Class Generation for Numerical Wind Atlases

Risø National Laboratory
Wind Energy Department

and

The Technical University of Denmark
Informatics and Mathematical Modelling Department

Nicholas J. Cutler
s000144

30^{th} June, 2005

Contents

List of Figures

Abstract

A new optimised clustering method is presented for generating wind classes for mesoscale modelling to produce numerical wind atlases. It is compared with the existing method of dividing the data in 12-16 sectors, 3-7 wind speed bins and dividing again on the stability of the atmosphere.

Wind atlases are typically produced from many years of on-site measurements. Numerical wind atlases are the result of mesoscale model integrations based on synoptic scale wind climates and can be produced in as quickly as a day. 40 years of twice daily NCEP/NCAR Reanalysis geostrophic wind data (200 km resolution) is represented in typically around 100 classes, each with a frequency of occurrence. The mean wind speeds and directions in each class is used as input data to force the mesoscale model, which downscales to 5 km resolution while adapting to the local topography. The number of classes is to minimise the computational time for the mesoscale model while still representing the synoptic climate features.

Only tried briefly in the past, clustering has traits that can be used to improve the existing class generation method by optimising the representation of the data and by automating the procedure more. The Karlsruhe Atmospheric Mesoscale Model (KAMM) is combined with WAsP to produce numerical wind atlases for two sites, Ireland and Egypt. The model results are compared with The New Irish Wind Resource Atlas and wind atlases made from meteorological station measurements in Egypt.

The new clustering method has the ability to include wind data from different heights and thermal stability for the classification. The results show that the clustering method is able to produce results at least equivalent to the existing method results for both sites. A refined, general clustering procedure is devised which could improve the results for both sites, where the existing method requires two different parameter settings.

Resumé

En nye clustering metode er en del af den numeriske vindatlas metode (NWA). Det er sammenlignet med denne eksisterende metode at inddele vind data i 12-16 sektorer, 3-7 vind hastigheds grupper og dele igen efter stabilitet af atmosfæren.

Et vindatlas er typisk taget baseret på mange års on-site målinger. Numeriske vind atlasser er resultatet af mesoskala modellering på grundlag af vind klimaer på synoptisk skala og kan være produceret i som hurtigt som en dag. 40 år af to gange per dag NCEP/NCAR Reanalysis data (200 km opløsning) er repræsenteret i typisk omkring 100 klasser, hver med en frekvens og vejrsituation. Gennemsnit lige vind hastigheder af retninger i hver klasse er brugt som input data at drive modellen, og det skaleres ned til 5 km opløsning afhængig af topografien. Antallet af klasser begrænser beregningen til et minimum mens en god repræsentation af klimaet stadig opnås.

Clustering har kun været prøvet kortvarigt tidligere, og kan bruges til at forbedre den eksisterende klassegenereringsmetode. Clustering kan forbedre repræsentationen af data og automatisere fremgangsmåden. Karlsruhe Atmospheric Mesoscale Model (KAMM) er kombineret med WAsP er brugt til at lave numeriske vindatlasser for to steder, Irland og Egypten. Modellerens resultater er sammenlignet med den New Irish Wind Atlas og vindatlasser lavet fra meteorologiske målinger i Egypten.

Den nye clustering metode har fordelen at kunne inkludere vind-data fra forskellig højder samt termisk stabilitet for klassifikationen. Resultater viser clustering metoden kan opnå mindst lige så gode resultater som den eksisterende metode resultater for begge lokaliteter. En udvidet generel clustering metode, som kan forbedre resultaterne for begge lokaliteter er forslået. Den eksistenende metode behover to forskellige parametersæt for at opnå tilsvarende resultater.

Acknowledgments

Firstly, I'd like to thank Bo Hoffmann Jørgensen and Jake Badger at Risø National Laboratory for inviting me to undertake this masters thesis topic. I have been privileged to do a masters project with such close contact with two supervisors from the wind energy industry. This thesis could not have been done within six months without your expertise in numerical wind atlas method. You allowed me to concentrate on my part of the method, the classification, whiel you did the other parts of the method when required. This also allowed me to test the clustering method on two sites, which is critical for the conclusions in this thesis. As I hand over the new clustering method computer programs to you both, I wish you all the best for future numerical wind atlas constructed at Risø.

Thank you Bo for running my Introduction to Mesoscale Modelling special course last Autumn. You have been a very supportive and encouraging supervisor.

Thank you Jake for your support, particularly in running KAMM for the "second site", Ireland, which was critcial for this project. Also, your help in the last few days to discover a problem with the varying frequency calculation made a very big difference.

To my supervisor at the Technical University of Denmark, Bjarne Ersbøll, thanks for your assistance in clustering techniques and some very good advice and ideas at different times along the way. It turned out that image analysis *did* have something to offer to the wind class generation task (a.k.a. colour quantisation method)!

A special thanks goes to Stefan Heiske, who put in several hours to translate the significant parts of the Frey-Buness article [12] to English, as it was only available in German.

I would also like to thank the people involved in the data gathering and construction of the existing data for the Egypt and Ireland met station sites. To name a few people these are Ib Troen, Erik Lundtang Petersen, Lars Landberg and Niels Gylling Mortensen.

Finally, thank you to all my colleagues at Risø National Laboratory for the their help now and then, and making my time working at Risø National Laboratory pleasant and enjoyable.

Chapter 1

Introduction

This report is submitted as the final thesis for the two year masters program in wind energy at the Technical University of Denmark (DTU). The thesis was hosted by the Wind Energy Department of Risø National Laboratory in Roskilde, Denmark. The two supervisors from Risø, Bo Hoffmann Jørgensen and Jake Badger, have both been working on constructing numerical wind atlases for several years. This thesis is part of the ongoing development in numerical wind atlas construction at Risø. The supervisor from the Informatics and Modelling Department at DTU, Bjarne Ersbøll, has a great experience in statistics, including clustering techniques.

The method used at Risø to construct numerical wind atlases involves representing the large scale wind data with around 150 classes, each with a corresponding frequency of occurrence. This reduces the computation time greatly and possibly provides averaging of the large scale wind data cancelling out some errors. The task for this thesis is to explore clustering as an alternative to the existing method used at Risø to generate wind classes. This existing method is described in detail in section 3.2. The aim is to improve the accuracy of the resulting wind atlas, which is evaluated by comparing wind atlases made from measurements at particular sites. The potential reasons why clustering could improve on the existing method, are as follows.

A more automated classification procedure. Producing a good set of classes from the existing method at Risø requires much experience and specific knowledge about the site. One aim of the clustering method is to simplify the classification procedure at Risø. The eventual goal for Risø is to devise a classification method that can be used directly on any site. This thesis takes the first step in that direction using a new clustering method with results and conclusions from two sites, Ireland and Egypt.

Easier to tune. The existing classification program at Risø requires experience to set the parameters so that the program gives a valid output. The parameters for the new clustering algorithm can be easily tuned to achieve a desired objective in the clustering result.

Overall optimised representation of the wind climate. The existing wind classification method divides the data into a preset number of wind direction sectors evenly spaced. This part of the existing method does not consider the actual data for making the class boundaries. Hence, there is room for improvement in the classification scheme to make a better representation of the wind climate.

A representation considering different heights in the data The existing classification method divides the data considering only the wind speed, direction and inverse Froude number (which describes thermal stability, see section 3.1) at the lowest height. Even though these three variables are considered the most important, a clustering algorithm can take advantage of the possibility to consider other variables, such as the wind speed and direction at the next height above the ground. These wind speeds and directions at the second height could be weighted less important so that the original three variables still have the greatest importance in the classification outcome. By doing this, the wind shear between the first and second heights would be captured in the classes to some extent. Also, the second height, usually around 1500 m, could be the elevation in some parts of the domain used in the mesoscale model (domain size is typically around 500 $\times$ 500 km). In this case, the wind at the second height is also important for the wind flow around the entire domain.

Generating a higher number of classes. As computer technology advances, Risø's computer resources improve, including the speed with which mesoscale modelling simulations can be run at Risø. Hence, using up to 500 classes or possibly more is not as time consuming as it once was. The way the existing method is set up at Risø makes an increase of the number of classes to over 200 difficult. This would not be the case for the new clustering algorithm program, which simplifies Risø's numerical wind atlas procedure.

Clustering has been tried as the synoptic wind classification method for mesoscale modelling a few times before. However, most of these attempts have been brief and poorly documented as they, for example, do not state what clustering algorithm was used. In most cases the clustering method tried was shown to be inferior compared to methods similar to the existing method at Risø. Many of them note that a more detailed and carefully done clustering algorithm might improve the results. These are described in more depth in section 4.6.

In this report, chapter 2 introduces what is involved with constructing a numerical wind atlas and how it works. Section 2.2 discusses the status with numerical wind atlas construction today and describes some achievements that have been made. Chapter 3 discusses what is thought to be important for representing a wind climate for mesoscale modelling. Section 3.4.1 describes how a representation can be evaluated. Chapter 4 introduces the different clustering methods that could be used and section 4.6 discusses how clustering methods

have been applied to this and similar applications previously. Chapter 5 introduces the two sites used for testing the new clustering method, Ireland and Egypt. Chapter 6 compares the clustering results from a number of different clustering methods that were introduced in chapter 4. These results assist the selection of the clustering method to be used. In chapter 7 the chosen clustering method is described along with the procedure to use for wind classification. In chapter 8 the results for the actual simulations are shown for Ireland and the performance of the new clustering method is compared with the existing method. Chapter 9 compares the simulation results for Egypt. Finally, the conclusions are in chapter 10.

Chapter 2

Constructing a Numerical Wind Atlas

2.1 Introduction

For planning a wind farm it is important to know something about the wind resource of the specific site. A wind atlas is normally used to obtain this information. Wind atlases provide average annual wind speed and direction information for specific sites over a large area. Each wind atlas contains information about the distributions and magnitudes of the wind speed in different sectors. These are defined for standard heights above the ground and standard ground roughnesses. It is usually constructed from many years of wind measurements on the sites from meteorological weather stations (met stations). An example from the European Wind Atlas is shown in figure 2.1 for a site in Ireland. On roughness class is shown where the wind distribution is defined in 12 sectors and at 5 different heights. The Weibull distribution is used to describe a wind climate. For each height and sector as shown in figure 2.1 there are two numbers. The upper number is the A parameter and the lower number is the k parameter. These describe the magnitude and the shape of teh distribution respectively. The larger the k value, the more spread the data, as shown in figure 2.2.

Constructing a wind atlas is quite expensive, requiring many met stations and at least 10 years of measurements from them to obtain reliable climatic data. Even afer that, the wind resource is only accurately defined at the specific sites of the met stations. A microscale wind flow model is usally used to interpolate the wind atlas site to the wind farm location. If the nearest wind atlas station is not close enough, measurements from a new met station are required.

Another method to construct a wind atlas is mesoscale modelling. The wind atlas produced from this method is commonly referred to as a "Numerical Wind Atlas". Large-scale weather data (around 200 km resolution) is available on the internet over the whole world. This data is from the Reanalysis project [17] and is available from the NCEP/NCAR website [26]. The data is available

Dublin

53° 26′ 00″ N	06° 15′ 00″ W	UTM 29	E 682689 m	N 5924125 m	64 m a.s.l.

Situated 8.5 km N of the city centre of Dublin, with the suburbs extending to within 2.5 km of the site. The open sea lies 8 to 12 km away, between 040° and 150°. The Dublin/Wicklow mountains lie between 155° and 225°. The hills start about 18 km S of the airport and extend a further 60 km to the S. The highest peak rises to 930 m. The anemometer is well exposed except in the SSW where there is an enclosure with houses and trees. It is placed on top of a hut (3 × 3 × 3 m).

Height of anemometer: 12.0 m a.g.l. Period: 70010103-79123124

Roughness Class 1

z	0	30	60	90	120	150	180	210	240	270	300	330	Total
10	4.7	4.9	5.6	4.8	4.8	5.9	6.3	7.5	7.2	6.6	5.8	5.4	6.2
	1.48	1.53	1.72	1.56	1.51	1.74	1.83	2.16	2.15	1.99	1.85	1.83	1.82
25	5.6	5.9	6.7	5.8	5.7	7.0	7.5	8.9	8.5	7.8	6.9	6.5	7.3
	1.59	1.64	1.83	1.68	1.62	1.84	1.93	2.25	2.25	2.10	1.98	1.98	1.92
50	6.6	6.8	7.7	6.8	6.7	8.1	8.5	10.0	9.6	8.9	7.9	7.6	8.4
	1.78	1.82	2.01	1.88	1.81	1.99	2.08	2.39	2.41	2.27	2.20	2.22	2.08
100	7.8	8.1	9.0	8.1	7.9	9.3	9.9	11.4	11.0	10.3	9.3	9.0	9.7
	1.90	1.94	2.16	2.01	1.93	2.14	2.24	2.56	2.60	2.44	2.35	2.37	2.24
200	9.6	9.9	11.0	10.0	9.8	11.2	11.7	13.3	13.0	12.3	11.5	11.2	11.7
	1.81	1.86	2.07	1.92	1.85	2.06	2.15	2.49	2.51	2.35	2.25	2.26	2.18
Freq	3.1	4.2	5.4	5.7	6.9	9.0	5.2	9.5	17.5	17.4	10.3	5.7	100.0

Figure 2.1: Parts of the two pages for Dublin Airport from the European Wind Atlas

in 6-hourly averages at different heights (air pressure levels) since 1957. It is compiled from a large range of weather measurements from cup anemometers, weather balloons, satellites, buoys, etc.

The NCEP/NCAR data is derived from the geopotential height. The geopotential height is the height of a given pressure level, which depends on the air pressure and the vertical temperature distribution for that column of air. Essentially it is what is shown on a common weather map with the high and low pressure zones. At a high enough elevation, say above 3 km, the only two forces contributing the geopotential height are the pressure gradient from high to low pressure and the coriolis force, which is due to the earth's rotation. At these heights the wind derived from the geopotential height is in "geostrophic balance", meaning that the coriolis force and the pressure gradient are in equilibrium. Here, the geostrophic wind derived for NCEP/NCAR always flows parallel to the isobars in circles around the high and low pressure zones. Figure 2.3 demonstrates the direction of the geostrophic wind.

However, close to the surface the geopotential height is also associated with horizontal temperature gradients (from solar heating) and friction on the ground (roughness). Here the geostrphic wind is derived for the NCEP/NCAR data in the same way but it is not likely to be in geostrophic balance. Also, at higher

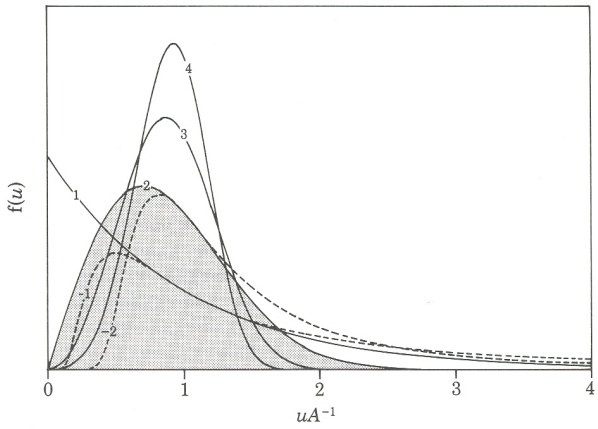

Figure 2.2: The shape of the Weibull distribution for different values of the shape parameter, k. This figure is taken directly from [33].

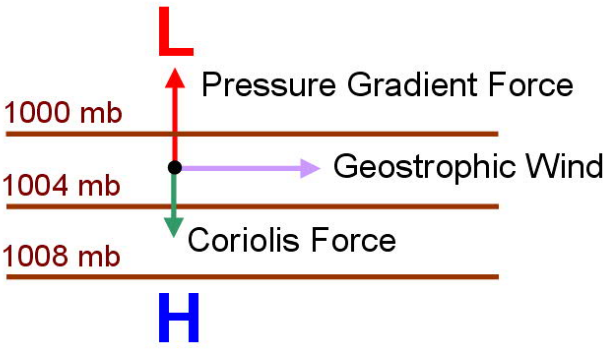

Figure 2.3: The geostrophic wind direction

latitudes, the coriolis force is stronger and the height with which geostrophic balance is observed does not need to be as high.

The true wind speed, or surface wind speed at a specific site would need to have the local elevation and roughness taken into account with the raw NCEP/NCAR data point wind data. Furthermore, this also depends on the surrounding weather conditions. Under influence of topography (ground roughness and orography), the surface wind tends to spiral in towards the low pressure points.

A mesoscale wind flow model, such as the Karlsruhe Atmospheric Mesoscale Model (KAMM) [2], is used to downscale the large scale weather to a mesoscale resolution (around 5 km). KAMM is a three-dimensional, non-hydrostatic atmospheric mesoscale model which assumes non-divergent wind fields in order not

to simulate sound waves. The topography [32] and roughness [31] for a specific site are available on the internet for as good as a 1 km resolution on the surface. This data is averaged to a 5 km resolution for KAMM. This is found to be the optimum resolution for Ireland [11]. The KAMM model requires these and one large scale weather situation as input (forcing) to give 5 km resolution wind speeds and wind directions. The single geostrophic weather situation contains a wind speed and direction at different heights, the temperature at different heights and the air pressure. The model uses this weather information as an initial condition, and iterates the application of computational fluid dynamics (CFD) until convergence occurs with the mesoscale wind conditions. Thus, in time the flow adapts to the topography. The model is run with stationary forcing, i.e. without radiation. The soil and water surface temperatures are given by the difference to the initial air temperature at the surface. Separately values are used for over water and land. An example of a KAMM output shown in figure 2.4 for Ireland. The KAMM model gives generally lower wind speed over the country where the ground is more rough than over the sea. The KAMM model also captures the wake of land masses as can be seen by the reduced wind speed over the sea as the wind emerges from Ireland at the bottom of the map and from the hills on edge of Scotland in the top right corner of the map. Note how the wind direction is referred to by the direction the wind comes *from*. The tallest mountain on Ireland is on the east coast at coordinates (300, 200). The KAMM model output shows a speed up the wind over this mountain since the wind speeds shown are at a constant 50 m above ground level. This is the standard way to refer to a wind direction, and every wind direction mentioned or plotted in this report follows this definition.

A collection of geostrophic weather situations are made for the KAMM model to be run on each. The results are compiled together to construct the Numerical Wind Atlas for each 5 km grid. Some mesoscale models are run on every data point in the data set. This requires a very fast model, and fast computer resources. It is thought that classifying the wind data tends favourably average out the innacuracies in the reanalysis data. Both methods have their advantages and disadvantages. The KAMM model uses a great deal of computer resources and it is therefore impractical to run the model on each NCEP/NCAR data point over the available 40 years as this amounts to the order of 50000 data points. It is intuitive that it would be a waste of resources to run the KAMM model separately on near-identical weather situations that may occur in the NCEP/NCAR data. Thus, the climate from the NCEP/NCAR data is represented in classes, each with a frequency of occurrence. The members of each class are similar enough such that they can be represented by one mean weather situation.

The existing method to make this classification used at Risø in recent times is the similar to the method currently used by other parties concerned with Numerical Wind Atlas construction. The basis of this method and the extra innovations added by Risø are explained in section 3.2. This thesis investigates the use of a statistical technique, clustering, to make more optimal classes and improve on the results of the existing method.

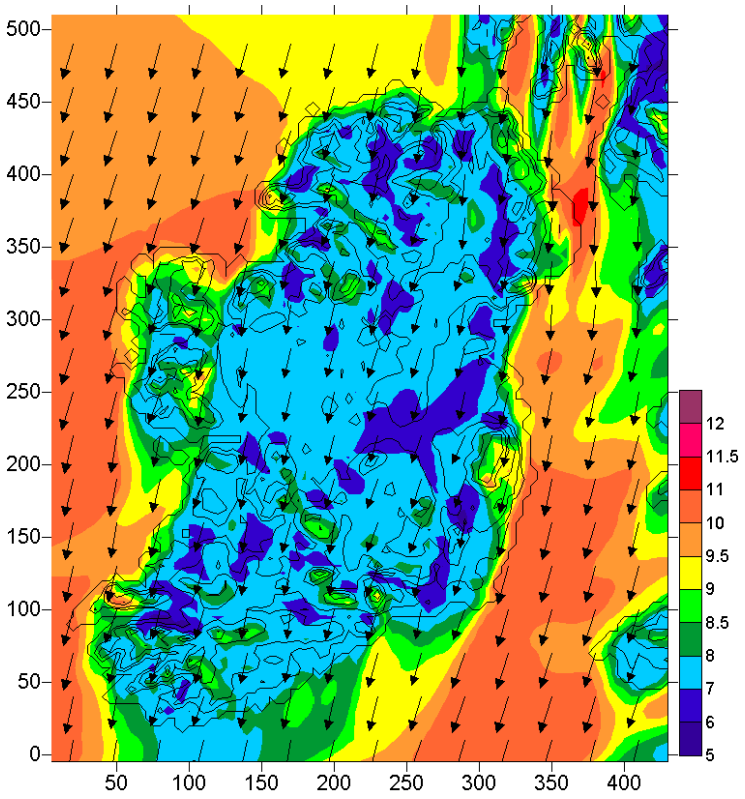

Figure 2.4: An example KAMM output at 50 m height over Ireland with a wind forcing of 13.3 ms^{-1} and from 47° (NE).The orographic contour lines are every 100 m and also include the 50 m line. The colours represent the wind speed and the legend is in ms^{-1}. The axis values are in km. The arrows represent the wind direction and their length also represents the wind speed. Each arrow represents a 5 km grid point in the KAMM domain, but only 1 in 36 arrows are shown.

2.1.1 Mesoscale modelling procedure summary

In short, the procedure to construct a numerical wind atlas with mesoscale modelling is summarised in the following.

1. Collect synoptic wind data (NCEP/NCAR Reanalysis) and topographic data for the region of interest.

2. Represent the wind climate data with a manageable number of classes, each with a frequency of occurrence. Some mesoscale models are run on the entire data set, so no classification is required.

3. Each of the class mean wind speeds and directions (class centroids) are

used as a forcing for a KAMM simulation. The flow in the model adapts to the topography and the result is obtained downscaled to a 5 km resolution after convergence.

4. Combine the results with the frequencies to construct the numerical wind atlas for the region.

2.1.2 Modelling weather phenomena

As described above, WAsP and KAMM are both wind flow models, but on different scales. WAsP models over a microscale domain (up to around 20 x 20 km) and KAMM models over a mesoscale domain (up to around 500 x 500 km). Different weather phenomena occur at these different scales and the models can only capture what they see. This is the reason why it is recommended to use both models in combination to assess a wind climate for a site [11].

On the mesoscale, weather phenomena such as wind channelling in mountain-valley systems occur. This strong streamline of wind can affect nearby locations in complex terrain, where the wind could be flowing in the opposite direction. These effects can be captured by mesoscale models such as KAMM but it is more difficult to capture them with WAsP as the domain required is very large. It might also involve adjusting WAsP's internal parameters such as the inversion layer height.

Mesoscale models also model the depth of the atmosphere and hence can capture effects on the wind such as atmospheric stability. If the temperature decreases with height, there is warmer, less dense air sitting on top of colder, heavier air. This situation is deemed as stable. The opposite situation is unstable. In the mesoscale, stability affects the wind flow around hills. If the atmosphere is stable, the wind is likely to flow directly over hills. However, if the atmosphere is unstable, the colder air above the surface wind flow pushes down and the wind tends to flow around the hills. Stratification is one type of stability where the atmosphere is structured in separate layers, with a distinct boundary. In this situation the wind behaviour in one layer could be quite different to the behaviour in the next layer.

On the micro-scale, other wind phenomena occurs. For example, the situation can occur where the air at the bottom of a mountain is heated by the sun. This warmer air flows up the slope of the mountain and eventually cools at the top, where it flows down again. This cycle of wind is thermally induced and depends on cloud cover, season and time of day. These affects would be measured on site by a met station and would produce energy with a wind turbine. However, neither WAsP or KAMM are designed to model complicated systems such as this. It would take a very complicated model to capture these local effects and it is likely the computational resources required is impractical. Hence, predicting a wind climate using a nearby met station and WAsP or using the geostrophic wind data and KAMM, is a difficult task. However, research continues in attempt to improve the accuracy to within acceptable limits.

2.2 Previous achievements

The numerical wind atlas method is currently being used around the world for wind energy applications. Not all methods are the same as described in section 2.1.1. There are many variations, including the mesoscale model used. The different mesoscale models all use the basic CFD flow equations but have different features (e.g. hydrostatic/non-hydrostatic, modelling tree canopies) and different computation times. Some mesoscale models are fast enough to allow for the possibility of running the model on each NCEP/NCAR data point. This is often done for a 10-year period rather than the 40 years however, and there are advantages and disadvantages with each type of method used.

Some the numerical wind atlasas constructed to date are described in the following. The method of statistical-dynamical downscaling is described by Frey-Buness in 1995 [13]. This method has been used and developed at Risø over the past 10 years. An original Irish Wind Atlas was published in the European Wind Atlas [33] in 1989. In 1994 a New Irish Wind Atlas was made individually using twice as many years of wind data as 20 years was now available [18]. In the same paper the mesoscale/microscale modelling combination using KAMM and WAsP is introduced. A basic cluster analysis was made in the u-v space of the lowest height on a two year period and initial KAMM simulations were made. The work was completed in 1997 [9]. Here the previous clustering attempt was shown to be inadequate, one reason being that the two year period used was a non-representative sample of overall climate of Ireland. The results from KAMM and WAsP showed that almost half of the power (43%) was lost with the results based on the clustering analysis. A new classification was made with 12 sectors, each with 5 or 6 wind speed bins, similar to the method used at Risø in recent times. This classification was performed on 10 years of data and the article does not say why the the clustering was not performed on the ten year data set. The new classification gave fair results, the biggest errors being an underprediction of the amount of weak winds and somewhat high wind speeds as well. The reason for this was put to the KAMM model's neglection of diurnal cycles, transient disturbances (like weather fronts), the poor grid resolution used (10 km) and due to the input geostrophic wind classes. These wind classes did not include any atmospheric stability definition and no thermal forcing was used in the KAMM simulations. It was suggested that using a higher grid resolution and producing the wind classes from a carefully done clustering algorithm, incorporating more dimensions to include thermal stability could improve the results. All of these suggestions are implemented in this report.

In 2001, Risø constructed numerical wind atlases for Denmark, Ireland, northern Portugal and Galicia and the Faroe islands [11]. They presented the mesoscale/microscale model combination technique using KAMM and WAsP. They employed Risø's existing classification method as described in section 3.2. They concluded that a 5 km resolution in the mesoscale is good enough for the relatively flat areas of Denmark and Ireland. However, better results were obtained for a higher resolution in northern Portugal. They also found it was more accurate to use WAsP to remove the topographic effects around the local

site for a direct comparison with the local wind atlas made from measurements.

Risø also initially constructed a numerical wind atlas for the Gulf of Suez in Egypt in 1999 using KAMM only. [8] describes the KAMM model set up in detail. The article mentions problems with southerly wind predicted in the gulf too often by KAMM, causing a rapid decrease in the wind speeds predicted at the southern end of the gulf. It also mentions that the gradients of wind speed in the results are quite steep and hence it is very important to use the exact position of the stations for comparison. [25] extends the results for Egypt using the KAMM-WAsP combination. However, as with the results using only KAMM in [8], the speeds and power densities are found to be somewhat underestimated.

Risø was also involved with predicting the wind climate in northern Finland in 1999 [10]. This project was concerned with the changing of the domain characteristics between the seasons due to snow and ice and level inversions during winter. An inversion is a high potential temperature gradient gradient in the immediate 500m above the surface. In this project the KAMM and WAsP models were not combined but compared. The classes were made using a very rough 8 sectors with fixed speed class boundaries giving valus of 3, 6, 10 and 16 ms^{-1} in each sector. An arbitrary 22 ms^{-1} class was also added to 3 sectors to represent the strong westerly winds in the data. These 35 classes were each split in 3 to make 105 classes - one in summer conditions, on in winter conditions and one for inversions. Despite the KAMM resolution being as high as 350m, it was still concluded to be too coarse for the specific locatoins, Pyhätunturi Fell and Sodankylä Observatory. The results found that WAsP did a better job at predicting the wind climates at Pyhätunturi since its higher, microscale resolution resolved the steep slopes of the terrain there. However, WAsP overpredicted the wind speeds at Sodankylä due to its inability to capture the extreme stratification occurring in the valleys. Many other challenges for prediction occur in polar regions including icing on the blades of the wind turbines.

In 1996, Risø was involved with the University of Karlsruhe to assess the wind climate of the Baltic Sea [1]. Here KAMM was used alone over a very large area of 1300 by 540 km. A clustering analysis was made over three dimensions, u-v space and the difference between air temperature and sea surface temperature, ΔT. 120 clusters were made. The results were not evaluated in comparison with measurements but qualitatively by observing the coastal and topographical effects.

In 1993, Frey-Buness performed the statistical-dynamical method to construct a numerical wind atlas for the Alpine region of mainland Europe [12]. The ECHAM mesoscale model was used used, which consists of time-mean hydrodynamic CFD equations for humidity, considering vorticity and divergence, temperature and ground pressure. Two classification schemes were used. One, labelled the "conventional method" consisted of dividing the geostrophic wind data into 8 sectors and dividing each of these into 4 groups based on season and humidity to give 48 classes. The other method used a complicated application of empirical orthogonal functions (EOF). (For more detail on this method, see section 4.6.) The results proved that the resolution of the mesoscale model was too coarse to capture the valleys and mountain tops in the Alps. Further the

EOF method was concluded to be not as good as the conventional classification method. On reason suggested for this was that the conventional method resolved the classes better spacially in the approaching flow to the regional model area.

In 1997, Mengelkamp performed statistical-dynamical downscaling to assess the wind resource of the Rhine valley [20]. The non-hydrostatic mesoscale model GESIMA (Geesthacht Simulation Model of the Atmosphere) was used alone without any microscale model. Cluster analysis was performed to make 143 clusters, which is described in more detail in section 4.6. This project was concerned with the modelling of forests and a few different ways of simulating this was tried. The results showed close estimations to the mean wind speed comparing with 7 meteorological stations (met stations) though the energy prediction errors were up to 20%. Three of the met stations only had three years of data and these were in less agreement with the simulation results. The results were very good considering that the height of the met station masts were only 10 m and that no microscale model was used to remove the local effects around the met stations.

2.3 Super computers

In 2004, Risø acquired a super computer, which they called Mary. With 240 processors, Mary is close to being in one of the fastest 500 computer servers in the World [28]. Mary's speed has allowed Risø, once the pre-processing is done, to perform KAMM simulations with 150 classes in around twenty minutes. Less than 10 years ago, some articles mention simulation times of around 24 hours for the same number of classes. Some technical specifications for Mary are shown in 2.1 and a picture of her is shown in 2.5.

Processors	240
Each Processor	Dell PowerEdge 750 3.2GHz Intel Pentium 1 MB cache, 2GB RAM
Login Management Server	Two Dell PowerEdge 2850's Intel Xeon 2.8 GHz processor, 4GB RAM
File Server	Two Dell PowerEdge 2850's Intel Xeon 2.8 GHz processor, 2GB RAM 2 Terabyte userdisk in EMC-SAN

Table 2.1: Technical Specification for Mary

2.4 The existing procedure at Risø

The following describes the pre-processing procedure used at Risø for collecting the data and generating the wind classes before running KAMM. The new

Figure 2.5: Mary's processors

clustering method replaces steps two and three below.

1. (a) A couple of shell scripts retrieve the NCEP/NCAR data from CD-ROMS [1]. The data finishes in binary format.

   (b) A Perl, C++ and 2 fortran programs convert the binary data files to text files, with a "*.d" extension. The data available is the year, month, day, hour, geostrophic wind speeds (U), geostrophic wind directions (DD), potential temperature (Tv), pressure (p) and humidity (q). All of these are given for the required number of heights, say, 0, 1500, 3000 and 5500m.

2. The text files are read in by the Perl and Fortran programs and the classes are made as described last week. A "*.dccc" file is produced containing all the statistics of the classification. A "*.lim" file is also produced containing a list of the classes. For each class the upper and lower boundary values are written for the wind direction, speed and inverse Froude number, along with the class number and frequency.

3. (a) The "*.lim" and original "*.d" files are read in by a Perl program (along with other programs). The class limits and frequencies were originally found for a certain height (typically 0m). The statistics (limits and means etc) for each class are found for the other heights required (typically 1500m, 3000m and 5500m) at the same NCEP/NCAR data point.

   (b) The frequencies are now found based on the original limits, for the surrounding NCEP/NCAR data points od interest. That is, the existing 40 years of data for these data points are used to find what frequency each of the chosen classes occur. New means are calculated for each of these. This is to be used as part of the post-processing,

[1] The CD-ROM data is only once every 12 hours. The NCEP/NCAR website [26] now has 6-hourly data but it was not ready for this project at Risø

when the wind atlas is put together. The frequencies of the classes are linearly interpolated from the centre, where they were created, to the frequency values calculated at the surrounding data points. The wind results for each class from KAMM are combined with the resulting varying frequencies of occurance across the domain to build the wind atlas. The program also produces a fixed frequency option, where the same frequencies as obtained with the original NCEP/NCAR grid point are used across the entire domain.

(c) A ".cl" file is produced which contains the centroids of each class. This file is used for initialising the KAMM simulations.

(d) A ".frq" file is produced containing the mean wind speed and direction for each class at the lowest height. It also contains the frequencies of all the surrounding NCEP/NCAR data points. When the KAMM simulation results are combined at each individual 5 km area, the frequency of each class is included. There are two options for creating these frequencies, fixed frequency and varying frequency. The fixed frequency option uses the same class frequencies of the original classes made across the entire domain. The second option is a feature of this existing procedure in that this frequency file can be used to make the frequencies vary across the domain. This means that the that when the KAMM output wind fields for each class are combined, the frequency used for each 5 km area will depend on its location, interpolated from the frequencies calculated at the nearest NCEP/NCAR data sets. This feature is particularly useful if the neighbouring NCEP/NCAR data points contain siginificantly different geostrophic wind data.

4. A ".ri" file is also produced which simply contains a list of the u and v wind speeds along with the frequency, the class number, the wind speed and directions for each class. This file is used directly for the KAMM simulations.

5. One KAMM simulation is made for each class. The initial wind speed, wind direction and temperature profiles are used to force the model. The flow in the model adapts to the topography and the result is obtained downscaled to a 5 km resolution after convergence.

6. The simulation results are combined at each 5 km area using a frequency file generated as a linear interpolation between the frequencies calculated in the ".frq" file.

7. Finally, WAsP is applied to the results at the specific sites to remove the effects of the mesoscale topography and create a wind atlas for the site region. Here, WAsP uses the same 5 km resolution maps that KAMM used. The same procedure is applied to the measurements at the sites to make the comparison wind atlases. Since the measurements were obtained at a point on a local scale, the maps used in WAsP in this case have a

much higher resolution and include nearby obstacles. This removes the local effects ensures a fair comparison between the numerical wind atlas results and the measured data at the sites.

Chapter 3

Representing a Wind Climate

When classes are made to represent a wind climate for mesoscale modelling, it is important that they represent the right variables that affect the outcome of the model simulations. Examples of such variables are wind speed and wind direction. It is not fully known, nor is it a simple answer as to how much each of these variables should be represented for optimum simulation results. The effect each variable has depends on the terrain features and atmospheric climate and hence, the location. The clustering method developed in this report allows for these variables to be easily tested for their influences on the model. The variables known to have some effect on the model are described below in section 3.1. Section 3.2 describes Risø's existing classification method and how the variables are treated. Section 3.3 outlines theoretically how the variables need to be treated in the clustering algorithm plus some other desired traits the algorithm could have. Section 3.4.1 describes how a representation is evaluated by looking at the variables individually.

3.1 The important variables for classification

The two most important variables for a geostrophic wind classification are wind speed and wind direction. The wind speed is the quantity being predicted for a wind atlas. For wind energy purposes, the wind energy is proportional to the cube of the wind speed. The directions are also important since the orography and roughness the wind flows over before arriving at a given site affects the wind speed. The orography and roughness is usually different in each direction from a given site.

Another important variable is the atmospheric stability. Atmospheric stability is related to the change in temperature with height and this affects the wind flow around the terrain. The actual temperature of the atmosphere almost always gets colder with height due to the effects of energy transfer and humidity.

It must be noted here that the air temperature referred to in the following is the virtual potential temperature. This is the temperature of the air with the effects of humidity and energy transfer removed. The virtual temperature can be first found from the temperature with the following equation.

$$T_v = \frac{T(1 + r_v/\epsilon)}{1 + r_v} \tag{3.1}$$

$$r_v = \frac{\epsilon e}{p - e} \tag{3.2}$$

where
ϵ is the ratio of the gas constants of air and water vapor ≈ 0.622, and
r_v is the mixing ratio of water vapour where,
e is the vapour pressure and,
p is the air pressure.

The virtual temperature is obtained from the geostrophic wind data in the raw data files. It is converted to virtual potential temperature with:

$$\theta_v = T_v(1/P)^{R_{cp}} \tag{3.3}$$

where
P is the air pressure in Pa, and
$R_{cp} = 287/1005 = 0.286$ is the ratio R/c_p where R is the gas constant and c_p is the specific heat.

The advantage of using the virtual potential temperature is that the problematic varying humidity and energy transfer in the air is removed so that the treatment of the thermal stability becomes simpler. The two extreme, but not uncommon cases of thermal stability are shown in figures 3.1 and 3.2 with the resulting wind behaviour. In the stable situation, the virtual potential temperature of the air rises with height. A air parcels close to the ground are colder, more dense, and hence heavier than the air parcels above. When it moves, it will tend to stay at the ground level and flow around hills. The effect is also more pronounced at lower wind speeds since the wind has more time to change direction to flow around hills.

In the unstable situation, a ground level air parcel will be lighter and less dense than air parcels above. This invites vertical flow and mixing and gravity pulls the heavier air down. In this situation, the wind will tend to flow over hills more as the vertical direction of flow is more natural. This behaviour is only a trend and the actual wind flow depends on the level of stability the steepness of the hill. If the atmospheric stability is neutral, the flow with hills will be a combination of over and around them. The mesoscale model can capture these effects if initialised with the right atmospheric stability conditions. Thus, the thermal stability is an important variable to capture in the clusters.

