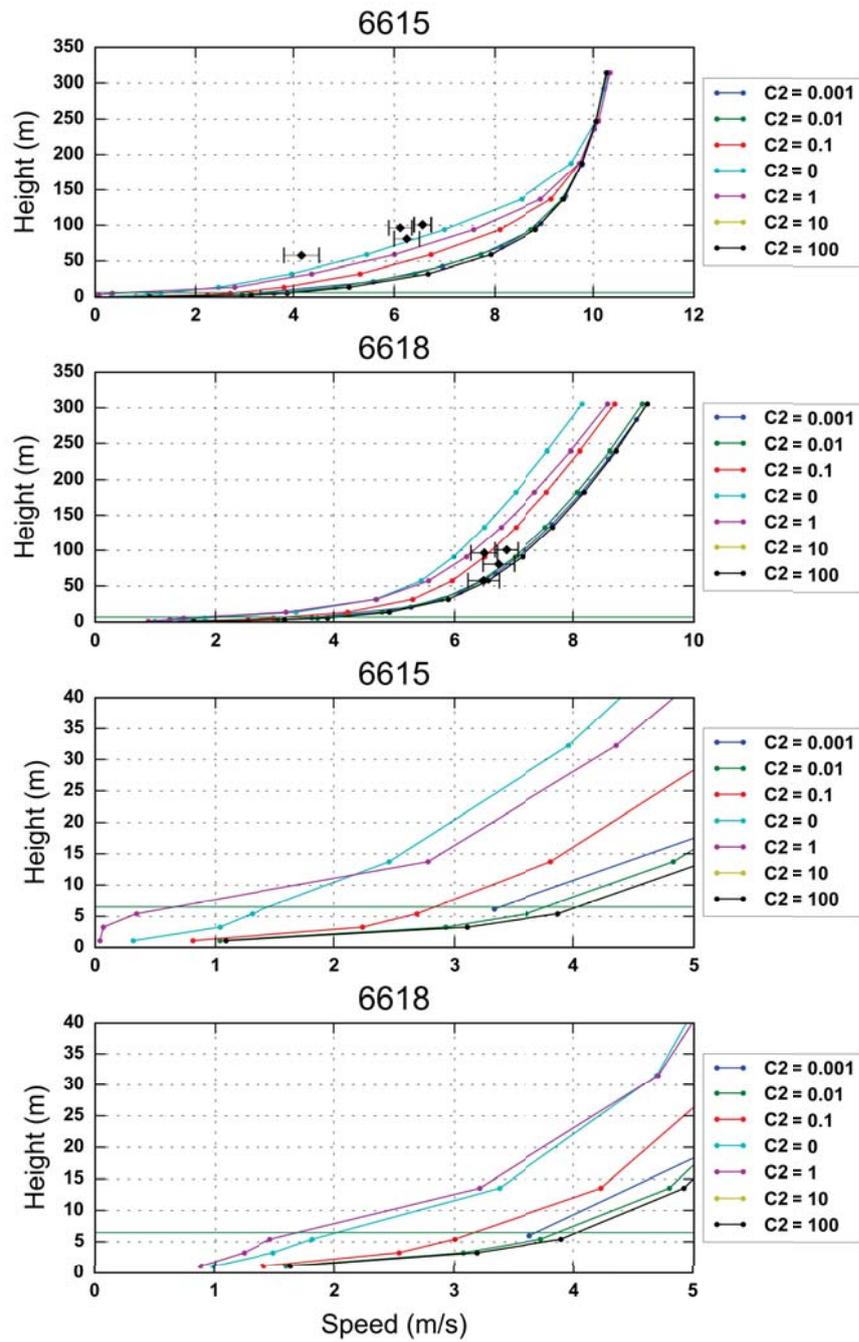


Figure 5.26: Comparison between measured and simulated data between the met. stations 6615 and 6618. The turbulence model standard “k- ϵ ”-model has been used.

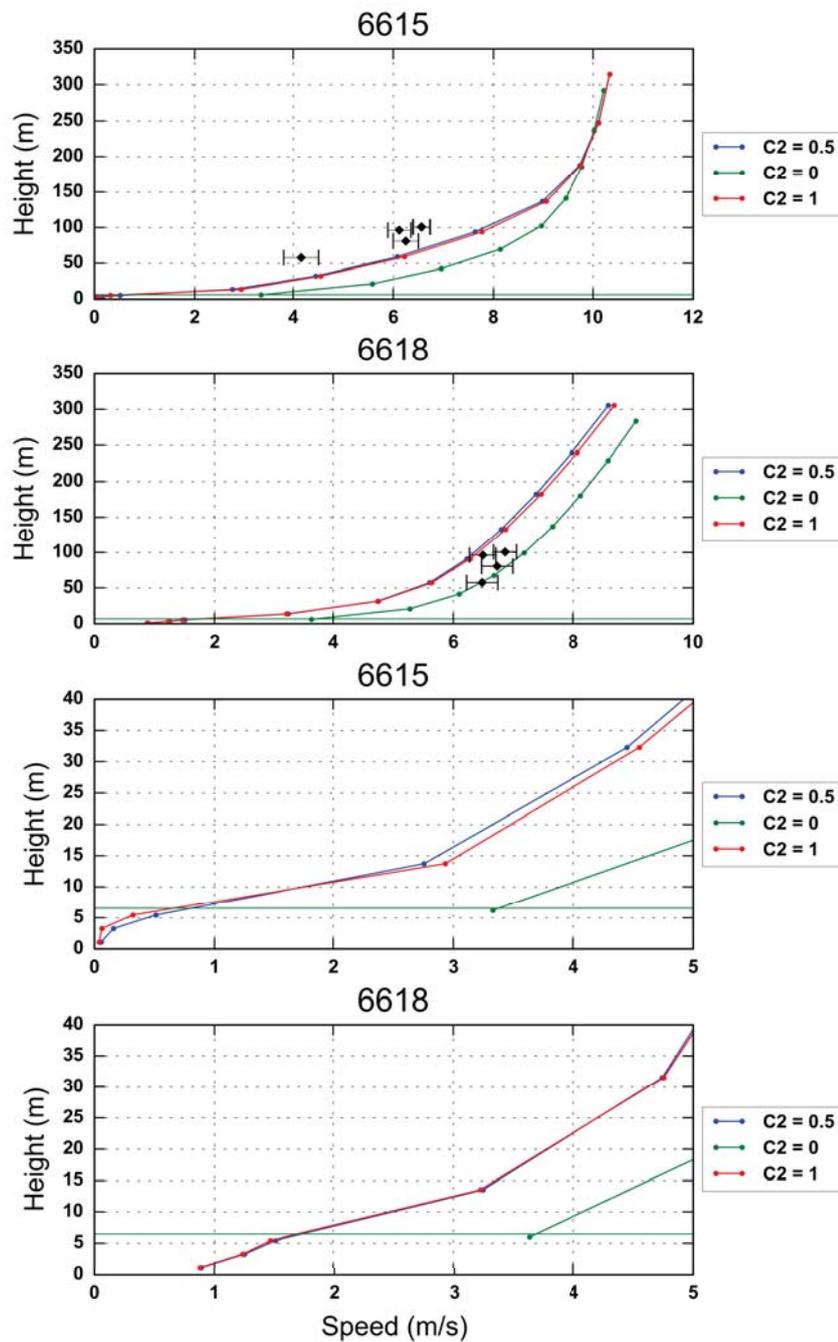
YAP-modified “k- ϵ ”

Figure 5.27: Comparison between measured and simulated data between the met. stations 6615 and 6618. The YAP-modified “k- ϵ ”-model has been used.