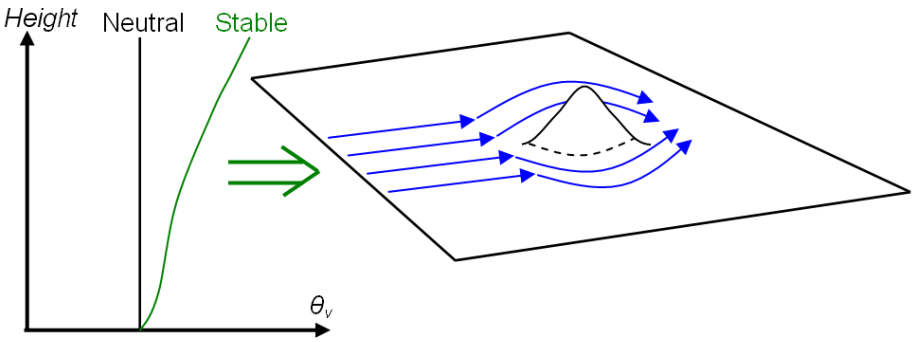

Figure 3.1: The wind will tend to flow around hills when the atmosphere is stable

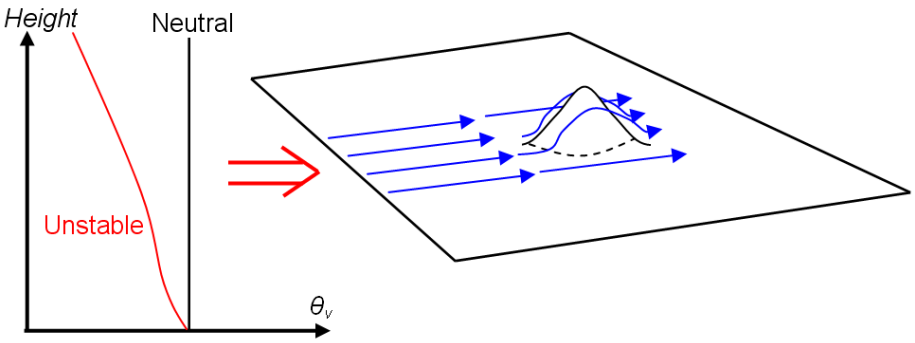

Figure 3.2: The wind will tend to flow over hills when the atmosphere is unstable

The thermal stability is described by the inverse Froude number. The definition used for the inverse Froude number is based on the ratio between buoyancy and inertia, and is shown in equation 3.4 for the first two heights in the domain. The formal definition for the Froude number is based on the ratio of inertia and gravity and is shown in appendix A.2.1 along with the derivation of equation 3.4 below. This quantity includes the square of the wind speed in the denominator which gives the situations with low wind speed more weighting. This is advantageous to the classification.

$$Fr_{1,2}^{-1} = \sqrt{\frac{gP(\theta_2 - \theta_1)}{S_1^2(\theta_2 + \theta_1)/2}} \tag{3.4}$$

where
g is acceleration due to gravity,
P is the pressure,

θ_i is the potential temperature at the ith height, and
S_0 is the wind speed at the lowest height of heights 1 and 2.

3.2 Risø's existing representation method

The existing wind class generation method used at Risø National Laboratory is described in detail below. This is also referred to as the "old method" in this report. To make the classes at Risø, a Perl program is run and it calls a Fortran 90 program which reads the NCEP/NCAR data and puts them into classes. The resulting class statistics are then written into an output file.

The important parameters set in the Perl program are as follows (the letters listing these are referred to in the following paragraphs).

A The number of sectors for the class divisions (Eg. 16).

B The nominal number of speed classes (Eg. 7).

C The number of stability classes (Eg. 2).

D The minimum frequency allowable for a class (Eg. 0.004%).

E The maximum number of classes per sector (Eg. 10).

F The minimum number of split speed classes per sector (Eg. 1).

G The frequency of the first speed bin class relative to the other classes (Eg. 0.7).

H The frequency of the last speed bin class relative to the other classes (Eg. 0.35).

I The minimum allowable wind speed, below which, the data are treated as "calms" (Eg. 0.1 ms^{-1}).

J The number of observations or greater (for setting the array size, eg. 25000).

Using these parameters, the Fortran 90 program divides the data in the following manner.

1 Evenly divides 360° into the number of sectors (A), centred at 0°. For example, if A is 16, the data is divided into the sectors where -11.25° $<$ direction $\leq$ 11.25°, 11.25° $<$ direction $\leq$ 37.75°, etc. The data with wind speeds below the calm threshold (I) are put into their own class at this stage. Statistics are calculated and stored for the data in each of the sectors. The data in each sector is sorted by increasing speed.

2 The decision is made as to how many classes there will be for each sector. Starts with the nominal number of speed classes, B, and takes into account C, D, E and F to produce:

- The number of speed classes, *NUCL*, and
- How many of these speed classes will be split into the desired number of stability classes (C), *NSPLIT*.

Hence the total number of classes for a sector is

$$NUCL + (C - 1) \times NSPLIT \tag{3.5}$$

For example, if $NUCL = 7$ and $NSPLIT = 2$, the total number of classes is 9 and they are divided as shown below.

	Speed Bins →						
Stability ↓	1	2	3	4	5	6	7
	8	9					

Table 3.1: Example class divisions for one sector

3 The measurements in each sector are divided such that each class has the same number of measurements, except:

- The first class (lowest wind speeds) has a weighting of G (Eg. 0.7), and
- the last class (highest wind speeds) has a weighting of H (Eg. 0.35), relative to the other classes.

Thus, the number of observations, NOT the speed values themselves, sets the speed boundaries between the speed classes within a sector. The data in each sector is hence divided into speed classes, where the first *NSPLIT* speed bins have more data then the others as these will be later split further into stability classes. For example, if B = 7, *NSPLIT* = 2, C = 2, G = 0.7, H = 0.35 and the number of observations in the sector was 805, the data would be divided into speed classes from low speeds to high speeds as follows in table 3.2.

Speed Bins →						
170	200	100	100	100	100	35

Table 3.2: Example number of data in speed bins

4 The lower speed bins are split into stability classes as atmospheric stability is thought to influence the mesoscale model more when the speed is low. The *NSPLIT* lowest wind speed classes are split into C stability classes. This is done in one of two ways as decided by the parameters in the Perl program.

From percentiles: The amount of data in the speed bin is divided evenly. This means the class boundaries are not dependant on the actual

Stability ↓	Speed Bins →						
	85	100	100	100	100	100	35
	85	100					

Table 3.3: Example number of data in speed bins with lowest two divided into two stability groups

Froude number values in the data. The example used in table 3.2 would become:

From values: Preset inverse Froude number limits from the Perl program are used as the boundaries for the classes. This means the number may not be evenly spread as follows.

Stability ↓	Speed Bins →					
	72 111	100	100	100	100	35
	98 89					

Table 3.4: Non-even spread example number of data in speed bins with lowest two divided into two stability groups

Finally the statistics are calculated for each class, stored, and all statistics are written to an output file.

3.2.1 Varying frequency calculation

Part of the numerical wind atlas procedure is that the class frequencies are recalculated for neighbouring NCEP/NCAR data sets when it is desired to have varying frequencies across the domain. The existing method stores the boundary values for each class for wind speed, wind direction and the inverse Froude number. The wind speeds in the new data set are transformed to allow for the change in the coriolis parameter with latitude as per equation 3.6.

$$S = cf \times S = \frac{\sin(lat)}{\sin(lat_0)} \times S \tag{3.6}$$

where
S is the speed values in the new data set,
lat is the latitude of the location of the new data set, and
lat_0 is the latitude of the original NCEP/NCAR data point on which the classification was made.

The frequencies are recalculated for the new transformed data set by simply using these boundaries and determining how many observations are assigned within them.

3.2.2 Interpolation for WAsP

In the final stage of the numerical wind atlas procedure, the numerical wind atlas is converted to a wind atlas using WAsP to remove the mesoscale topography effects. This is done at each 5 km grid of interest in the KAMM domain. Unfortuneately, around 150 classes is not enough data points to construct an accurate Weibull distribution. Furthermore, all the class centroids lie close to the middle of the sectors. Thus if 16 sectors are used, there are practically only 16 different values for the geostrophic wind direction in the representation. To solve this problem, "extra simulations" are created by splitting each geostrophic wind forcing into five values, the original data point and two on each side, direction-wise. A diagram of the splitting is shown in figure 3.3. The frequency is typically split evenly amongst the 5 new points. The new values lie along the line of interpolation between the original geostrophic simulation wind and the closest simulation wind in the next sector on each side. To find these closest centroids, the points in the region from 1/3 of the sector width to 4/3 of the sector width away are examined. The new data values lie 0.2 and 0.4 of the distance to the nearest data points along this line. The interpolations of these new data values are transformed to the corresponding interpolations of the simulation result winds at the specific 5 km site. Thus, this now gives 5 simulation "results" for every original one and this improves the resulting Weibull distribution constructed for the wind atlas.

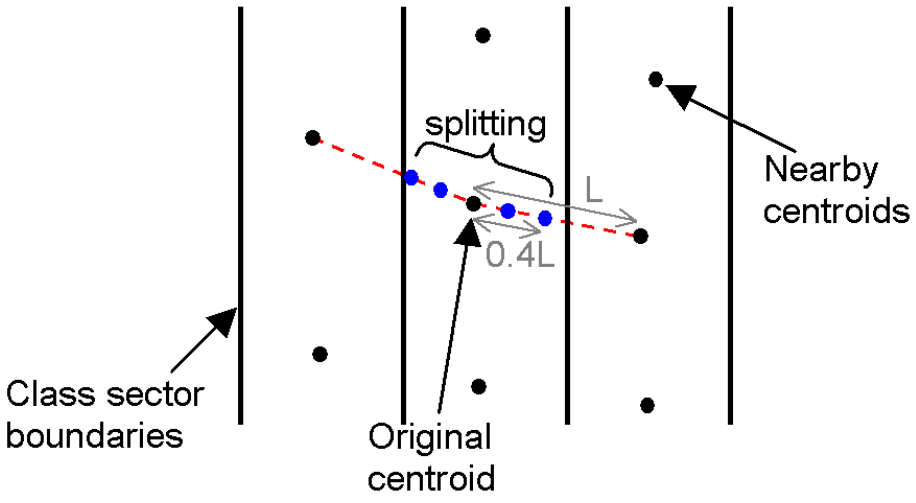

Figure 3.3: The method of interpolating extra wind classes for constructing the Weibull distribution in the wind atlas

3.3 Desired traits for a clustering algorithm

The data to classify are twice daily synoptic weather measurements. Each weather measurement contains wind speeds and directions, virtual temperature, pressure and humidity each at 4 different heights (typically around 0, 1500, 3000 and 5500m). The temperature, speeds and pressure are used to calculate the inverse Froude number between paris of adjacent heights using equation 3.4. The number of data units is 24836 in this case, which encompasses 34 years.

The general traits desired for a clustering algorithm are described in the following points. More specific discussions are presented in sections 3.3.1 - 3.3.3.

1 The clustering is able to consider up to eleven variables. Namely, these variables are the wind speeds and wind direction at four heights, plus three inverse Froude numbers calculated using the temperature difference between pairs of adjacent heights.

2 The user has control over the importance of each variable for the clustering. In other words, the user can control how much the clustering algorithm focusses on representing each variable.

3 The frequency of occurrence of each class is calculated for each NCEP/NCAR data point and the results are written to output files in the same formats as used now. Thus, this process would replace step three in the existing method (as described in section 2.4).

4 The efficiency needs to be very good since there are a large number of observations (24836). The computation goes up with the square of the number of observations in most clustering methods [14].

5 With finding wind classes for mesoscale modelling, it is important that the average (or centroid) of each class does not deviate much from the individual observations in that class. This suggests that the most important class criterion is that the error sum of squares within each class is minimised (see section 3.3.1, below). The inter-variance, or dissimilarity between classes does not seem important and perhaps is totally irrelevant. Thus many clustering techniques contain components which are possibly irrelevant to this study. In fact, this study is not actually about finding "clusters", but rather a representation of the data in which the error sum of squares within each class is minimised.

6 The clustering method output is able to calculate the frequencies of the clusters for other NCEP/NCAR data sets. This is so the varying frequency option feature in the existing procedure (see section 2.4) is possible to implement.

3.3.1 Distances and error sum of squares

When using clustering to classify a NCEP/NCAR data set, a measure of a "distance" between two data points is required. There are eleven variables

considered for this distance. These are the wind speeds at 4 different heights, the wind directions at 4 different heights and the 3 nverse Froude numbers describing the temperature profiles between adjacent heights.

The distance between 2 wind speeds or inverse Froude numbers is trivial. They are both linear continuous variables and the distance can always be based on the difference between the values. For example, the linear distance between two wind speeds, S_1 and S_2 is:

$$Distance = |S_1 - S_2| \tag{3.7}$$

The wind directions require some extra complication in the programming since they wrap around in a circular fashion (e.g. 1° is closer to 359° than to 10°). A condition needs to be applied which could be something like:

$$\text{if } |DD_1 - DD_2| > 180, \text{ then} \tag{3.8}$$

$$Distance = 360 - |DD_1 - DD_2|, \tag{3.9}$$

$$\text{otherwise, } Distance = |DD_1 - DD_2| \tag{3.10}$$

Sometimes the variance, or error sum of squares is required. This is related to the Euclidean distance. With wind speeds and the inverse Froude numbers, the standard linear formula can be used. For example, the error sum of squares for a set of m wind speeds is:

$$E = \sum_{j=1}^{m} (S_j - \bar{S})^2 \tag{3.11}$$

where
S_j is the speed value of the jth of m data points, and
$\bar{S}$ is the mean speed over the m data points.

Since this error sum of squares is the same as the variance multiplied by the number of data points, the standard variance formula (see equation A.1 on page 133) can be used so equation 3.11 is equivalent to:

$$E = \sum_{j=1}^{m} S_j^2 - (\sum_{j=1}^{m} S_j)^2/m \tag{3.12}$$

The variance of the set of speeds is calculated using equation 3.11 or 3.12 divided by the total number of values, in this case, m. The linear error sum of squares definition defined in 3.12 above can be applied to a set of directions on the condition that the directions are rescaled with additions of 360° so that the values do not cross 0°. For example, the set { 358°, 359°, 0°, 1°, 2° } would be scaled to { 358°, 359°, 360°, 361°, 362° }.

[6] describes another definition for the variance of a set of directions, the so-called angular variance. As described in Appendix A.1.2, the angular variance is defined as:

$$angular\ variance = 2(1 - r) \tag{3.13}$$

where
$r = \sqrt{\bar{u}^2 + \bar{v}^2}$,
$\bar{u}$ = mean wind speed in u direction, and
$\bar{v}$ = mean wind speed in v direction.

The angular error sum of squares could be defined as the value in equation 3.13 multiplied by the number of observations. The angular definition has the advantage that no checks, rescaling or additions of 360° are required before the formula is applied. The disadvantage is that it is usually not considered when general algorithms are written involving variances or error sum of squares. The linear and angular error sum of squares definitions are comapred on the Egypt data in figure 3.4. The plot shows that both are non-decreasing and either could be used depending on what is more convenient to implement in the chosen clustering algorithm(s).

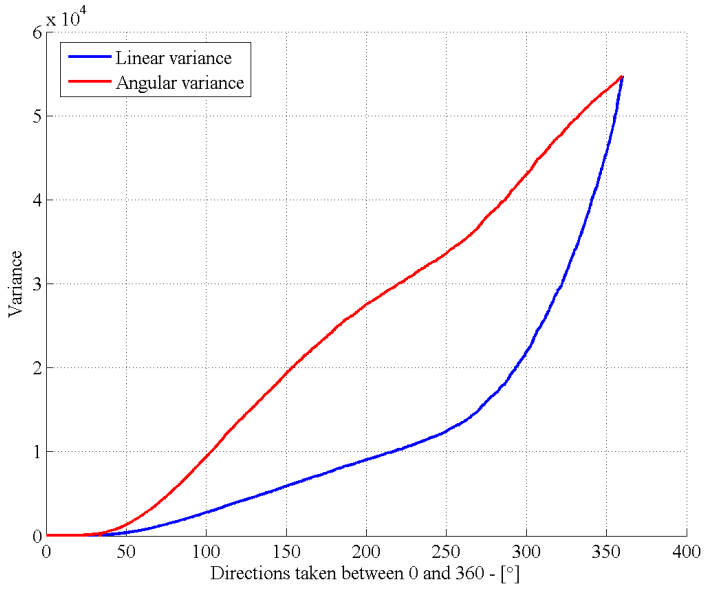

Figure 3.4: The difference between the angular variance and the linear varaince on sets of the Egypt data. For each direction value on the axis, the set of directions is taken between this value and 0°.

Another way the problem with directions can be dealt with is by transforming them to two new variables, $\sin(DD)$ and $\cos(DD)$. If the linear Euclidean distance is applied to these, the following result is obtained between two values, DD_1 and DD_2.

$$Distance = \sqrt{(\sin(DD_1) - \sin(DD_2))^2 + (\cos(DD_1) - \cos(DD_2))^2} \quad (3.14)$$

$$= \sqrt{1 + 1 - 2(\sin(DD_1)\sin(DD_2) + \cos(DD_1)\cos(DD_2))} \quad (3.15)$$

$$= \sqrt{2 - 2\cos[DD_1 - DD_2]} \quad (3.16)$$

The sin-cos definition has the advantage that the standard linear distance and variance formulae can be used directly on the values.

$\sin(DD)$ and $\cos(DD)$ are effectively the u and v components of the directions on the unit circle, since $\sin(DD) = u/S$ and $\cos(DD) = v/S$. DD_1 and DD_2 are thus plotted in figure 3.5. The figure shows that the distance between two data points with the directions represented with sin and cos, is the same as the straight distance across the unit circle.

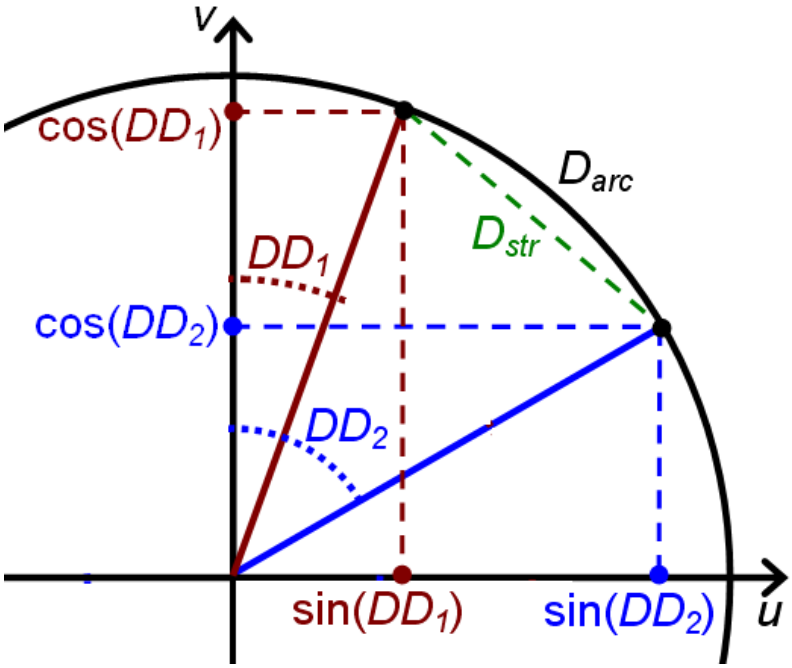

Figure 3.5: Two wind directions represented by sin and cos functions. Two ways to measure the distance between them are shown, the straight distance and the distance along the arc.

Hence, equation 3.16 represents the straight distance between two directions if placed on the unit circle. This is not the same as the arc length distance

between the directions, which would be obtained by using equation 3.10, above. However, individual directions will be ranked in the same order in terms of distance from each other using either definition. For directions within say, 30° of each other, these two definitions are nearly the same. The only difference is, as the span becomes greater than around 30°, the increase in distance using equation 3.16 is less than using 3.10. The effect is at its most extreme when the arc length distances are close to 180°. For instance, if the two arc length distances are 179° and 180°, equation 3.16 changes by only a very small amount, from 1.99992 to 2. This is a distance increase of 0.559% compared to only 0.0038%.

3.3.2 Combining the 11 variables

There are two things to consider when combining the eleven variables together for clustering, weighting and which norm to use.

Weightings are required on each variable since some variables are more important than others, and even more importantly, they have different scales (e.g. the units for wind speeds are metres per second while directions are in degrees or radians). Further, care must be taken as the wind speeds vary on a linear scale whereas the wind directions vary on a circular scale. The normal way to control the weightings of variables is to standardise the values to a larger or smaller standard deviation to make the importance more or less, respectively relative to the other variables. The more spread the values are on a variable, the further apart the clustering measures them to be. Hence, the clustering algorithm will tend to separate the data into more clusters based on this variable.

Once the weightings and hence the values of the variables have been established, the variables need to be combined in some way. The standard way to do this is to treat each variable separately and combine them with one big sum of Euclidean squares. This mean that each variable has its own dimension in the space that the clustering algorithm sees the data. This is called the 2-norm way to combine the variables and is written in general between two data points j and p in equation 3.17.

$$\text{2-norm } Distance_{j,p} = \sqrt{\sum_{i=1}^{n}(X_{i,j} - X_{i,p})^2} \tag{3.17}$$

where
$X_{i,j}$ is the value of the jth data point on the ith variable of n variables.

With profile data however, [30] uses a 1-norm method, by calculating the distance between profiles as the sum of the distances in each variable. Since the wind data also exist in profiles over 4 heights, the 1-norm distance could be calculated as:

$$\text{1-norm } Distance_{j,p} = \sum_{i=1}^{4} \sqrt{(S_{i,j} - S_{i,p})^2 + (DD_{i,j} - DD_{i,p})^2} \tag{3.18}$$

where
$S_{i,j}$ is the speed value of the jth data point on the ith height of 4 heights, and $DD_{i,j} - DD_{i,p}$ is calculated in an appropriate way for the difference in directions.

The two norm options are considered in the following. The eleven dimensions are simplified to just two, the wind speed at two different heights. Figure 3.6, shows two black profile lines and a reference profile line in red.

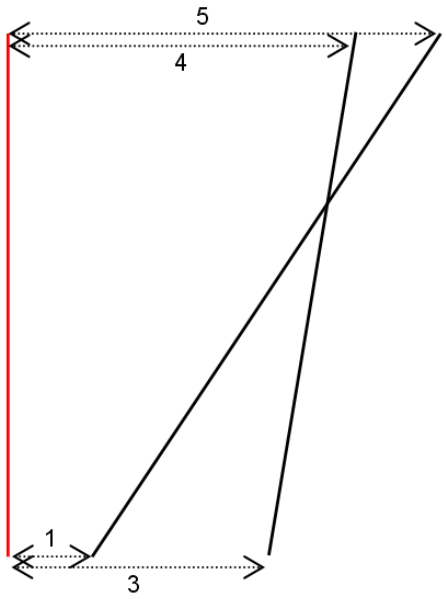

Figure 3.6: Simplified example of distance between profiles

Table 3.5 shows the distances between the black and red lines calculated with the 1-norm and 2-norm methods. The results show that the ranking switches as to which line is closer to the red one.

Distance from red line	Line 3-4	Line 1-5	Which is closer?
2-norm	5	5.1	3-4
1-norm	7	6	1-5

Table 3.5: Norm comparison

For clustering, the 2-norm is seen as the best way to measure this distance. This means that 3-4 is the line closest to the vertical red line. This is due to

the physics of the atmospheric flow. Considering the geostrophic wind, which is used to drive the mesoscale Model (KAMM) via the forcing terms, the 1-5 line would generate more shear than the 3-4 line. Hence the 1-5 line is further away from the near neutral shear generated by the vertical red line.

3.3.3 Treating the inverse Froude number

It is possible that the value of the inverse Froude number describes to what degree the effects discussed in 3.1 occur. However there is also a theory that the mesoscale model reacts in only three ways based on the inverse Froude number. These three situations are unstable, neutral and stable atmospheric stability, which are represented by negative, zero and positive inverse Froude number values respectively. Section 3.1 describes how the mesoscale model is thought to behave in these situations. It is possible that this 3-mode behaviour occurs more prominantly at certain sites compared to others.

If this 3-mode theory is true, the clustering algorithm should optimally consider the inverse Froude variables as only three different values. This could be done with a simple non-linear transformation of the variable to -1 if the number is negative and +1 if the number is positive. This is quite a harsh transformation however, and a softer transformation could be an inverse tangent or cube root function. The inverse tangent and cube root are compared in 3.7. The inverse tangent is flatter and hence gives a closer result to the desired effect.

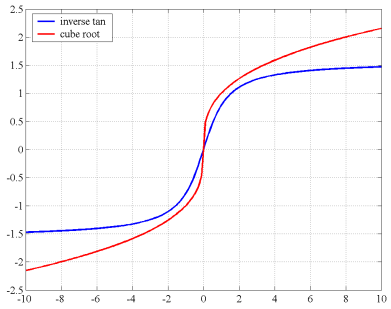

Figure 3.7: Comparison of behaviour betwen inverse tangent and cube root functions

3.4 Evaluation

3.4.1 Evaluating a representation

A good representation of a large data set by a small number of situations can be evaluated by the total error sum of squares. The total error sum of squares is the sum of all distances from each data point to the situation by which each is

represented. The error sum of squares function is minimised for a set of classes, if each class is represnted by the mean values on each variable (centroid) [3]. The smaller the total error sum of squares, the better the representation. It follows that the error sum of squares is zero if each data point is represented by itself (all distances in the sum are zero). Equation 3.19 describes the formula in general.

$$E = \sum_{k=1}^{h} \sum_{j=1}^{m_k} \sum_{i=1}^{n} (x_{i,j,k} - \bar{x}_{i,k})^2 \tag{3.19}$$

where
$x_{i,j,k}$ is the value of the ith variable of n variables for the jth of m_k data points in the kth of h clusters, and
$\bar{x}_{i,k}$ is the mean of the values on the ith variable over the m_k data points in the kth cluster.

The total error sum of squares value is useful to evaluate the overall representation. However, the equation relies on the distance between two points being defined, and hence the relative importance (and the weights - see section 3.3.2) of each variable must be known. Also, the total error sum of squares value does not contain information about how well the individual variables are represented by the classes. Thus, the total error sum of squares over just one variable is a good way to compare the representation of each variable. This number is made more meaningful, by dividing this by the total number of observations and taking the square root, thus giving the weighted average standard deviation of the values in each class for that variable. The equation for the average weighted standard deviation is shown in equation 3.20.

$$AS_i = \sqrt{\sum_{k=1}^{h} \sum_{j=1}^{m_k} (x_{i,j,k} - \bar{x}_{i,k})^2 / m} \tag{3.20}$$

Since the goal of a numerical wind atlas is usually to assess the wind energy resource over a large area, the classes should represent the wind energy content of the raw NCEP/NCAR geostrophic wind data as well. The wind energy is proportional to the cube of the wind speed. Hence the wind energy representation of a set of classes is evaluated by comparing the weighted mean of the cube of the mean wind speed for each class with the mean of the cube of the wind speed in the data. The equation for the wind energy content of the classes is shown in equation 3.21. The resulting value is always less than the mean of the cube of the wind speed in the data. The class representation of the wind energy is shown as the “percentage lost” due to this number being smaller, in this report.

$$\text{Weighted Mean} = \sum_{k=1}^{h} \left[\frac{\sum_{j=1}^{m_k} x_k^3}{m_k} \times \frac{m_k}{m} \right] \tag{3.21}$$

Incidently, if the wind speed content of the classes is compared in the same way, by comparing the weighted mean of the mean wind speed for each class with the mean wind speed in the data, the same value would be the result, regardless of the classes. This is proved in appendix B.

3.4.2 Evaluating a numerical wind atlas

As explained in section 2.4, WAsP is used to remove the local topography effects from the measurements and from the KAMM results. This produces comparable wind atlases. A wind atlas summarises the wind climate for a region. The wind atlases produced for this report contain the A and k parameters of the Weibull distribution for a set of wind speeds, along with the frequency of winds in each sector. These values are defined in 12 sectors, 4 roughness classes and at 5 heights. The second roughness class of 3 cm and the 3rd height of 50 m are chosen to compare the wind atlases as this is the common scenario for a wind turbine. The A parameter of a Weibull distribution describes the magnitude of the wind speed and the k parameter describes the distribution shape of wind speeds. The higher k is, the less spread are the wind speeds.

The A and k parameters can be converted to mean wind speed and wind energy with formulae given in [33]. These formulae is written in appendix A.3. Thus, the mean wind speed and wind energy is obtained for each sector. These can be compiled with the frequency of each sector to obtain the overall mean wind speeds and mean wind energy.

The mean wind energy is the most important quantity to compare for wind energy purposes. Predicting the annual wind energy at a given site is directly related to how much money a wind turbine can save by producing this energy. Accurate wind energy predictions are of vital importance to the planning of wind farms. As indicated in the formula in appendix A.3, the wind energy is proportional to the cube of the wind speed. The mean wind speed is naturally then also important to compare.

The sector frequencies and the mean wind speed in each sector and both also important for planning wind farms. A wind atlas has all local topography effects removed, but the wind received at a wind turbine is affected by the local surroundings. Accurate wind direction frequencies and mean wind speeds are required for siting a turbine as the wind speed at the site is affected by the roughness and orography it flows over before arriving at the site. The roughness and orography depends on the direction and the optimum turbine site is based on knowledge of the wind directions.

Chapter 4

Clustering Techniques

4.1 Introduction

Clustering encompasses many diverse techniques for classing objects or data units into groups. The grouping is usually based on the objects as members of a group (*cluster*) resembling each other as much as possible, and the objects as members of different groups being as dissimilar as possible.

Each clustering technique requires a way to measure how similar two objects of the data set are, and this is usually in the form of a distance. There needs to be at least one "variable" on which the distance can be calculated. The variable could be binary, ordinal or interval. For the weather measurements, an example of a binary variable could be stability, i.e. each data unit is evaluated on whethere the temperature profile is stable or unstable. Stability could also be evaluated on a interval variable, by evaluating each data unit by the inverse Froude number [1]. If the stability was evaluated with an ordinal variable, ranges of the Froude number could be used to rank the data units in a finite number of categories. An ordinal variable describes an order for the data, but gives no actual values. The distance calculation becomes complicated when the data is evaluated over variables of different types. Clustering with multivariables is often performed by selecting an order for the dimensions. In this way, a "principal axis" (on one of the variables) is chosen on which the data is most spread. After the data is divided on the principal axis, the next most principal axis or variable is analysed. Another important term is "outlier" which refers to a data unit that is isolated from the bulk of the data units, as defined by the distance.

Indeed, clustering could be used to categorise the different types of clustering methods. The various techniques can be divided into two types of methods, *hierarchical* and *non-hierarchical*. Some example clustering algorithms of each

[1] The inverse Froude number at a location and point in time is based on the temperature at two different heights. It describes in some sense, the stability of the atmosphere. See section 3.4 on page 19

type are described below in sections 4.2 and 4.4. The list of methods is extensive but by no means exhaustive. Each clustering method has advantages and disadvantages depending on the application. Thus it is important to use the method that best suits the application. The clustering methods used previously in similar applications is described in section 4.6. Chapter 6 compares the results obtained from different clustering methods on actual wind data to assist the selection of the best method for representing a wind climate.

4.2 Hierarchical methods

Hierarchical algorithms describe a method that builds the clustering arrangement of the data, step by step, with no previous information about the nature of the clusters. The algorithms provide clustering possibilities on different levels, from every object being in its own cluster to one big cluster containing all the objects. The decision on which clusters to merge is made based on the situation at each step of the algorithm, without any information about the situations and previous or future steps. Due to this nature of hierarchical algorithms, they are not guarateed to give the absolute optimum solution for the objective function. There are two types of hierarchical methods, *agglomerative* and *divisive*.

Agglomerative methods involve building a set of clusters by starting with each entity separately and merging them one by one. They continue until all objects are classed in one big cluster, unless some stop condition is included. Some stopping conditions are described in section 4.5. Some examples of agglomerative methods are described in sections 4.2.1 - 4.2.11). Sections 4.2.1 - 4.2.5 merge the closest two clusters in any step, and differ only by the definition of the distance between clusters. Methods 4.2.7 - 4.2.11 are all variations of the Ward Method [34]. They merge clusters based on the criteria of maximising some objective function, instead of only the basic distance between clusters. Thus, these methods differ by the definition of the objective function. Ward also suggested a specific objective function, and this is the first fo these methods.

In divisive methods, the clusters are built in the opposite way. Initially, all the data points are in one big cluster and then they are partitioned into smaller clusters based on the best separation between the clusters. This is different to the agglomerative notion of grouping together the data points that are the most alike. Some divisive methods are described in sections (4.2.12 - 4.2.15).

4.2.1 Single Linkage

In single-linkage the distance between two clusters are measured as the distance between the closest members from each cluster. The single-linkage method has the trait that the phenomena of “chaining” is likely to occur, as shown in figure 4.1. This trait can be a disadvantageous as the two objects at the ends of the chain may be markedly dissimilar, yet they are members of the same cluster. It depends on if this sort of clustering is desired or not.

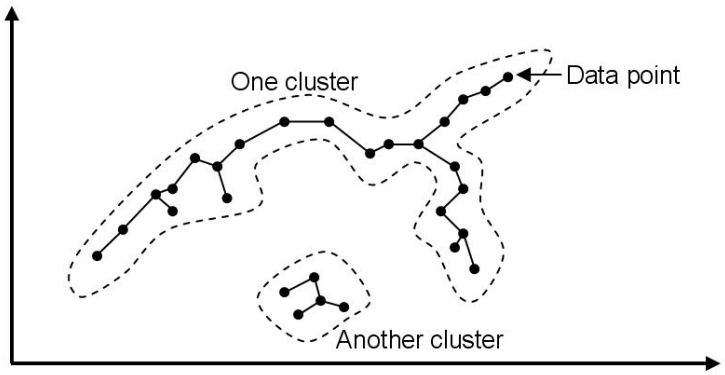

Figure 4.1: An example of chaining, affecting how clusters would be formed with single linkage

4.2.2 Complete Linkage

This method is the same as single-linkage except the distance between clusters is defined as the distance between the most distant members of the two clusters. This results in clusters where interpretation is only really possible within the clusters and not between them.

4.2.3 Average Linkage within the New Group

Instead of using the two extreme distances between clusters as with the two methods above, the distance can be calculated as the average distance between all possible combinations of two objects from two different clusters. This includes the distances between pairs of objects in the same cluster. The dependence on extreme values is removed with this method, but conversely, it is not possible to use the extreme maximum or minimum similarity within a cluster. In selection of which clusters to merge, this method favours the tightly grouped clusters ahead of the more spread clusters as the tight groupings would help to reduce the average distance.

4.2.4 Average Linkage between Merged Groups

This method is almost the same as above, except that only the distances between pairs of objects from the two different clusters are included for the average. This method would treat all existing clusters equally for selecting which clusters to merge, since the within-cluster distances are ignored in the average. However, the results produced are not radically different from the New Group method [3].

4.2.5 Centroid Method

The centroid of a cluster is the mean point or vector of all members of that cluster. The distance between two clusters in the basic centroid method is the distance between the centroids of the two clusters. A charactersitic of using centroids is that the closest similarity between the clusters for merging may rise and fall from step to step in the algorithm. This is because the centroid is recalculated for each new cluster, possibly bringing the new cluster closer to another than any two clusters were in the previous step. This phenomena does not occur in the aforementioned methods and could be seen as a problem.

4.2.6 Density Linkage

The term, *density linkage* encompasses a few clustering methods that use non-parametric probability density estimates. As described in [14], density linkage can be described in two steps:

1 A dissimilarity measure, $d*$, is computed. $d*$ is based on density estimates and adjacencies, which depend on the method of density estimation. If two data points, x_i and x_j are adjacent, $d*$ is the reciprocal of an estimate of the density midway between x_i and x_j.

2 A single linkage cluster analysis is done using $d*$.

Two types of density linkage are described in the following.

***k*th nearest neighbour** Let $r_k(x)$ be the distance from a point x to the kth nearest observation, and C be a sphere around point x with radius $r_k(x)$. Then, the estimated density at x, $f(x)$, is the proportion of observations within the sphere, C divided by the volume of the sphere. The disimilarity measure is then computed as:

$$d^* = \frac{1}{2}\left[\frac{1}{f(x_i)} + \frac{1}{f(x_j)}\right] \text{ if } d(x_i, x_j) \leq \max(r_k(x_i), r_k(x_j)) \tag{4.1}$$

$$= \infty \text{otherwise} \tag{4.2}$$

where
$d(x_i, x_j)$ is the distance between x_i and x_j.

Thus, $d*$ is infinity if neither of the points, x_i nor x_j, lie within the sphere, C, of the other point.

Uniform-kernel method Instead of specifying the parameter k for defining the radius, the radius is specified as a fixed value, r. The dissimilarity measure is then calculated in the same way as equation 4.2 except the maximum expression is replaced by r.

Both methods have a smoothing parameter (k and r). [14] emphasizes that when using density linkage, the analysis should be repeated for many different values of the smoothing parameter since the result is very sensitive to it. Density linkage methods are known to be capable of recovering clusters of irregular or elongated shapes. This is not a desired feature for mesoscale modelling classes however.

4.2.7 The Ward Method

As mentioned in the introduction to this chapter, Ward []Ward described a general type of hierachical clustering methods, where clusters are merged based on maxmising an objective function. Ward suggested his own objective function for his general method, that is, minimising the Euclidean error sum of squares. The squared Euclidean distance is calculated between each object and it's corresponding cluster's centroid. The objective function is the sum of these values for the whole data set, which can be considered in some sense as the innaccuracy of the cluster centroid appoximation to the data. Thus, at the beginning, when each cluster contains only one element, this objective function has the value, zero. The pair of clusters to be merged at each step is chosen as the merge which increases this error sum of squares objective function by the least amount. This method differs from the centroid method above, in that it weights the distances between the centroids. The error function is nondecreasing and the method is not subject to the rise and fall problem as with the simple centroid method above. As mentioned above, this algorithm is not guaranteed to find the optimum solution, i.e. the least possible value for the objective function. However, it is generally one of the best hierarchical algorithms.

For the objective of representing many points with as few a possible, minimising the Euclidean error sum of squares from each cluster is thought to be the desired goal. The general equation is described in equation 3.19 on page 31. Thus, the Ward method is a good candidate for wind climate representation.

4.2.8 Minimum Total within Group Sum of Squares in the New Cluster

This is a variation of Ward's method, where at each step, the total error sum of squares in only the newly formed cluster is minimised. This is instead of minimising the increase in the error sum of squares from a merge. Thus at each step, only the error sum of squares for the new cluster to be formed is considered and all other clusters are ignored. The clusters formed with this method tend to have approximately equal error sum of squares values. The size of clusters cannot grow much larger than others and isolated data is likely to be merged into clusters earlier in the algorithm.

4.2.9 Minimum Average within Group Sum of Squares in the New Cluster

By minimising the average contribution of each object to the error sum of squares, the variance within only the newly formed cluster is minimised. Thus, this method tends to produce clusters with approximately equal variance. Since the "true" clusters in a data set are not usually homogeneous in variance, this method is likely to fail in finding them properly.

4.2.10 Parks' Clustering Algorithm

[27] describes a heirarchical algorithm, which is implemented in a Fortran computer program. The program is old and was designed to handle a large data set (up to 200 variables and 1000 data units) such that computers from that time (1969) could cope with the clustering task in a reasonable time. However, the algorithm is still valid today and has a couple of interesting features not previously mentioned. Firstly, all the variables are normalised to a range between 0 and 1. Within these transformed variables, two options are available for creating a similarity matrix:

1 use the product moment correlation coefficient, or

2 use the complement of the mean square difference between all pairs of variables, summed over all objects.

Various criteria is applied to the constructed similarity matrix, to choose the principal components, or axes. The distances between the objects are then computed over the principal axes, and a distance matrix is constructed. After this, merging finally takes place using the ordinary centroid method, except that at each merge the centroid is updated with a weighting according to the number of members in each of the two original clusters. The original program saved time by using only a certain number of the smallest distances in the first instance. The distance matrix was recalculated and resorted only a few times, the third time being after 80% of the merges were completed. This may be avoidable today with modern fast computers. Further, Parks' algorithm has been criticised for:

- not allowing the scope for the normalised variables to be weighted,
- the fact that complemented mean square differences between variables are uninterpretable, and
- the fact that the objective function can rise and fall during the progres of the algorithm since it uses the common centroid method.

4.2.11 The EML method

The EML method was developed by W.S. Sarle of SAS Institute Inc. for disjoint clustering. The distance between two clusters, p and q, is defined as:

$$d_{pq} = mn \ln(1 + \frac{E_j - E_p - E_q}{E_t otal}) - 2P\left[m_j \ln(m_j) - m_p \ln(m_p) - m_q \ln(m_q)\right] \tag{4.3}$$

where
P is some penalty factor, m is the total number of observations,
n is the total number of variables,
j denotes the potential cluster formed by merging clusters p and q,
m_k is the number of observations in the kth cluster,
E_k is the error sum of squares for the kth cluster,
E_{total} is the total error sum of squares for the current set of clusters at the current stage in the hierarchical method.

The EML method joins clusters to maximise the likelihood at each level of the hierarchy. [14] claims the EML method method to be similar to Ward's minimum variance method but that the bias towards equal-sized clusters is removed. The lavel of bias the EML method has towards unequal-sized clusters can be adjusted with the penalty factor, P. [14] states the computational time of this method is proportional to the cube of the number of observations. This might be too much for the large number of geostrophic wind data.

4.2.12 Monothetic Division

Monothetic division is the simplest divisive clustering algorithm. The data is organised with many binary variables and is divided based on one of the binary variables such that the similarity between the two groups is minimised. The two groups formed are then split on the remaining binary variables until satisfactory clusters are found. Of course, the way of measuring the between group similarity and the conditions for a satisfactory result would need to be devised. [29] describes a variation of the basic monothetic division method, by combining it with Ward's method. The data is split based on minimising the within cluster error sum of squares.

4.2.13 Minimise total sum of squares

In the same way as with section 4.2.8, the minimum total method, this method minimises the total error sum of squares for each step to partition the data. [3] describes how the method was originally suggested to consider all possible partitions for each step. The problem with this is that there are some $2^{m-1} - 1$ possibilities for m data points. This number is completely impractical for the wind climate application with 25000 data points. [3] mentioned a discovery that the actual number of relevant partitions is somewhat less than this amount of $2^{m-1} - 1$, but it had not been formulated yet for use. This would be an interesting topic of further clustering research.

4.2.14 Colour Quantisation

Colour Quantisation (CQ) is a method that arised from a completely different field, image analysis. It was devised to represent an image which may contain millions of colours with a 256-colour palette or less. The colour for each pixel in the original image is described by certain levels of the three primary colours: red, green and blue (RGB). Thus, the quantisation algorithm finds 256 clusters, made in RGB-space and the mean of each cluster are the colours for the new simplified palette. The image application is virtually redundant today, since most computers are capable of displaying 16 million colours in an image. However, the algorithm may be applicable to the mesoscale model clustering task. An algorithm for the colour quantisation method is described in [35] and it has the same properties as a hierarchical divisive clustering method. The method is similar to the minimise total method in section 4.2.13 above, but it does not have the problem of exorbidant computational needs. The precise algorithm is not mentioned in the clustering texts reviewed ([3], [30], [23], [19], [21] and [14]).

The method has the same objective as the Ward and minimise total methods (sections 4.2.7 and 4.2.13), that is minimising the error sum of squares of the clusters. More explicitly, the error sum of squares is the sum of the squares of the distance between all the data points and their respective cluster means.

$$E = \sum_{k=1}^{h} \sum_{j=1}^{m_k} \sum_{i=1}^{n} (x_{i,j,k} - \bar{x}_{i,k})^2 \tag{4.4}$$

where
$x_{i,j,k}$ is the value of the ith variable of n variables for the jth of m_k data points in the kth of h clusters, and
$\bar{x}_{i,k}$ is the mean of the values on the ith variable over the m_k data points in the kth cluster.

Since this error sum of squares is the same as the variance multiplied by the number of data points, the standard variance formula (see equation A.1 on page 133) can be used so equation 4.4 is equivalent to:

$$E = \sum_{k=1}^{h} \sum_{i=1}^{n} \left[\sum_{j=1}^{m_k} x_{i,j,k}^2 - (\sum_{j=1}^{m_k} x_{i,j,k})^2 / m_k \right] \tag{4.5}$$

The initial condition in the algorithm is to have all data points assigned to one big cluster. The steps for the original CQ algorithm in RGB-space (red-green-blue) are then as follows.

1 Sweep a cutting plane perpendicularly along each of the R,G and B axes. Since the RGB-space is a discrete space the increment distance along each exis is simply to the next value. For each position of the cutting plane,

calculate the error sum of squares from the two new clusters formed on each side of the cutting plane.

2 Make a division of the data using the cutting place at the position where the total error sum of squares was a minimum.

3 If the number of clusters formed is the desired amount (e.g. 256) go to step 5. Otherwise go to step 4.

4 From all the current clusters formed in the data, find the cluster with the highest error sum squares. Go to step 1, and repeat the procedure, using only the data points in this cluster.

5 Represent each data point as the centroid of the cluster to which it belongs.

[35] continues to describe an efficient way to implement this algorithm in RGB-space using update equations, since it would be time consuming to calculate the whole error sum of squares on each side of the cutting plane at every new position. However, these equations rely on the property that the space is discrete and that there are only 3 dimensions. Neither of these traits hold for classifying a wind climate at different heights. To implement the colour quantisation method for a wind climate, the following adjustments need to be made to the algorithm.

1 The cutting plane is swept along all axes, which could be up to 15. It is also possible that the principal axis, the axis along which is the largest variance in the data, could be used to sweep along instead. The principal axis method is explained in the following section, 6.1. Figure 4.2 shows the cutting plane sweeping from data point to the next on one axis, with only one other axis shown.

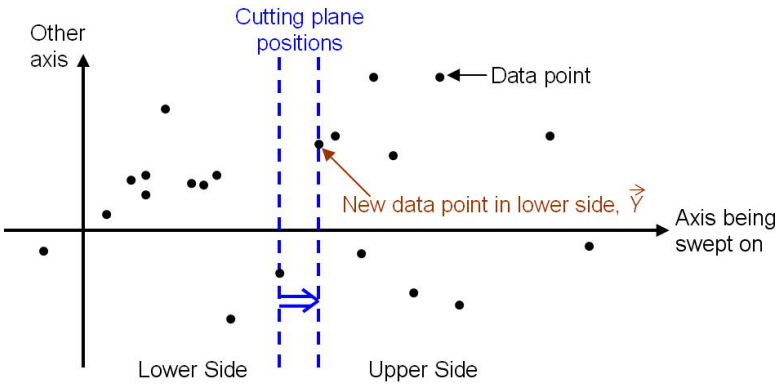

Figure 4.2: The cutting plane sweeping from one data point to the next along one axis to the next with one other axis shown

2 The space is continuous instead of discrete. The individual data points are sorted along the sweeping axes and the cutting plane steps from data point to data point, instead of value to value.

3 Since theer is one new (or one less) data point on each side of the cutting plane with each step, a set of update equations are formulated to calculate the error sum of squares on each side. The error sum of squares is initially calculated for the whole data set. It is formulated using the sum of the values and the sum of squares of the values on each variable, as in equation 4.5 as follows.

$$E = E_L + E_U \tag{4.6}$$

$$= sX_L^2 - \frac{s\vec{X}_L \cdot s\vec{X}_L}{m_L} + sX_U^2 - \frac{s\vec{X}_U \cdot s\vec{X}_L}{m_U} \tag{4.7}$$

$$= 0 + \sum_{i=1}^{n} \left[\sum_{j=1}^{m_U} x_{i,j,U}^2 - (\sum_{j=1}^{m_U} x_{i,j,U})^2 / m_U \right] \tag{4.8}$$

where

$sX_U^2 = \sum_{i=1}^{n} \sum_{j=1}^{m_U} x_{i,j,U}^2$,

$s\vec{X}_U = \sum_{j=1}^{m_U} x_{i,j,U}$,

$x_{i,j,U}$ is the value of the ith variable of n variables for the jth of m_U data points,

L represents the "lower" side of the cutting plane (which does not contain any data points at the start),

U represents the "upper" side of the cutting plane (which contains all data points at the start), and

m_U is the number of data points in the upper side of the cutting plane.

The values sX_U^2, sX_L^2, $s\vec{X}_U$, $s\vec{X}_L$, m_U and m_L are stored and updated as the cutting plane as the cutting plane steps to the next data point. The update equations are as follows.

$$sX_U^2 = sX_U^2 - \vec{Y} \cdot \vec{Y} \tag{4.9}$$

$$sX_L^2 = sX_L^2 + \vec{Y} \cdot \vec{Y} \tag{4.10}$$

$$s\vec{X}_U = s\vec{X}_U - \vec{Y} \tag{4.11}$$

$$s\vec{X}_L = s\vec{X}_L + \vec{Y} \tag{4.12}$$

$$m_U = m_U - 1 \tag{4.13}$$

$$m_L = m_L + 1 \tag{4.14}$$

where
$\vec{Y}$ is the vector the values on each variable for the new data point.

The new error sum of squares is then simply recalculated with a few sums and two divisions.

$$E = E_L + E_U \tag{4.15}$$

$$= sX_L^2 - \frac{s\vec{X}_L \cdot s\vec{X}_L}{m_L} + sX_U^2 - \frac{s\vec{X}_U \cdot s\vec{X}_L}{m_U} \tag{4.16}$$

$$\tag{4.17}$$

The position of the cutting plane when the minimum E is found during this process is where the data points are divided into two new clusters. The values E_L and E_U are stored so the cluster with the maximum E can be easily found for each new sweep. The algorithm stops when the desired number of clusters is reached. Other possible stopping criteria is suggested in section 4.5.

Since the CQ algorithm tries to minimise the same objective function as the Ward minimum variance method (section 4.2.7), it is also a suitable candidate for representing a wind climate. The CQ algorithm also would give the most similar results to the old method (section 3.2), particularly if the speed and direction axes are scanned for splitting. The resulting clusters would have defined sector and speed bin limits, as with the old method, but would focus better on the denser areas of data points, thus giving a lower error sum of squares. Neither the CQ or Ward minimum variance methods find the optimum clustering to minimise E.

4.2.15 Discriminant Analysis

This is a third divisive approach to clustering. An initial partition is made and a linear discriminant function is computed. The data units are reassigned iteratively and the linear discriminant function recalculated, until the groups are separated with maximum dissimilarity.

4.3 Preparation for non-hierarchical methods

The non-hierarchical clustering methods require an initial partitioning of the data, and involves altering the memberships in this partitioning iteratively minimising some objective function. The number of clusters is typically specified *a priori* but can be also be decided as part of the algorithm. Non-hierarchical algorithms are in general faster and use less computer space than hierarchical algorithms. This is because it is not necessary to calculate a large similarity matrix. An initial partition is either formed randomly or from a set of seed points

around which the clusters are built. The initial partition is then optimised by altering the data memberships in the clusters, often until a local optimum of the objective function is found. The various non-hierarchical algorithms below differ by how they consititute the objective function and its optimum. Before using an algorithm, an initial set of seed points, or an initial partition of the data is required. The following two sections, 4.3.2 and 4.3.3, give some examples on how this might be done. This is followed by some non-hierarchical clustering algorithms.

4.3.1 Initial data division

Before applying a nonhierarchical algorithm, an initial method to divide the data is required. The two main ways of doing this are with seed points or some initial partition. It is not uncommon for a nonhierarchical method to be used in combination with a initial hierarchical method to find the seeds.

4.3.2 Seed Points

A set of k seed points can be used as nuclei, around which partitions of the data are formed. Some representative methods for generating seed points are explained in the following:

1 **Basic.** Use the first k data units in the data as the k seed points. Alternatively, if there is concern for the way the data is ordered, choose the k data points evenly spread throughout the data, choose k random data points or subjectively choose k data points. These methods are the simplest and easiest and can be used if the choice of initial seed points does not affect the algorithm's outcome. An advantage is that every seed point is a data point so every resulting cluster would contain at least one data point.

2 **Random on variable.** From the range of an important variable, k random values (or vectors) are chosen to be the seed points. This could result in some of the seed points being quite distant from the bulk of the data points. Some seed points may even be distant from any of the data points and the resulting clusters using them would be empty.

3 **Centroids of partitions.** Any desired partitioning is made on the set and the initial seed points are the centroids of each of these partitions. Possible ways to make initial partitions are described in section 4.3.3.

4 **Spanning the data set.** Intuitively, a good set of seed points would span the data set. In other words, the seed points are evenly spread around the data units, within the space (possibly multi-dimensional) in which the distances are calculated. [5] (pp.72-74) describes a simple method to achieve this. In their method, the first seed point is chosen as the overall mean vector of the entire data set. Subsequent seed points are chosen by examining the data units in the order they are provided, and choosing

any data unit that is at least a distance d_{min} from all previously chosen seed points. The procedure is terminated when either the desired k seed points are found or the data set is exhausted. Due to the simplicity of the method, the value for d_{min} could be found with only a few trials to obtain close to the desired k seed points. A criticism for this method is that the seed points chosen may lie anywhere in the data set, and some are likely to be outliers if outliers exist.

5 **Spanning the data set weighted in density.** [4] suggests the following method to find more representative seed points than the above method.

(a) Calculate the "density" around each data point as the number of data points within a specified distance, d_{dens}.

(b) Make the first seed point the data point with the highest density.

(c) Choose subsequent data points in order of decreasing density, with the added condition that each new seed point is at least a specified distance, d_{min} from all other previously chosen seed points.

(d) Continue finding new seed points until the density of the next candidate data point has zero density, i.e. it is at least d_{dens} from all other data points. Thus, it is avoided to make outliers seed points.

(e) If this procedure produces more seed points than desired, the seed points are grouped hierarchically using the centroid method.

In theory Astrahan's method is probably ideal. In practice, evaluating the distances d_{dens} and d_{min} require experience, good judgement and probably need to be found with trial and error. In general d_{dens} should be less than d_{min}, otherwise too many data points are included in the density calculation, compared to the eventual size of the clusters. This method is considerably more complicated than the previous methods, but Astrahan used it in 1970 with 3231 data units at apparently acceptable cost [3]. On that occasion Astrahan was actually intersted in clustering about 16,000 data units, but found the seed points based on 3231 as a compromise. Today, such compromises may not be needed if super computers are available (see section 2.3).

4.3.3 Initial Partitions

Some non-hierarchical clustering methods require dividing the data into k mutually exclusive partitions instead of k seed points. Alternatively, the centroids of a partitioning can be used as the seed points. Some initial partitioning methods are described in the following, the first two using seed points.

1 **Closest seed.** Given an initial set of seed points, assign each data unit to the cluster initiated with the closest seed point [7]. This method is the simplest and the result is independant of the order in which the data units are assigned. The resulting clusters would have the shape of polyhedrons

(possibly multidimensional), since recalling from geometry that the locus equidistant from two points is a straight line, perpendicular to the line joining the two points (see figure 4.3 on page 47).

2 **Closest updated centroid.** This method is suggested by [19] adds one extra complication to the closest seed method above. Each time a new data unit is assigned to a cluster, the centroid is recalculated. Thus, since the centroids may move considerably as new data units are assigned, the result of this method depends on the order of the data units examined. This effect resembles the hierarchical centroid and Ward methods in sections 4.2.5 and 4.2.7.

3 **Use hierarchical method.** An initial partition for use in a nonhierarchical algorithm can be found quite satisfactorily by using a hierarchical algorithm. Hierarchical grouping will give relatively distinct group centroids, and the centroids will be tailored towards the desired objective function depending on the type of clustering algorithm used. Using a hierarchical method may require more computation effort than the rest of the clustering analysis. In the past subsets of the data have been used to reduced this burden, but again, the availability of a super computer could mean this is not necessary.

4 **From judgement.** It could be benficial to make an initial partition based on one's own judgement. The data could be sorted on single concept or variable/principal axis and could thereby be deliberately biased towards a particular aspect of favour.

4.4 Some non-hierarchical methods

4.4.1 Forgy's Method (1965)

Given a set of seed points, this method simply loops through each observation assigning each observation to the closest seed point. After this is completed, new centroids are calculated based on the new cluster memberships. The new centroids are used as seeds for a reassignment of the observations. This process is repeated until convergence; that is, no membership changes during an assignment compared to the last assignment. Five iteratons is generally sufficient, and it is rare that more than ten iterations are needed [3]. The SAS program [14] has a procedure called Fastclus, which implements Forgy's method with only one iteration by default.

4.4.2 Jancey's Variant (1966)

Independently, Jancey devised the same method as Forgy, but with a small variant. He suggested that the updating of the centroids is retarded for each iteration, and that the update should be in the same direction as the new

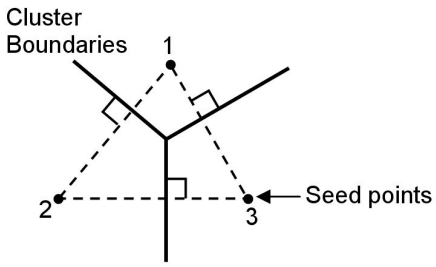

Figure 4.3: With the Forgy method, the cluster boundaries would be equidistant from the seed points

centroid compared to the old seed point, but twice as far away from the old seed point. Figure 4.4 below demonstates this.

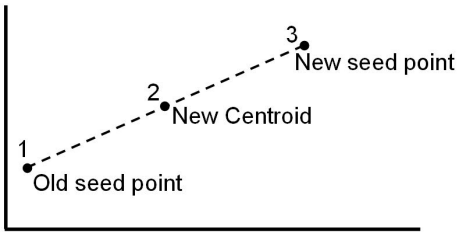

Figure 4.4: Jancey's seed update method

4.4.3 MacQueen's *k*-means

MacQueen's method has the same basis as the Forgy method. It uses the basic method of finding seed points by taking the first k points in the data set. With those seed points, only two pases are made through the data using the same method as Forgy.

4.4.4 Convergent *k*-means

[3] describes a convergent k-means clustering method which is a little different to the previous non-hierarchical methods mentioned. After the first partition is made, each data point is checked in sequence for its distance from the existing cluster centroids. If the data point is closer to a different centroid from that which it is assigned, it is reassigned to the closer centroid, and the corresponding centroids of the gaining and losing clusters are updated. This data points are continually checked in sequence until no further changes are made, i.e. convergence is achieved.

The K-means method described in [30] is similar to this method, except that each data point is temporarily reassigned to each and every other cluster to see

if the objective function decreases.

4.4.5 MacQueen's *k*-means with Coarsening and Refining Parameters

As a further complication to MacQueen's method, two extra parameters are introduced along with k, C and R. It is described in [3]. The algorithm starts with making the first k data units the initial seeds. The remaining algorithm steps are as follows.

1 All pairwise distances are computed between the seeds, which are to be treated as clusters on one member each. If the smallest distance is less than the "coarsening parameter" C, merge the two associated clusters and calculate their centroid. This process is repeated until all centroids are separated by a distance of at least C.

2 Assign all the remaining data units to the cluster with the nearest centroid. At each and every assignment, the cluster centroid is updated and the process in step 1 is checked using C. An extra check is now done using the "refining parameter" R. If the distance of the new centroid to the nearest centroid is greater than R, the new data unit is assigned to its own new cluster instead of the cluster with the nearest seed point.

3 After all data units are assigned, the existing centroids are used as fixed seed points and the whole data set is reassigned to the closest seed point.

This method also does not continue until convergence and is hence efficient. The disadvantages are that the final number of clusters made is not controllable, and that the final centroids are not necessarily separated by at least the distance, C.

4.5 Stopping conditions

The hierarchical clusters methods will continue to merge all clusters until only one cluster remains. This is a trivial result and the number of clusters desired is always more than 1. Thus, the algorithms require a condition to decide when to stop merging clusters. Stop conditions include:

1 The desired number of clusters is known *a priori*. The hierarchical algorithm can then stop when the number is reached.

2 The pseudo T-test is a test ratio which is checked against some critical value. The ratio is the error sum of squares for two clusters over the error sum of squares for 1 cluster when they are merged. Thus there is a new pseudo T-test value for each step in the hierachical algorithm where clusters are merged. The algorithm stops when this value falls below a critical value.

3 The pseudo F-test is also a test ratio calculated at each step in the hierarchy. The formula for the pseudo F-test ratio is:

$$pseudo\ F = \frac{trace\frac{B}{h-1}}{trace\frac{W}{m-h}} \tag{4.18}$$

where
m is the total number of observations,
h is the number of clusters in the current solution,
B is the between and pooled within cluster sum of squares, and
W is the cross product matrix.

The best number of clusters is when the value is at a maximum.

There are many more stopping rules. The two Milligan articles [21] and [22] both do Monte Carlo studies examining many different stopping rules. One is about finding the correct number of clusters in a data set and the other compares 15 different clustering techniques as to how they perform in allocating the data. The data sets for both of these studies were generated hypothetically with very distinct clusters. The number of clusters used is between 2 and 5 and the number of data points is only 50. Both conclude that the pseudo T and pseudo F tests are the best two to use for finding clusters. [23] also studies different clustering algorithms and stopping rules, mentioning that the Ward method performs very well. However, the application of these stopping rules is concerned with finding distinct clusters in the data. Since this project is about finding the best *representation* of a data set rather than distinct clusters, these studies and stopping rules in general probably do not apply. Also, for this report, the clustering is made to be compared with the old classification method, and will be set to produce the same number of clusters as the old method. Hence, the simplest stopping rule, number 1, above will be used.

For interest, a short study is made to check if an optimum representation can be found. Figure 4.5 shows the percentage change in the error sum of squares with increasing number of clusters on the Egypt data. The clustering criteria only looks at the wind speed and direction at the lowest height. The percentage change in the error sum of squares is similar to the pseudo T test, as described above. The hypothesis is that an improvement in the error sum of squares decreases with an increasing number of classes might indicate an optimum number of classes. The error sum of squares function always improves but when the improvement is not as much compared to the last increment in the number of clusters, the last number of clusters could be said to be more optimal. The plot with both the regular axes and principal axis methods is shown below for making between 30 and 400 classes. A local minimum in the curve, where the improvement was minimal compared to the improvement made with one less number of clusters shows a possible optimum. These could be at 47, 55, 108, 162 or 193 classes. However, the effect is very small. Furthermore,

the goal is to make better classes for producing better numerical wind atlases and this optimum probably has no effect on the results.

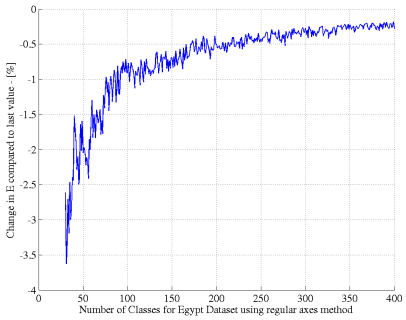

Figure 4.5: The percentage change in the error sum of squares, E, with increasing number of clusters on Egypt data

4.6 Clustering Techniques previously used in similar applications

Some examples of clustering used previously for mesoscale modelling are described in the following in section 4.6.1. Some other related applications where clustering has been used are described in section 4.6.2. These clustering methods previously used are summarised in table 4.1 on page 53.

4.6.1 Mesoscale modelling

There are a few documented applications of clustering for mesoscale modelling. However, clustering has only been tried briefly in the past and most of the documents suggest that a more sophisticated and more carefully done clustering could improve the results. The information available on the details of the clustering methods used is brief. For example, none of them mention the time resolution of the geostrophic wind data so it is impossible to know how many data points were used to make clusters. Also attempts were made to find out what clustering methods were used, but these failed it was difficult to contact the right author who made the clustering algorithm. The information that was obtainable is summarised in the first five entries of table 4.1 on page 53.

Of the five clustering analyses mentioned, three performed the clustering in u-v space. This was also tried originally for this report, but it is not the optimal way to view the data. Clustering in u-v space gives an unwanted weighting in that the higher the wind speed, the greater the weight of the wind direction. For example, the distance measured between two points 10 degrees apart is greater at 10 ms^{-1} than at 1 ms^{-1}. It is also likely that one cluster would form in

the middle containing data from all directions, which are cancelled out in the class centroid. Also the cluster means are focussed on mean vector winds, which is not optimal for wind energy which depends on the cube of the mean wind speed (as also mentioned in [9]). Thus, it is thought that clustering in wind speed-direction space is better suited for mesoscale modelling. This is also the space in which the existing method works by dividing the data into sectors and speed bins.

Also, two of the previous clustering attempts were made on only a two year data sample of geostrophic wind. In these two articles the clustering method is dismissed as unsuccessful, and along with the clustering being done in u-v space, it is easy to see why. The third of the u-v space clustering articles, [1] does not the state the period of geostrophic wind data used, but it is the only article to at least mention that an hierarchical clustering method was used. The application in [1] was a specially large area covering the Baltic Sea. A third variable was used here with u and v, ΔT. ΔT represents the temperature difference between the sea surface and the air.

The Mengelkamp clustering attempt in 1997, [20] uses a 12 year time series of geostrophic wind data to make 143 clusters. A third variable was used in the form of 3 fixed stability classes but the article does not mention whether the clustering was performed in u-v space or with speeds and directions. It does say however, that the cluster representation was evaluated based on the standard deviation of the speeds and directions in each of the individual clusters. This is similar to the method used for evaluating a representation in this report (see section 3.4.1). Of the five mentioned mesoscale modelling attempts using clustering, Mengelkamp's results seem the most accurate comparison with measurements.

The fifth article tried two classifications, a conventional one similar to the existing method described in section 3.2, and an unusual clustering method using empirical orthogonal functions (EOFs). The conventional classification involves splitting the data on 4 weather conditions first - summer/winter and then humid/dry. These four classes are each then divided in 30° sectors (12) and this makes $4 \times 12 = 48$ classes. Briefly, an EOF describes a detailed axis, along which the data is the most spread in the energy sense. The eigenvalue spectrum of the EOFs describes the order of the EOFs in terms of the variance of the data. In the EOF-clustering method, they also first split the data into 4 seasonal classes. They then find three EOFs as principal axes and make a special tailored clustering analysis. The three EOFs are summarised from [12] as follows.

- EOF1 describes a coupling of exclusively positive variations for January. In this way, relatively dry and cold conditions with a NW flow as well as warm and humid conditions with a W-SW flow are covered. The same meteorological correlation is described by the profile of eigen vectors for July.
- EOF2 describes warm temperatures close to the ground with cold SW currents in the higher layers as well as cold temperatures close to the

ground and warm NW currents in the higher layers. Variations of humidity are unimportant.

- EOF3 describes variations of the zonal component and to a certain extent also variations of the meri-diurnal component of the geostrophic wind. In this way, weak SW winds and strong NW winds are covered.

They conclude that the conventional classification method is better than clustering on EOF modes. Since EOF modes are an energy approach to the data, great care needs to be taken to filter out noise. When EOF is performed, it is possible that some important features that contribute little to the energy, but affect the mesoscale model results, will be filtered out. Even if this problem could be avoided satisfactorily, many more modes than three (possibly 100) would be required to classify the data for mesoscale modelling ([16] and [15]).

4.6.2 Other related applications

The clustering methods used in related articles to mesoscale modelling and the number of clusters found are shown in table 4.1 from the 6th row down. A wide variety of clustering methods are used. The number of clusters used is typically not very high, around 10, which is much less than the figure typically used for mesoscale modelling. However, the amount of data points used are also often much less than the 25000 data points used for this report.

Of main interest is the method used by Kaufmann and Weber for classing mesoscale wind fields. Their clustering methods used are a good recommendation for this report. They use a two-stage process. The first stage uses a complete linkage method to find an optimal number of clusters, and find seed points to be used as input for the second stage. The k-means method is used to allocate the data to each seed minimising the error sum of squares.

Author(s)	Year	Topic	Principal Method(s)	No. Data	No. Clstrs
Landberg Watson	1994	Mesoscale Modelling	In u-v space only	2 years	120
Adrian Dotzek Frank	1996	Mesoscale Modelling	Hierarchical method in u-v and ΔT (sea - air) space	Unknown	120
Mengelkamp Kapitza Pflüger	1997	Mesoscale Modelling	3 fixed stability classes Evaluated by standard deviations of speed & direction	12 years	143
Frank Landberg	1997	Mesoscale Modelling	In u-v space on unrepresentative two-year period.	2 years	60
Frey-Buness	1993	Mesoscale Modelling	Uses empirical orthogonal functions.	20000	48
Davis and Kalkstein	1990	Synoptic climatological classific.	Two-stage: K-means using the seeds from the average linkage method.	2 million	90
DeArmon	2004	Severe Weather Days	K-means and Ward's. Used "hclust" in Splus.	197	18
Eisen	1998	Genome-wide expression patterns	Pairwise average-linkage	1000 to 10000	11
Fernau	1990	Weather transport patterns	Ward. Number decided arbitrarily.	1093	7, 18
Glascoe	2004	Regional-scale wind fields	K-means. Number decided arbitrarily.	2920	5
Kaufmann Weber	1996	Mesoscale wind fields	K-means w/ Ward method w/ seeds from complete linkage.	8784	12
Kaufmann Whiteman	1999	Wind pattern in Grand Canyon	Same as above.	883	12
Mimmack	2000	Defining rainfall regions	Mahlanobis distance vs euclidean on Ward min variance.	360	5

Table 4.1: Summary of some related applications using clustering

Chapter 5

The Sites

Two sites are used for testing the new classification method, Ireland and the Gulf of Suez in Egypt. Here forth, the term, Egypt, refers only to the Gulf of Suez region. The two sites are described in detail, along with reasons for their selection in the following. For both sites, the NCEP/NCAR Reanalysis [26] geostrophic wind data used is from the same period of time, complete years from 1965 to 1998, inclusive. Hence there are 24836 data points in the 34 years, one every 12 hours. Risø plans to update their data to 6-hourly as it is available but this was not done for this report.

5.1 Ireland

Ireland's terrain consists of rolling hills, with a relatively homogeneous roughness. It is chosen to test the new classification method since it is relatively simple and has a high level of neutral thermal stability. There isn't much local wind phenomena (see section 2.1.2) occurring to which modelling errors can be attributed.

5.1.1 Geostrophic wind data

The wind rose for the geostrophic wind data used for classification is shown in the left plot of figure 5.1. The wind rose for the mean wind speed in each sector is shown in the right plot in this figure. The plots show that the majority of the wind comes from between W and SW, and this is also where the highest mean wind speeds are. The overall mean wind speed of 11.8 ms^{-1} and mean wind energy of 2175 Wm^{-2} for the whole period are also given.

The wind atlases from measurements use twenty years of data from 1970 to 1989. Figure 5.1 compares the wind roses and mean wind speeds and energy for the shorter twenty year period of the geostrophic wind data compared to the full 36 years availble. There is no significant difference for the wind roses or mean values. Hence, although the measurement wind atlases are made with

twenty years of data, they can be compared directly against the numerical wind atlases made with 36 years of geostrophic wind data.

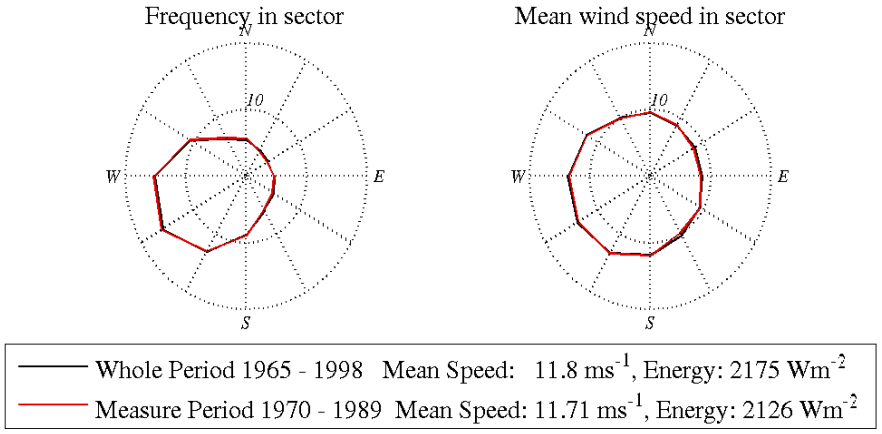

Figure 5.1: Comparing the geostrophic wind data period with the measurement wind atlas data period for Ireland.

5.1.2 KAMM

The model domain for KAMM consists of 90 × 108 grid points with a resolution of 5 km. It covers the whole island including some grid points over water in each direction. The domain also includes a part of the terrain of Scotland which might influence the flow in Northern Ireland. 25 levels are used from the surface to a height of 4000m. The lowest grid levels are at 15, 43, 84 and 138 m a.g.l. for a grid point at sea level. The maximum terrain height resolved is 522 m over the Wicklow mountains. If a 2.5 km resolution was used, this would rise to 698 m and the maximum terrain height in the original data set is 1002 m. The 5 km orographic map of Ireland is shown in figure 5.2. The highest peak in the map at the Wicklow mountains can be seen just south of Dublin. Ten sites are shown on the map and these are explained in more detail in section 5.1.3.

The roughness map used for Ireland is shown in figure 5.3. The higher the roughness length, z_0, the more "rough" the ground is. Values for roughness lengths range from 0 to 1 m. Typical roughless length values are zero for water, 0.1 for grass, 0.7 for forests and 1 for buildings. The roughness length is described in more detail in [33]. Since the roughness is averaged over each 5 km grid area, the range of roughness values in the map is only from 0 to 0.4.

The principal NCEP/NCAR data set used as the data for classification, was created based on the data in four NCEP/NCAR grid points. The centre between these four points is the location of the resulting data for the procedure. This is shown in figure 5.4.

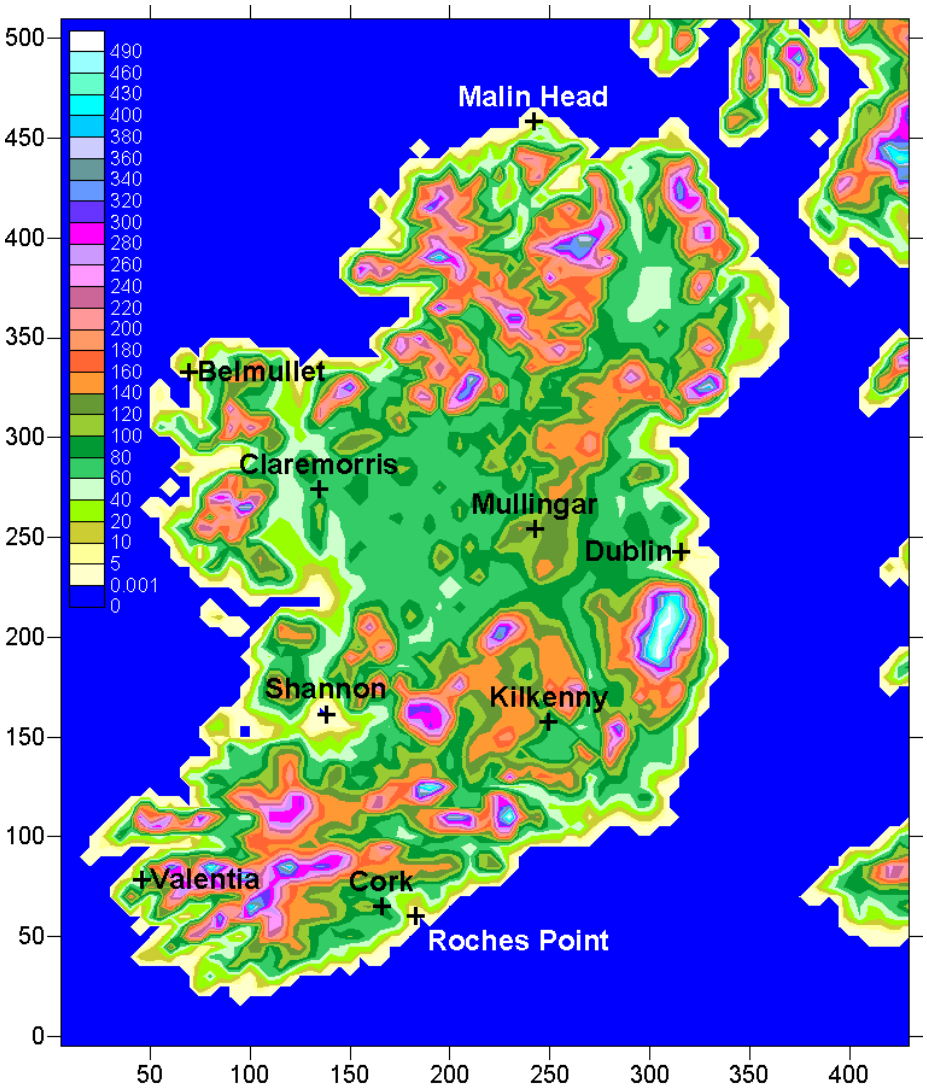

Figure 5.2: The 5 km resolution orographic map used for Ireland. The map also shows the locations of the ten met stations used for comparison. Elevations are in metres and axes are in kilometres.