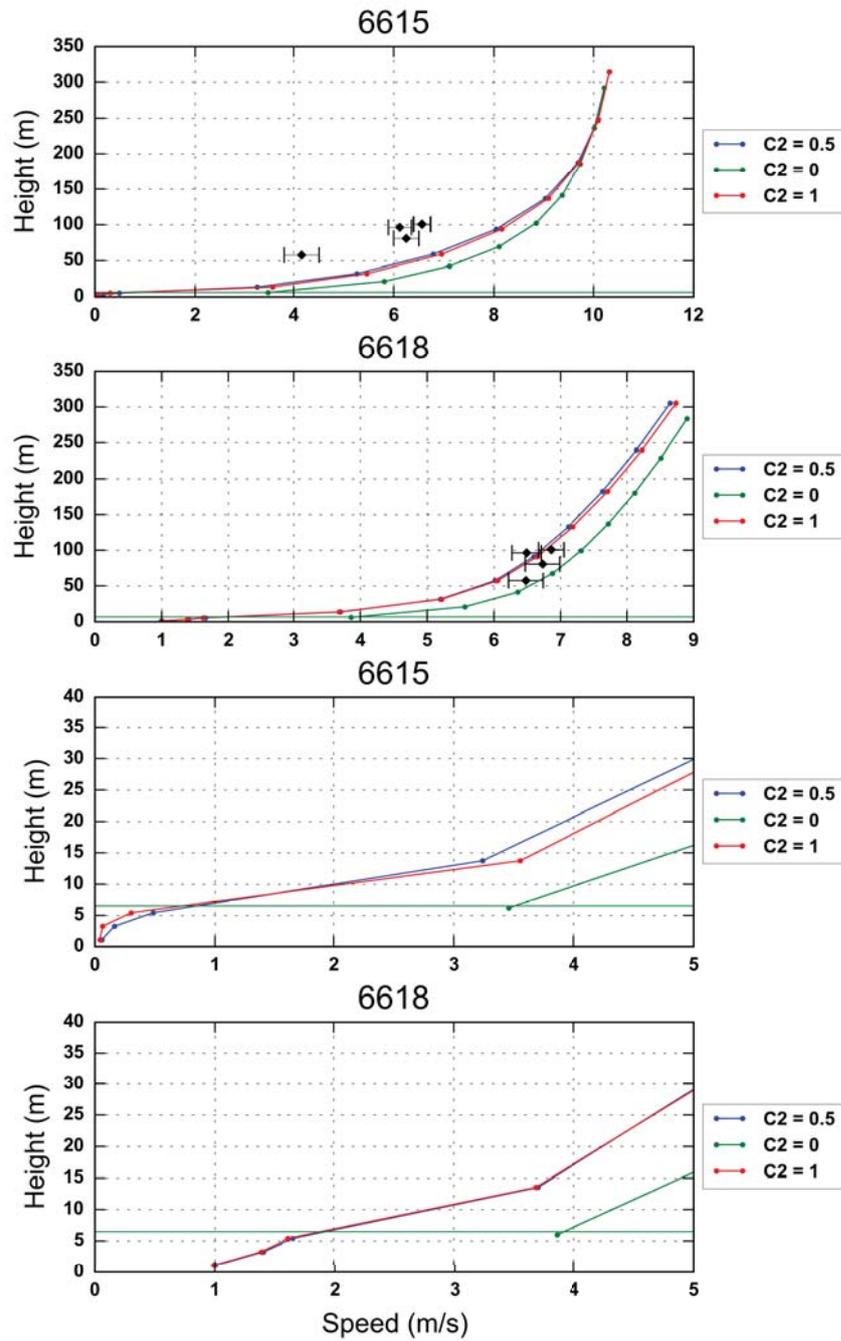
RNG- $k-\epsilon$ 

Figure 5.28: Comparison between measured and simulated data between the met. stations 6615 and 6618. The RNG“ $k-\epsilon$ ”-model has been used.

Modified

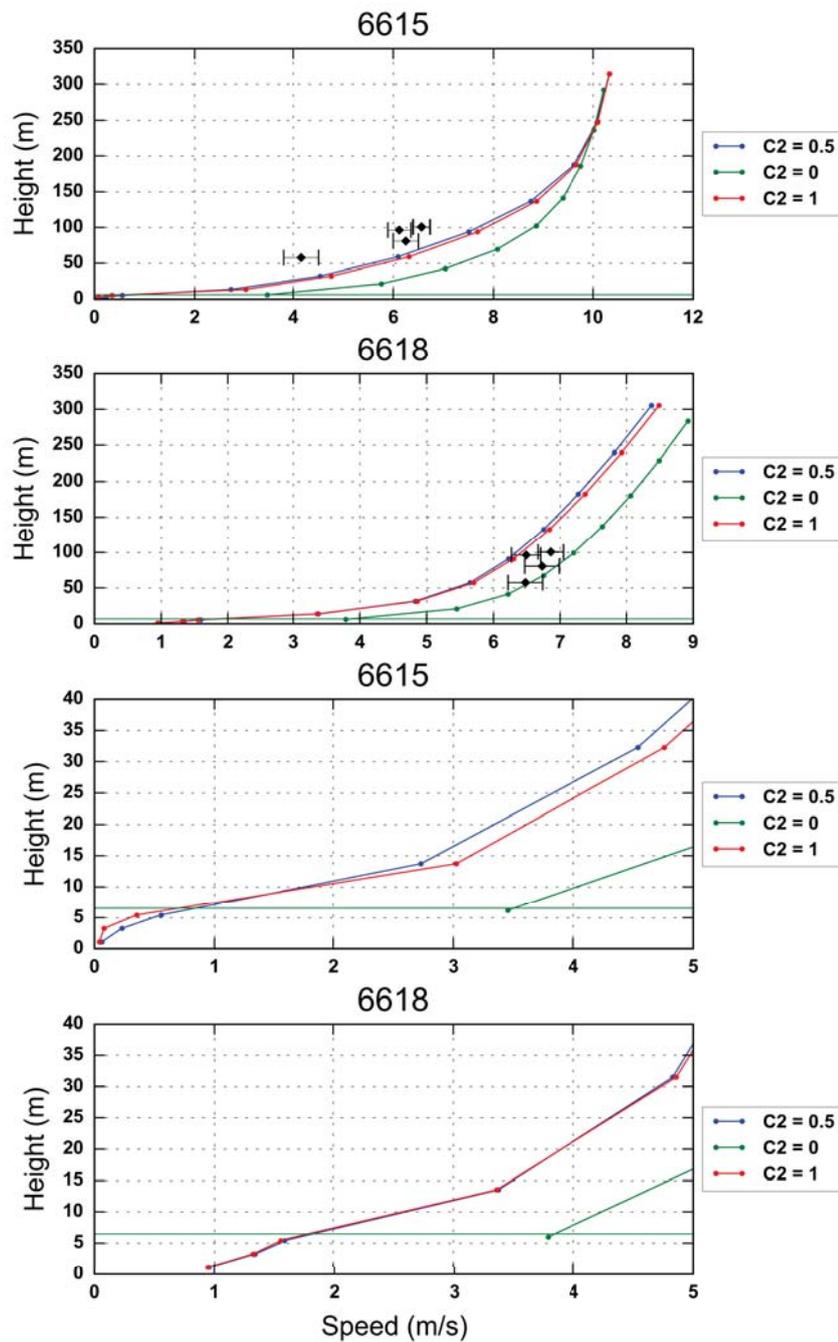


Figure 5.29: Comparison between measured and simulated data between the met. stations 6615 and 6618. The modified “k- ϵ ”-model has been used.

5.6 Atmospheric stability

Table 5.11: Data table retrieved from the stability parameter (equation 4.28).

Station number	Sample number	Unstable (%)	Neutral (%)	Stable (%)	Frequency (h^{-1})	Cycles	Time span (h)
6603	30079	21,7	0	78,3	0,8	6470	7716,2
6604	34165	23,5	0	76,5	1,0	8014	7692,2
6605	32307	21,6	0	78,4	0,8	6978	8502,5
6606	36502	22,6	0	77,4	1,0	8245	8553,5
6610	38224	23,2	1,3	75,4	1,1	8787	7705,5
6611	27015	18,2	1,2	80,6	0,6	4914	8510,2
6612	31157	20,3	1,1	78,6	0,8	6315	7950,7
6613	40784	18,4	1,3	80,3	0,9	7509	8542,7
6614	29104	16,5	1,5	82	0,6	4539	7640,8
6615	35047	22,2	1,1	76,8	1,0	7732	7697,2
6616	29893	21,6	1,2	77,2	0,8	6463	8500,5
6617	32054	21,3	1,2	77,5	0,9	6523	7637,8
6618	33954	18,8	1,5	79,7	0,7	6374	8587,5
6619	36707	23,5	1	75,4	1,0	8628	8504,8

Discussion

Terrain

Case I and II diverges when choosing a total number of cells of about 1 000 000 cells with the use of the forest turbulence model with a C2-value of 0.5. That may be due to a complex terrain (fig-5.1a) which causes “holes” in the grid i.e. there is a build of stress. The digital terrain model of the grid in all cases is smoothed out compared to the DEM generated with interpolated point in contour lines. The simulation of 6619 in case V deviates strongly from the measurements which suggests that the terrain effects are dominant which can be shown with the peak between the met. mats in figure 5.15a.