There are 32 NCEP/NCAR data sets used throughout the domain based on interpolations from the grid points. These are used to create more accurate varying frequencies of the classes throughout the domain (see section 2.4 for a description on varying frequencies). Since the elevation is not very high, these NCEP/NCAR points can all be considered valid. In Ireland the geostrophic wind increases from the south-east to the north-west [11]. Four NCEP/NCAR

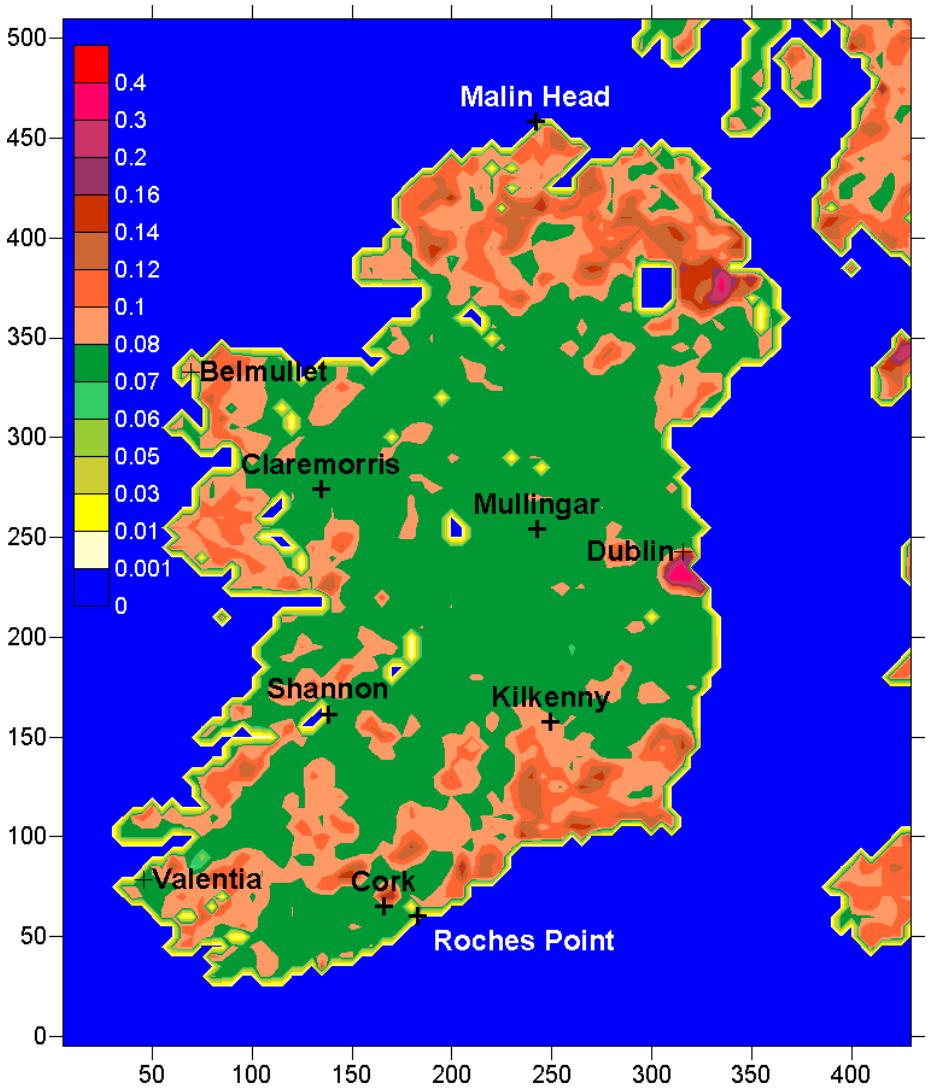

Figure 5.3: The 5 km resolution roughness map used for Ireland. The map also shows the locations of the ten met stations used for comparison. Roughness length is in metres and axes are in kilometres.

data points are compared in figures 5.5 and 5.6. The "cluster" data point refers to the data point used for the classification. The location of this point is close to Claremorris (see figure 5.2). The other three points are chosen to be near other sites in 5.2. Figure 5.5 shows two sites in the north-west area of Ireland. The mean wind speed and wind energy is comparable. The same comparison conclusions can be made for the two sites in the south of Ireland, figure 5.6.

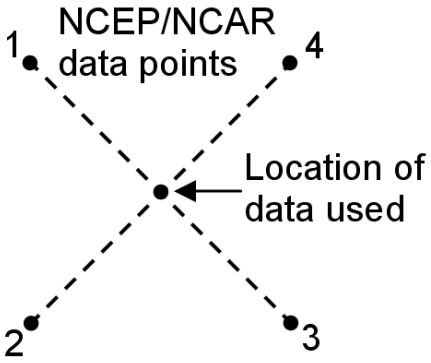

Figure 5.4: The data from four NCEP/NCAR grid points are used to make the NCEP/NCAR data used for clustering. The arbitrary site of the resulting data is shown.

However, on comparing the two regions, the wind speed is 6% less and the wind energy is 17% less in the south of Ireland. Also, a small decrease in the wind frequency from the south and south-south-west directions can be observed. Varying frequencies are used to capture this change in geostrophic wind speed across the domain.

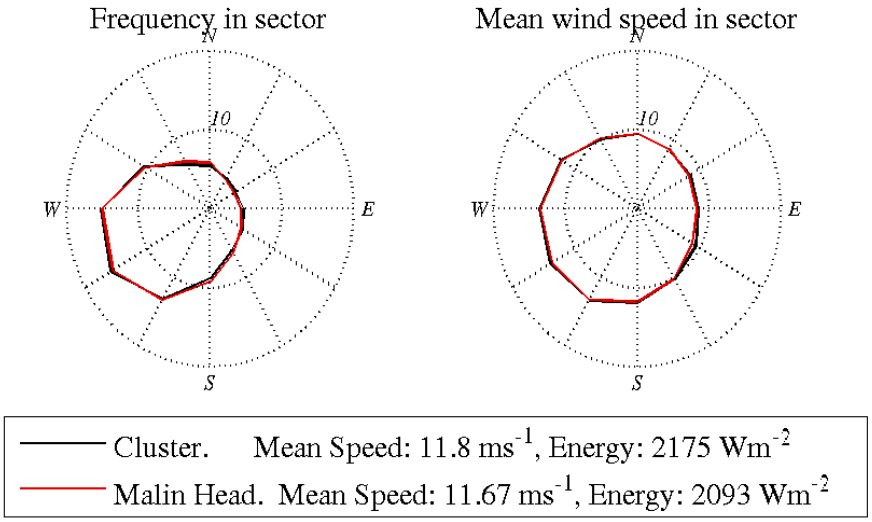

Figure 5.5: Comparing the geostrophic wind data for two sites in the north-east region of Ireland. The cluster site represents the data point used for classification and is located near Claremorris.

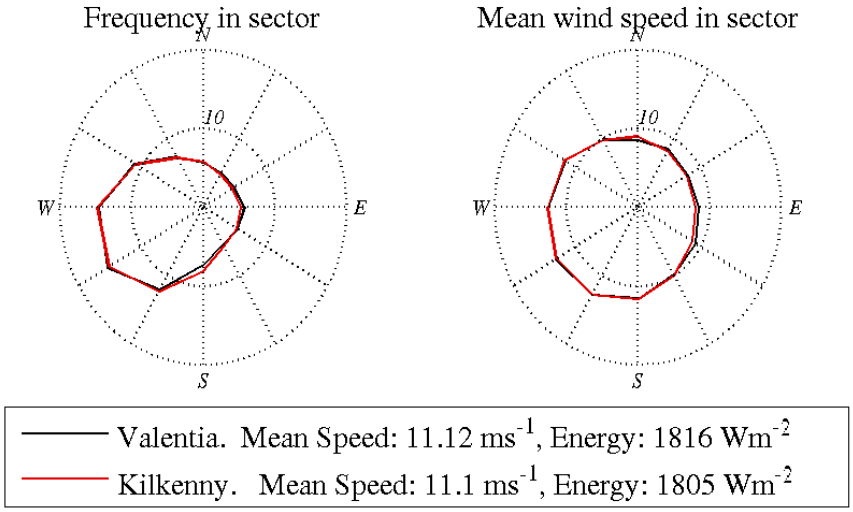

Figure 5.6: Comparing the geostrophic wind data for two sites in the south region of Ireland.

5.1.3 Measurement locations

Ten stations are chosen for comparison, which all have twenty years of data from 1970 to 1989. This geostrophic wind data in this period is comparable to the full 36 years. The wind atlases from measurements for comparison come from the New Irish Wind Atlas [18]. The site locations of the stations are shown in figure 5.2.

The stations are listed in the following, each with a short description of the surroundings as obtained directly from [33]. The elevation of each station is added. The coordinates used for the stations are in table E.1 in appendix

Belmullet Situated at the N end of Blacksod Bay, 3.5 km E of the Atlantic coastline and about 1 km WNW of the town of Belmullet. The long irregular promontory of The Mullet, on which the station stands, is almost an island, bounded on the N and W by the Atlantic Ocean, on the S by Blacksod Bay and on the E by Broad Haven. There are distant hills and mountains from 040° - 205° and at 040° - 080° there are some isolated hills 10 km away. Elsewhere the 150 m contour line lies at least 20 km away. Up to 1976 the site was very well exposed with only a few buildings in the SE sector. The anemometer is placed above a 3 m high flat-roofed building with a horizontal diagonal dimension of 12 metres. The elevation is 9 m a.s.l.

Claremorris Situated in gently rolling countryside 1 km SSE of the small town of Claremorris. There are no hills higher than 150 m within 13 km. There

are mountains, hills and higher ground in most directions beyond 13 km, but this does not seem to influence the measurements. The countryside is characterised by large areas of peat bogs, hedges, roads, railways, lakes and some small buildings. The anemometer is placed on a hut ($3 \times 3 \times 3$ m). The elevation is 69 m a.s.l.

Cork Location 17 km inland from the south cost with the centre of Cork City about 6 km to the N. The suburbs extend to within 2.5 km of the site. The airport lies on a long ridge extending almost due eastwards from a mountainous region of West Cork. The general elevation of the ridge is about 170 metres above sea level. About 30 km of W and N of the site peaks of 600-400 metres occur. The anemometer is situated on the SW-facing slope of the highest crest in the locality. It is placed on a 3 m high hut and is well exposed in the sectors bewteen 030° - 260°. To the W the airport buildings lie 400 m upwind and the sectors W to NNE are obstructed by houses and trees. The countryside beyond the airfield is strongly undulating with local steep slopes. The elevation is 162 m a.s.l.

Dublin Situated 8.5 km N of the city centre of Dublin, with the suburbs extending to within 2.5 km on the site. The open sea lies 8 to 12 km away, between 040° and 150°. The Dublin/Wicklow mountains lie between 155° and 225°. The hills start about 18 km S of the airport and extend a further 60 km to the S. The highest peak rises to 930 m. The anemometer is well exposed except in the SSW where there is an enclosure with houses and trees. It is placed on top of a hut ($3 \times 3 \times 3$ m). The elevation is 64 m a.s.l.

Kilkenny Location 2 km to the NW of the town centre of Kilkenny, the outskirts of the town being close in the SE quadrant. The region is surrounded by hills and mountains in all directions with the exception of the Nore Valley downstream. This river valley distorts the airflow so that the winds blow from preferentially from the NNW or from the S. The Centic Sea lies 58 km to the S and the Irish Sea 68 km to the E. The mountain ranges, whose foothills are no more than 40 km away, have peaks reaching heights of 500 - 900 m. Between the SSW and WSW the winds have already crossed several mountain ranges. Closer to the station the countryside is strongly rolling. The anemometer is above a 3 m high building with a horizontal diagonal dimension of 12 m. The elevation is 63 m a.s.l.

Malin Head Malin Head, the most northerly headland of Ireland is at the extremity of a narrow peninsula about 2 km wide, extending in a WNW to ENE line from the much larger Inishowen peninsula. The adjacent regions - within 100 km to the SW and S - are mostly mountainous. Close to the station the surrounding countryside is barren but not flat. The station is situated about 4 km to the E of Malin Head on a narrow ridge running SW-NE. The general WNW-ESE line of the coast is broken by small 400-m wide cove facing NE, 150 m to the E of the station. The shore

is generally rocky with steep inclines. Due to the height of the anemometer various buildings in the vicinity do not appear to seriously obstruct the airflow around the anemometer. The elevation is 24 m a.s.l.

Mullingar Situated in rolling countryside in the Central Plain of Ireland, about 1.7 km to the NW of the town centre. The Irish Sea lies more than 75 km away to the east. The nearest mountain range is 50 km away to the SSW. Its highest peak reaches 530 m a.s.l. In the other sectors the closest land above 300 metres is 70 km to the SE and NW/NNW and 85 km to the NE. The anemometer is situated above the roof of teh station building which is 3 m high with a longest horizontal diagonal of 11 m. In the sectors from W through N to ENE there are several obstacles, mainly buildings. Behind these, there is open countryside. Lough Owel lies at a distance of about 2 km between 320° and 350°, with a further overwater fetch of about 5 km. From 165° through S to 265° the exposure of the station is good with open countryside. The elevation is 101 m a.s.l.

Roches Point Located near the south coast, on the eastern bank of the mouth of Cork Harbour. The distance to water is 500-900 m in the sectors 105° - 210° and 200 - 300 m in sectors 210 ° - 330°. The land/water boundary is generally a steep incline of even a cliff in places. However, a study of win ddirection traces indicates that flow separation does not occur. In the sctor 330° - 005° there is a complex series of land/water fetches. A 70 m high bluff lies 1500 m away. The anemometer is placed on top of a 3 m high building with a horizontal diagonal dimension of 12.5 m. The elevation is 40 m a.s.l.

Shannon Situated on the N bank of the 3 km wide Shannon Estuary. The airport is bounded on the W by the 6 km wide estuary of the river Fergus. The nearest point on the Atlantic seaboard is 40 km to the W. The airport is built on very low-lying land - there is no land over 75 m within 11 km. There are mountains, hills and higher ground in almost all directions at different distances. At low tide vast expanses of mudflats are exposed beyond 4 m high embankments in the sectors SSE to WNW. From WSW to N the tidal mudflats are more than 2.5 km away. The anemometer is placed S of the runways on a hut ($3 \times 3 \times 3$ m). Nearby buildings appear in sectors N to E. The elevation is 8 m a.s.l.

Valentia Location on the SW coast at the Valentia Observatory, which is situated ENE of Valentia Island and on the S bank of the estuary of the Valentia River. Although it is within 3.5 km of the open sea the Observatory is surrounded by hills on almost all sides. The surrounding countryside is mostly rather barren and there are peaks in almost all directions up to 500 metres height and at distances beyond 2 km. Winds from directions between N and SSE through E have crossed at least one mountain ridge before reaching the local area. The anemometer is situated above a hut with dimensions ($3 \times 3 \times 3$ m). The main buildings appear to the NW at distances of 120 - 180 m. The elevation is 18 m a.s.l.

5.2 The Gulf of Suez, Egypt

In contrast to Ireland, the Gulf of Suez region in Egypt has complex terrain. Channelling of the wind is observed between the steep mountains on either side of the gulf. At nearby locations at higher heights or on the other side of the mountains, completely different wind directions can be observed to those in the gulf.

On the micro-scale, thermally induced wind circulations, as described in section 2.1.2 have been observed. Egypt is also closer to the equator than Ireland, so the coriolis effect plays a smaller role in influencing the wind flow. Egypt is chosen as the second site to test the new classification due to its complexity and that it represents different circumstances to Ireland.

5.2.1 KAMM

The model domain has 60 × 81 grid points with a grid cell size of 5 km and thus covers an area of 300 × 400 km. The model uses 28 levels in the vertical from the surface to a constant height of 6000 km a.s.l. The resolution is higher near the surface than at the top of the model. The 5 km orographic map of Egypt is shown in figure 5.7. The map has been rotated 30 degrees to align the Gulf of Suez parallel to the side boundaries of the map. The gulf to the right of the map is the Gulf of Aqaba. Four sites are shown on the map and these are explained in more detail in section 5.2.3. The highest peak in the map is around 2000 m on the opposite side of the gulf to the four sites shown. Other peaks close to the sites exceed 1000 m making the terrain around the Gulf of Suez more complex than for Ireland.

The terrain for the Gulf of Suez is also plotted in 3D in figure 5.8. The view shows the Gulf of Aqaba to the right and the Gulf of Suez straight ahead. The four sites are again shown. Note that the vertical axis scale is around 25 times the other axes' scales for visual clarity. This means the mountains are 25 times smaller in reality than shown.

The roughness map used for Egypt is shown in figure 5.9. The higher the roughness length, z_0, the more "rough" the ground is. The range of roughness values on the map are much smaller than for Ireland. They range from 0 to only 0.04 and the majority of the region is assigned a roughness length of just 0.002. This indicates that most of the ground is bare, and probably desert.

5.2.2 Geostrophic wind data

The NCEP/NCAR geostrophic wind data was calculated based on four surrounding NCEP/NCAR grid points in the same way as Ireland (see figure 5.4). For the mesoscale model, the location of the data is shifted to the left in the domain, as its natural position is invalid due to being inside a mountain. The final position of the NCEP/NCAR data point is not very important as long as the four surrounding data points used to make it are valid. Its modified location is at sea level over the Gulf of Suez. The wind rose for the geostrophic wind data

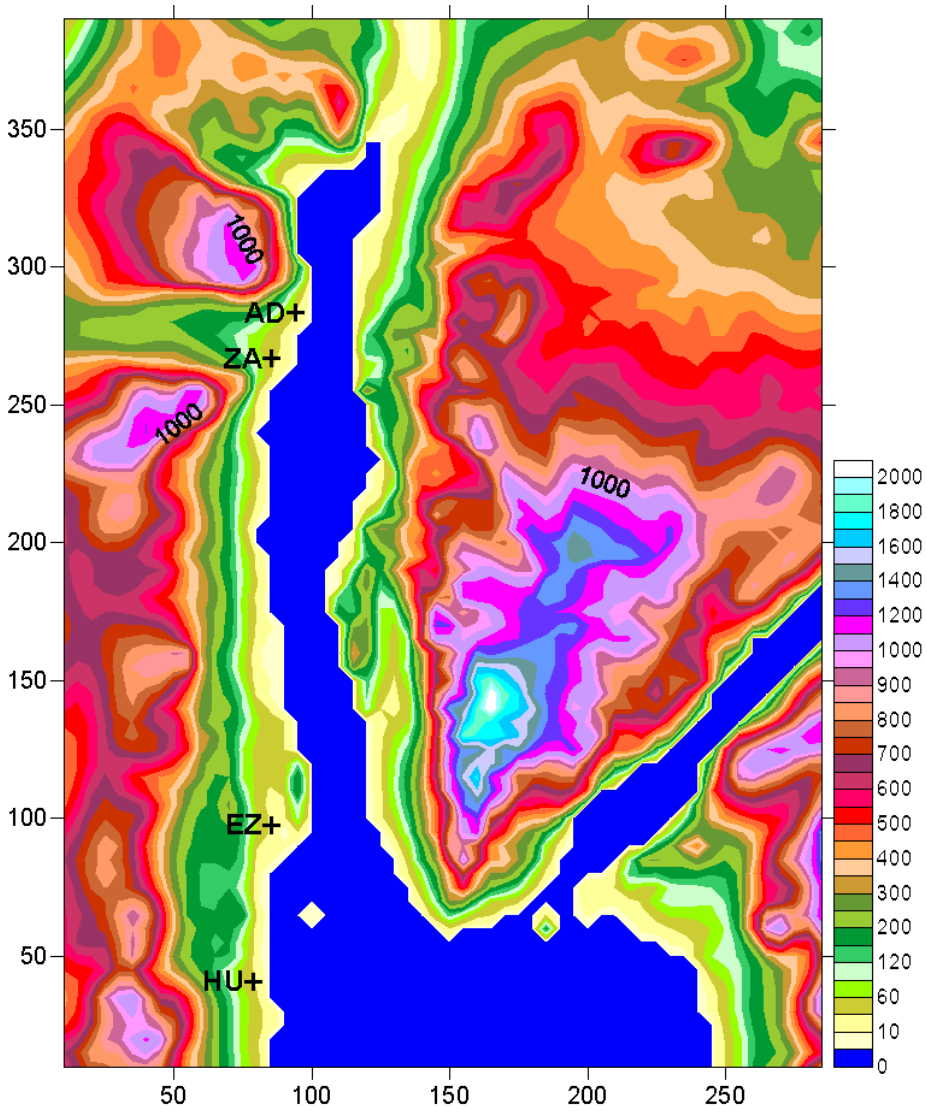

Figure 5.7: The 5 km resolution orographic map used for Egypt. The map also shows the locations of the four met stations used for comparison. AD = Abu Darag, ZA = Zafarana, EZ = Gulf of El-Zayt and HU = Hurghada. Elevations are in metres and axes are in kilometres.

is shown in the left plot of figure 5.10 and the mean wind speed in each sector is shown on the right. The figure shows that the data almost unidirectional with nearly 70% of the wind coming from bewteen N30°E and E. The mean wind speeds are also higher in these sectors. The overall mean wind speed of 8.01

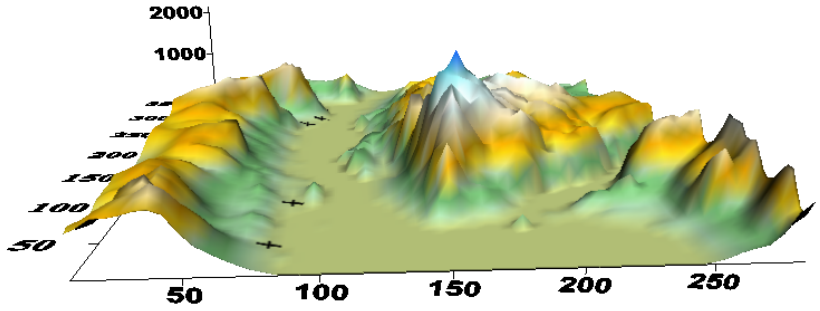

Figure 5.8: The 5 km resolution orographic map used for Egypt in 3D. The map also shows the locations of the four met stations used for comparison. Elevations are in metres and other axes are in kilometres.

ms^{-1} and mean wind energy of 515 Wm^{-2} for the whole period are also given. This is significantly lower than Ireland, which is a result of weaker pressure gradients in Egypt due to a weaker coriolis force. Note that the actual wind speed at the surface could still be higher in Egypt since this depends on the topography.

The data for the nearest eight NCEP/NCAR data points to this location are available for making the class frequencies vary in the post-processing stage of the procedure. However, the results for Egypt were made using a fixed frequency across the domain. This was done since many of the nearby grid points used to create the data are inside mountains at the surface level and are hence invalid. Thus, applying the same calculations to combine four data points as in figure 5.4 also gives an invalid result. The KAMM set up for Egypt is described in greater detail in [8].

These measurements for comparison were made over a ten year period from 1992 to 2001 inclusive. This twenty year period is shorter than the full 36 years of geostrophic wind data used for the classification, 1965 to 1998. Figure 5.10 compares the wind roses and mean wind speeds and energy for the shorter ten year period. There is no significant difference for the wind roses or mean values. Hence, although the measurement wind atlases are made with just ten years of data, they are compared directly against the numerical wind atlases made with 36 years of geostrophic wind data.

5.2.3 Measurement locations

Four stations from the Wind Atlas for the Gulf of Suez [24] with the longest period of data are selected for comparing the numerical wind atlas results for Egypt. Ten years of data for each station is between 1992 and 2001 with the exception of the Gulf of El-Zayt, which uses data from 1995 to 2001. The site locations of the stations are shown in figures 5.7 and 5.8.

The stations are listed in the following, each with a short description of

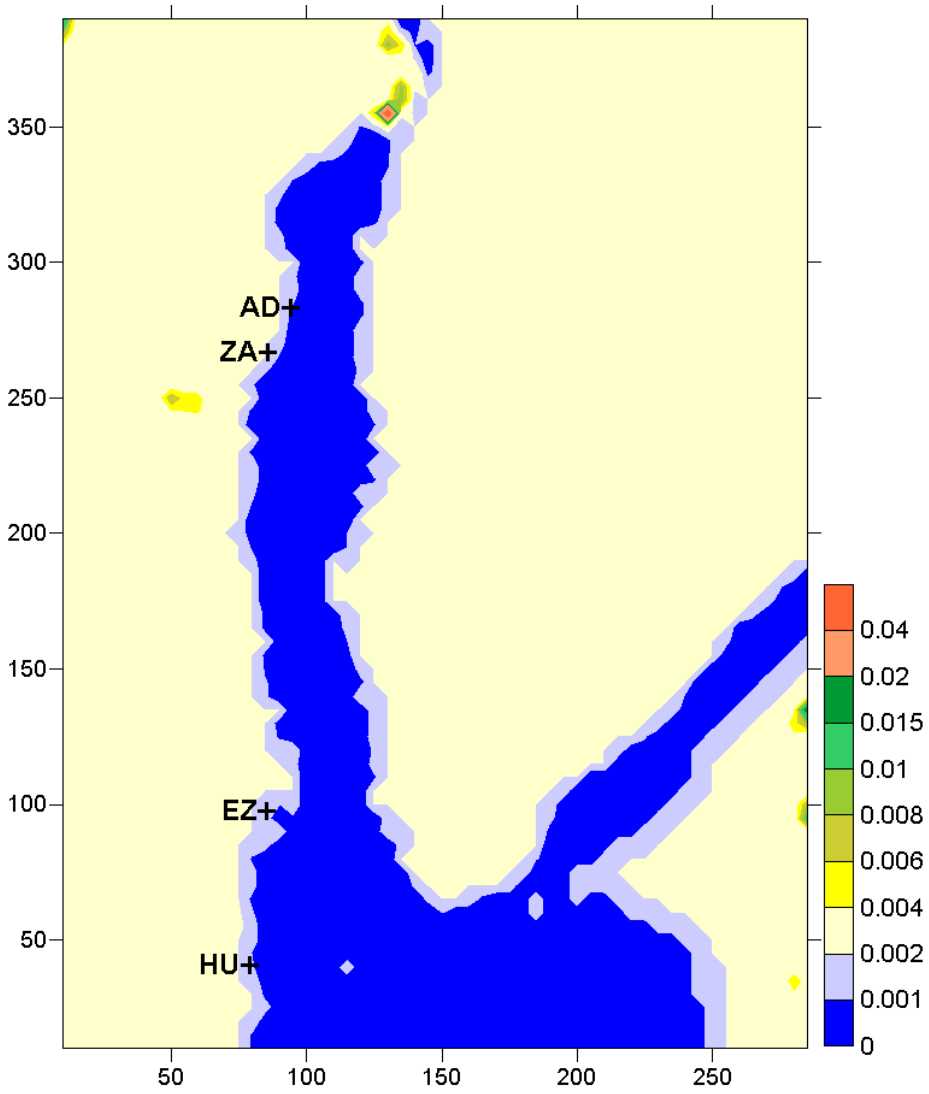

Figure 5.9: The 5 km resolution roughness map used for Egypt. The map also shows the locations of the four met stations used for comparison. Roughness length is in metres and axes are in kilometres.

the surroundings as obtained directly from [24]. The coordinates used for the stations are in table E.2 in appendix

Abu Darag The Abu Darag mast is situated about 100 m W of the Suez-Hurghada road, approximately 20 km N of Zafarana. The distance to the coastline of the Gulf of Suez is 500 m in an easterly direction. There are

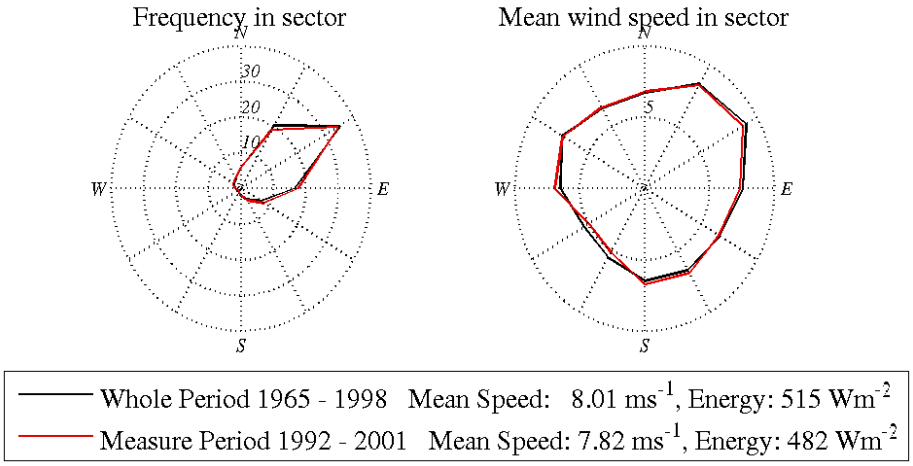

Figure 5.10: Comparing the geostrophic wind data period with the measurement wind atlas data period for Egypt.

no sheltering obstacles close to the mast. The surface consists of mostly of sand dunes with a roughness length of about 0.01 m. The terrain rises to the W and reaches 200 m at a distance of 7.5 km. The highest peak to the NW is about 700 m a.s.l. Further to the W (20 km) the North Galala Plateau rises to more than 1000 m a.s.l.

Zafarana The Zafarana mast is situated about 80 m S of the Zafarana-El Wasta road, approximately 5 km W of Zafarana along this road. The distance to the coastline of the Gulf of Suez is 5000 m in an easterly direction. There are no sheltering obstacles close to the mast. The surface consists mostly of sand and gravel with a roughness length of less than 0.01 m. To the NW and S the wide valley is bordered by the North and South Galala Plateaus, respectively, which rise to more than 1000 m.

Gulf of El-Zayt The Gulf of El-Zayt is situated just W of the Hurghada-Zafarana road, approximately 60 km NNW of Hurghada along this road. The distance to the coastline of the Gulf of El-Zayt is 1300 m in a north-easterly direction. There are no sheltering obstacles close to the mast. The surface consists mostly of sand and gravel with roughness length of less than 0.01 m.

Hurghada The Hurghada mast is situated at the Wind Energy Technology Center of NREA, about 175 m SW of the Zafarana-Hurghada road and approximately 12.5 km N of Hurghada. The distance to the coastline of the Gulf of Suez is 650 m towards the NE. There were no sheltering obstacles close to the mast during the period when data was collected. The surface consists mostly of sand with a roughness length of less than

0.01 m. The station is situated on a coastal plateau which rises gently towards the SW, where it reaches 300 m a.s.l. at a distance of 20 km. Further to the SW the mountains rise to more than 1000 m. To the NW the terrain rises abruptly at a distance of 10 km - to heights of almost 200 m.

5.3 Comparison

The NCEP/NCAR wind speed data for Ireland is summarised in 5.1.

Height (m)	Mean	Standard Deviation	Minimum	Maximum
0	11·80	6·81	0·03	51·29
1450	12·15	6·70	0·01	46·44
3000	14·24	7·54	0·02	50·56
5500	19·42	10·79	0·02	74·06

Table 5.1: NCEP/NCAR wind speed data summary for Ireland

Table 5.2 summarises the wind speed data for Egypt. The geostrophic wind speeds are on average lower than for Ireland. This is because the pressure gradients exert a greater force over Ireland. The standard deviation of the wind speeds are also lower than for Ireland, particularly at the lower heights. The mean wind speeds show such an unusually consistent negative shear, that the mean wind speeds at not only 1500 m, but at the third height, 3000 m, are less than the mean wind speed at the ground. Normally the wind speed increases with height since it is further away from the roughness of the terrain. Negative shear can be caused by a number of effects. One is from solar heating causing the pressure gradient at the surface to be greater than at the second height of 1500 m. This stronger pressure gradient gives a stronger geostrophic wind.

Height (m)	Mean	Standard Deviation	Minimum	Maximum
0	8·01	3·61	0·11	45·61
1500	5·05	3·05	0·01	30·08
3000	7·42	5·03	0·06	42·90
5500	14·87	10·51	0·11	130·04

Table 5.2: NCEP/NCAR wind speed data summary for Egypt

Table 5.3 shows that the standard deviation of the directions is less for Egypt at 3 out of 4 heights (for formula used see A.1.2). This means the geostrophic wind for Egypt is essentially more uni-direction than for Ireland, or rather, that the wind directions are more evenly distributed around the circle for Ireland than for Egypt. This is most pronounced at the bottom and top heights.

Ireland's mid-latitudes location means that the geostrophic balance between the coriolis force and the lateral pressure gradient is stronger than for Egypt.

Height (m)	Ireland	Egypt
0	1·16	0·79
1450 / 1500	1·09	1·09
3000	1·05	0·87
5500	1·02	0·63

Table 5.3: Comparing the standard deviation of the wind directions between heights of the NCEP/NCAR data for Ireland and Egypt

This should mean that the geostrophic weather patterns take place in a deeper volume of air compared to the horizontal direction, since local effects on the ground level has less influence. This is demonstrated by the correlation between adjacent heights as shown in tables 5.4 and 5.5.

Correlation of wind speed	Ireland	Egypt
0 - 1450/1500 m	0·751	0·169
1450/1500 - 3000 m	0·781	0·401
3000 - 5500 m	0·800	0·299

Table 5.4: Comparing the correlation (Pearson) of the wind speed between heights of the NCEP/NCAR data for Ireland and Egypt

Since all the numbers are positive, the speeds between two adjacent heights are positively correlated for both countries, as expected. (The wind speed occurring at one height should be somewhat related to the wind speed occurring at a different height.) However the wind speeds for Ireland are much more correlated between heights than for Egypt. The correlation for the speeds between 0 and 1500 m is very weak, and this is due to Egypt's location closer to the equator than Ireland.

The results obtained for the correlations for the wind directions between heights (for formula see appendix A.2) are shown in table 5.5.

Correlation of wind direction	Ireland	Egypt
0 - 1450/1500 m	0·925	0·483
1450/1500 - 3000 m	0·941	0·535
3000 - 5500 m	0·936	0·831

Table 5.5: Comparing the correlation of the wind direction between heights of the NCEP/NCAR data for Ireland and Egypt

The directions are statistically positively correlated for both sites between all pairs of heights as with the speeds. Again, a much higher correlation is observed for Ireland.

The significance of this is that when classes are made at the bottom height for Ireland, the second height should follow and be better grouped with wind speed and direction than for Egypt. Hence, for the construction of wind classes,

there is less to gain by including the second height in the clustering criteria for Ireland than for Egypt. Including the second height (and the other heights) should have a greater effect for Egypt.

The inverse Froude number describing the atmospheric stability (see section 3.1), is summarised for Ireland and Egypt in table 5.6. The last two columns show that the Egypt site has many more unstable stratification occurrences than for Ireland between the lowest two heights.

Heights (m)	Mean		Standard Dev.		% negative	
	Ireland	Egypt	Ireland	Egypt	Ireland	Egypt
0 - 1450/1500	2·25	2·75	3·74	3·79	0·99	6·58
1450/1500 - 3000	3·23	5·70	6·23	25·67	0·00	1·06
3000 - 5500	3·00	6·78	4·30	10·94	0·00	0·01

Table 5.6: NCEP/NCAR inverse Froude number data summary for Ireland and Egypt. Note that one outlier has been removed for each of the Ireland values.

Chapter 6

Clustering Technique Comparison

To compare the performance of different clustering methods, a simple two-dimensional case is used. The old classification method is used to construct a set of classes for the bottom height on the Egypt data. The stability splitting is disabled such that the old method classifies the data based on wind direction and speed bins only. The resulting number of classes is 86 and the result is plotted in figure 6.1 below on the direction and speed axes. Each plot of this kind shows all the data points in the given data set, unless otherwise specified. The colour and symbol for each point uniquely defines the class to which it belongs. Where the symbols cannot be distinguised in the areas where the data is dense, the colours have been interchanged so that no two neighbouring clusters have the same colour. If one class contains data points close to 0° and close to 360° the data is automatically shifted so that the data points are displayed together on the plot.

The figure shows how the old method first divides the data into sectors of equal width and then divides the data into speed bins depending on the density of the wind speeds in that direction bin. In this case there are 16 sectors, giving each a width of 22.5°, centred at 0°, 22.5°, 45° and so on. The classes centred at 0° are all displayed with a centre of 360°. The number of speed bins ranges from 3 to 7 in each direction bin. The same set of classes is plotted in figure 6.2 below on the u and v axes. In this plot the direction of the data point can be read as the angle it makes with the positive v axis from the origin, (0,0). For comparing the different clustering techniques the speed-direction axes are used for the plots since these are the parameters on which the clustering is made. Hence it is much easier to see how well the clustering techniques perform.

The Statistical Analysis System (SAS) program [14] is used to make 86 clusters using different standard clustering techniques (see code in I.1) and the results are compared. One exception is the colour quantisation (CQ) method which was implemented using a fortran 90 program. Theoretically, the aim is to

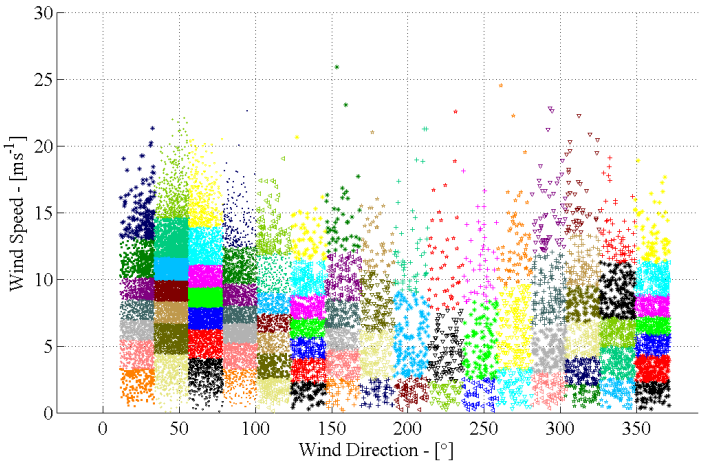

Figure 6.1: The Egypt data in 86 classes with old method, plotted on speed and direction axes

represent the speeds and directions equally well or better compared to the old method. The results for the average within new group, centroid, ward, colour quantisation, kth nearest neighbour density linkage, single linkage and complete linkage methods are shown in figures 6.3 - 6.18 below. The EML method was also attempted but the SAS procedure could not complete the algorithm due to lack of resources, even when a random sample of 10% of the data was used. [14] states that the computational time for the EML method is proportional to the cube of the number of observations where for all other methods it is the square of the number of observations. There are two figures for each method, with the exception of the CQ method. The first shows the resulting clusters from the method alone. The second shows the resulting clusters when the method is used as the first stage of a two-stage algorithm. the centroids from the method are used as seeds for the Fastclus method from the SAS program [1] (see section 4.4.1).

The data is transformed in the same way for each clustering method, with the exception of figure 6.12. The sin-cos method (see section 3.3.1) is used to represent the wind directions. Thus the clustering is made on three dimensions: the wind speed, S, the sinus of the wind direction, $\sin(DD)$ and the cosinus of the wind direction, $\cos(DD)$. The wind speed has been scaled down such that its standard deviation is 0.5 with the following formula.

[1] The result for each hierachical method can be improved by using the centroids of the clusters as seeds for a non-hierachical clustering algorithm and a second stage clustering refinement (see point 3 of section 4.3.3).

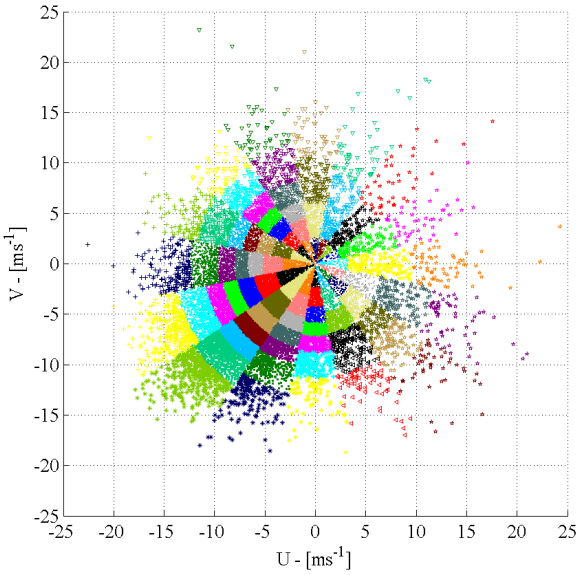

Figure 6.2: The Egypt data in 86 classes with the old method, displayed with the u and v wind velocities on the axes

$$S = \frac{S \times 0.5}{stdv(S)} \tag{6.1}$$

where
S is the wind speed, and
$stdv(S)$ is the standard deviation of the wind speed.

On the plots, the speed and direction axes have been scaled to match this standardisation so that the clustering distance is visual. This means that a ruler could be used to measure the distance used by the clustering algorithm on the page. The single and complete linkage methods proved to use too much resources on the server used, and had to be run on a random sample of 50% of the data. However, the corresponding second stage Fastclus algorithm uses all the data as only the seeds are based on 50% of the data. The statistics to show the total error sum of squares for each method plus how well they represent the wind speeds and directions on average are displayed in table 6.1. The methodology behind these statistics is described in section 3.4.1.

The average and centroid methods show very similar results. This can be explained by the fact that a centroid is the average of a set of data points. Thus, the concept of the average distance between all pairs of points (average method) is somewhat related to the distance between all pairs of points (centroid

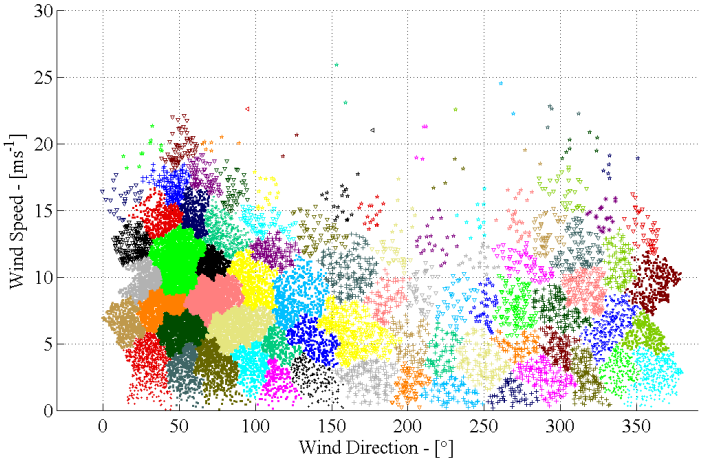

Figure 6.3: The Egypt data in 86 classes with the average linkage within new group method

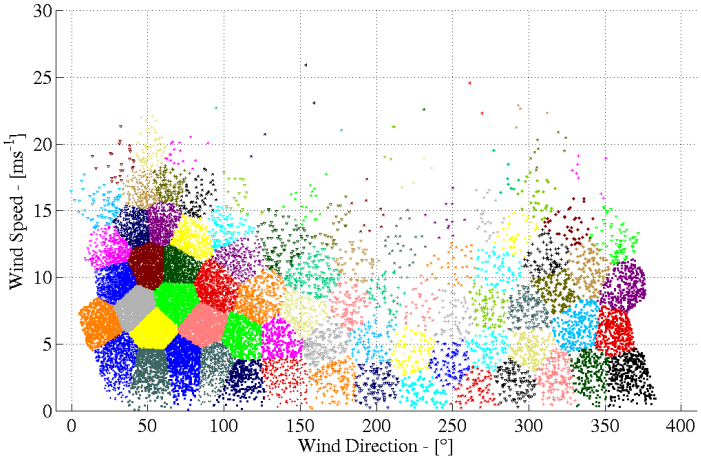

Figure 6.4: The Egypt data in 86 classes with the Fastclus method using the average linkage method seeds

method). Both methods give quite large clusters where the data is most dense around 50° wind direction. These clusters have a very big frequency compared to the existing method (see table 6.2 in appendix D). This is not the desired

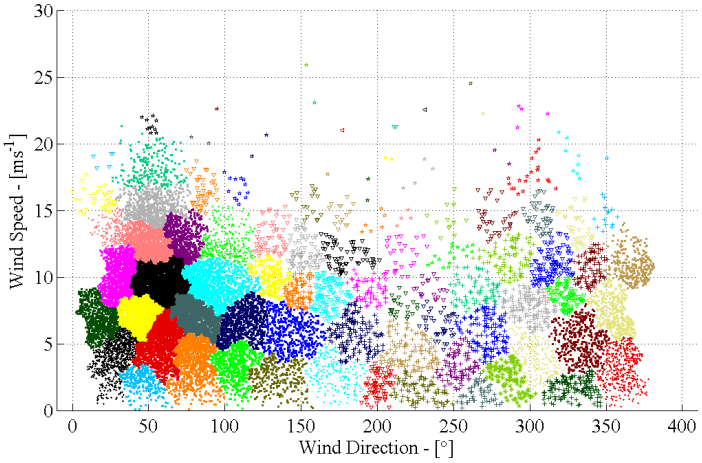

Figure 6.5: The Egypt data in 86 classes with the centroid method

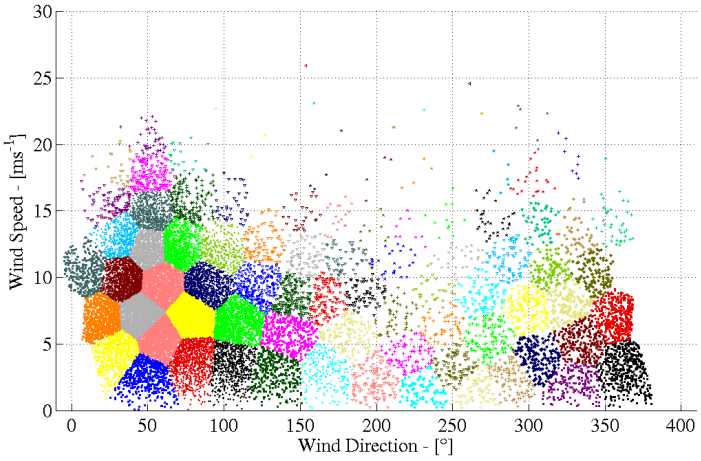

Figure 6.6: The Egypt data in 86 classes with the Fastclus method using the centroid method seeds

effect for mesoscale modelling since the bulk of the data is not well represented. This is also reflected in the total error sum of squares values for these method in table 6.1. The Fastclus algorithm improves this somewhat and then the clusters for these methods look nearly indentical (figures 6.4 and 6.6), although the error sum of squares values show that the average method is slightly better

at reducing the error sum of squares.

As expected, figure 6.7 shows that the ward method does give the desired trend of smaller clusters in the dense data area. However, the cluster shapes are a little elongated which is a characteristic of agglomerative algorithms, and one of the reasons why the don't necessarily find the optimum clusters. The Fastclus method does find the optimum clusters for a given set of seeds and the result is shown in figure 6.8. The total error sum of squares values in table 6.1 are comparatively low for the ward method.

The resulting clusters produced from the colour quantisation method in figure 6.9 look similar to the existing classification method (figure 6.1). This is because the CQ method divides up the data with boundaries perpendicular to the either the wind speed or wind direction axes. One difference with the CQ clusters is that the width of the clusters direction-wise is not fixed. This helps the CQ method produce a lower total error sum of squares than the existing method (see table 6.1). The Fastclus method does not change the look of the clusters very much. This is because the Fastclus method produces clusters with straight boundaries mid-way between the centroids(see section 4.4.1), and the boundaries using the CQ method alone in 6.9 are already straight. The Fastclus method only applies one iteration of the Forgy algorithm and hence, the already straight boundaries do not move much.

The full Forgy algorithm is applied to the CQ seeds in figure 6.12. Here, a total of 74 iterations were applied for convergence and the clusters now look similar to the Fastclus with ward seeds clusters in figure 6.8. Another difference in this Forgy method applied is that the wind direction is represented directly rather than using the sin-cos method. This means that an extra parameter is used to scale the direction values so they are weighted comparatively compared with the wind speed. The direction values in degrees are thus divided by 55. This value is chosen so that the percentage improvement of the speeds and directions is equal when the Forgy algorithm is applied. Thus, the directions are treated as values from 0 to 6.55 (instead of 0 to 360) with circular characteristics so that 6.55 is the same as 0. The result is shown in figure 6.12.

Note that the colours and symbols of the clusters in the Forgy with CQ seeds result match with the colours of the clusters from the CQ algorithm alone. This indicates how the clusters migrate with each iteration as the centroid gets updated. The CQ algorithm result is shown a second time in figure 6.11 for easy comparison between them on the same page.

The resulting clusters when using the kth nearest neighbour density method, with $k = 28$ are shown in figures 6.13 and 6.14. The value of 28 for k is the optimum value to minimise the error sum of squares and give the best looking clusters. It is found through trial and error as recommended in [14]. The resulting clusters are very sensitive to k and values larger or smaller than this gave inconsistent cluster shapes and sizes. The result with $k = 28$, gives even smaller clusters in the dense region of the data than the ward and CQ methods. However the clusters in other areas seem rather large and the resulting centroids would be too sparse to represent this data well. This is reflected in a high error sum of squares values in table 6.1.

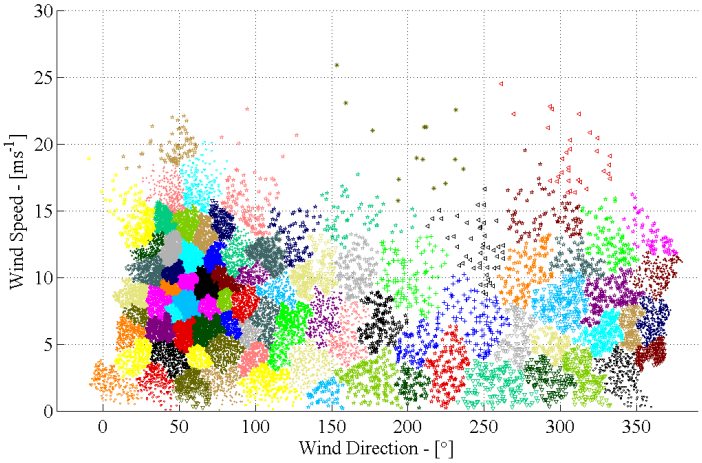

Figure 6.7: The Egypt data in 86 classes with the ward method

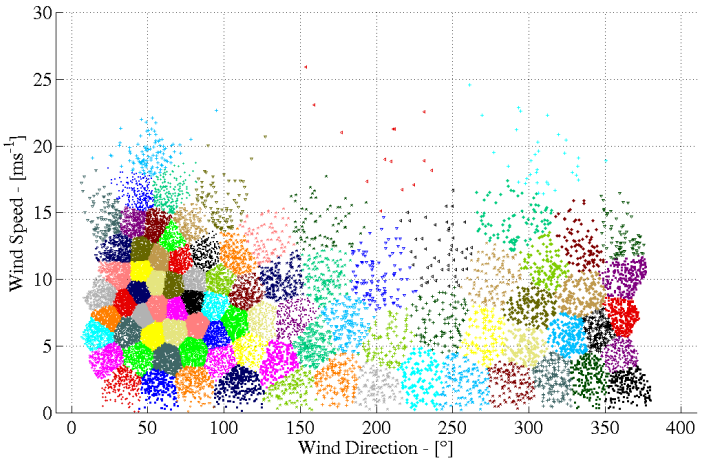

Figure 6.8: The Egypt data in 86 classes with the Fastclus method using the ward method seeds

The single linkage result in figure 6.15 is very bad, with a severe case of chaining occurring to give one very large cluster containing over 98% of the data. Furthermore, the single linkage method could only be run on a random 50% sample of the data due to its lerge computational needs. The Fastclus

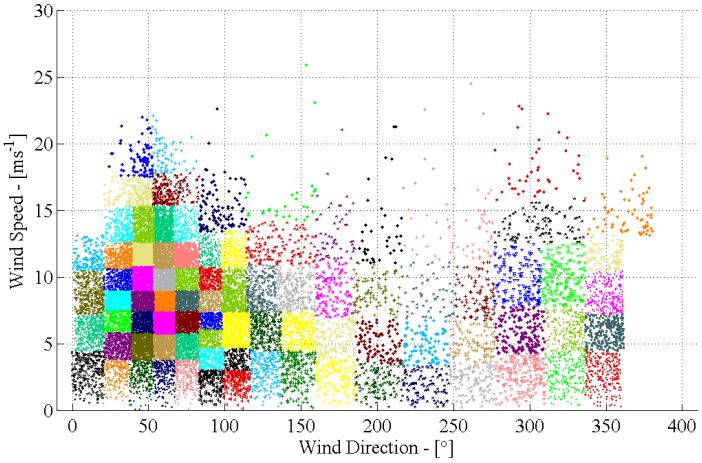

Figure 6.9: The Egypt data in 86 classes with the colour quantisation method

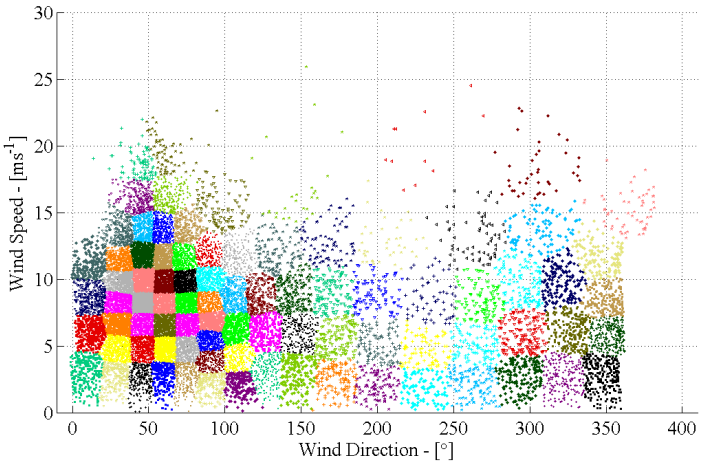

Figure 6.10: The Egypt data in 86 classes with the Fastclus method using the CQ method seeds

algorithm improves the resulting clusters a little but the single linkage method is not a good method for representing data. As explained in section 4.2 the

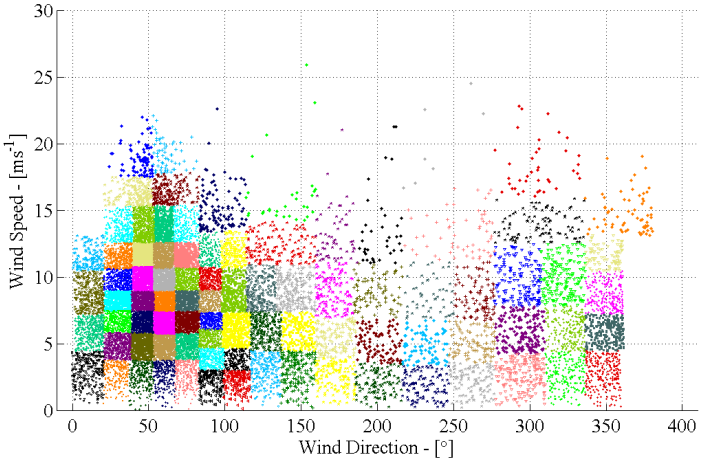

Figure 6.11: The Egypt data in 86 classes with the colour quantisation method

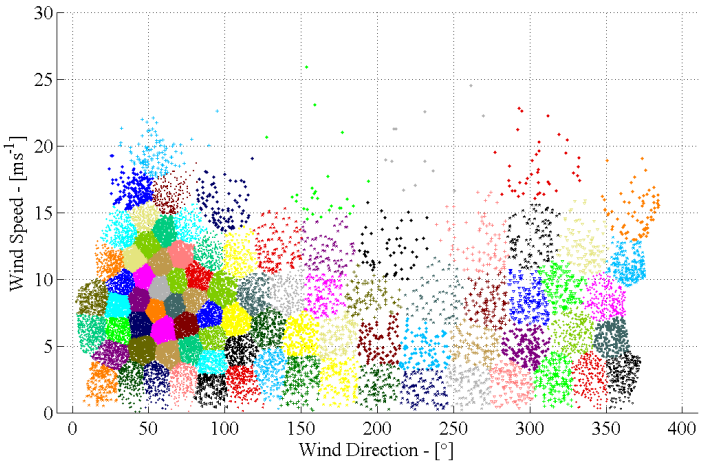

Figure 6.12: The Egypt data in 86 classes with the Forgy method using the CQ method seeds

single linkage method is good for finding irregular shaped clusters, which is quite a different task.

The complete linkage method also had to be run with a random 50% sample of the data. The resulting clusters from the method alone and with Fastclus

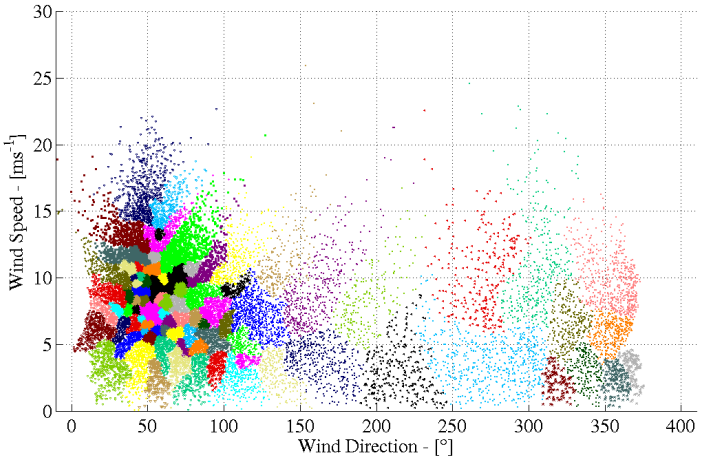

Figure 6.13: The Egypt data in 86 classes with the kth nearest neighbour method, $k = 28$

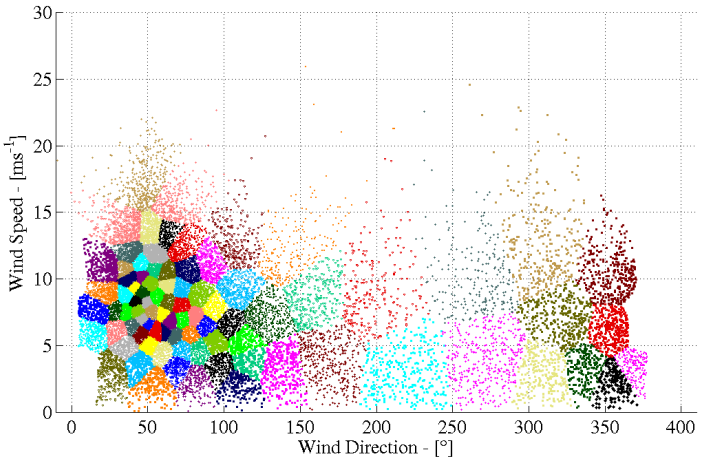

Figure 6.14: The Egypt data in 86 classes with the Fastclus method using the density method (K = 28) seeds

look similar in physical size. The total error sum of squares after Fastclus is shown in table 6.1. It is the third best result for the methods, and it is better than the old method alone.

Method	**Total error sum of sqrs**		**Mean stdv Speeds**		**Mean stdv Directions**		**Mean U^3 (% lost)**	
	Only	**Fc**	**Only**	**Fc**	**Only**	**Fc**	**Only**	**Fc**
Old	632	547	0.83	0.71	6.31	6.42	2.56	1.75
Average	926	711	0.99	0.88	7.86	6.78	2.83	2.09
Centroid	977	783	1.00	0.90	8.15	7.30	2.69	2.17
Ward	533	454	0.76	0.70	5.85	5.46	1.74	1.44
CQ	499	467	0.74	0.70	5.66	5.55	1.63	1.45
CQ,Forgy	499	449	0.74	0.70	5.66	5.34	1.63	1.45
Density	1098	781	1.14	0.93	8.10	6.67	4.39	2.90
Single$*$	(9168)	3061	(3.48)	1.78	(51)	14.61	(36)	7.53
Complt$*$	(386)	585	(0.88)	0.78	(7.3)	6.27	(2.2)	1.74

Table 6.1: Comparison of the clustering methods, used alone (only) and used with Fastclus (Fc), except "CQ,Forgy" which is CQ only and CQ with the full Forgy algorithm. The error sum of squares values are based on the sin-cos reprsentation of the data since this the axes used by all of the clustering methods (except Forgy). If the error sum of squares is based on the wind direction values divided by 55, the same comparitive trend is the numbers is observed. $*$ denotes clustering was performed on only 50% of the data. Hence, values in parentheses cannot be directly compared with the other values.

Method	**Max cluster freq**		**Total top 5**		**Max dir range**	
	Only	**Fc**	**Only**	**Fc**	**Only**	**Fc**
Old	4·4	4·2	22·1	18·9	22·4	81·4
Average	14·1	7·9	43·6	33·9	45·3	35·8
Centroid	12·8	9·2	39·9	24·9	51·5	44·1
Ward	3·6	3·0	16·5	13·5	82·8	89·4
CQ	3·3	3·3	15·1	15·1	59	63·8
CQ Forgy	3·3	2·8	15·1	13·3	59	66·8
Density	4·6	2·2	18·5	10	87·7	88·9
Single$*$	98·5	36·5	99	69	360	96·9
Complete$*$	9·4	6·7	34·6	29·2	38·5	39·6

Table 6.2: The maximum frequency and total frequency of the top five clusters in %. The maximum direction range for the classes is also given in the last 2 columns.

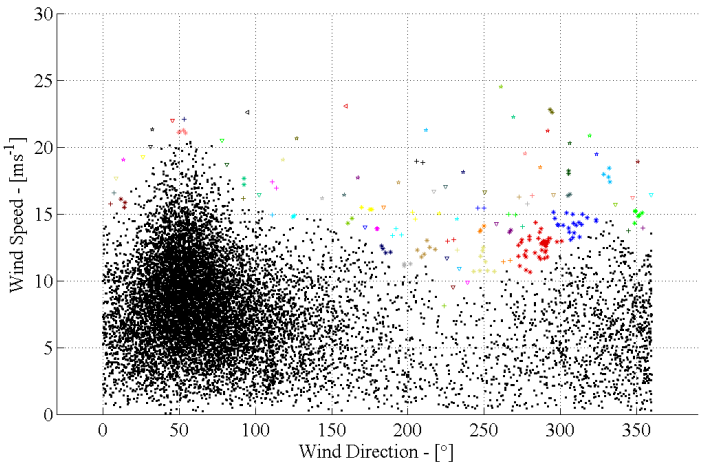

Figure 6.15: The Egypt data in 86 classes with the single linkage method, using a random sample of 50% of the data

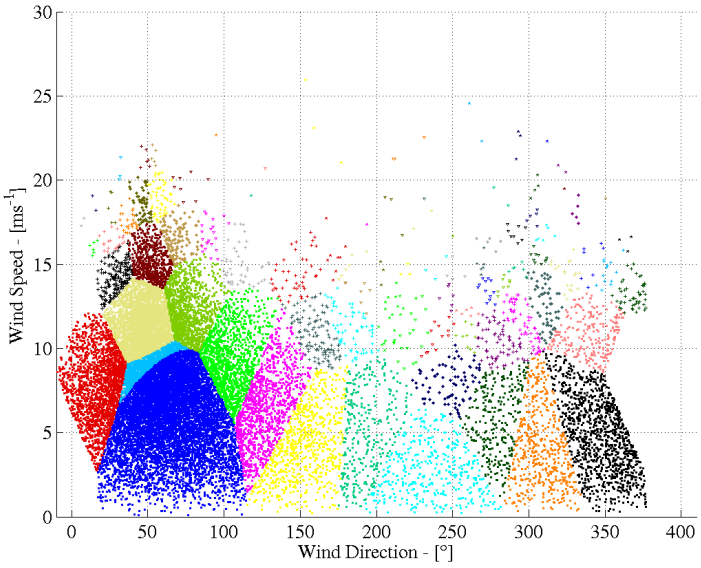

Figure 6.16: The Egypt data in 86 classes with the Fastclus method using the single linkage method seeds

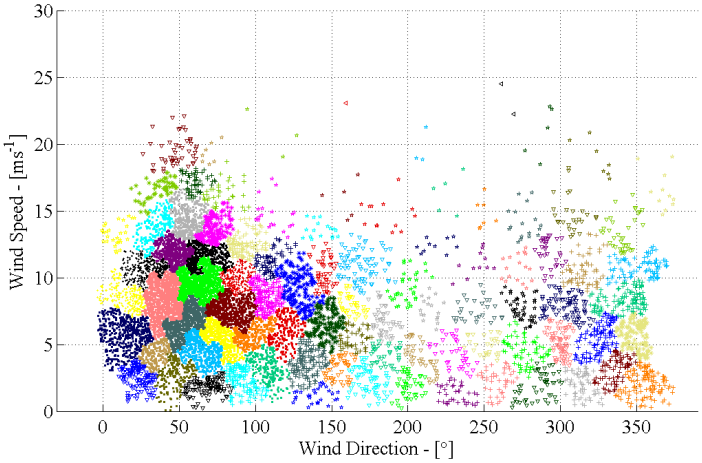

Figure 6.17: The Egypt data in 86 classes with the complete linkage method, using a random sample of 50% of the data

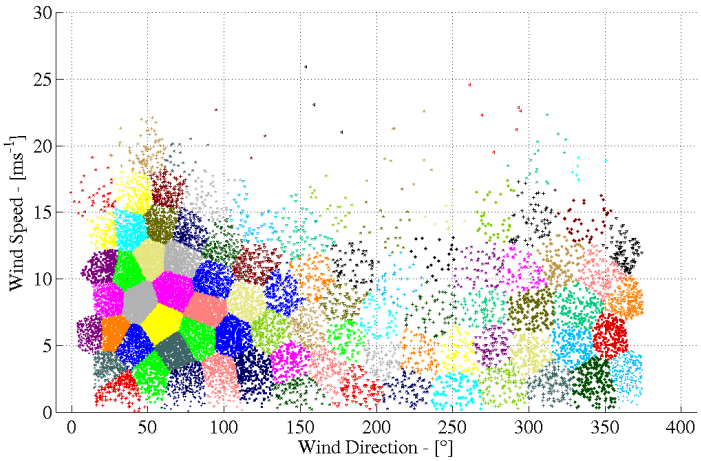

Figure 6.18: The Egypt data in 86 classes with the Fastclus method using the complete linkage method seeds

The comparable total error sum of squares values are plotted for all methods in this chapter except the single linkage method in figure 6.19. The single linkage method is left out because its values are extreme. The CQ and Ward methods give the best representations used alone and when improved by Fastclus or the full Forgy methods. The CQ-Forgy method is chosen for the mesoscale modelling experiments for two main reasons.

1 The CQ-Forgy method gives the lowest error sum of squares of all the methods.

2 The nature of the divisive CQ method is similar to the old method. This nature of dividing up the data means that the CQ method gives clusters where the direction range is not too large and lower compared to the Ward method (see table 6.2). It is thought that a lower maximum wind direction range in the clusters should give a better result in the numerical wind atlas.

It is possible that the Ward method could give a lower error sum of squares result when combined with the full Forgy method though this is not expected to be very different to the CQ-Forgy result. Hence, given that the Ward and CQ methods give similar error sum of squares results, the second reason above is why the CQ method is chosen over the Ward method.

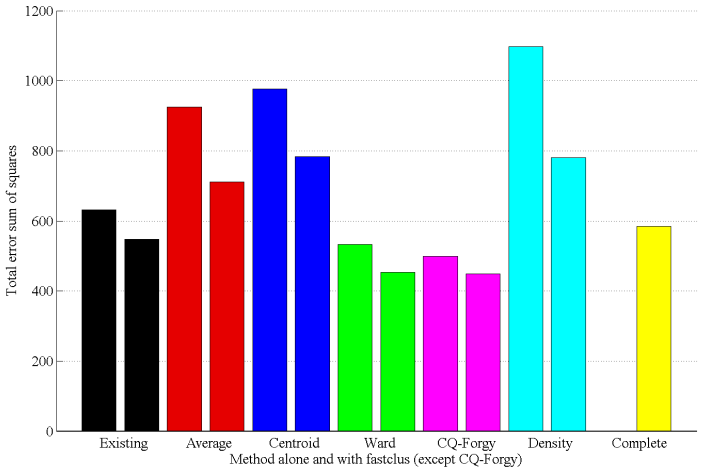

Figure 6.19: The different hierarchical clustering methods compared for the total error sum of squares. For each method, the results for the method alone, and the method in combination with Fastclus, are shown.

6.1 Using the principal axis for CQ

The colour quantisation (CQ) algorithm is published in [35] with the cutting axis being swept along each of the three axes in the colour spectrum to find the best place to partition the data. This is a logical way to do it for CQ since the axes are discrete. For geostrophic climate data, the axes are continuous, yet the algorithm can be done the same way as shown in section 4.2.14. However, another logical option exists to scan the cutting plane on the principal axis of the data. The principal axis is the axis through the data, along which the data has the most variance. Splitting the data along this axis might give a better total error sum of squares than just using the regular axes. This was implemented and tested in the following.

The CQ algorithm was applied to the Egypt data as in the previous section 6. Figure 6.20 shows the percentage difference the principal axis method gives in the weighted variance compared to the regular axes method for the Colour Quantisation algorithm. The variances are compared at each step in the hierarchical algorithm. If this value is less than zero, the principal axis method is gives a lower error sum of squares. The plot shows that an improvement in the error sum of squares is obtained for the first eight partitions (up to nine classes). When the 10th class is made, the regular axes method becomes the better of the two methods and there it remains up to at least four hundred classes.

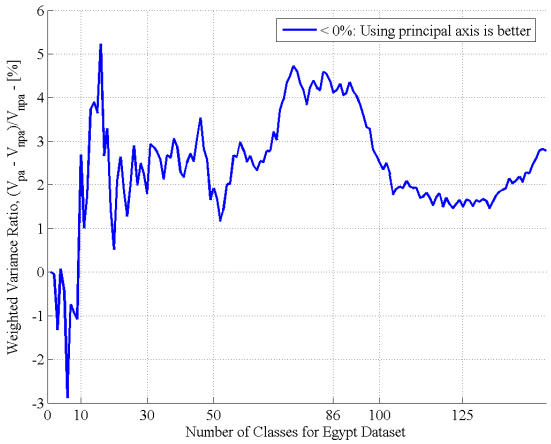

Figure 6.20: The percentage improvement in the error sum of squares by using the regular axes compared to using the principal axis in the CQ method on the Egypt data

The actual weighted variance values for 9 and 10 classes are shown in table 6.3 for each method.

The following 2 figures, 6.21 and 6.22 show the 10 classes made with each method and indicate which of 9 classes was partitioned to make the tenth. It is

Number of classes	All regular axes	Principal axis
9	16984	16800
10	14964	15367

Table 6.3: Error sum of squares comparison between the axis methods at as 9th and 10th clusters are made

not obvious why the regular axes error sum of squares should suddenly become better than the principal axis method on this 9th splitting. Either way, the difference between the methods is not very big, and this is only the first stage of the clustering. Using the regular axis method is simpler and it is decided to use this method.

6.2 Data transform theory - wind speeds

Two non-linear transforms are considered for the wind speed. Both are rejected as explained below.

- It was suggested that perhaps the wind energy should be used instead of the wind speed for clustering. This could be done by cubing the wind speeds before standardisation. This however results in too spread low wind speed clusters since the data points at lower speeds are closer than before, relative to higher wind speeds. Thus clustering on the cube of the wind speed does not give the desired result.

- It is known that at high wind speeds, say, above 10 ms^{-1}, dividing the speed into different clusters becomes less important for mesoscale modelling. One of the reasons for this is that the shear on the wind speed is consistently positive for high wind speeds at the ground. The clustering principle automatically gives some bias to larger clusters at higher wind speeds since the data points are generally more spread than for the more common lower wind speeds. The bias effect could be magnified by taking the cube root of the wind speed. The results from this give very small wind speed bins at the lowest wind speeds. Although this make the speed clusters bigger at higher speeds, it distorts the clustering at speeds less than 10 ms^{-1} which should all be treated equally.

Thus it is decided to rely only on the intrinsic small bias effect due to the decreasing density at higher wind speeds, and only cluster on the wind speed directly. This also gives the most direct comparison with the old method, as it also considers the plain wind speed.

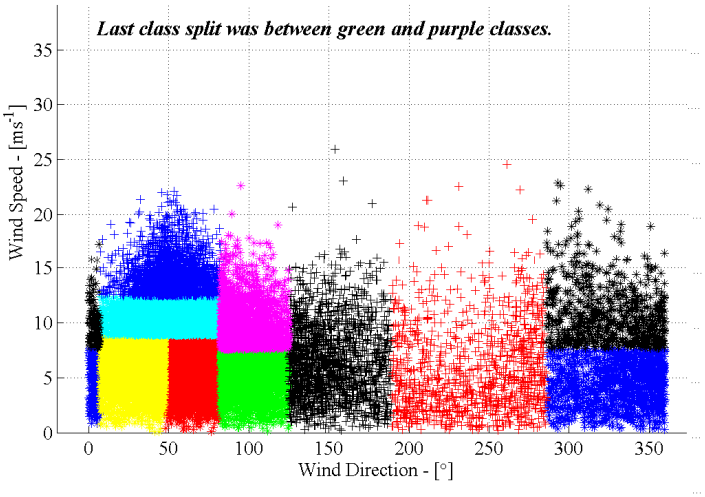

Figure 6.21: 10 clusters using the CQ method with all regular axes

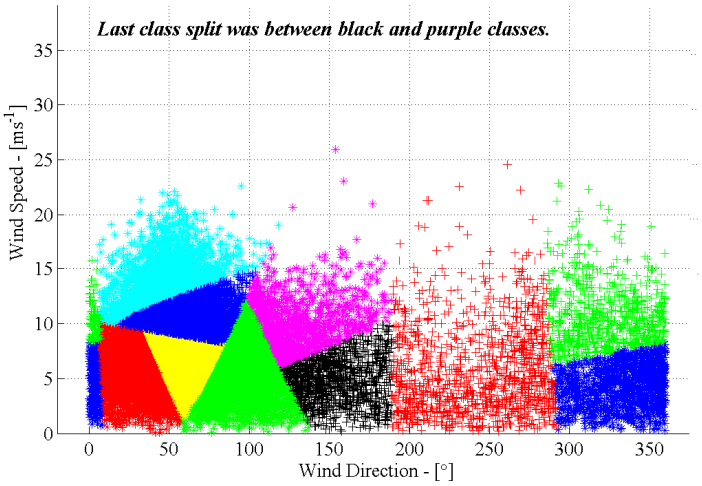

Figure 6.22: 10 clusters using the CQ method with the principal axis

Chapter 7

Clustering Technique Used: Colour Quantisation with Forgy

A two-stage clustering method is chosen as the best method to represent a synoptic climate with clustering. The first stage is that the hierarchical colour quantisation (CQ) method (see section 4.2.14) is used. In the second stage, the non-hierarchical Forgy method (see section 4.4.1) is performed using the centroids of the first stage result as seeds. This method was selected based on the results from the previous chapter (chapter 6). All the important variables as described in section 3.1 are considered. While the old method considers the wind speed and direction and inverse Froude number at the lowest level, the clustering method considers the wind speeds and directions at the three more available heights, as well as the inverse Froude number from two more height differences. This amounts to 11 variables for the clustering. The sin-cos method is used to represent the directions in the CQ method to avoid the circular characteristics. This means the directions are represented with two dimensions as described in section 3.3.1. Thus for the CQ method, 15 variables are used. The number of variables becomes 11 again when the Forgy algorithm is applied. For each stage there are three different types of variables with different units. Furthermore, amongst the variables with the same units, some are more important than others. Thus a set of parameters is devised for controlling the weighting of the variables to obtain an optimal clustering. In section 7.1 these parameters are listed and described. Figure 7.3 shows the main parameters and what variables they improve if they are increased or deceased. Section 7.2 describes a the procedure used to set the parameters, in order to achieve the desired representation of the climate. Finally, section 7.3 describes how the varying frequencies are calculated for neighbouring NCEP/NCAR data.