Tendency

The method of validation in this thesis is to find the rate of interaction between the turbulence models and the forest model hence the tendencies (Busch 2015; Borás 2015). Another method to validate a forest model is to use measurements at one mast and to reproduce the mean wind speed for strong wind speed (Zixiao et al. 2013). The justification of using a model with a grid resolution of 90 000 cells is to find the tendencies of the interaction between the forest model and the turbulence model. The tendency for the simulated vertical profile is to shift from lower to higher mean wind speed, ie. from the left part of the plots to right. The first met. station affected by the air flow in a direction is close to the vertical wind speed distribution, but the met mast downstream tends deviate the most. The reason may be a complex terrain and long distance. The suitable C2-value is between 0.5 and 1 for all cases. In case 1 there is a sign of turbulence at height 57,8m at station 6615. However the difference

in speed direction at nearly the same height indicates that the flow between the preceding measurement was not stationary and unidirectional. In case 2 both met. stations are affected by the canopy height is affected by the canopy height around where $C2 = 1$. The interaction between the forest model and the turbulence would be clear if the speed affected by the same $C2$ values with different turbulence models were plotted with each other.

Atmospheric stability

The atmospheric stability around the project sites shifts once an hour (see table 5.11) and neutral stability is a transition state between stable and unstable atmospheric stability (*Email correspondence with the Norwegian met. office 07/06/2016 2016*). Neutral atmospheric stability is rare phenomena (Arya 1988, see table 5.11) and subsequently instantaneous nor time averaged data with hourly temporal resolution would most probably not register neutral boundary layer condition in these project sites.

6.1 Uncertainty

Data set

To find the variation between the recording intervals of 10minutes the time averaged values speed should ideally compared with instantaneous speed data with a temporal resolution of seconds. It is probable that the amount of validation cases could have been higher if the data values had been instantaneous and would show less uncertainty. The difference in speed direction (tables 5.4,5.6b, 5.8, 5.10) records measured at a height difference of 0.1m reveals that (table 5.8a) there has been rapid shift in the wind direction with an effect by the boom. A forest effect can be either be damped or enhanced with the use of time averaged values.

6.2 Is the model validated?

The ideal case for validation is to test a CFD-model of air flow, as implemented in PHOENICS, with statistical tools i.e. test of significance (Moore et al. 2006) or omnidirectional wind speed (Zixiao et al. 2013). The sample rate in the data set is 3-4 observations over span of 100m. The average difference from the top is 25% and 22% for the meteorological towers of 58 meters and 100,7 meters ¹. The anemometers

¹Observations at 25,75,100m gives an average difference from top of 37,5%.

can be installed too high in order to capture a forest effect (Zixiao et al. 2013; Borás 2015). There has been a tendency that the simulation towards the cases, but the complexity of the terrain between the masts, low sample rate and time averaged values of unsteady flow give rise to a great deal uncertainty about the reliability about the cases.

Conclusion

A measurement campaign for wind speed, direction, humidity, temperature and pressure for a long period and short height intervals with high temporal resolution specially designed for validating a forest model would give validation cases closer to ideal. In order to increase the precision of the drag coefficients C_1 and C_2 in a validation case with given settings a suitable program should be developed and it as a validation module with a finer mesh. The distances between the cases, the complex terrain, the time averaged values and low uncertainty give rise to uncertainty about the reliability of the reliability of the cases. Because of the low sample rate and complexity there has been difficulties with extracting numerical data for comparing simulated and simulated data. An alternative method is to compared simulated and measured mean wind speeds over a long observation period. Thus the forest model needs further investigation.

Case selection module

Data-New-Format/caseselectionmodule.py

```

1  # -*- coding: utf-8 -*-
2
3  import numpy as np
4  import datetime,os,sys
5  print datetime.datetime.today()
6  reload(sys)
7  sys.setdefaultencoding('utf-8')
8
9  def dataset(st_no, type, criterion, startdate="2009-02-23",enddate="2010-0
    2-22"):
10  ___#setting __doc__
11  ___"""
12  ___Criterion is the number of elements.
13  ___The dates must be set on the form YYYY-MM-DD as a string.
14  ___"""
15  ___path_n1 = "/Users/pederwessel/Documents/NMBU/Masteroppgave-Data/"
16  ___path_n = path_n1 + "Skogsvalidering WindSim SCA/wind data/TIL SSVAB"
17  ___for file in os.listdir(path_n):
18  ___     if file.find(type) >= 0 and file.find("%s"%str(st_no)) >= 0:
19  ___         filename = os.path.abspath(path_n + "/" + file)
20  ___     if file.find(type) >= 0 and file.find("%s"%str(st_no)) >= 0:
21  ___         filename = os.path.abspath(path_n + "/" + file)
22  ___infile = open(filename, 'r')_____#open file for reading
23  ___unstrippedlines = infile.readlines()
24  ___lines = []
25  ___for line in unstrippedlines:
26  ___     ___lines.append(line.strip('\n'))
27
28  ___#fetching the number of lines in the header
29  ___headernumb = 0
30  ___indexcount = 0
31  ___while headernumb < len(lines):
32  ___     ___l = lines[indexcount]

```

```

33 _____if len(l.split(","))>criterion:
34 _____break
35 _____headernumb +=1
36 _____indexcount +=1
37 _____firstline = lines[headernumb]
38 _____lastline = lines[-1]
39
40 _____#deleting the header
41 _____headerlines = lines[:headernumb]
42 _____UTM_coor,heights = [],[]
43 _____for l in headerlines:
44 _____if l.find("Elevation")>= 0:
45 _____h = l.split('-')
46 _____del h[0]
47 _____heights.append(h)
48 _____elif l.find("Latitude") >= 0:
49 _____g = l.split('-')
50 _____del g[0]
51 _____UTM_coor.append(g)
52 _____elif l.find("Longitude") >= 0:
53 _____f = l.split('-')
54 _____del f[0]
55 _____UTM_coor.append(f)
56 _____for i in range(headernumb):
57 _____del lines[0]
58
59 _____#setting number of total amount of measurements
60 _____comments = []
61 _____comments.append("The total amount line numbers in the header is %d."
        _____%headernumb)
62 _____comments.append("The total number of measurements before filtering is
        _____%d." % len(lines))
63
64 _____#For crosschecking the first and the last measurement in the
        _____dataserries
65 _____if firstline != lines[0] or lastline != lines[-1]:
66 _____comments.append("The data has been read wrongly.")
67 _____else:
68 _____comments.append("The data has been read correctly.")
69
70 _____#neglecting the time series without a wind speed value
71 _____newlines = []
72 _____counter = 0
73 _____for i in range(len(lines)):
74 _____if lines[i].find('NaN') < 0:
75 _____newlines.append(lines[i])
76 _____counter +=1
77 _____comments.append("The number of measurements after deleting"+\
78 _____" wind speed measurements with NaN is %d." % counter)
79
80 _____#categorizing data into columns
81 _____import numpy as np
82 _____rows = []
83 _____for line in newlines:

```

```

84 _____row = line.split(',')
85 _____rows.append(tuple(row))
86
87 _____#Fetching the field names in the header
88 _____col_names = headerlines[headernumb-len(rows[0])-1:-1]
89
90 _____#Storing the data in a structured array
91 _____dtype_list = ['i4', 'a10', 'a8']
92 _____for i in range(3, len(rows[0])):
93 _____dtype_list.append('<f8')
94 _____dt = np.dtype(zip(col_names, dtype_list))
95 _____cols = np.array(rows, dt)
96
97 _____#Selecting the dataset period. The periods must be on the form YYYY-
          _____MM-DD on a string.
98
99 _____indexlist = []
100 _____for i in range(len(cols['Date Field'])):
101 _____date = stringtodate(cols['Date Field'][i])
102 _____if stringtodate(startdate) <= date <= stringtodate(enddate):
103 _____indexlist.append(i)
104
105 _____start, stop = indexlist[0], indexlist[-1]
106 _____del indexlist
107
108 _____#Writing the field names in this file
109 _____comments.append("The fields in this file are:")
110 _____for i in range(len(dt.names)):
111 _____comments.append([i+1, dt.names[i]])
112
113 _____#Counting number of days of measurements
114 _____for name in dt.names:
115 _____if name == 'Date Field':
116 _____datefield = cols[name]
117 _____counter = 0
118 _____daycounter = 0
119 _____while counter < len(datefield):
120 _____day1 = int(datefield[counter-1][-2] + datefield[counter-1][-1])
121 _____day = int(datefield[counter][-2] + datefield[counter][-1])
122 _____if day1 != day:
123 _____daycounter +=1
124 _____counter+=1
125 _____comments.append("The measurement period has extended for %d days." %
          _____daycounter)
126 _____infile.close()
127 _____#defining the output
128 _____return cols[start:stop], comments, heights, UTM_coor
129
130 def RH_ar(filename, startdate = "2009-02-23", enddate = "2010-02-22"):
131 _____"""
132 _____This function returns moisture data into structured arrays.
133 _____"""
134 _____import datetime
135 _____infile = open(filename, 'r')

```

```

136 ____linelist = infile.readlines()
137 ____lines = []
138 ____for line in linelist:
139 ____    ____lines.append(line.split(';'))
140     #Fetching the field names
141 ____fieldnames = lines[9][0:3] #9 and 10 or 12 and 13
142
143 ____#Deleting the header
144 ____del lines[0:10]
145 ____dates , times, rhs = [],[],[]
146 ____for row in lines:
147 ____    dates.append(row[0])
148 ____    times.append(row[1])
149 ____    rhs.append(row[2])
150
151 ____#Categorizing the header
152 ____dt = np.dtype([(fieldnames[0], 'S10'), (fieldnames[1], 'S8'), (
153     fieldnames[2], float)])
154
155 ____dataarray = np.asarray(zip(dates,times,rhs), dt)
156
157 ____#Filtering the data by date
158 ____indexlist = []
159 ____for i in range(len(dataarray[fieldnames[0]])):
160 ____    date = stringtodate(dataarray[fieldnames[0]][i])
161 ____    if stringtodate(startdate) <= date <= stringtodate(enddate):
162 ____        indexlist.append(i)
163 ____start, stop = indexlist[0], indexlist[-1]
164 ____del indexlist
165 ____return dataarray[start:stop]
166
167 def new_matrix_reader(varname, st_no):
168     """
169     This function reads the synchronized matrices of moisture, pressure,
170     temperature,
171     temperature difference wind direction, wind speed, wind speed
172     standard deviation and
173     the measure of stability parameter.
174     """
175     import os
176     if varname == "RH":
177         foldername1 = "/Users/pederwessel/Documents/NMBU/Masteroppgave/
178         Data-New-Format/"
179         foldername2 = foldername1 + "Matrices_newest/rhmatrices"
180         for file in os.listdir(foldername2):
181             if file.find("%s"%str(st_no)) >= 0:
182                 infile = open(os.path.abspath(foldername2 + "/" + file),'
183                 r')
184                 lines = infile.readlines()
185                 date,time,var = [],[],[]
186                 for line in lines:
187                     row = line.strip('\r\n').split(',')
188                     date.append(row[0])
189                     time.append(row[1])
190                     var.append(row[2])

```

```

185         dt = [('Date', 'S10'),('Time', 'S8'),('RH',float)]
186         return np.asarray(zip(date,time,var),dtype = dt)
187     if varname == "P":
188         foldername1 = "/Users/pederwessel/Documents/NMBU/Masteroppgave/
Data-New-Format/"
189         foldername2 = foldername1 + "Matrices_newest/pmatrices"
190         for file in os.listdir(foldername2):
191             if file.find("%s"%str(st_no)) >= 0:
192                 infile = open(os.path.abspath(foldername2 + "/" + file),'
r')
193                 lines = infile.readlines()
194                 date,time,var = [],[],[]
195                 for line in lines:
196                     row = line.strip('\r\n').split(',')
197                     date.append(row[0])
198                     time.append(row[1])
199                     var.append(row[2])
200                 dt = [('Date', 'S10'),('Time', 'S8'),('P',float)]
201                 return np.asarray(zip(date,time,var),dtype = dt)
202
203     if varname == "t":
204         foldername1 = "/Users/pederwessel/Documents/NMBU/Masteroppgave/
Data-New-Format/"
205         foldername2 = foldername1 + "Matrices_newest/tmatrices"
206         for file in os.listdir(foldername2):
207             if file.find("%s"%str(st_no)) >= 0:
208                 infile = open(os.path.abspath(foldername2 + "/" + file),'
r')
209                 lines = infile.readlines()
210                 date,time,var = [],[],[]
211                 for line in lines:
212                     row = line.strip('\r\n').split(',')
213                     date.append(row[0])
214                     time.append(row[1])
215                     var.append(row[2])
216                 dt = [('Date', 'S10'),('Time', 'S8'),('t',float)]
217                 return np.asarray(zip(date,time,var),dtype = dt)
218     if varname == "dt":
219         foldername1 = "/Users/pederwessel/Documents/NMBU/Masteroppgave/
Data-New-Format/"
220         foldername2 = foldername1 + "Matrices_newest/dtmatrices"
221         for file in os.listdir(foldername2):
222             if file.find("%s"%str(st_no)) >= 0:
223                 infile = open(os.path.abspath(foldername2 + "/" + file),'
r')
224                 lines = infile.readlines()
225                 date,time,var = [],[],[]
226                 for line in lines:
227                     row = line.strip('\r\n').split(',')
228                     date.append(row[0])
229                     time.append(row[1])
230                     var.append(row[2])
231                 dt = [('Date', 'S10'),('Time', 'S8'),('dt',float)]
232                 return np.asarray(zip(date,time,var),dtype = dt)

```

```

233     if varname == "wd":
234         foldername1 = "/Users/pederwessel/Documents/NMBU/Masteroppgave/
Data-New-Format/"
235         foldername2 = foldername1 + "Matrices_newest/wdmatrices"
236         for file in os.listdir(foldername2):
237             if file.find("%s"%str(st_no)) >= 0:
238                 infile = open(os.path.abspath(foldername2 + "/" + file),'
r')
239                 lines = infile.readlines()
240                 date,time,var = [],[],[]
241                 for line in lines:
242                     row = line.strip('\r\n').split(',')
243                     date.append(row[0])
244                     time.append(row[1])
245                     var.append(row[2])
246                 dt = [('Date', 'S10'),('Time', 'S8'),('wd',float)]
247                 return np.asarray(zip(date,time,var),dtype = dt)
248     if varname == "ws":
249         foldername1 = "/Users/pederwessel/Documents/NMBU/Masteroppgave/
Data-New-Format/"
250         foldername2 = foldername1 + "Matrices_newest/wsmatrices"
251         for file in os.listdir(foldername2):
252             if file.find("%s"%str(st_no)) >= 0:
253                 infile = open(os.path.abspath(foldername2 + "/" + file),'
r')
254                 lines = infile.readlines()
255                 date,time,var = [],[],[]
256                 for line in lines:
257                     row = line.strip('\r\n').split(',')
258                     date.append(row[0])
259                     time.append(row[1])
260                     var.append(row[2])
261                 dt = [('Date', 'S10'),('Time', 'S8'),('ws',float)]
262                 return np.asarray(zip(date,time,var),dtype = dt)
263     if varname == "ws_std":
264         foldername1 = "/Users/pederwessel/Documents/NMBU/Masteroppgave/
Data-New-Format/"
265         foldername2 = foldername1 + "Matrices_newest/ws_stdmatrices"
266         for file in os.listdir(foldername2):
267             if file.find("%s"%str(st_no)) >= 0:
268                 infile = open(os.path.abspath(foldername2 + "/" + file),'
r')
269                 lines = infile.readlines()
270                 date,time,var = [],[],[]
271                 for line in lines:
272                     row = line.strip('\r\n').split(',')
273                     date.append(row[0])
274                     time.append(row[1])
275                     var.append(row[2])
276                 dt = [('Date', 'S10'),('Time', 'S8'),('ws_std',float)]
277                 return np.asarray(zip(date,time,var),dtype = dt)
278     if varname == "s_par":
279         foldername1 = "/Users/pederwessel/Documents/NMBU/Masteroppgave/
Data-New-Format/"