7.1 The parameters

To have full control over a two-stage clustering method, where the number of variables change from 15 to 11, and where there are three different variable types, many parameters are required. The clustering methods consider all variables simultaneously when deciding which data points are closer or further from each other. The standard deviation of the values on a variable is scaled to control its influence on the distance compared to the other variables. The objective function the clustering algorithm tries to minimise, is calculated from the scaled variables. If the values on one variable are further apart than another (i.e. with a greater standard deviation), the clustering algorithm will divide that variable up into more different classes than it will for the other variable. In a sense, the clustering algorithm will favour the variable with the higher standard deviation. For CQ there are two sinusoidal variables to represent the directions (the cosinus and sinus of the direction angle) and one speed variable for each height (see section 3.3.2). In this way, the clustering algorithm sees the wind data on the surface of a cylinder, where the wind speed is the height of the cylinder and the wind direction is the position around the curved surface. The calculated distance between two data points is phsyically the direct path through the middle of the cylinder. When Forgy is used, this distance becomes the length along the arc, which is seen to be more accurate for the best classification. The following explains each of the parameters in the perl program (see appendix F on page 177) and what they control. Parameters 7.1.1 - 7.1.4 are needed both stages of the method and 7.1.5 - 7.1.6 are required for the translation from CQ to Forgy. As these parameters are introduced the appropriate variables are updated with the equations shown, always keeping their values from the previous equations. Section 7.1.7 describes the other important parameters for running the clustering algorithm.

7.1.1 *R*

Since the wind speed and direction are in different units, a factor is required to scale the speed values. The parameter, R, sets the ratio of importance between the wind speeds and the cos-sin variables in the CQ algorithm. Specifically, the wind speeds at each height are standardised to have a standard deviation of R, while the cos and sin variables are untouched. Physically, the value of R controls the height of the cylinder as seen by the clustering algorithm.

$$S_i = \frac{S_i \times R}{stdv(S_i)} \tag{7.1}$$

$cos(D_i)$ and $sin(D_i)$ unchanged, for $i = 1$, 2, 3 and 4 where
S_i is the wind speed at the ith height,
$stdv(S_i)$ is the standard deviation of the wind speeds at the ith height, and
D_i is the wind direction at the ith height.

7.1.2 Height weights, *wgt(1-4)*

It is assumed that once set for the bottom height, the parameter R should be the same for all heights. Physically this means that the ratio of the radius to the height of each cylinder is the same. However, the size of each cylinder needs to be controlled to give some heights more weighting than others in the clustering algorithm. The parameters *wgt(1-4)* are each multiplied by the wind speeds and cos-sin values for the four heights respectively.

$$S_i = S_i \times wgt(i) \tag{7.2}$$
$$cos(D_i) = cos(D_i) \times wgt(i) \tag{7.3}$$
$$sin(D_i) = sin(D_i) \times wgt(i) \tag{7.4}$$

for i = 1, 2, 3 and 4.

The figure below demonstrates a physical representation of the variables for calculating the distance between data points. In this example the weighting on the bottom height is much more than on the other heights. The figure shows how the distance between the points at the bottom height will dominate the clustering.

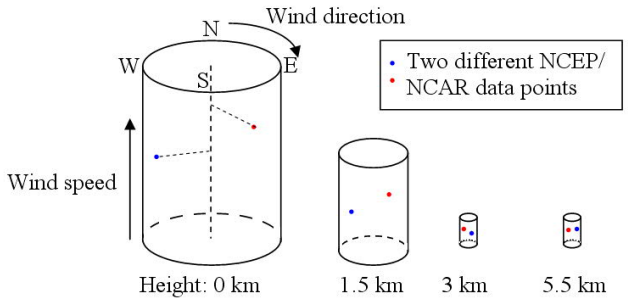

Figure 7.1: A physical representation of the 8 speed and direction variables for teh clustering algorithm. Tow data points are shown with height weights, wgt(1-4) = [8, 4, 1, 1].

7.1.3 Relation between the wind and inverse Froude number, *sd_invFr*

The inverse Froude number is used to describe the stability of the atmosphere from the temperature difference between two heights. The standard deviation of the inverse Froude number is set to the inverse of the *sd_invFr* parameter. This means the larger *sd_invFr* is, the more weighting the speeds and directions have, compared to the inverse Froude numbers. The formula for calculating the inverse Froude number is equation 3.4 in section 3.1.

$$Fr^{-1}_{i,i+1} = \frac{Fr^{-1}_{i,i+1}}{sd\_invFr \times stdv(Fr^{-1}_{i,i+1})} \tag{7.5}$$

for $i = 1$, 2 and 3 where
Fr^{-1}_{i-i+1} is the inverse Froude number calculated from the temperature difference between the ith and $i+1$th height, and
$stdv(Fr^{-1}_{i-i+1})$ is the standard deviation of the ith inverse Froude number.

7.1.4 Height weights for the Froude numbers, *wgtFr(1-3)*

As with the speeds and directions, the inverse Froude numbers also need to be scaled to weight each height for the clustering. Theer are three inverse Froude numbers calculated between each pair of neighbouring heights (1-2, 2-3 and 3-4). The weighting on the inverse Froude number at bottom height is already controlled by the *sd_invFr* parameter, hence the weighting from *wgtFr* is scaled so that *wgtFr(1) = wgt(1)* and the other two heights follow.

$$Fr^{-1}_{i-i+1} = Fr^{-1}_{i-i+1} \times wgtFr(i) \times \frac{wgt(1)}{wgtFr(1)} \tag{7.6}$$

$$\tag{7.7}$$

for $i = 1$, 2, 3 and 4.

7.1.5 *RF*

Since Forgy works on the basis of the distance along the arc, it only uses the speeds and directions directly, rather than cos and sin. This means that it requires a different factor to *RF* to scale the directions compared to the speeds. The diagram below demonstrates the change in the variables as seen by the clustering algorithm from the CQ stage to the Forgy stage.

This would seem to be pose large problems since the Forgy algorithm is supposed to improve the clustering results obtained by CQ. The optimal result is obtained if both methods cluster the data points with equivalent weightings on each variable. This would mean that the Forgy method would simply improve on the already good result obtained by CQ. Thus, a parameter is required to scale the new direction variable in Forgy so that it is weighted equivalently to the sin-cos variables in CQ. This task is not as hard as it seems, since the sin and cos variables are directly calculated from the directions and are highly correlated (i.e. the behave as one dimension). The biggest change is that the distance between directions is now the arc length around the circle, rather than the straight distance across the circle. For simplicity, the wind speeds remain as they were in CQ, with a standard deviation of R. The directions are then divided by the new parameter, *RF*.

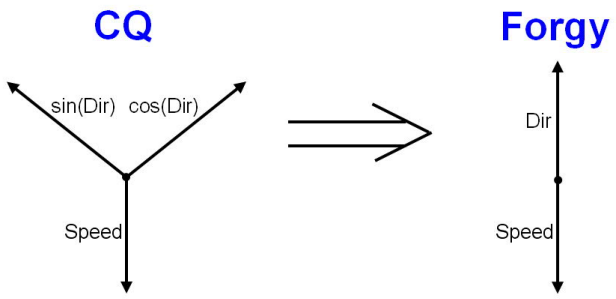

Figure 7.2: The change in the speed and direction variables at one height, between the CQ method to the Forgy method. In total, this transformation occurs at all 4 heights, and there are 3 inverse Froude number dimensions that are unchanged.

$$D_i = \frac{D_i}{RF} \tag{7.8}$$

for $i = 1$, 2, 3 and 4.

7.1.6 *sd_invFr_factor*

Since each inverse Froude number was one of 15 variables for CQ and now is one of 11, and that the directional distance is now calculated differently, a parameter is used to rescale the inverse Froude numbers for the new situation. However, due to the fact that the speeds are unchanged and the rescaling of the directions is already done, this new parameter is usually close to 1.

$$Fr^{-1}_{i-i+1} = \frac{Fr^{-1}_{i-i+1}}{sd\_invFr\_factor} \tag{7.9}$$

$$\tag{7.10}$$

for $i = 1$, 2 and 3.

7.1.7 Other parameters

The two-stage algorithm currently works to finding a specific number of clusters. This is set by *nCL*. The number of variables to be used can be set with *nDim*. There are 3 options for this parameter, 2, 8 or 11. If 2, the clustering is performed on the speeds and directions at the lowest height only. If 8, the clustering is performed on the speeds and directions at all heights, only. If 11,

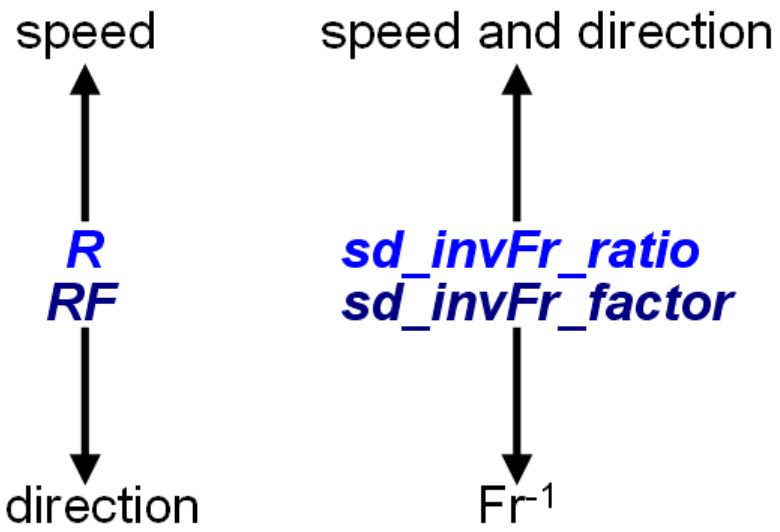

Figure 7.3: The four main parameters showing which variables are improved if the parameter is increased (up arrow) or decreased (down arrow). The two top parameters are used in the CQ and Forgy algorithms, and the two matching lower parameters are only used in the second stage in Forgy algorithm.

the clustering is performed at all heights with the speeds, directions and inverse Froude numbers.

The *forgy* can be set to 0 or 1. If 0, only the CQ algorithm is used. If 1, the full two-stage algorithm is used.

The *outputType* variable can be set to 0, 1 or 2. If 0, an output file is generated with "txt" extension displaying the statistical summary of the clustering result and each data point with a cluster ID. This file can be read into the MatLab programs (see appendix H) to analyse the data further and plot the clusters. If 1, the output file with extension "cl" is made for Risø's mesoscale modelling procedure. If 2, the output file with extension "frq" is made for Risø's mesoscale modelling procedure. If 3, the output file with extension "nsi" is made for Risø's mesoscale modelling procedure.

7.2 Procedure to set the parameters

A careful procedure is used to set the parameters, as the clustering result is very sensitive to them. Since the algorithm is in two stages it is important to set the parameters for CQ right first, before setting the translation parameters, *RF* and *sd_invFr_factor*, even though these remain almost fixed once found the first time. There are 3 steps to follow to set the parameters right for a classification of a wind climate with the CQ-Forgy method.

1 Before starting, the location specific parameters need to be set correctly in the perl program (see appendixF to view the full program). This includes:

- The file names of the NCEP/NCAR data to be used (the main one for clustering on, and the other files for the varying frequencies option).
- The associated file paths and other file parameters.
- The gesotrophic wind data heights for the site (e.g. 0, 1500, 3000 and 5500 m).
- The *azklm* factor, which is the angle of rotation of the KAMM domain.

2 Run the CQ algorithm using only the speed and sin-cos dimensions at the lowest height ($nDim = 2$ and $forgy = 0$). The lowest height is usually the most important. The resulting mean standard deviation of the speeds and the directions are printed in the output text file ($outputType = 0$). If the speed standard deviation is is too low, or the direction standard deviation is too high, R should be decreased and the program run again. Through trial and error, the value for R can be found, though it may need a slight adjustment when the other variables are introduced in the next step. Experiments have shown that for the data to be divided into around 16 sectors, a typical value for the mean standard deviation of the directions is around 6.4. The value for R is usually around 0.5 and this is a good starting value. Note that this step is only to find the rough value for R so not much time should be spent on it as the situation changes with the next step.

3 Increase $nDim$ to 11 and run the CQ algorithm on all variables ($forgy = 0$ still). Now parameters 7.1.1 - 7.1.6 are all being used and their value depends on the desired values for the 11 mean standard deviations of the variables. Again, trial and error is used to find the values for these parameters. Note here though, that the values obtained for all the mean standard deviations will all improve when Forgy is added in the last step (unless they are given an insignificant weighting). For most of the mesoscale model runs in the next section, the top two heights of all of these parameters are considered completely unimportant. Thus, their weights are 1000 times smaller than the lowest height weight in the *wgt* and *wgtFr* parameters. With this simplification, only *sd_invFr* and *wgt(2)* need to be found at this step, though a small adjustment to R is probably required since the inverse Froude number is often slightly correlated with the directions and/or speeds.

4 The Forgy method can now be added to improve the standard deviation values further ($forgy = 1$). Trial and error is used to set the RF and *sd_invFr_factor* parameters so that the percentage improvement in the standard deviation for each variable is as equal as possible.

5 Once the parameters are set, change *outfileType* to 1, 2 and 3 and rerun the program for each to produce the output files required for KAMM. Make sure the *freqType* parameter is set for the fixed or varying frequencies option if outfile type 2 is used. Note also, that outfile type 3 is only required if the *.nsi files are to be used.

7.3 Varying frequency calculation

The use of varying frequencies to improve a numerical wind atlas is described in section 3.2.1 for the existing procedure. The existing method uses the class boundaries to calculate the frequencies for the neighbouring NCEP/NCAR data as described in section 3.2.1. Since the clustering method described above finishes with the Forgy method assigning each data point to the closest seed after refining the values for these seeds, this can be used to recalculate frequencies for a different data set. The seeds are fixed and one run of the Forgy algorithm is used to assign each data point in the new set to the closest seed, taking care to transform the data with the same factors used in the original data set, and also to include the coriolis factor as the old method. The frequencies of the resulting clusters are the frequencies to be used for the new data set. This is a more optimal way to calculate the frequencies for the new data set than the existing method.

7.4 Interpolation for WAsP

As described in section 3.2.2, around 150 classes does not give enough simulation results to construct an accurate Weibull distribution for making a numerical wind atlas. The original procedure assumes that the sector width of each class is constant. Since this is not the case for clusters, a new interpolation scheme is divised as follows. For each cluster centroid (geostrophic wind forcing), the distance is calculated to all other cluster centroids using the same scaled centroid values for each variable from the parameter values as described in section 7.2, above. The closest centroid with a greater wind direction at the bottom height, and the closest centroid with a smaller wind direction[1] at the bottom height are the two selected closest wind classes. The splitting is then performed on these two wind classes in the same manner as described for the old method in section 3.2.2. This new method is referred to as the "NSIB" method (Nearest Simulation Interpolation on Both sides with respect to wind directions).

The old interpolation can also be applied directly on the clusters. In this case, the old program typically usually chooses 24 as the "number of sectors" for the clusters. This is equivalent to a sector width of 15°. This means that the nearest cluster centroid is chosen from all centroids between 5° and 20° way from the centroid being split (see section 3.2.2). This may not be the optimum

[1] Greater means in a clockwise direction around the circle and visa versa for a smaller direction

method for every cluster in a representation, yet it is a valid method. Hence, both this old method, and the new NSIB method above are tried in the results.

Chapter 8

KAMM Simulation Results for Ireland

The CQ-Forgy method is used to make 13 sets of clusters using the NCEP/NCAR geostrophic wind data for Ireland and Egypt. Each set of clusters is designed to focus on certain variables using the parameters described in 7.1. The existing classification method is used a make a set of the classes the old way. The KAMM/WAsP method [11] is applied to each case using the centroids and frequencies of the classes or clusters. The resulting wind atlases for the specific sites described in chapter 5 are compared with the wind atlases from measurements.

The existing classification method is used to divide the NCEP/NCAR geostrophic wind data into 151 classes. The data is divided into 16 sectors, with between 3 and 7 wind speed bins. Finally, the classes with low wind speeds are divided again in up to 3 inverse Froude number bins. The classes are plotted in figure 8.1. The different colours for the lower wind speeds show how the classes are split into 2 or 3 inverse Froude number bins.

All cluster attempts use the same number of classes as the existing method, 151, except for the fourth batch where higher and lower numbers of clusters are tried. The first batch of clusters, batch A, only focusses on the lowest height. The first run is a general improvement and the other three each focus as much possible on one of the three variables, speed, direction and inverse Froude number, while keeping the other two variables equivalently represented as the old method. This is to analyse the sensitivity of the simulation results to the variables at the lowest height. The second batch, B, experiments with the other heights and the maximum allowed direction range. The third batch, C, tries extreme focus on the three variables at the lowest height, disregarding the representation of all the other variables. The fourth batch, D, tries smaller and larger numbers of clusters. Table 8.1 below, summarises the features of each attempt. Table 8.2 shows the statistics on the important variables for each KAMM run made, including the existing method. Figures C.1 - C.13 in

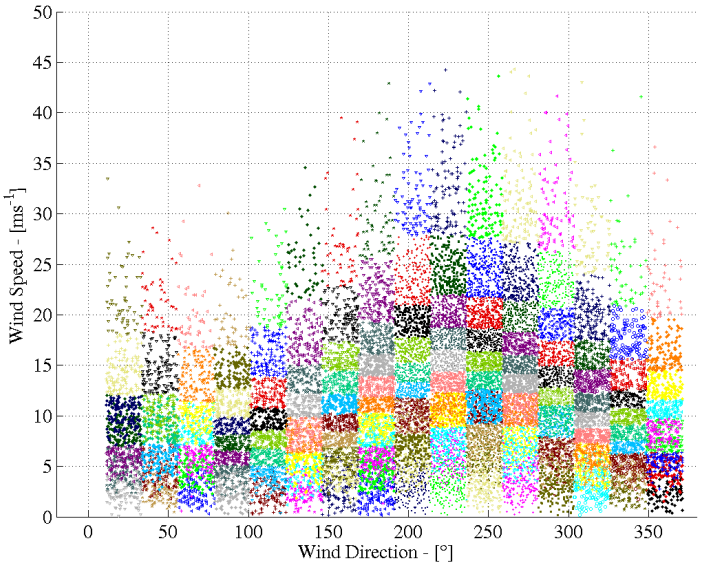

Figure 8.1: The Ireland data in 151 classes with existing method, plotted on speed and direction axes at the lowest height

appendix C.1 show the clusters made for each run on the direction and speed axes at the lowest height. Figures C.14 and C.15 compare the classes for the existing method with the clusters for KAMM run B1 at the second height. On examining these figures, the colours hint that there is extra clarity for the clustering run at the second height. The actual parameter values used for these runs are listed in appendix D.

Run	Features
A1	General improvement on the wind speed, wind direction and inverse Froude number at the lowest height
A2	Improvement on the wind speed at the lowest height while keeping the wind direction and inverse Froude equivalent
A3	Improvement on the wind direction at the lowest height while keeping the wind speed and inverse Froude equivalent
A4	Improvement on the inverse Froude number at the lowest height while keeping the wind speed and direction equivalent
B1	Improvement on the wind speed and direction at the second height while keeping all lowest height variables equivalent
B2	Improvement on the wind speed and direction at all other heights while keeping all lowest height variables equivalent
B3	Improvement on the wind direction at the lowest height with a 22.15° direction range limit for all clusters while keeping the wind speed at the lowest height equivalent and keeping the inverse Froude number within 10%
C1	Extreme improvement in the wind speed at the lowest height disregarding all other variables
C2	Extreme improvement in the wind direction at the lowest height disregarding all other variables
C3	Extreme improvement in the inverse Froude number at the lowest height disregarding all other variables
D1	Equivalent to run B1 except with 100 classes
D2	Equivalent to run B1 except with 300 classes
D3	Similar to run B1 except with less weighting on the second height and more weighting on the wind direction at the lowest height, and with 300 classes

Table 8.1: Summary of objectives with the clustering attempts for Ireland. Each referral to a variable refers to the mean standard deviation of that variable in the clusters. Each comparison is referring to the existing method.

Run	Height: 0 m				Height: 1450 m	
	Mean standard deviation			Mean U^3	Mean stand. dev.	
	Speeds	Dirs	Fr^{-1}	(% lost)	Speeds	Dirs
Old	1·29	6·44	2·78	2·5	2.27	23·83
A1	1·03	4·94	2·13	1·3	2.16	23·48
A2	0·79	6·28	2·74	0·7	2.08	23·83
A3	1·21	4·07	3·02	1·8	2.23	23·57
A4	1·25	6·34	0·59	1·9	2.25	23·34
B1	1·23	6·18	2·68	1·7	1.90	11·04
B2	1·23	6·42	3·02	1·6	1.92	15·17
B3	1·28	4·43	3·04	2·3	2.25	23·64
C1	0·21	23·70	6·19	0·0	2.05	23·70
C2	5·57	0·78	3·60	35·6	5.27	23·87
C3	2·80	23·27	0·01	10·2	3.13	30·43
D1	1·43	7·32	2·93	1·9	2.01	12·53
D2	1·01	4·64	2·19	1·1	1.77	8·73
D3	1·26	3·18	2·57	1·8	2.08	9·48

Table 8.2: The statistics on the variables for all the Ireland KAMM runs

8.1 Example KAMM output files

Five representative clusters are chosen from Ireland run B3 to show the output from KAMM when forced with these clusters. Four of the clusters have centroids with wind speeds around 13 ms^{-1} and different directions approximately 90° apart. The remaining cluster has the same wind direction as the first cluster, but a weaker wind speed. The actual centroid wind speeds and wind directions for each of the five clusters are shown in table 8.3.

Cluster ID	Speed (ms^{-1})	Direction	Frequency (%)
137	13·3	47°	0·3
89	13·2	327°	0·9
111	13·3	128°	0·7
5	13·9	234°	1·4
81	3·4	236°	0·9

Table 8.3: The wind speed and wind direction of the centroids of 5 clusters chosen as representative examples for the KAMM output for Ireland

The result for cluster 137 is shown and described as an example in figure 2.4 on page 9. The other four KAMM output grids are shown in the following figures 8.2 - 8.5. Each figure shows the wind speed at a 50 m height above ground level. The orographic contour lines shown are every 100 m and also include the 50 m line. The colours represent the wind speed and the legend is

in ms^{-1}. The axis values are in km. The arrows represent the wind direction and their length also represents the wind speed. Each arrow represents a 5 km grid point in the KAMM domain, but only 1 in 36 arrows are shown with the exception of figure 8.5 where 1 in 16 arrows are shown. All plots show a generally reduced wind speed over the land, with a speed up over the hills and mountains.

A wind forcing from the NW, as shown in figure 8.2 does not turn very much, due to the high wind speeds and the fact that the wind is aligned with some straits, like between Ireland and Scotland in the top right corner. A small delay in the slowing of the wind speed as it hits the land can be seen around the NW coastline, which is in agreement with the fact that sites near the coast are usually more windy than inland sites.

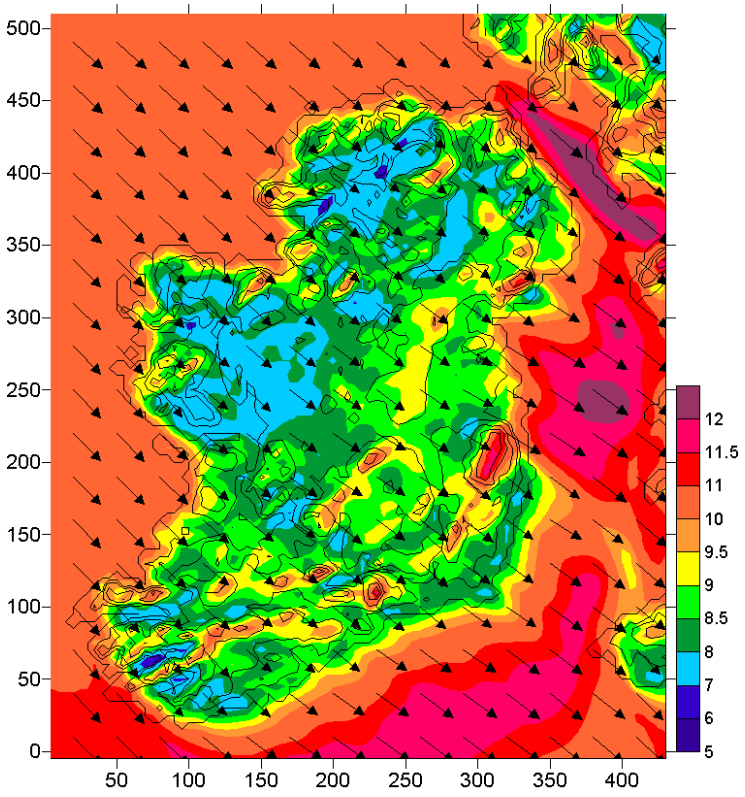

Figure 8.2: The KAMM result for cluster 89 from run B3. The wind forcing is 13.2 ms^{-1} and from 327° (NW).

The south-easterly wind forcing example in figure 8.3 shows mostly easterly winds in the KAMM output. From this direction, some turning of the wind is observed around after the Wicklow Mountains at coordinates (300,200) and around the hills of northern Ireland. Extreme wake effects are modelled here,

marked by the blue streaks of lower wind speed as the wind emerges from the parts of the british coast in the domain off the west coast of Ireland. The overall wind speed is lowest for this case of the four 13 ms^{-1} examples.

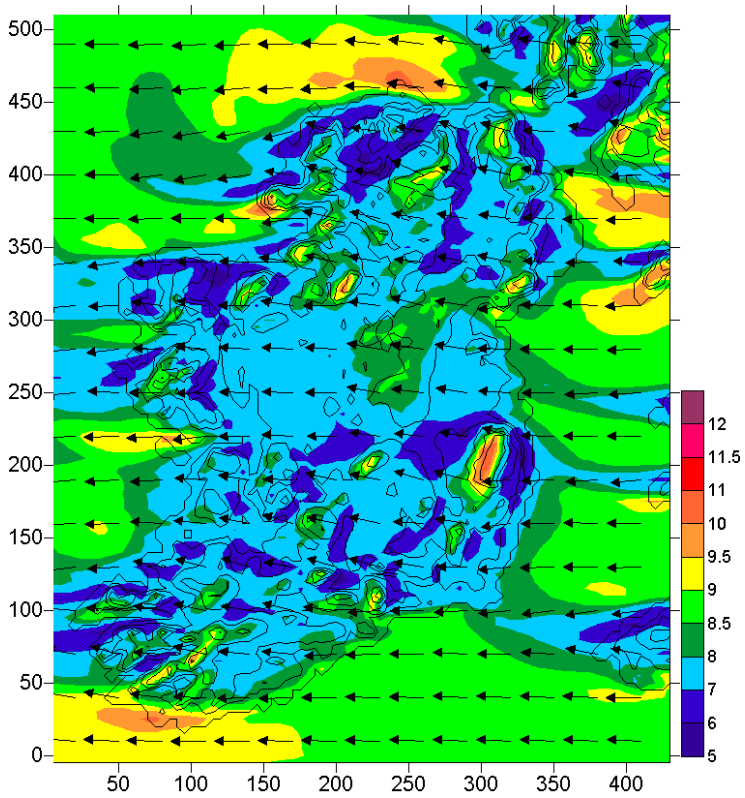

Figure 8.3: The KAMM result for cluster 111 from run B3. The wind forcing is 13.3 ms^{-1} and from 128° (SE).

The 13.9 ms^{-1} wind from the SW example is shown in figure 8.4. In this direction, the wakes of the hills and mountains on Ireland give a significant reduction in the wind speed. This is the most extreme for the Wicklow Mountains where the wind speed is 11 ms^{-1} over the top of the mountain and decreases to 6 ms^{-1} immediately after the mountain. The contour lines shown, and the detailed orographic map in figure 5.2 on page 57, indicate that the steepest incline up to the peak of the Wicklow mountains is to the NE from the peak. This explains why the wake effect is the most extreme for a wind from the SW.

The same direction as the previous case, but with a much lower wind speed is used as the final KAMM ouput example, as shown in figure 8.5. Due to the lower wind speed, a much greater wind turning effect can be seen particularly around the south-west hills of Ireland and in the north-east corner in the strait between Scotland and Ireland.

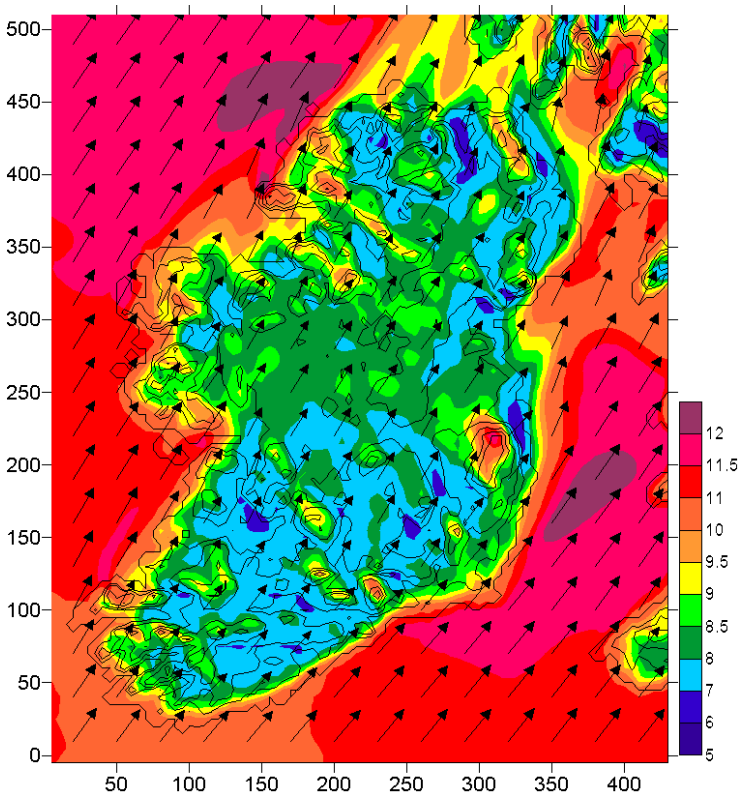

Figure 8.4: The KAMM result for cluster 5 from run B3. The wind forcing is 13.9 ms^{-1} and from 234° (SW).

8.2 Wind atlas results

Comparing wind atlases is introduced in section 3.4.2. Each of the following figures show the comparison of the numerical wind atlas results with the "true" wind atlases based on the measurements at the ten sites. The sites have been placed on the figure to match as close as possible with their locations in Ireland. Each figure compared the KAMM runs to the old method, which is always the right, red bar. The results are all first made using the old method to interpolate the simulations for constructing the Weibull distribution in WAsP. The new NSIB method is also tried on a eight of the runs and the results are included in the summary table of the results in table 8.4 on page 113. The results for batch A are compared to the old method in the following three figures 8.6 to 8.8, for the mean wind energy, mean wind speed and wind direction sector frequencies respectively. These show the percentage error of the numerical wind atlases compared to the wind atlases made from the measurements. The task is to find differences in the run performances, which can be attributed to the

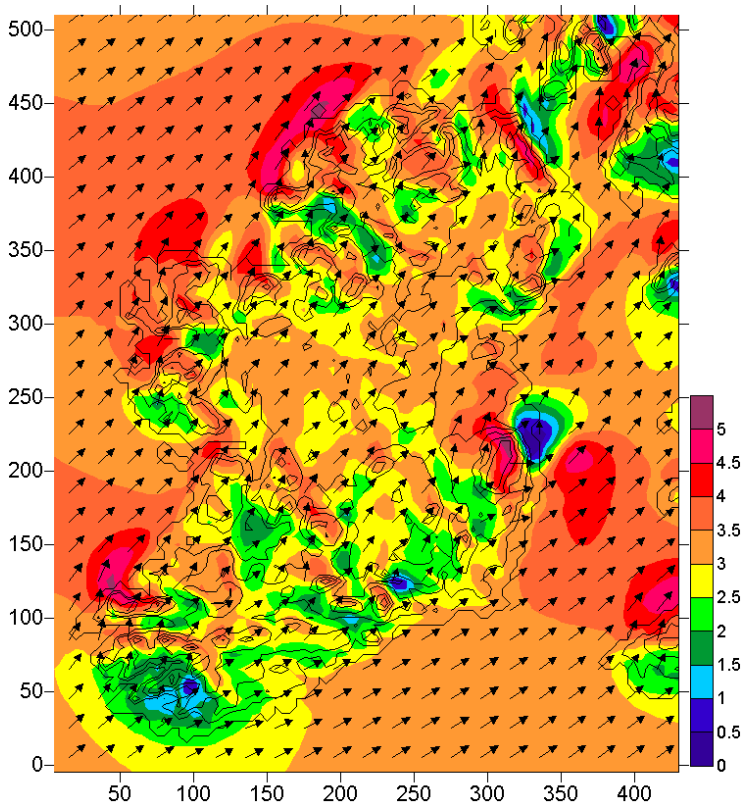

Figure 8.5: The KAMM result for cluster 81 from run D1. The wind forcing is 3.4 ms^{-1} and from 236° (SW). Note that due to the lower wind speed forcing, the colour scale is different on this map compared to the others.

classification, and not to the various other errors in the data, the measurements and the modelling. Hence, with the aim to average out the station specific errors, the mean percentage error over the ten stations is displayed next to the name of the clustering attempt in the legend. This value describes the overall performance of the method. The smaller the number, the less mean error, and the more accurate the method at predicting the quantity examined (energy, speed or direction).

In figure 8.6, the mean wind energy measured for the sites are shown in parentheses next to the name of each station. The figure shows that Shannon Airport and Dublin Airport are predicted very accurately for the wind energy with errors less than 4% for all runs in batch A. The worst stations are Mullingar and Malin Head with close to 40% underprediction and 40% overprediction respectively. Varying frequencies have been used for 32 geostrophic wind data points to capture the known gradient in the wind energy, decreasing from NW to SE. The achieved result is possibly over estimating this gradient

since all three stations in the south are underpredicted. The variation of the wind energy predictions at each station between the runs is very minor, as is the variation between the overall mean error values. This shows that the objectives for these runs did not significantly change the mean wind energy prediction. Marginally, A4, with some focus on the inverse Froude number, has the best overall result, but this is achieved by a slightly lower wind energy prediction for every station. This shows that on average, the KAMM/WAsP method with the old classification method over predicts the wind energy for the ten stations.

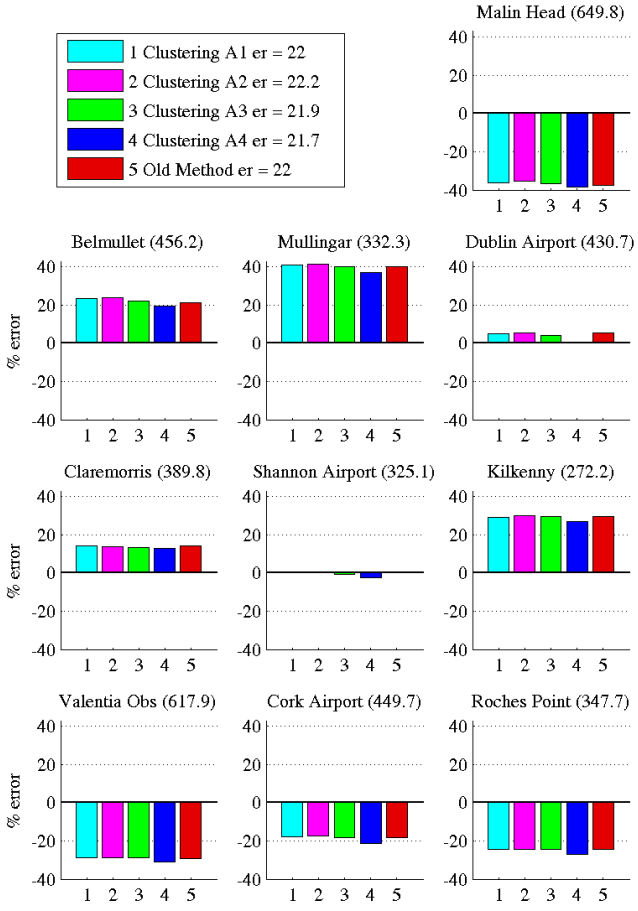

Figure 8.6: Mean wind energy comparison for clustering batch A

The batch A results for the mean wind speed are shown in figure 8.7. The absolute mean wind speed for each station is shown in parentheses next to the station name. The overall percentage error in the wind speeds are smaller than for the wind energy, which is expected since the wind energy is proportional to

the cube of the wind speed, which tends to magnify wind speed errors. However, the variation in the wind speed results between the runs is more distinct than for the wind energy results. A lower wind speed prediction for run A4 is quite visible at all stations. This lower prediction gave an improvement in the overall mean wind energy error, but the overall wind speed error is much higher than for the other runs in batch A. The wind speed predictions for runs A1 and A3 are also visibly less for every station compared to the old method, but the difference bewteen A1 and A3 them varies. Run A2 has the overall best wind speed prediction, due to an improvement in the prediction at Cork, which disagrees with the trend of lower predictions from the clustering runs. Run A3, with a focus on wind directions is the only run in batch A to give a better result for the wind energy and wind speed than the old method.

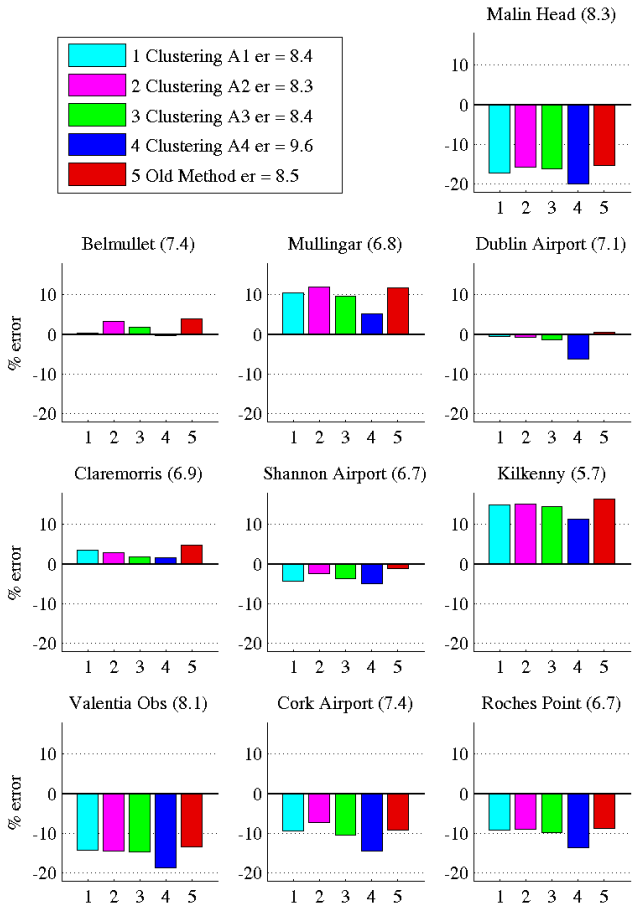

Figure 8.7: Mean wind speed comparison for clustering batch A

The wind direction sector frequency comparison for batch A is shown in figure 8.8. The error value presented here is the total absolute frequency error over the 12 sectors. The mean of this error is calculated in the same way as for the wind energy and speed. The results show that run A3, with extra focus on wind directions, gives the best direction prediction of the 4 clustering runs in batch A. However, the old method performs marginally better overall with the wind directions. Run A4 is the worst method in batch A, with greater error in the wind direction frequencies at nine out of ten stations. Since run A4 predicts wind speed and wind directions with significantly more error than the othe methods, its improved wind energy result is likely to be by chance.

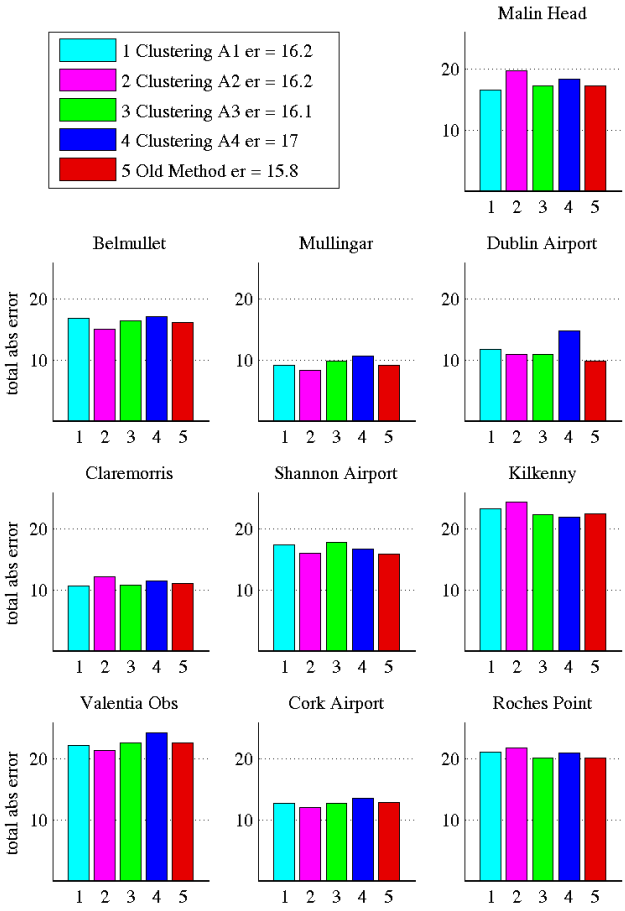

Figure 8.8: Wind direction comparison for clustering batch A. The bar graph values represent the total absolute frequency error in % over the 12 sectors.

The actual wind direction frequency roses for the New Irish Wind Atlas

are shown for each station along with the wind roses for batch A and the old method are plotted in figure 8.9. The visibly most accurately predicted station is Mullingar which is consistent with the total frequency errors shown in figure 8.8. The wind roses for each numerical wind atlas result are all very similar so that only the top line colour can be seen clearly. This suggests that the wind direction frequencies are dominated by the NCEP/NCAR data, the coarse resolution 5 km maps or the flow models, rather than the classification method used. For example, the site description for Kilkenny in section 5.1.3 explains how a nearby river system distorts the wind so that it flows preferentially from the S or the NNW. This effect is seen in the result from measurements in figure 8.9 but the numerical wind atlas results fail to capture this effect strongly enough. Thus for this station the wind direction errors from the model are likely to be dominating the total error, so that the changes offerred by the new clustering algorithm are hard to distinguish.

The deliberately bad clusters in batch C show some variations in the wind roses for some sites as shown in figure 8.10. However, considering how poor the clusters are, the resulting wind roses do not differ that much. Hence the coarse 5 km resolution model maps and the model itself is likely to be dominating the wind rose dirction shape.

The remaining equivalent plots for batch B, C and D are in appendix C.3. The overall mean error for the wind energy, wind speed and wind directions, for each KAMM run are listed in table 8.4. Most of the runs are also tested with the NSIB method for the last step when the Weibull distribution is made, and these results are also shown in the table.

The results for batches B, C and D without using the NSIB method are discussed first. In batch B, a very good result is produced by run B2 which has some focus on all four heights. B2 produces the best wind energy prediction and equal best wind speed prediction of all the KAMM runs without NSIB including the old method. The wind directions are reasonable at 16.3. Run B1, focussing on the second height only, produces a surprising result with a relatively poor wind energy prediction. The wind speed for B2 is equal best and the wind directions are probably at the boundary of an acceptable error compared with the old method. The Run B3 clusters have a maximum direction range of 22.15°, slightly less than the sector width for the old method, 22.5°. Thus, these clusters are considered the as the most similar to the old method. Consequently, the results are very similar to the old method, with a 0.1 change in the wind energy and wind speed mean errors and the same mean error in wind directions. B3 is the only run to achieve a result at least as good as the old method for wind direction. This could be due to the fact that the interpolation of simulations method (section 3.2.2) was originally designed specifically for the old method, and hence works well with the most similar clusters in run B3.

A greater variation is obtained in the results from the deliberately bad clusters in batch C. These runs show the importance of a reasonable representation of at least the wind speeds and wind directions. Runs C1 and C2 perform surprisingly well for their respective dominating variables, considering the representation of all other variables is almost random. Run C1 focusses wholly on

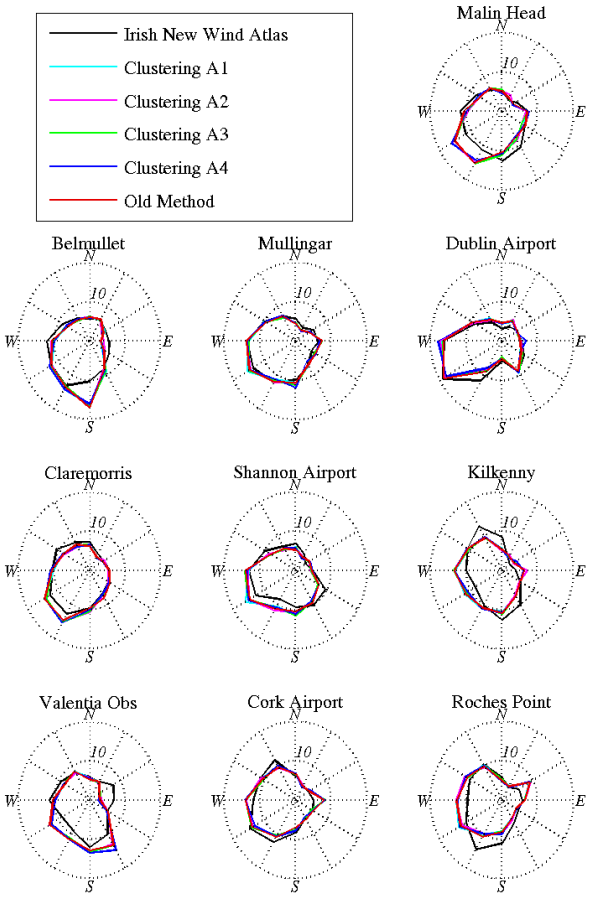

Figure 8.9: Wind direction rose comparison for clustering batch A

speeds, and the resulting wind speed predictions are better than the old method and the wind energy is only 0.1 worse. However, C1 performs poorly with the wind directions. This makes sense since the clusters for run C1 are actually divided a little on wind directions usch that the mean direction range for the clusters is around 90°. Run C2, focussing wholly on wind direction, performs only a little worse in wind direction than the runs from the other batches, and still better than run A4. However, the wind speeds and energy are both comparitively poor. Run C3 with a similar mean standard deviation on the wind directions to run C1 (see table 8.2) and with a much worse focus on wind speed, gives the worst prediction by far in wind speeds and is second worst in energy and direction, only to run D1.

Batch D tries different numbers of clusters. Run D3 is almost equivalent to

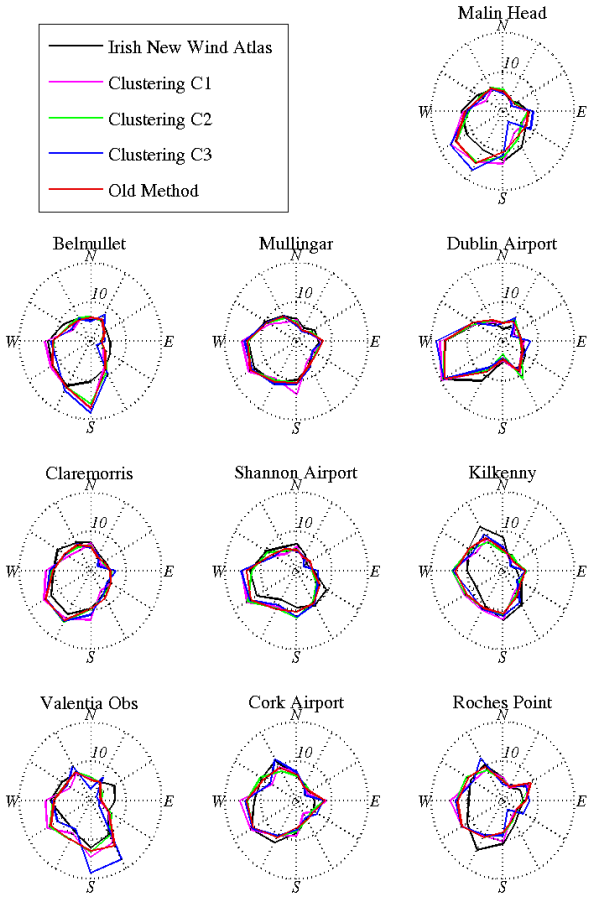

Figure 8.10: Wind direction rose comparison for clustering batch C

run A3, focussing on directions, but with 300 classes. The run does not use the same parameters as A3. Instead the weighting on directions are increased to give a very big improvement in the representation of the wind directions and only small improvements in the wind speed and inverse Froude numbers (see table 8.2 on page 102). The results show an small improvement in the wind directions, but slightly worse wind speeds and a worse wind energy prediction.

Run D1 and D2 are equivalent to run B1, except they have 100 and 300 clusters, respectively. Hence, run B1 is included again in the table so that the effect of increasing the number of clusters on the same parameters can be viewed easily. It is seen the wind directions improve with an increasing number of classes, yet the final result is still not as good as the old method. The wind energy and speeds improve between 100 and 151 clusters, but then get slightly

Run	Wind Energy		Wind Speed		Direction	
	norm	**NSIB**	**norm**	**NSIB**	**norm**	**NSIB**
Old	22	-	8·5	-	15·8	-
A1	22	22·4	8·4	8	16·2	16·3
A2	22·2	-	8·3	-	16·2	-
A3	21·9	22·3	8·4	8·1	16·1	16·1
A4	21·7	-	9·6	-	17	-
B1	24·5	24·8	8·3	8·1	16·4	16·8
B2	21·5	21·7	8·3	8·1	16·3	16·7
B3	22·1	22·6	8·6	8·7	15·8	16·1
C1	22·1	-	8·4	-	19·6	-
C2	24·5	-	9·2	-	16·7	-
C3	30·2	-	16·3	-	21·8	-
D1	44·4	21·9	9	9·2	22·6	22·6
B1	24·5	24·8	8·3	8·1	16·4	16·8
D2	24·6	24·6	8·4	8·2	16·2	16·3
D3	22·7	23·2	8·5	8·3	16	16·1

Table 8.4: The Ireland KAMM run results showing mean error on wind energy, wind speed and wind direction. The result for using the old way to interpolate the results for fitting a Weibull distribution is shown (norm) as well as the results from using the NSIB method with the clusters.

worse with the 300 cluster case. This discreprency could be due to the fact that the original cluster parameters for 151 clusters, B1, does not produce a good result. Run D3 has 300 clusters, but focusses heavily on the wind directions, thus being equivalent to run A3. Again, compared to A3, D3 produces a slightly better wind direction result but slightly worse wind speeds and wind directions. Run D1 has unusual results, with large wind energy prediction errors in only some stations (see C.30 in appendix C.3). The meaning behind this is discussed in the next section 8.3.

Applying the NSIB method to many of the results to produce the wind atlas from WAsP produces generally better wind speed predictions but worse wind energy and direction predictions. However, the 100 cluster run, D1 is an exception to this, as the NSIB method improves the wind energy dramatically to be better than the old method. The NSIB method only replaces one part of the existing simulation interpolation procedure, that is, calculating the two closest centroids. The NSIB method could be studied further and the entire procedure could be rewritten to be more tailored to clusters. This may give better results.

8.3 Mean Energy plots

The mean energy results for the wind atlases generated by KAMM and WAsP are shown in figures 8.11 and 8.12 on page 115. Note that in these plots the interpolation of simulations is not required as no Weibull distribution is made. Four examples from the KAMM runs are chosen: the old method, one of the closest clustering runs A3, the 100 cluster run with extreme errors at 3 stations and the worst bad run, C3. Very little discernable difference can be made between the old method and the A3 cluster mean energy over the entire domain. Both show a general decresae in the energy from the NW to the SE. Since the wind flows predominantly from the NW, the mean energy is highest on the NW coastline of Ireland and weakest around the Irish coastline to the E and S. This region is in the wake of Ireland's hills and mountains from the predominant wind directions, and so on average has the lowest wind energy. One difference between the two is that the A3 result seems to have smoother energy contour lines. This is particularly visible over the ocean, where the contours would be expected to be smooth due to the fact that the ocean surface is flat. The contour lines should be more smooth, the further away from land.

The D1 mean wind energy result shows a higher wind energy in general across the domain, including over the ocean. However, the wind energy prediction for D1 is only notably increased for 4 stations out of the ten: Claremorris, Cork, Mullingar and to a lesser degree, Kilkenny. Comparing the D1 and A3 plots, these 4 stations are in positions of higher energy and are all inland. All the other stations are on the coast. This suggests that the numerical wind atlas procedure is more sensitive to the clusters made for inland locations and over the sea but less sensitive along coastlines.

The final example is C3 which focusses only on the inverse Froude number, disregarding all other variables. The result shows a big reduction in the mean wind energy over the domain, which is in agreement with the 35.6% loss of energy represented in the clusters themselves (see table 8.2).

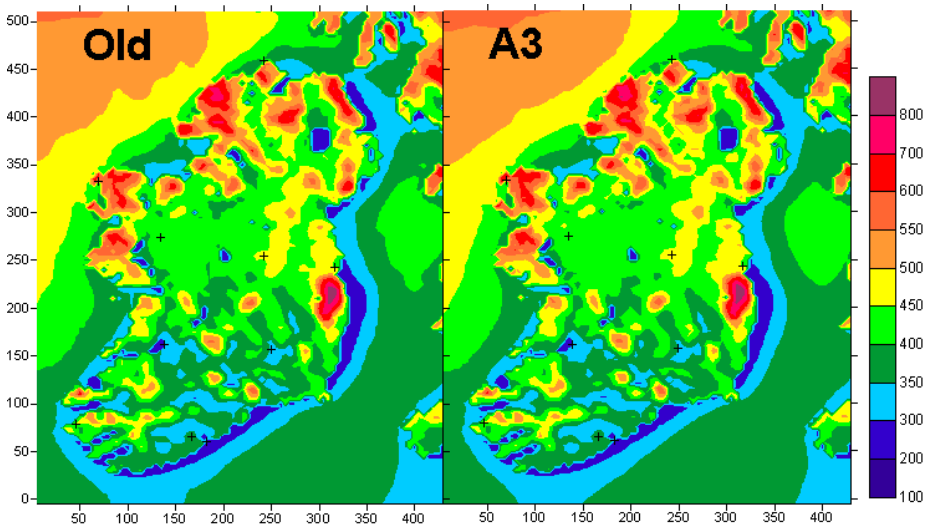

Figure 8.11: The wind atlas result mean energy across the entire KAMM domain, for the old method and clustering run A3. The ten stations locations are shown for comparison. The energy scale units are Wm^{-2}.

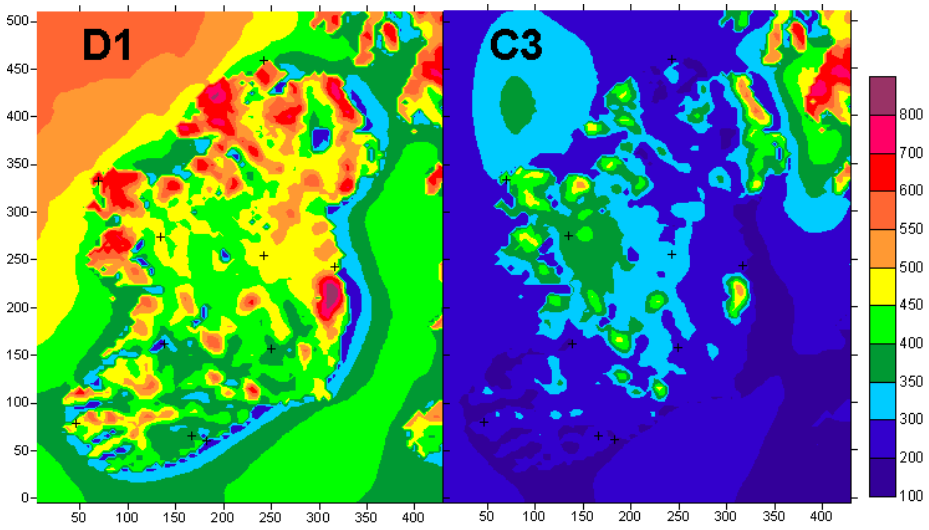

Figure 8.12: The wind atlas result mean energy across the entire KAMM domain, for clustering runs C3 and D1. The ten stations locations are shown for comparison. The energy scale units are Wm^{-2}.

Chapter 9

KAMM Simulation Results for Egypt

The existing classification method is used to divide the NCEP/NCAR geostrophic wind data into 126 classes for the Gulf of Suez in Egypt. The data is divided into 16 sectors, with between 2 and 6 wind speed bins. Finally, the classes with low wind speeds split up in 2 inverse Froude number bins, one with negative inverse Froude numbers and one with positive inverse Froude numbers. Thus the classification is performed slightly differently than with Ireland. This difference has developed at Risø over the past 5 years and has been shown to produce better results for each location. The wind in the Gulf of Suez is more sensitive to the stability of the atmosphere than for Ireland. The classes are plotted in figure 9.1. The different colours for the lower wind speeds show how the classes are split into 2 inverse Froude number bins.

Three clustering attempts are made for Egypt. All cluster attempts use the same number of classes as the existing method, 126. The objectives for the three Egypt runs are outlined in table 9.1. Table 9.2 shows the statistics on the important variables for each KAMM run made, including the existing method. Figures C.16 - C.18 in appendix C.2 that follow show the clusters made for each run on the direction and speed axes. Figures C.19 and C.20 compare the classes for the existing method with the clusters for KAMM run 1 at the second height. The clusters produced for Egypt are all relatively focussed at the lowest height. Thus, the corresponding centroid wind directions near the surface are spread yet dominated by easterlies and northerlies. At the higher heights the data in each class becomes well spread and the mean direction gives only westerly winds.

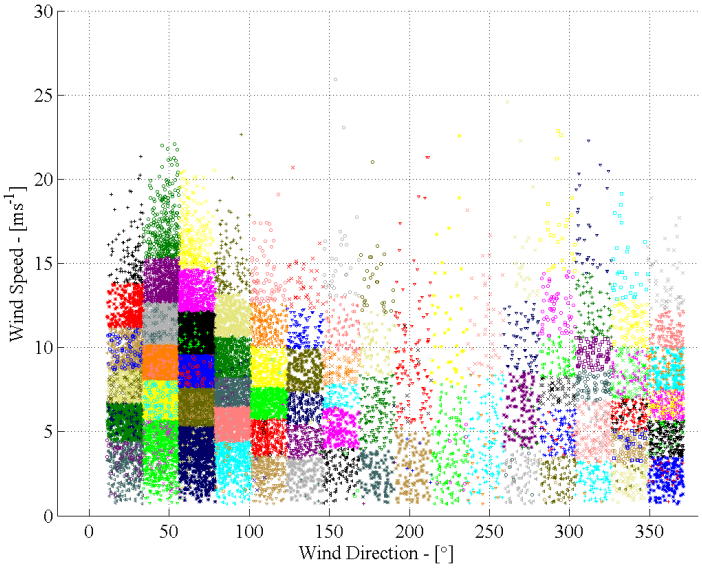

Figure 9.1: The Egypt data in 126 classes with existing method, plotted on speed and direction axes at the lowest height

Run	**Features**
1	The equivalent of run B1 for Ireland, an improvement on the wind speeds and directions at the second height while keeping the lowest height variables equivalent
2	Transforming the inverse Froude number by multiplying it with 0.1 and taking the inverse tangent before standardisation. Also to obtain more negative inverse Froude clusters, all negative inverse Froude values are multiplied by 2. These transformations allow the positive and negative values to be more separated in the resulting clusters.
3	Improvement on the inverse Froude number at the lowest height while keeping the wind speed and direction equivalent. This is an attempt to capture the negative inverse Froude numbers well without using any special conditions or transformations like in run A4.

Table 9.1: Summary of objectives with the clustering attempts for Egypt. Each referral to a variable refers to the mean standard deviation of that variable in the clusters. Each comparison is referring to the existing method.

	Height: 0 m					1500 m	
Run	**Mean stand. dev.**		**Fr^{-1}**	**Mean U^3**		**Mean st. dev.**	
	Speeds	**Dirs**	**Fr^{-1}**	**s-u (%)**	**(% lost)**	**Speeds**	**Dirs**
Old	0.88	6.29	1.78	92.9-4.6	2.4	2.40	59.44
1	0.87	6.23	1.71	92.1-0	1.8	2.31	26.20
2	0.88	6.28	1.79	93.2-4.9	1.9	2.39	59.05
3	0.89	6.28	0.77	93.6-4.8	2.4	2.39	59.31

Table 9.2: The statistics on the variables for the three Egypt KAMM runs. The Fr^{-1} s-u column refers to the percentage of the data that is represented by a stable or unstable atmosphere, with neutral atmosphere being defined as the inverse Froude number between -0.5 and 0.5.

The mean standard deviations of the wind direction at the second height are much larger than for Ireland, around 59 instead of 23. With some weight on the second height for Egypt, this can be more than halved as shown in the table for run 1, however the wind speed and the second height does not improve much.

An extra column is added to show how well the classes are split on the positve and negative Froude numbers. Three numbers are given here. The first is the percentage of the data that is represented by a postive inverse Froude number above 0.5 (stable), the second is the percentage of data that is represented by negative Froude numbers below -0.5 (unstable). By this definition, in the raw data, 92.1% are stable and 5% are unstable. Classes with inverse Froude numbers between -0.5 and 0.5 are considered to be neutral and may contain both positive and negative inverse Froude numbers in the data. The better the inverse Froude splitting, the higher these numbers in the table. Since the existing method splits about half of the lower wind speed bins into positive and negative inverse Froude values, this gives 46 classes with only negative inverse Froude numbers. The resulting percentage of the data represented by an unstable atmosphere is 4.6%, nearly the same as the percentage of the data that is unstable. The percentage of stable situations is slightly over represented at 92.9%. Using no special weighting or transformation on the inverse Froude number, the clustering produces no wholly negative inverse Froude clusters and the 0% the of data is represented by an unstable atmosphere. If the inverse tangent is applied to separate the negative and positive inverse Froude numbers more (see run 2 in table 9.1 for the details and section 3.3.3 for general description of the transformation), 21 negative clusters are achieved. The percentage of the data represnted with an unstable atmosphere is up to 4.9%, even closer to 5 than the existing method. It is desired for the clustering method and parameter settings to be simple and applicable to many locations around the world without special knowledge about the wind climate of the specific site. Hence, a third run is made without any tranformations, but with a greater weighting on the inverse Froude number. Twenty of the resulting clusters have wholly negative inverse Froude values and the unstable percentage is 4.8%, still better than the existing method. The actual parameter values used for these runs are listed in appendix D.

9.1 Wind atlas results

The wind atlas results for Egypt are compared in the same way as for Ireland. The previous numerical wind atlas results for the Gulf of Suez are published with a general underprediction in the wind energy and wind speeds [8]. Further, it was published that the wind speed results from KAMM are channeled too much down the gulf from the north, and that the wind speeds decreased too much at the southern end of the gulf. The same trends can be seen in these results in figures 9.2 - 9.6.

The results for the mean wind energy and mean wind speed for the three clustering attempts and the old method is shown in figures 9.2 and 9.3 respectively. Run 1, equivalent to run B1 for Ireland, gives a worse wind energy prediction overall and at 3 of the 4 stations. It also gives worse wind speed and wind direction predictions at almost every station. Improving the second height is expected to improve results for regions where the height of the terrain is closer to the second height than the lowest height. However the four stations used in the Gulf of Suez are all within 25 m of sea level (see table E.2). Thus the effect of focussing on the second height is an opportunity for further research.

Run 2, with the special stability separation, gives the best wind energy result, with a small improvement at 3 stations and a very big improvement at Hurghada. It also gives better wind speed result at the 2 south stations with an overall better wind speed result. However, this run includes special knowledge about the site and is difficult to tune the parameters to achieve the desired clusters with the inverse tan transformation of the inverse Froude number. The NNW wind direction frequency for Egypt is overpredicted from the KAMM model, making the error in wind directions very high. This model error could be dominating the direction results. At the cost of a slightly worse wind direction prediction, the best wind speed prediction is obtained in run 3. It is equivalent to run A4 for Ireland, which gives the best wind energy result for Ireland. A good wind energy result is obtained Egypt as well for run 3, with improvements at 3 of the 4 stations and no change at Zafarana.

The wind directions for the Gulf of Suez are poorly modelled by KAMM as seen by the 50 - 110% total frequency errors in figure 9.4. All methods performed very similarly compared to the magnitude of the error and no conclusions can be made. The wind roses in figure 9.5 show that wind directions are modelled to be almost unidirectional at the stations as was previously published in earlier Gulf of Suez KAMM simulation results in [8]. The numerical wind atlases give over 60% frequency at due north sector for Zafarana at the N30°W sector for the other three stations. Thus, these sectors can be compared for the mean wind speed in each sector, but the other sectors are not allocated enough data to give the mean wind speed a reliable result. The mean wind speeds in each sector are shown in figure 9.6. In the Gulf of El-Zayt and Hurghada roses, the wind speed is underpredicted in the N30°W sector, this giving the overall 20% underpredictions in figure 9.3. The north sector for Zafarana is only a little underpredicted and thus is also the most frequently occurring sector from the measurements. This is reflected in the smaller underprediction of the wind

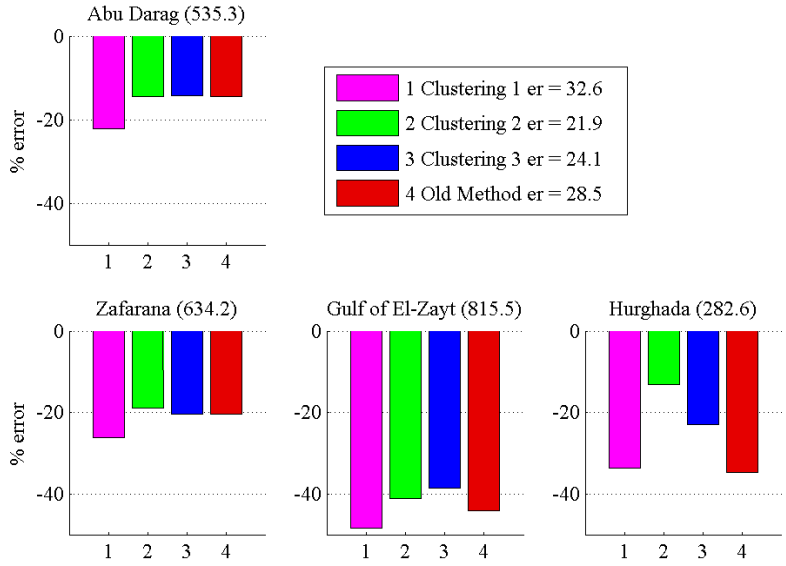

Figure 9.2: Mean wind energy comparison for the four stations in the Gulf of Suez

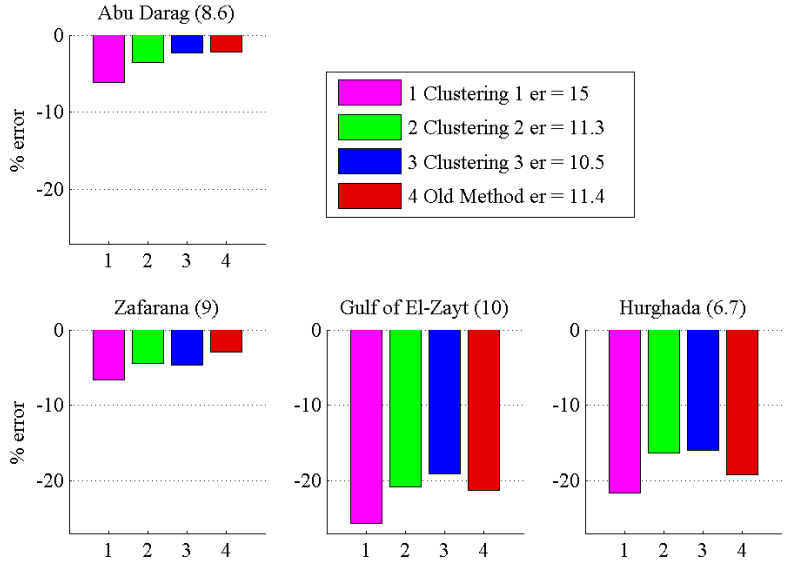

Figure 9.3: Mean wind speed comparison for the four stations in the Gulf of Suez

speed in figure 9.3.

As with Ireland, table 9.3 summarises the mean error values for each run

with Egypt.

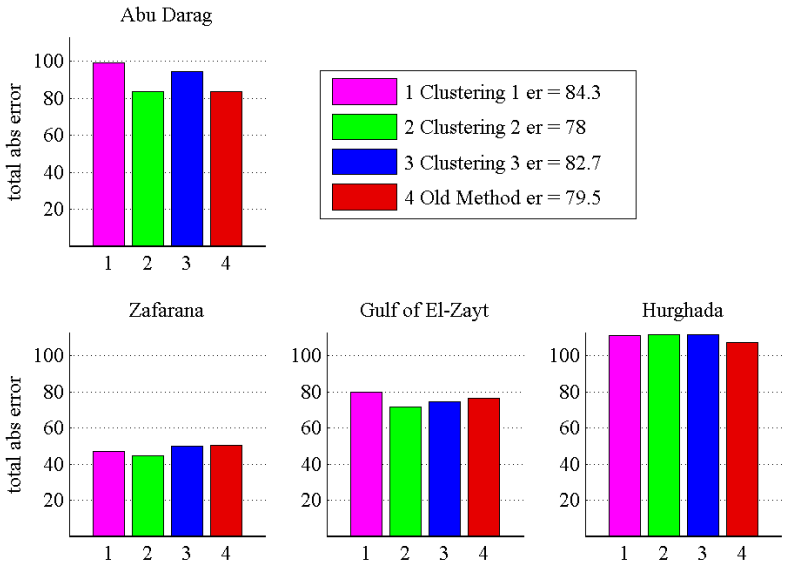

Figure 9.4: Wind direction comparison for the four stations in the Gulf of Suez

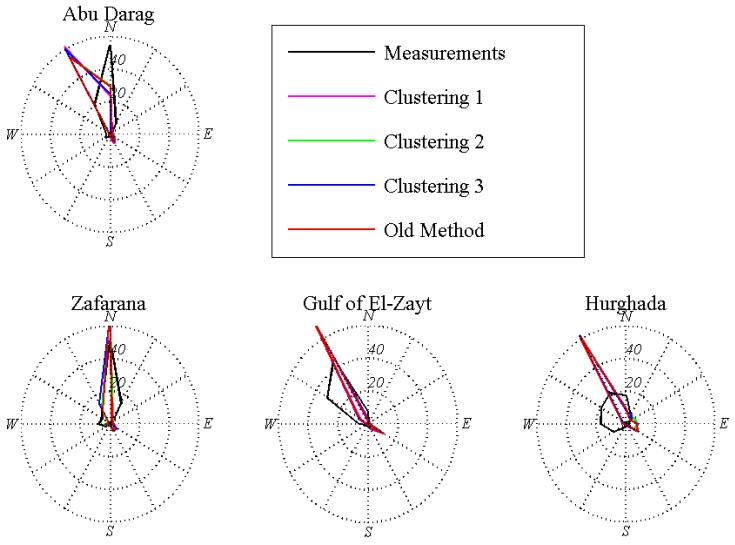

Figure 9.5: Wind direction frequency rose comparison for the four stations in the Gulf of Suez

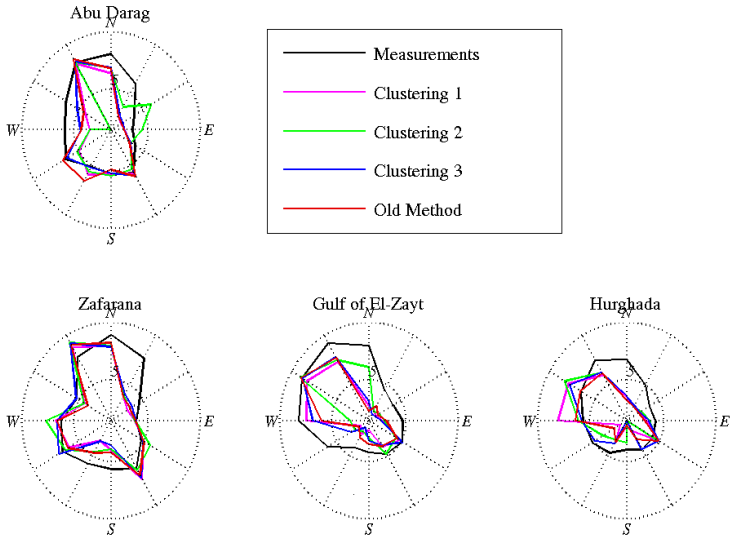

Figure 9.6: Sector mean wind speed rose comparison for the four stations in the Gulf of Suez

Run	Wind Energy	Wind Speed	Direction
Old	28·5	11·4	79·5
1	32·6	15	84·3
2	21·9	11·3	78
3	24·1	10·5	82·7

Table 9.3: The Egypt KAMM run results showing mean error on wind energy, wind speed and wind direction.

9.2 Results comparison between Ireland and Egypt

As with Ireland, clustering is able to achieve an at least equivalent result to the old method. Focussing on the inverse Froude number improves the wind energy predictions for both countries. However, this method only improves the wind speed predictions for Egypt, while the wind speeds in Ireland are worse. For both countries the wind rose plots look very similar between the methods and the old method mean error is hard to beat. One reason for this could be the use of a non-optimal simulation interpolation method, which is designed for classes made from the old method. The new NSIB method devised to test this theory gives improvements in the wind speed but wind directions and wind energy errors get slightly bigger. The NSIB method only replaces the part of the old interpolation method where the closest centroids are found. Further exploration into a fully cluster oriented simulation interpolation method is required.

Adding weight on the second height gives worse results in wind energy for

both Ireland and Egypt. However, in Ireland the energy is higher for every station and in Egypt the energy is lower for every station. The opposite effect to the inverse Froude number focus case is observed where the wind speeds are predicted better for Ireland and worse for Egypt.

9.3 Suggested parameters

Based on the results obtained for both countries, a set of parameter values is devised that may produce improved, or at least equivalent wind energy, speed and direction predictions to the old method, for both sites simulataneously. The general idea is to focus on wind directions and inverse Froude number at the lowest height, while keeping the mean standard deviation of the speed reasonable. A reasonable standard deviation for the wind speed depends on the mean wind speed of the geostrophic wind data. The mean wind speed for Ireland is 11.8 ms^{-1} for Egypt is 8.01 ms^{-1}. The mean standard deviations obtained by the old method classes for these data sets are 1.29 and 0.88 respectively, which in both cases is the mean wind speed divided by 9.1.

The inverse Froude number describing thermal stability proves more important for Egypt than for Ireland. However a reasonable focus on the inverse Froude number improves the wind energy predictions for both countries. An indication of how much the inverse Froude number will affect mesoscale modelling in a region, could be obtained by analysing the percentage of values from the geostrophic wind data that are below -0.5. The greater this percentage, the more weighting on the inverse Froude number could be required. However, without this pre-analysis, the following general set of parameters may produce good results, at least for Egypt and Ireland. Further sites need to be tested to make more general conclusions.