```

```

280     foldername2 = foldername1 + "Matrices_newest/s_parmatrices"
281     for file in os.listdir(foldername2):
282         if file.find("%s"%str(st_no)) >= 0:
283             infile = open(os.path.abspath(foldername2 + "/" + file), '
r')
284             lines = infile.readlines()
285             date,time,var = [],[],[]
286             for line in lines:
287                 row = line.strip('\r\n').split(',')
288                 date.append(row[0])
289                 time.append(row[1])
290                 var.append(row[2])
291             dt = [('Date', 'S10'),('Time', 'S8'),('s_par',float)]
292             return np.asarray(zip(date,time,var),dtype = dt)
293
294     else:
295         return "Please choose between RH, P, dt,s_par, and specify the
met mast."
296 def combinatoric(filename):
297     """
298     This function finds the combination between two meteorological masts.
299     """
300     latitude = heights(filename)['Latitude']
301     longitude = heights(filename)['Longitude']
302     #Making the UTM coordinates into a vector.
303     COOR = np.array(zip(latitude ,longitude))
304     Station_no = heights(filename)['Station Number']
305     Elevation = heights(filename)['Elevation']
306     counter = 0
307     comblist1, comblist2, distancelist, El_diff, columnlist= [],[],[],[],
[]
308     for i in range(len(COOR)):
309         for j in range(len(COOR)):
310             if i!=j:
311                 B = COOR[i]-COOR[j]
312                 El_diff.append(eval("%.2f" % (Elevation[i]-Elevation[j]))
)
313                 comblist1.append(Station_no[i])
314                 comblist2.append(Station_no[j])
315                 distancelist.append(eval("%.2f" % (np.linalg.norm(B)/1000
.)))
316                 counter +=1
317     print "There are %d combinations between 2 selections and %d
possibilities." \
318     % (counter, len(COOR))
319     dt = [('Met Mast 1', 'int'),('Met Mast 2', 'int'),('Distance','float'
),\
320     ('Elevation difference (1-2)','float')]
321     array_ = np.asarray(zip(comblist1, comblist2,distancelist,El_diff),
dtype = dt)
322     sort_arr = np.sort(array_,order = ('Distance','Elevation difference (
1-2)'))
323     #Returning two lists for cross checking
324     return sort_arr[0::2] , np.delete(sort_arr, np.s_[0::2],0)

```

```

325
326 def comb_RT90_array():
327     """
328     This function reads the combinations associated with direction and
329     geographical
330     position.
331     """
332     infile = open("RT90-comb-dir.txt", 'r')
333     lines = infile.readlines()
334     M1, la1, lo1, M2, la2, lo2, dir1, dir2 = [], [], [], [], [], [], [], []
335     for line in lines:
336         row = line.strip('\r\n').split(' ')
337         M1.append(row[0])
338         lo1.append(row[1])
339         la1.append(row[2])
340         M2.append(row[3])
341         lo2.append(row[4])
342         la2.append(row[5])
343         dir1.append(row[6])
344         dir2.append(row[7])
345     dt = np.dtype([('M1', 'i'), ('lo1', float), ('la1', float), ('M2', 'i'), ('lo2', float), ('la2', float), ('Dir1', float), ('Dir2', float)])
346     return np.asarray(zip(M1, lo1, la1, M2, lo2, la2, dir1, dir2), dtype = dt)
347
348 #The functions needed to read raster data.
349 def pixel(file, x, y):
350     """
351     This function returns the closest pixel-values of a coordinates in a
352     os.geo-instance.
353     Inspired by the lecture slides from the following website:
354     http://www.gis.usu.edu/~chrisg/python/2009/lectures/ospy\_slides4.pdf
355     """
356     xOrigin = file.GetGeoTransform()[0]
357     yOrigin = file.GetGeoTransform()[3]
358     pixelWidth = file.GetGeoTransform()[1]
359     pixelHeight = file.GetGeoTransform()[5]
360     rasterx = int(round((x - xOrigin) / pixelWidth))
361     rastery = int(round((y - yOrigin) / pixelHeight))
362     return rasterx, rastery
363
364 def elevraster(linestring, gridresolution, file, data):
365     """
366     This function returns the interpolated values along a line on a
367     rasterized image.
368     """
369     from shapely.geometry import LineString
370     resx, resy = gridresolution
371     line = LineString(linestring)
372     length = line.length
373     #diagonal = np.sqrt(resx**2+resy**2)
374     diagonal = resx
375     resolution = int(round(float(length)/diagonal))
376     x, y, z, color, dist = [], [], [], [], []
377     arlist = np.linspace(0, length, resolution)

```

```

375     for distance in arlist:
376         point = line.interpolate(distance)
377         xp,yp = float(point.x), float(point.y)
378         rasterx,rastery = pixel(file,xp,yp)
379         x.append(xp)
380         y.append(yp)
381         z.append(data[rasterx,rastery])
382         dist.append(distance/1000.)
383     return x,y,z,dist,round(diagonal,2)
384
385 def transformer(filename):
386     """
387     This function reads the raster file.
388     """
389     import osgeo.gdal as gdal
390     from gdalconst import GA_ReadOnly
391     dataset = gdal.Open(filename, GA_ReadOnly)
392     gt = dataset.GetGeoTransform()
393     band = dataset.GetRasterBand(1)
394     nodata = band.GetNoDataValue()
395     #bandtype = gdal.GetDataTypeName(band.DataType)
396     array = band.ReadAsArray(0, 0, dataset.RasterXSize, dataset.
RasterYSize).astype(np.float)
397     if nodata is not None:
398         array_n = np.zeros((np.shape(array)))
399         for i in np.arange(np.shape(array)[0]):
400             for j in np.arange(np.shape(array)[1]):
401                 if array[i,j] != nodata:
402                     array_n[i,j] = array[i,j]
403             else:
404                 array_n[i,j] = 0.0
405         return gt[1], gt[5], dataset, array_n
406     else:
407         return gt[1], gt[5], dataset, array
408
409 def Static_para_array(st, z_t, z_dt, precision=5):
410     """
411     This function returns the array of the static stability parameter
412     """
413     g0 = 9.80665 # m/s2
414     p1 = pressure_reduced(st)*100.#Pa
415     t = new_matrix_reader("t",st) ['t'] #degrees Celsius
416     date = new_matrix_reader("t",st) ['Date']
417     time = new_matrix_reader("t",st) ['Time']
418     dt = new_matrix_reader("dt",st) ['dt'] #degrees celsius
419     RH = new_matrix_reader("RH",st) ['RH']/100. #mixing ratio
420     if z_t == 2.0:
421         z1 = z_t
422         z2 = z_dt
423         t1 = t
424         t2 = t1 - dt
425     if z_t != 2.0:
426         z1 = 2.0
427         z2 = z_t

```

```

428         t2 = t
429         t1 = dt + t2
430         rho = -0.005150475461058842*t2 + 1.296159304053393 #kg/m3
431         p2 = p1 - rho*g0*z2 #Pa
432         Q1 = 0.622*RH*sat_vap_p_ar("liq",t1)/p1
433         Q2 = 0.622*RH*sat_vap_p_ar("liq",t2)/p2
434         static_par = (t1*(1 + 0.611*Q1)-(t2*(1 + 0.611*Q2)))/(z1 - z2)
435         return date,time, z1, z2, np.around(static_par,decimals = precision)
436
437 def CaseSelection_loop(path, stolerance = 0.1, wdtolerance = 5.):
438     """
439     This function reads all the files to find a matching pair of
440     combination.
441     """
442     import datetime, csv
443     infile = open('zlist.txt', 'r')
444     lines = infile.readlines()
445     st_z, z_t, z_dt = [], [], []
446     for line in lines:
447         row = line.strip('\n').split(' ')
448         st_z.append(eval(row[0]))
449         z_t.append(eval(row[1]))
450         z_dt.append(eval(row[2]))
451     tif_filename_f = "Elevation/HEIGHT_XX_P_10_CLIP.tif"
452     resx_f, resy_f, set_f, dataset_f = transformer(tif_filename_f)
453
454     c = comb_RT90_array()
455     for i in np.arange(len(c)):
456         filename = path + "/" + "%d-%d-cases_stol_%.2f_wdtol_%.1f.txt" %
457         (c['M1'][i], c['M2'][i], stolerance, wdtolerance)
458         print filename
459         print i+1, c['M1'][i], c['M2'][i]
460         rf = [resx_f, resy_f]
461         a = [np.array((c['lo1'][i], c['la1'][i])), np.array((c['lo2'][i],
462         c['la2'][i]))]
463         x_f, y_f, z_f, dist_f, diagonal_f = elevraster(a, rf, set_f, dataset_f)
464         dist = round(dist_f[-1] - dist_f[0], 5)
465         std = np.std(z_f)
466         z_M1 = z_f[0]
467         z_M2 = z_f[-1]
468
469         s_d1 = new_matrix_reader("s_par_new", c['M1'][i])['Date']
470         s_t1 = new_matrix_reader("s_par_new", c['M1'][i])['Time']
471         s1 = new_matrix_reader("s_par_new", c['M1'][i])['s_par']
472         wd1 = new_matrix_reader("wd", c['M1'][i])['wd']
473         ws1 = new_matrix_reader("ws", c['M1'][i])['ws']
474
475         s_d2 = new_matrix_reader("s_par_new", c['M2'][i])['Date']
476         s_t2 = new_matrix_reader("s_par_new", c['M2'][i])['Time']
477         s2 = new_matrix_reader("s_par_new", c['M2'][i])['s_par']
478         wd2 = new_matrix_reader("wd", c['M2'][i])['wd']
479         ws2 = new_matrix_reader("ws", c['M2'][i])['ws']
480
481     #Setting boundaries

```

```

479     g0 = 9.80665 # m/s2
480     cp = 1003.78597374 # J/kgK
481     varGamma = round(g0/cp,5)
482     us = - varGamma + round(varGamma*stolerance,5)
483     ls = - varGamma - round(varGamma*stolerance,5)
484     u1 = c['Dir1'][i] + wdtolerance
485     l1 = c['Dir1'][i] - wdtolerance
486     u2 = c['Dir2'][i] + wdtolerance
487     l2 = c['Dir2'][i] - wdtolerance
488
489     counter = 0
490     with open(filename,'wb') as outfile:
491         cw = csv.writer(outfile, delimiter = ' ')
492         cw.writerow([datetime.datetime.today()])
493         cw.writerow(['Mast 1', 'Mast 2', 'Date (1)', 'Date (2)', '
Static stability parameter (1)', 'Static stability parameter (2)', '
Wind direction (tol_1)', 'Wind direction (tol_2)', 'Wind direction (1)
', 'Wind direction (2)', 'Wind speed (1)', 'Wind speed (2)', 'Forest
height (1)', 'Forest height (2)', 'Standard deviation (s)', 'Distance'
])
494         for j in np.arange(len(wd1)):
495             for k in np.arange(j,len(wd2)):
496                 if l1 <= wd1[j] <= u1 or l2 <= wd1[j] <= u2 or l1 <=
wd2[k] <= u1 or l2 <= wd2[k] <= u2:
497                     if ls <= s1[j] <= us and ls <= s2[k] <= us:
498                         c1 = datetime.datetime.combine(stringtodate(
s_d1[j]), stringtotime(s_t1[j]))
499                         c2 = datetime.datetime.combine(stringtodate(
s_d2[k]), stringtotime(s_t2[k]))
500                         if c1 == c2:
501                             cw.writerow([c['M1'][i],c['M2'][i], c1, c
2, s1[j], s2[k], wd1[j], wd2[k], c['Dir1'][i], c['Dir2'][i], ws1[j],
ws2[k], z_M1, z_M2,std, dist])
502                             counter+=1
503                             break
504                             break
505                             break
506                 cw.writerow(['Total number of cases: ', counter])
507     if counter == 0:
508         os.remove(filename)
509
510 def CaseSelection_main():
511     """
512     This function initiates the case selection loop.
513     """
514     import os, shutil
515
516     parentfolder = "Case selection_new"
517     #if os.path.isdir(parentfolder):
518     #    shutil.rmtree(parentfolder)
519     #os.mkdir(parentfolder)
520
521     s_tol = np.arange(0,11,5)/100.
522     dir_tol = np.arange(0,16,2.5)

```

```
523     for st in s_tol:
524         for di in dir_tol:
525             foldername = parentfolder + "/" + "CaseSelection_stol_%.2
f_dirtol_%.1f" % (st,di)
526             os.mkdir(foldername)
527             counter = CaseSelection_loop(foldername, st, di)
528             if len(os.listdir(foldername)) == 0:
529                 shutil.rmtree(foldername)
```



Bibliography

- 107, 108 & 105T Temperature Probes & 41303-5 Radiation Shield (2005). Campbell Scientific. URL: http://web2.mendelu.cz/af_217_multitext/meteo/arboretum/doc/107_108_105t.pdf (visited on 03/06/2016) (cit. on pp. 9, 10).
- Arya, S. Pal (1988). *Introduction to micrometeorology*. 1st ed. Academic Press (cit. on pp. 26, 32, 36, 76).
- Atkins, Peter and Julio de Paula (2009). *Elements of physical chemistry*. 5th ed. Oxford University Press (cit. on pp. 11, 22, 34, 38).
- Borás, Jan (2015). “Forest model validation”. In: *WindSim user meeting*. URL: http://www.windsim.com/documentation/UM2015/1506_WindSim_UM_WS_Jan_Borras.pdf (visited on 08/06/2016) (cit. on pp. 75, 77).
- Buck, Arden L. (1981). “New equations for computing vapour pressure and enhancement factor”. In: *Journal of Applied Meteorology* 20, pp. 1527–1532 (cit. on p. 38).
- Busch, Karsten (2015). “Forest Modeling inside WindSim 7.0 – interim report”. In: *WindSim user meeting*. URL: http://www.windsim.com/documentation/UM2015/1506_WindSim_UM_WKN_Karsten_Busch.pdf (visited on 08/06/2016) (cit. on p. 75).
- Church, Gemma (2016). “Where to put a wind farm”. In: *Scientific Computer World*. URL: <http://cmr.no/applications/wind-farm-layout-planning/> (visited on 05/06/2016) (cit. on p. 1).
- Corine Land Cover 2006 raster data* (2012). European Environment Agency. URL: <http://www.eea.europa.eu/data-and-maps/data/corine-land-cover-2006-raster-2#tab-gis-data> (visited on 07/06/2016) (cit. on p. 18).
- Council, Global Wind Energy (2015a). *Global Wind Report 2014*. Brussels, Belgium (cit. on p. 2).

- Council, Global Wind Energy (2015b). *Global Wind Statistics 2014*. Brussels, Belgium (cit. on p. 2).
- Crasto, Giorgio (2005). "Forest modeling —A canopy model for WindSim 4.5". WindSim user meeting. URL: http://www.windsim.com/documentation/papers_presentations/0506_tonsberg/um_crasto.ppt (visited on 02/06/2016) (cit. on pp. 25, 26).
- Daugherty, Robert L., Joseph B. Franzini, and E. John Finnemore (1989). *Fluid Mechanics with engineering applications*. SI metric. McGraw Hill and Company (cit. on p. 23).
- Devore, Jay L. and Kenneth N. Berk (2012). *Modern Mathematical Statistics with Applications*. 2nd ed. Springer-Verlag New York (cit. on pp. 12, 39).
- Dingman, S. Lawrence (2002). *Physical Hydrology*. 2nd ed. Waveland Press Inc. (cit. on pp. 25, 39).
- Email correspondence with the Norwegian met. office 07/06/2016* (2016). yr.no (cit. on p. 76).
- Enhetligt geodetiskt referenssystem* (2009). Lantmäteriet. URL: https://www.lantmateriet.se/globalassets/kartor-och-geografisk-information/gps-och-matning/referenssystem/inforande_av_nya_referenssystem/info_blad-3.pdf (visited on 03/06/2016) (cit. on p. 18).
- Finnemore, E. John and Joseph B. Franzini (2009). *Fluid Mechanics with Engineering Applications*. 10th ed. McGraw-Hill Education (cit. on pp. 23, 34).
- Fischer, Ronald A. (1934). *Statistical methods for research workers*. 5th ed. Olivier and Boyd. URL: http://www.haghigh.com/resources/materials/Statistical_Methods_for_Research_Workers.pdf (visited on 03/06/2016) (cit. on pp. 12, 13).
- Geomatica Core - Technical Specifications* (2016). PCI geomatics. URL: <http://www.pcigeomatics.com/pdf/geomatica/techspecs/2016/Geomatica-Core.pdf> (visited on 03/06/2016) (cit. on p. 7).
- Gravdahl, A. R. and K. Harstvedt (2000). "Assessments of wind resources along the Norwegian coast". In: *German Wind Energy Conference*. URL: http://www.windsim.com/documentation/papers_presentations/0006_dewek/dewek_2000_proceedings.pdf (visited on 03/06/2016) (cit. on p. 3).
- Gravdahl, Arne (2016). "Personal Communication" (cit. on p. 30).
- Greenpeace International Global Wind Energy Council, SolarPowerEurope (2015). *Energy (R)evolution. A sustainable world. Energy Outlook 2015. 100% Renewable Energy for all*. (Cit. on p. 2).

- GSD-general map, vector - product description* (2016). Lantmäteriet. URL: <https://www.lantmateriet.se/globalassets/kartor-och-geografisk-information/kartor/produktbeskrivningar/overshmi.pdf> (visited on 03/06/2016) (cit. on pp. 6, 18).
- HMP35D Humidity and Temperature Probe - Data sheet* (1991). Vaisala. URL: https://s.campbellsci.com/documents/us/manuals/vaisala_hmp35a_manual.pdf (visited on 03/06/2016) (cit. on p. 15).
- Holton, James R. (2004). *An introduction to dynamic meteorology*. 4th ed. Elsevier Academic press (cit. on pp. 22, 26).
- How do you measure air moisture?* (2016). SMHI. URL: <http://www.smhi.se/kunskapsbanken/meteorologi/hur-mats-luftfuktighet-1.6959> (visited on 03/06/2016) (cit. on p. 16).
- How do you measure air pressure?* (2015). SMHI. URL: <http://www.smhi.se/kunskapsbanken/hur-mats-lufttryck-1.23830> (visited on 03/06/2016) (cit. on p. 16).
- IEA (2009). *Technology Roadmap - Wind Energy*. URL: http://www.iea.org/publications/freepublications/publication/wind_roadmap.pdf (visited on 03/06/2016) (cit. on p. 1).
- Installation av väderradar vid Leksand* (2002). SUNET. URL: <http://proj.sunet.se/TP8/pres/smhi2.pdf> (visited on 03/06/2016) (cit. on p. 16).
- Jacob, Daniel J. (1999). *Introduction to atmospheric chemistry*. 1st ed. Princeton University press (cit. on p. 33).
- Kato, Yukitaka, Michihisa Koyoma, and Yasuhiro Fukushima Takao Nagakaki (2016). *Energy technology roadmaps of Japan - Future energy systems based on feasible technologies beyond 2030*. Springer Japan (cit. on p. 1).
- Katul, Gabriel G., Larry Mahrt, Davide Poggi, and Christophe Sanz (2004). "ONE- and TWO-Equation Models for Canopy Turbulence". In: *Boundary-Layer Meteorology* 113.1, pp. 81–109. URL: <http://dx.doi.org/10.1023/B:BOUN.0000037333.48760.e5> (cit. on p. 27).
- Langtangen, Hans Petter (2011). *A primer on scientific programming with Python*. 2nd ed. Springer (cit. on p. 29).
- Moore, David S. and George P. McCabe (2006). *Introduction to the practice of statistics*. 5th ed. W.H Freeman and Company (cit. on p. 76).
- Murphy, D. M. and T. Koop (2005). "Review of the vapour pressures of ice and supercooled water for atmospheric applications". In: *Quarterly Journal of the Royal Meteorological Society* 131.608, pp. 1539–1565. URL: <http://dx.doi.org/10.1256/qj.04.94> (cit. on p. 38).

- NRG #40C Anemometer datasheet (2010). NRG systems. URL: <http://www.windandsun.co.uk/media/277060/NRG-40C-Anemometer-data-sheet.pdf> (visited on 03/06/2016) (cit. on p. 9).
- Persson, Olov (2000). “Hur mäts väderet?” URL: http://www8.tfe.umu.se/courses/elektro/elmat1/v35_00_da/Hemuppgifter/olov/SMHI%5C%20hur%5C%20m%5C%E4ter%5C%20man%5C%20v%5C%E4der.htm (visited on 03/06/2016) (cit. on p. 15).
- REmap 2030: A Renewable Energy Roadmap, June 2014 (2014). International Renewable Energy Agency. Abu Dhabi. URL: www.irena.org/remap (visited on 03/06/2016) (cit. on p. 1).
- Right, Anton (2011). “Renewable energy facts : The driving forces for turning to renewable energy technologies”. In: *Renewable green energy power*. URL: <http://www.renewablegreenenergypower.com/renewable-energy-facts-the-driving-forces-for-turning-to-renewable-energy-technologies/> (visited on 08/06/2016) (cit. on p. 2).
- Roache, P. J. (1997). “Quantification of uncertainty in computational fluid dynamics”. In: *Annual Review of Fluid Mechanics* 29.1, pp. 123–160. eprint: <http://dx.doi.org/10.1146/annurev.fluid.29.1.123>. URL: <http://dx.doi.org/10.1146/annurev.fluid.29.1.123> (cit. on p. 3).
- Saidi, Kais and Sami Hammami (2015). “The impact of {CO₂} emissions and economic growth on energy consumption in 58 countries”. In: *Energy Reports* 1, pp. 62–70. URL: <http://www.sciencedirect.com/science/article/pii/S235248471500013X> (cit. on p. 2).
- Sanz, Christophe (2003). “A Note on $k - \epsilon$ Modelling of Vegetation Canopy Air-Flows”. In: *Boundary-Layer Meteorology* 108.1, pp. 191–197. URL: <http://dx.doi.org/10.1023/A:1023066012766> (cit. on p. 27).
- SLU Forest Map (2010). Dept. of Forest Resource Management, Swedish University of Agricultural Science. URL: <ftp://salix.slu.se/download/skogskarta/2010/Data/Raster/Rt90/> (visited on 05/06/2016) (cit. on pp. 17, 45, 52, 59, 66).
- Sørensen, Bent (2004). *Renewable Energy — its physics, engineering, environmental impacts, economics & planning*. 3rd ed. Elsevier academic press (cit. on p. 2).
- The world of weather data* (2011). Thies Clima. URL: http://www.enercorpsensors.com/PDF/Wind_Instrumentation.pdf (visited on 03/06/2016) (cit. on p. 9).
- Tsilingiris, P.T. (2008). “Thermophysical and transport properties of humid air at temperature range between 0 and 100 °C”. In: *Energy Conversion and Management* 49.5, pp. 1098–1110. URL: <http://www.sciencedirect.com/science/article/pii/S0196890407003329> (cit. on pp. 34, 37).

“Turbulence models in Phoenix” (2008). In: *PHOENICS Encyclopaedia*. CHAM. URL: http://www.cham.co.uk/phoenics/d_polis/d_enc/turmod/enc_tu.htm (visited on 03/06/2016) (cit. on p. 24).

Versteeg, H K and W Malalasekera (2007). *Introduction to Computational Fluid Dynamics*. 2nd ed. Pearson Education Limited (cit. on pp. 3, 19–23).

Wyllen, Gordon J. Van, Richard E. Sonntag, and Claus Borgnakke (1994). *Fundamentals of Classical Thermodynamics*. 4th ed. Vol. 1. John Wiley & Sons (cit. on p. 35).

Zixiao, Jiang, María Bullido-Garcia, Jean-Claude Houbart, and Céline BEZAULT (2013). “CFD modeling of forest canopy flows: Input parameters, calibration and validation”. In: *EWEA 2013, Vienna, Austria: Europe’s Premier Wind Energy Event*. URL: http://www.meteodyn.com/wp-content/uploads/2013/03/PO_208_CFD-MODELING-OF-FOREST-CANOPY-FLOWS.pdf (visited on 08/06/2016) (cit. on pp. 75–77).

List of Figures

2.1	The location of the fourteen meteorological masts where wind speed and the associated wind direction and temperature have been measured from december 2007 and throughout 2011 in the wind farm areas “Björkhöjden”, “Ögonfågeln” and “Bodhögarna”.	7
2.2	The maps used to retrieve topographical data used in the case selection.	8
2.3	The roughness map used to as input data for the simulation.	8
2.4	The anemometers used to measure the wind speed at the different locations (<i>The world of weather data 2011; NRG #40C Anemometer datasheet 2010</i>).	9

2.5	The temperature sensors used to measure the temperature difference between 2m above ground level and the top of the met masts (<i>107, 108 & 105T Temperature Probes & 41303-5 Radiation Shield</i> 2005).	10
2.6	The location of the meteorological masts installed by SMHI and the wind measurement firm.	11
2.7	The distribution of relative humidity at the different meteorological stations around the four project sites. The number on top of the columns is sample number related to the potential dimension of the data set in percentage if it there were no data loss which is 8760.	15
2.8	The radiation shields protecting the temperature probe and the humidity sensor. The HMP35D is mounted in the same way as the HMP155 in the radiation shield (<i>How do you measure air moisture?</i> 2016).	16
2.9	The Vaisala PTB201A-barometer. The pressure sensor is the unit within the yellow rectangle (<i>How do you measure air pressure?</i> 2015).	16
2.10	A visualization of the raster file of forest height data in Sweden provided by SLU.	18
3.1	The flow of matter in and out of a fluid element (Versteeg et al. 2007).	20
3.2	The structure of the porosity of a canopy model (Crasto 2005).	25
4.1	Web of combinations of 14 meteorological masts and 2 possibilities.	30
4.2	The thermodynamic cycle of an air parcel Jacob 1999. The roman letter refers to each part of the cycle. t is temperature and p is pressure.	33
4.3	The specific heat capacity as a function of relative temperature, relative humidity plotted (Tsilingiris 2008).	34
4.4	A visualization of the regression analysis of tabulated data points for specific enthalpy related to temperature with a validity from $-33, 15^{\circ}\text{C}$ to $+46, 15^{\circ}\text{C}$	35
4.5	The visualization of the density formula (equation 4.19).	37
4.6	The structure of the case selection module that is used to find a case to perform the validation of the forest model (see A for the essential source code).	40
5.1	The top figure shows elevation and canopy profile of the terrain between the masts 6615 and 6605 with data from SLU (<i>SLU Forest Map</i> 2010). The red pole is the meteorological masts with the corresponding height. The bottom figure shows the profile of the grid.	45
5.2	A comparison of corresponding two forest models.	46

<i>List of Figures</i>	99
5.3 The grid of the flow model.	46
5.4 Comparison between measured and simulated data between the met. stations 6605 and 6615. The turbulence model standard “k- ϵ ”-model has been used. The green horizontal line is the canopy height.	47
5.5 Comparison between measured and simulated data between the met. stations 6605 and 6615. The YAP-modified “k- ϵ ”-model has been used. The green horizontal line is the canopy height.	48
5.6 Comparison between measured and simulated data between the met. stations 6605 and 6615. The RNG“k- ϵ ”-model has been used. The green horizontal line is the canopy height.	49
5.7 Comparison between measured and simulated data between the met. stations 6605 and 6615. The modified “k- ϵ ”-model has been used. The green horizontal line is the canopy height.	50
5.8 The top figure shows elevation and canopy profile of the terrain between the masts 6618 and 6609 with data from SLU (<i>SLU Forest Map 2010</i>). The red pole is the meteorological masts with the corresponding height. The bottom figure shows the profile of the grid.	52
5.9 A comparison of two corresponding forest models.	53
5.10 The grid of the flow model.	53
5.11 Comparison between measured and simulated data between the met. stations 6618 and 6619. The turbulence model standard “k- ϵ ”-model has been used.	54
5.12 Comparison between measured and simulated data between the met. stations 6605 and 6615. The YAP-modified “k- ϵ ”-model has been used.	55
5.13 Comparison between measured and simulated data between the met. stations 6618 and 6619. The RNG“k- ϵ ”-model has been used.	56
5.14 Comparison between measured and simulated data between the met. stations 6618 and 6619. The modified “k- ϵ ”-model has been used.	57
5.15 The top figure shows elevation and canopy profile of the terrain between the masts 6605 and 6619 with data from SLU (<i>SLU Forest Map 2010</i>). The red pole is the meteorological masts with the corresponding height. The bottom figure shows the profile of the grid.	59
5.16 A comparison of two corresponding forest models.	60
5.17 The grid of the flow model.	60
5.18 Comparison between measured and simulated data between the met. stations 6605 and 6619. The turbulence model standard “k- ϵ ”-model has been used.	61

5.19	Comparison between measured and simulated data between the met. stations 6605 and 6619. The YAP-modified “k- ϵ ”-model has been used.	62
5.20	Comparison between measured and simulated data between the met. stations 6605 and 6619. The RNG “k- ϵ ”-model has been used.	63
5.21	Comparison between measured and simulated data between the met. stations 6605 and 6619. The modified “k- ϵ ”-model has been used.	64
5.22	The top figure shows elevation and canopy profile of the terrain between the masts 6615 and 6618 with data from SLU (<i>SLU Forest Map</i> 2010). The red pole is the meteorological masts with the corresponding height. The bottom figure shows the profile of the grid.	66
5.23	A comparison of two corresponding forest models.	67
5.24	The grid of the flow model.	68
5.25	The grid data for the simulation in the area surrounding met. masts 6615 and 6618.	68
5.26	Comparison between measured and simulated data between the met. stations 6615 and 6618. The turbulence model standard “k- ϵ ”-model has been used.	69
5.27	Comparison between measured and simulated data between the met. stations 6615 and 6618. The YAP-modified “k- ϵ ”-model has been used.	70
5.28	Comparison between measured and simulated data between the met. stations 6615 and 6618. The RNG “k- ϵ ”-model has been used.	71
5.29	Comparison between measured and simulated data between the met. stations 6615 and 6618. The modified “k- ϵ ”-model has been used.	72



List of Tables

2.1	The geographical positions of the meteorological masts which has been used for recording wind speed and temperature. The elevation is the height above sea to the bottom of the meteorological mast. The coordinates refers to the map datum WGS84 and is the UTM-zone 33N.	6
2.2	The distances in kilometers between the most northern and southern meteorological stations and the SMHI automatic weather stations.	11
2.3	Statistics of the correlation between the temperature at SSVAB meteorological stations and Krångede A.	13
2.4	Statistics of the correlation between the moisture content in the air, as relative humidity, recorded 2m above ground level at the three weather stations operated and maintained by SMHI which are in the proximity of planned the wind farm.	14
2.5	Statistics of the correlation between the pressure at 2m above ground level at the three weather stations operated and maintained by SMHI and are located in the vicinity of the wind farm project site. The pressure is reduced to sea level pressure	14
4.1	An exempt of the table of the total combinations used in the case selection module with the focus on station 6611.	31
4.2	The values of C_p derived from linear interpolation between two values of the specific heat enthalpy of dry air as ideal gas property at 1 bar pressure retrieved from (Wyllen et al. 1994).	35
4.3	Results of the regression analysis of the enthalpy related to temperature with the python function <code>scipy.stats.linregress</code>	35

4.4	Results of the regression analysis of the density related to temperature with the python function <code>scipy.stats.linregress</code> . The r- and p-values and the standard error are truncated.	38
5.1	The 4 of 9 results from the case selection loop, sorted by distance. The number in the parenthesis refers to the mast number. The letter “z” and “s” symbols forest height and stability parameter.	42
5.2	The last of 9 results from the case selection loop, sorted by distance. The number in the parenthesis refers to the mast number. The letter “z” symbols forest height and stability parameter.	43
5.3	The seven closest pairs of meteorological masts.	43
5.4	The speed data in case 1.	44
5.5	The grid data for the for simulation in the area surrounding met. masts 6605 and 6615.	44
5.6	The speed data in case 2.	51
5.7	The grid data for the for simulation in the area surrounding met. masts 6618 and 6619.	51
5.8	The speed data in case 5.	58
5.9	The grid data for the for simulation in the area surrounding met. masts 6605 and 6619.	58
5.10	The speed data in case 6.	65
5.11	Data table retrieved from the stability parameter (equation 4.28).	73



Norwegian University
of Life Sciences

Postboks 5003
NO-1432 Ås, Norway
+47 67 23 00 00
www.nmbu.no