Parameter	Guideline
nCL	200. To allow for simultaneous focus on the wind directions and inverse Froude number, the number of classes required is probably around 200.
R	Around 0.4. This has been shown in the clustering attempts to be a value where the wind directions receive some more weighting than the speed compared to the old method. The overall aim is clusters with a mean standard deviation of the speeds to be equal to the mean geotrophic wind speed divided by 9.1. The mean standard deviation for the wind directions can be quite small, probably around 4 or 5. If the old simulation interpolation method is to be used, a maximum wind direction range of 22.5 could be used which will increase the number of clusters set by *nCL*.
sd_invFr	Start with 3. If the standard deviation of the wind speeds is too high as per the value calculated above, increase this number to 4 and try again, or visa versa. Stop when the standard deviation of the wind speeds is the same or up to 5% less than desired value.
wgt	(10, 0.01, 0.01, 0.01). For the Ireland and Egypt cases in this report the met stations for comparison are all closer to the lowest height. Hence *wgt* this should be set to a full weighting on the lowest height
wgtFr	(10, 0.01, 0.01) for the same reasons as above.
RF	Start with 60.
sd_invFr_factor	1.2.

Table 9.4: A guideline common to Ireland and Egypt, to set the parameters to make a set of clusters for Ireland and Egypt to produce good results. Note that this guideline should be used in conjuction with the suggested procedure in section 7.2.

Chapter 10

Conclusions

A clustering method is established for generating wind classes as part of numerical wind atlas construction. It is more automated than the existing classification method since a step-by-step procedure is published in this report for how to tune the parameters. The clustering method produces a desired number of clusters which the existing method could not do. It also has the ability to include other heights in the geostrophic wind data for generating the clusters.

Statistically, the clustering method produces a more optimal representation of the data than the existing method. The results from the KAMM/WAsP method show for the first time that the clustering method is able to produce results at least equivalent to the existing method results.

A set of classes can be evaluated before performing KAMM simulations by the weighted mean standard deviations of the speeds, directions and inverse Froude number from the classes. The smaller each of the numbers, the more focus the classification has on the corresponding quantity. For regions where the percentage of negative inverse Froude numbers in the data is high (around 5%), the percentage of the data represented by a negative Froude number in the centroids is also used for class evaluation. This percentage should be similar to the percentage for the raw data. The clustering method provides the ability to test the importance of the various quantities in the clustering. The results for Ireland show that the wind directions are more important than the wind speed compared to the importance placed on them by the existing method. Improving the focus on the wind direction in the clusters produced improved wind speed and direction predictions. The inverse Froude number is shown to be important for classification in Egypt, which was known from previous numerical wind atlas constructions for Egypt.

The clustering method is used with different parameter settings to produce different set of clusters for running KAMM simulations and creating numerical wind atlases. One of these runs marginally improves the wind energy and wind speed predictions for Ireland, while predicting the wind direction frequencies only slightly worse than the existing method. This run represents the wind directions better compared to the existing method. The wind energy prediction

change is also minimal for other cluster runs where the wind speed or the inverse Froude number are represented better than the existing method. Improving the representation on the second height produces a worse wind energy than most other runs, but the wind speed prediction is good. The second height is 1450 m and the highest elevation of the measurement stations used on Ireland is only 162 m. Hence, the effect of including the second height in the clustering is likely to be a non-optimal representation for Ireland. Including the second height could be tested on a different site using stations with higher elevations. Alternatively, a lower level for the second height could be used in the geostrophic wind data.

Three different numbers of clusters, 100, 151 and 300 are tested using the same parameters, focussing on the second height as well as the lowest height. The results show a general improvement in the wind direction predictions, and a distinct improvement in wind speeds and wind directions between 100 and 151 clusters. However the wind speed and energy get slightly worse for 300 clusters, which could be due to the method parameters not being optimal for this run. This could be tested further using different parameters.

For Egypt, one set of clusters produces more accurate wind energy, speed and direction results than the existing method. However this method involves a complicated transformation of the inverse Froude number. The wind directions for Egypt are modelled to be channeled in the Gulf of Suez as northerly winds too often. Thus this modelling error is likely to be dominating the wind direction errors. At the cost of slightly worse wind direction predictions, improved wind energy and wind speed results are obtained for a simpler clustering algorithm focussing more on the inverse Froude number.

The cluster method takes the first step towards a fully automated classification procedure that can be applied to any site around the world. A general guideline is devised with the aim of further automation where the same clustering procedure can be applied to Ireland or Egypt to produce better or at least equivalent results than the existing method. An opportunity for further work is to test this method to see if it is possible. The next step towards the automation goal is to try the clustering method on other sites around the world. Another possibility for more automation is to design a feed-back loop into the clustering algorithm so that it automatically finds the parameter values required to obtain desired standard deviations of the variables.

A modification to the simulation interpolation method is also tried to improve the Weibull distribution fitted to the results from simulations. This method, NSIB, could be refined to replace the existing interpolation and be more tailored to clusters.

Bibliography

[1] G. Adrian, N. Dotzek, and H. Frank. Influence of Thermally Induced Wind Systems on the Wind Climate of the Baltic Sea Analysed by Numerical Simulations. *1996 European Union Wind Energy Conference, Göteberg*, pages 608–610, 1996.

[2] G. Adrian and F. Fielder. Simulation of Unstationary Wind and Temperature Fields over Complex Terrain and Comparison with Observations. *Beiträge zur Physik der Atmosphäre*, February 1991:27–48, 1991.

[3] M. R. Anderberg. *Cluster Analysis for applications*. Academic Press, 1973.

[4] M. M. Astrahan. *Speech Analysis by Clustering, or the Hyperphoneme Method*. Stanford Artificial Initelligence Proj. Mem. AIM-124, AD 709067. Stanford Univ., Stanford, California, 1970.

[5] G. H. Ball and D. J. Hall. *PROMENADE - An On-Line Pattern Recognition System*. Rep. No. RADC-TR-67-310, AD 822174. Stanford Res. Inst., Menlo Park, California, 1967.

[6] E. Batschelet. *Circular Statistics in Biology*. Academic Press, 1981.

[7] E. W. Forgy. Cluster Analysis of Multivariate Data: Efficiency Versus Interpretability of Classifications. 1965.

[8] H. Frank. Wind Simulations for the Gulf of Suez with KAMM. 2000.

[9] H. P. Frank and L. Landberg. Modelling the Wind Climate of Ireland. *Boundary-Layer Meteorology*, 85:359–378, 1997.

[10] H. P. Frank, E. L. Petersen, R. Hyvönen, and B. Tammelin. Calculations on the Wind Climate in Northern Finland: the Importance of Inversions and Roughness Variations during the Seasons. *Wind Energy*, 1999.

[11] H. P. Frank, O. Rathmann, N. G. Mortensen, and L. Landberg. The Numerical Wind Atlas - the KAMM/WAsP Method. Technical Report Risø-R-1252(EN), Risø National Laboratory, Wind Energy and Atmospheric Physics Department, Roskilde, Denmark, June 2001.

[12] F. Frey-Buness. Ein statistisch-dynamisches Verfahren zur Regionilsierung globaler Klimasimulationen. (English Title: A Statistical-dynamical for the Regionalization of Global Climate Simulations). *Dissertation Universität München*, page 149 pp., 1993.

[13] F. Frey-Buness, D. Heimann, and R. Sausen. A Statistical-Dynamical Downscaling Procedure for Global Climate Simulations. *Theoretical and Applied Climatology*, 50:117–131, 1995.

[14] SAS Institute Inc. *SAS/STAT User's Guide, Version 6*. Cary, NC, USA, 4 edition, 1989.

[15] B. H. Jørgensen. Low-dimensional modeling and dynamics of the flow in a lid driven cavity with a rotating rod. Technical report, Technical University of Denmark, Department of Energy Engineering, Fluid Mechanics Section, 2000.

[16] B. H. Jørgensen. Personal Communication, 2005.

[17] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, and L. Gandin. The NCEP/NCAR 40-Year Reanalysis Project. *Bulletin of the American Meteorological Society*, 77(3):437–471, March 1996.

[18] L. Landberg and R. Watson. The New Irish Wind Resource Atlas. *Proc. EWEA'94, Thessaloniki, Greece*, I:233–237, 1994.

[19] J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. *Proc. Symp. Math. Statist. and Probability, 5th, Berkeley*, 1:281–297, 1967.

[20] H. Mengelkamp, H. Kapitza, and U. Pflüger. Statistical-dynamical downscaling of wind climatologies. *Journal of Wind Engineering and Industrial Aerodynamics*, 67&68:449–457, 1997.

[21] G. W. Milligan. An examination of the effect of six types of error peturbation on fifteen clustering algorithms. *Psychometrika*, 45:325–342, 1980.

[22] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

[23] R. Mojena. Hierarchical grouping methods and stopping rules: An evaluation. *The Computer Journal*, 20(4):359–363, 1977.

[24] N. G. Mortensen, U. S. Said, H. P. Frank, L. Georgy, C. B. Hasager, M. Akmal, J. H. Hansen, and A. A. Salam. Wind Atlas for the Gulf of Suez. Technical report, New and Renewable Energy Authority, Cairo, Egypt and Risø National Laboratory, Roskilde, Denmark, April 2003.

[25] N. G. Mortensen, U. S. Said, H. P. Frank, L. Georgy, C. B. Hasager, M. Akmal, J. H. Hansen, and A. A. Salam. Meso- and Micro-Scale Flow Modelling in the Gulf of Suez, Arab Republic of Egypt. 2005.

[26] National Oceanic, Atmospheric Administration (NOAA), and Cooperative Institute for Research in Environmental Sciences. NCEP/NCAR Reanalysis data home page. http://www.cdc.noaa.gov/.

[27] J. M. Parks. Classification of Mixed Mode Data by R-Mode Factor Analysis and Q-mode Cluster Analysis on Distance Functions. *In Numerical Taxonomy*, pages 261–223, 1969.

[28] M. V. Ryde. Mere regnekraft med Mary. *RIposten, Personaleblad for Forskningscenter Risø(in Danish)*, 5/6:26–27, 2004.

[29] J. A. Sonquist and J. N. Morgan. *The Detection of Interaction Effects.* Survey Res. Center, Inst. for Social Res., Univ. of Michigan, Ann Arbor, 1964.

[30] H. Späth. *Cluster Analysis Algorithms: For data reduction and classification of objects.* Ellis Horwood Limited, 1980.

[31] U.S. Geological Survey and NASA. Global Land Cover Classification, Land Process Distributed Active Archive Center. http://edcdaac.usgs.gov/glcc/glcc.asp.

[32] U.S. Geological Survey and NASA. The USGS GTOPO30 data set, EROS data centre, Land Process Distributed Active Archive Center, Sioux Falls, South Dakota. http://edcdaac.usgs.gov/gtopo30/gtopo30.asp.

[33] I. Troen and E. L. Petersen. *European Wind Atlas.* Risø National Laboratory, 1989.

[34] Jr. J. H. Ward. Hierarchical Grouping to Optimise an Objective Function. *J. Amer. Statist. Assoc.*, 58(301):236–244, 1969.

[35] X. Wu. Efficient Statistical Computations for Optimal Color Quantization. *Graphics Gems II*, pages 126–133, 1991.

Appendix A

Formulae

A.1 The variance of a set of directions

There are two accepted methods to find the variance of a set of directions, the linear variance and the angular variance.

A.1.1 Linear Variance

The linear variance is made by first splitting the directions (as explained above) and then adding 360 degrees to all the directions with values less than the split direction. The variance can then be calculated directly on the list of directions with the standard linear formula.

$$Linear\ Variance = \bar{x^2} - \bar{x}^2 \tag{A.1}$$

where
x represents the directions,
$\bar{x}$ is the mean of x, and $\bar{x^2}$ is the mean of x^2.

A.1.2 Angular standard deviation

The angular variance is defined in [6] as one minus the length of the vector mean, multiplied by two. Thus, this displays the desired property that the more evenly spread the directions are around the circle, the smaller the vector mean becomes, and the larger the variance. Also, it can be applied to any set of directions without requiring ay manipulation of the direction values modulus with 360 degrees. The disadvantage is that it is not standard and cannot be implemented in the SAS program. The angular standard deviation is the square root of the variance, as follows.

$$stdDev = \sqrt{2(1-r)} \tag{A.2}$$

where
$r = \sqrt{\bar{u}^2 + \bar{v}^2}$,
$\bar{u}$ = mean wind speed in u direction, and
$\bar{v}$ = mean wind speed in v direction.

A.2 Circular correlation

The correlation between two sets of directions is not as straightforward as in the linear case. Circular correlation is described in [6] as a science still under development, and defines a positive and negative correlation as:

$$r = \frac{1}{n}\sqrt{(\sum cos\delta_i)^2 + (\sum sin\delta_i)^2} \tag{A.3}$$

where
$\delta_i = \psi_i - \phi_i$ for positive correlation giving r_+, and
$\delta_i = \psi_i + \phi_i$ for negative correlation giving r_-, and
ψ_i, ϕ_i are a pair of angles for comparison.

Due to the strange fact that positive and negative correlation can exist at the same time with circular samples, the actual correlation coefficient is defined as:

$$r = max(r_+, r_-) \tag{A.4}$$

A.2.1 Formal definition for the Froude number

The formal definition for the Froude number is based on the velocity of a fluid over the square root of the acceleration due to gravity multiplied by the height, h.

$$Fr = \frac{V}{\sqrt{gh}} \tag{A.5}$$

The Froude number squared, multplied by the air density on the top and bottom of the equation, is shown in equation A.6. This can now be described as inertia divided by gravity.

$$Fr^2 = \frac{\rho V^2}{\rho g h} \tag{A.6}$$

The friction factor is defined as follows in equation A.7, along with the equivalent expression if the gas is assumed to be hydrostatic and ideal. The final expression obtained is similar to the inverse of the expression in equation A.6. Hence the "inverse Froude number" used in this report is actually the friction factor, and similar to the inverse Froude number squared.

$$f = \frac{gh}{V^2}\frac{\Delta\rho}{\rho} \simeq \frac{\rho h}{V^2}\frac{\Delta\theta}{\theta} \tag{A.7}$$

A.3 Conversion of Weibull parameters

The formulae to convert the Weibull parameters, A and k, to mean wind speed and wind energy are as follows.

$$\text{mean wind speed} = A\Gamma(1+\frac{1}{k}) \text{mean wind energy} \quad = \frac{\rho A^3}{2}\Gamma(1+\frac{3}{k}) \tag{A.8}$$

where
Γ is the gamma function, and
$\rho = 1.225$ is the air density for a temperature of 15 °C and standard pressure of 1013 mb.

Appendix B

Proof of weighted means

The mean of a set of values is the same as the weighted mean of groups of the same set. For classification, if the centroid of each class is calculated and summed up multiplying each with its class frequency, the centroid of the entire data set is obtained. This is proved in the following equations.

Let $x_{j,k}$ be the value on a variable of the jth data point in the kth cluster of h clusters. The mean of this variable in the entire data set of m data points is then:

$$\text{Mean} = \frac{\sum_{k=1}^{h} \sum_{j=1}^{m_k} x_k}{m} \tag{B.1}$$

The frequency of the kth cluster is simply m_k/m. Hence the weighted sum of the individual means of each cluster is:

$$\text{Weighted Mean} = \sum_{k=1}^{h} \left[\frac{\sum_{j=1}^{m_k} x_k}{m_k} \times \frac{m_k}{m} \right] \qquad = \frac{\sum_{k=1}^{h} \sum_{j=1}^{m_k} x_k}{m} \tag{B.2}$$

since the m_k cancels for each cluster. Hence, the weighted sum of the means of each cluster is the same as the mean of the whole data set.

Appendix C

KAMM run figures

C.1 The clusters for the Ireland runs

The figures showing the clusters made on the speed and direction axes for the Ireland KAMM runs are shown in the following figures. Each are displayed at the lowest height unless otherwise stated.

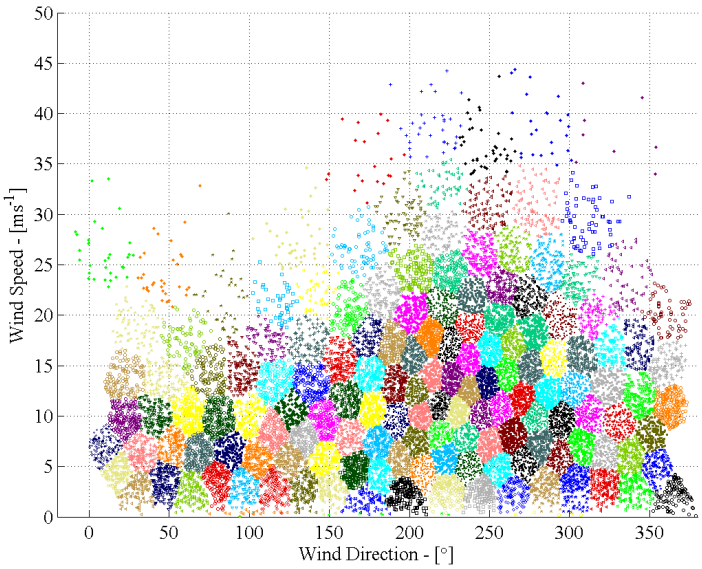

Figure C.1: The Ireland data in 151 clusters for KAMM run A1

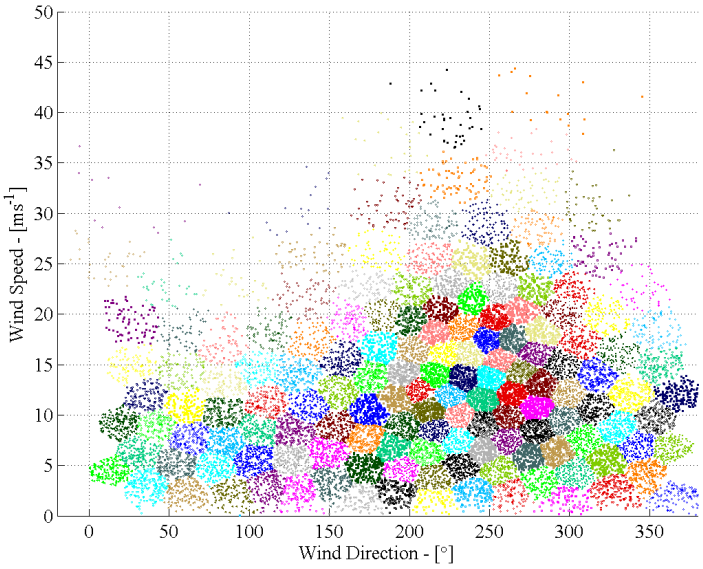

Figure C.2: The Ireland data in 151 clusters for KAMM run A2

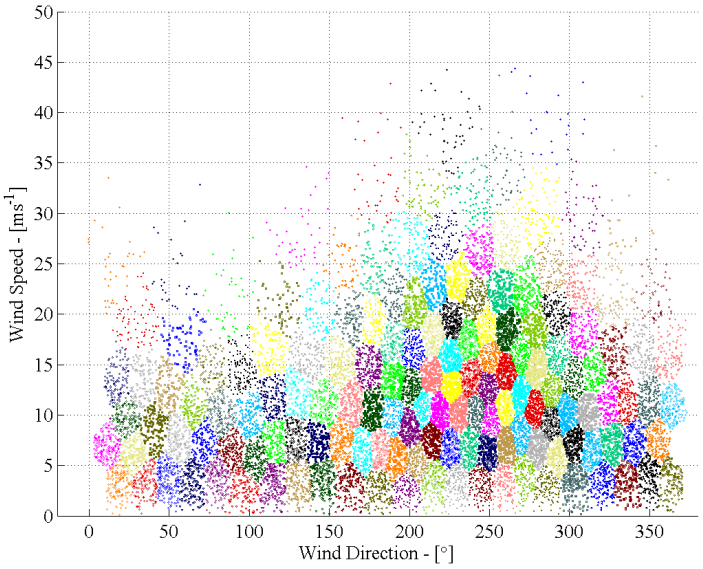

Figure C.3: The Ireland data in 151 clusters for KAMM run A3

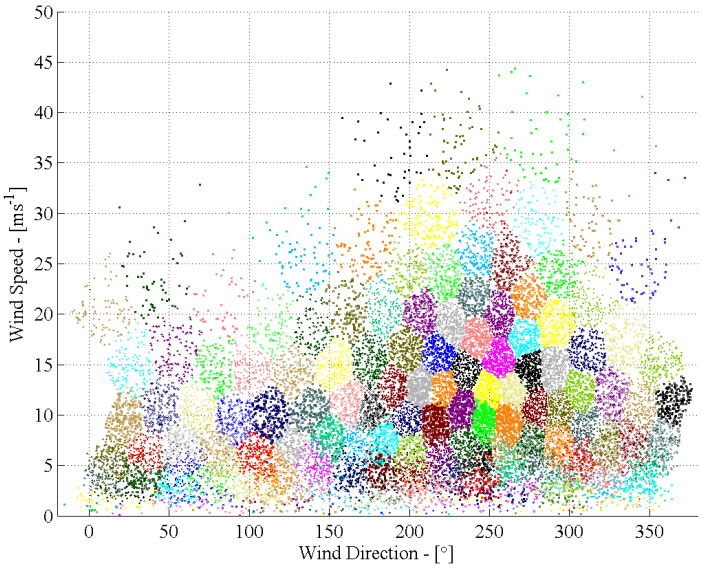

Figure C.4: The Ireland data in 151 clusters for KAMM run A4

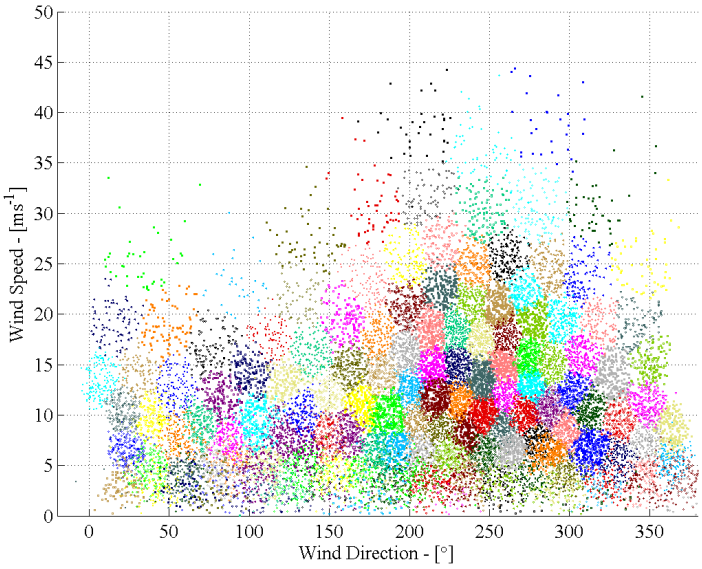

Figure C.5: The Ireland data in 151 clusters for KAMM run B1

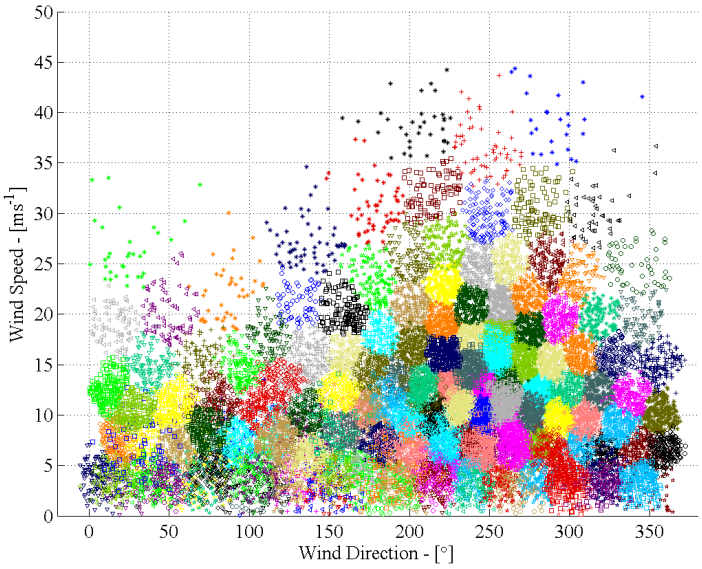

Figure C.6: The Ireland data in 151 clusters for KAMM run B2

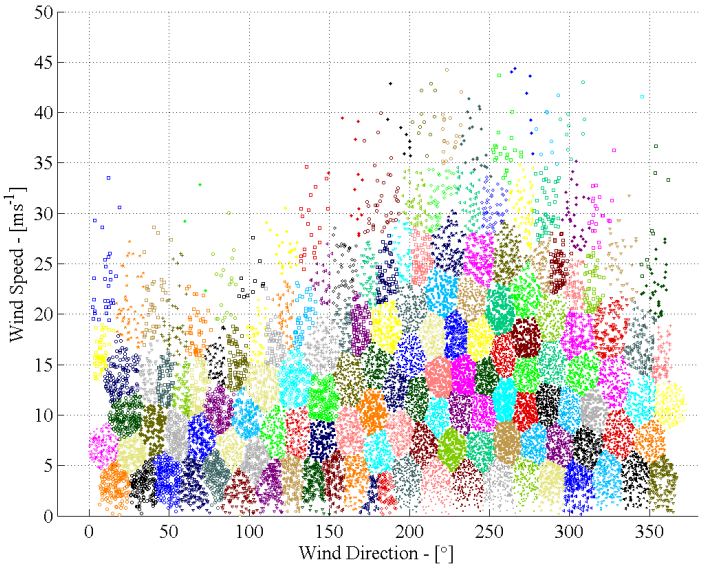

Figure C.7: The Ireland data in 151 clusters for KAMM run B3

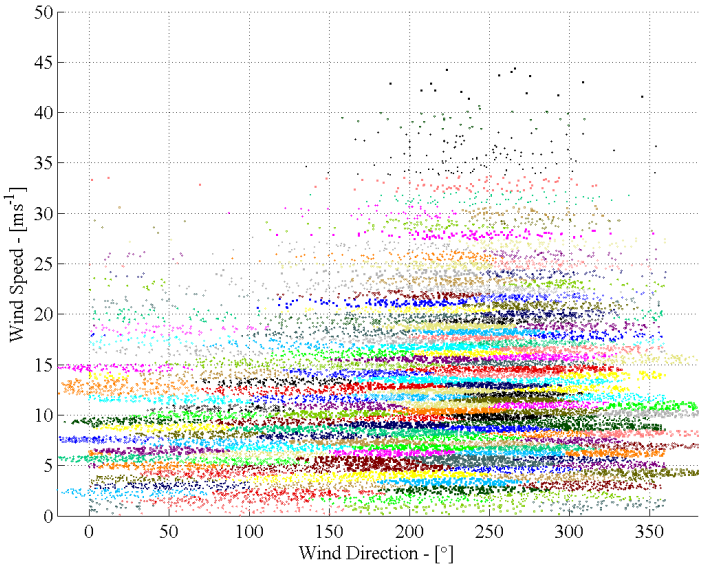

Figure C.8: The Ireland data in 151 clusters for KAMM run C1

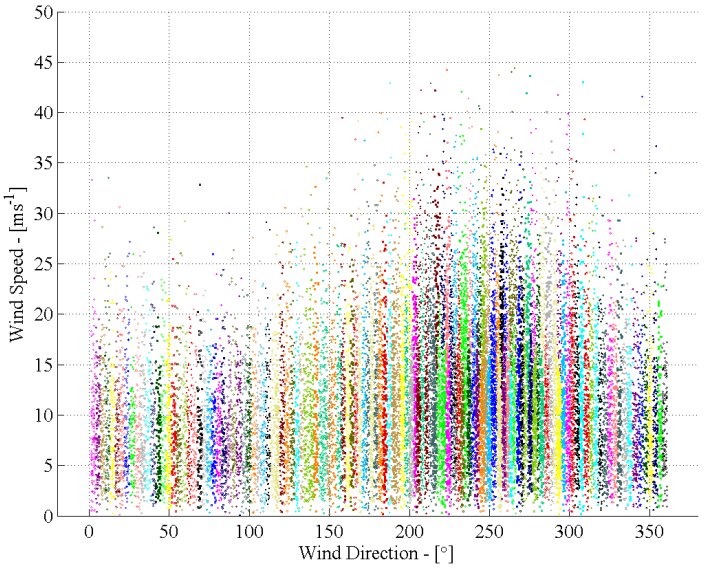

Figure C.9: The Ireland data in 151 clusters for KAMM run C2

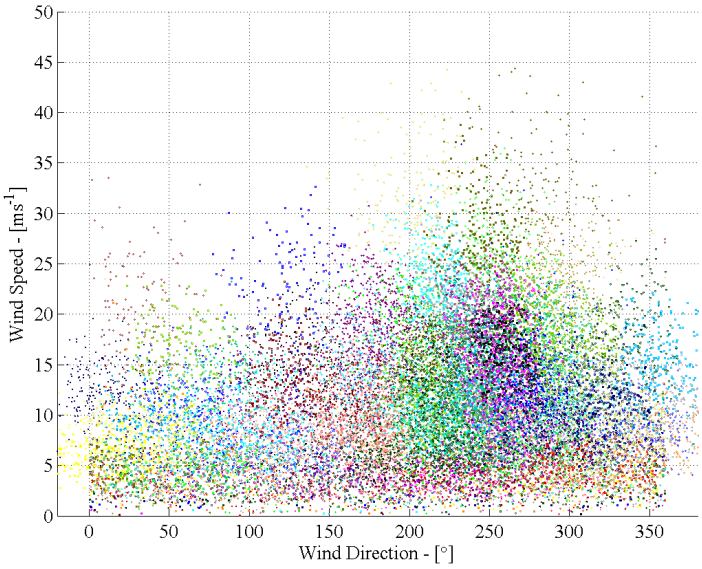

Figure C.10: The Ireland data in 151 clusters for KAMM run C3

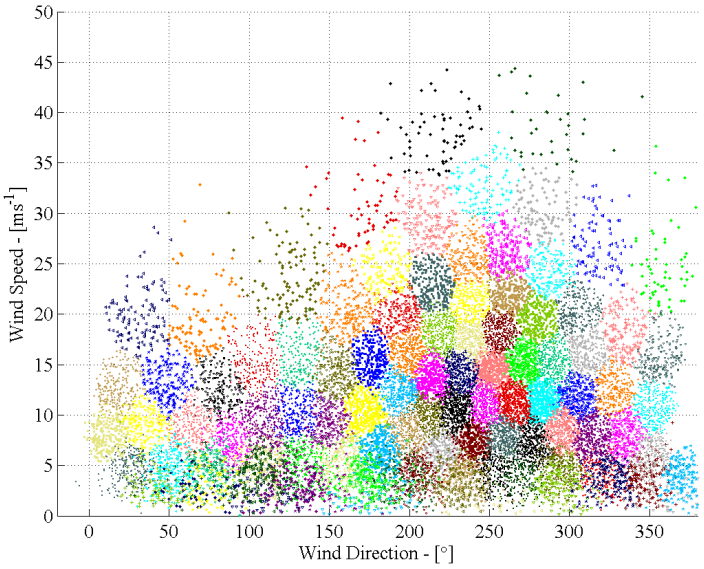

Figure C.11: The Ireland data in 100 clusters for KAMM run D1

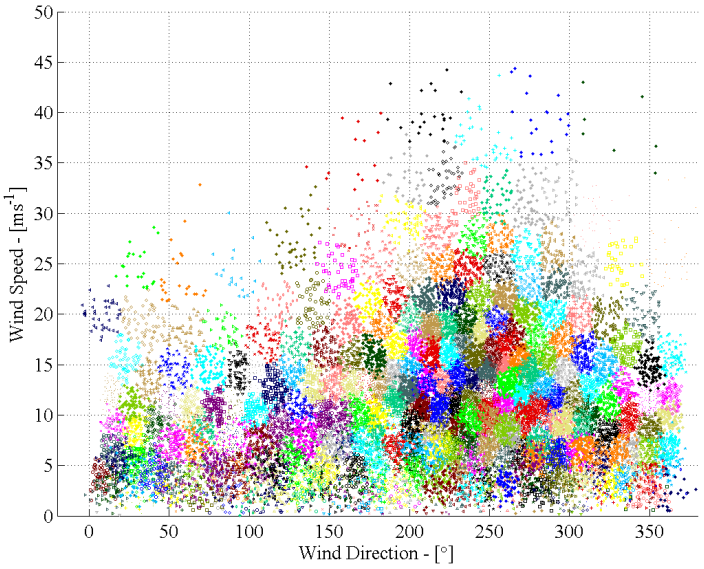

Figure C.12: The Ireland data in 300 clusters for KAMM run D2

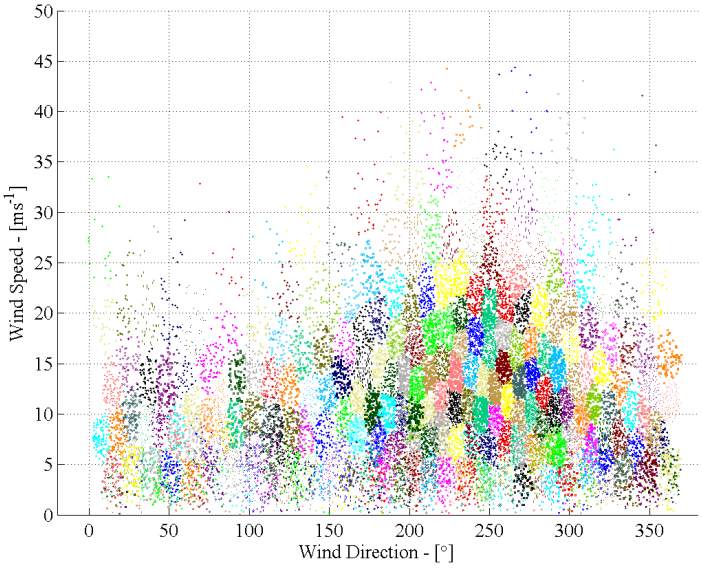

Figure C.13: The Ireland data in 300 clusters for KAMM run D3

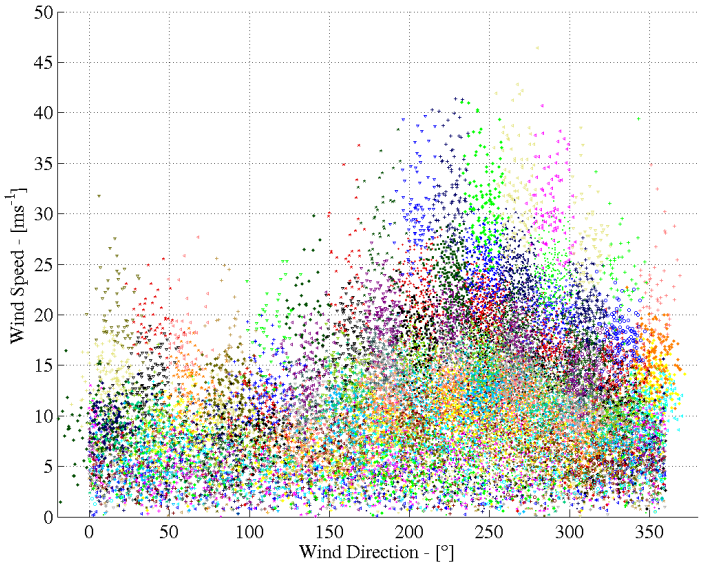

Figure C.14: The Ireland data in 151 clusters for KAMM from the existing method displayed at the second height of 1450 m

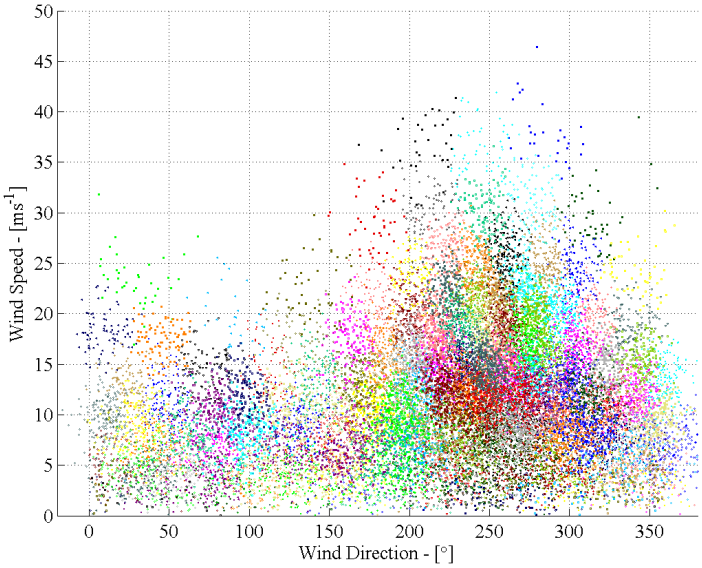

Figure C.15: The Ireland data in 151 clusters for KAMM run B1, displayed at the second height of 1450 m

C.2 The clusters for the Egypt runs

The figures showing the clusters made on the speed and direction axes for the Ireland KAMM runs are shown in the following figures. Each are displayed at the lowest height unless otherwise stated.

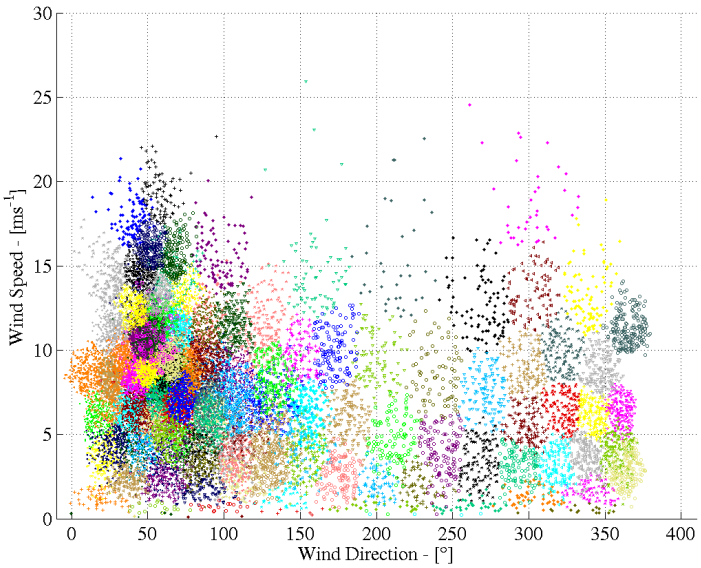

Figure C.16: The Egypt data in 126 clusters for KAMM run 1

The classes or clusters made for a classification can also be represented separately, showing the profiles of the wind speed for each observation in that class. As examples, figures C.21 and C.22 show the profiles for the first nine classes for the old method and Egypt clustering run 2, respectively. For both of these examples the classes are made with full focus on the lowest height and the inverse Froude number. The inverse Froude number helps to capture the shear in the wind, as can be seen in the figures. The wind speeds are quite spread at the top height.

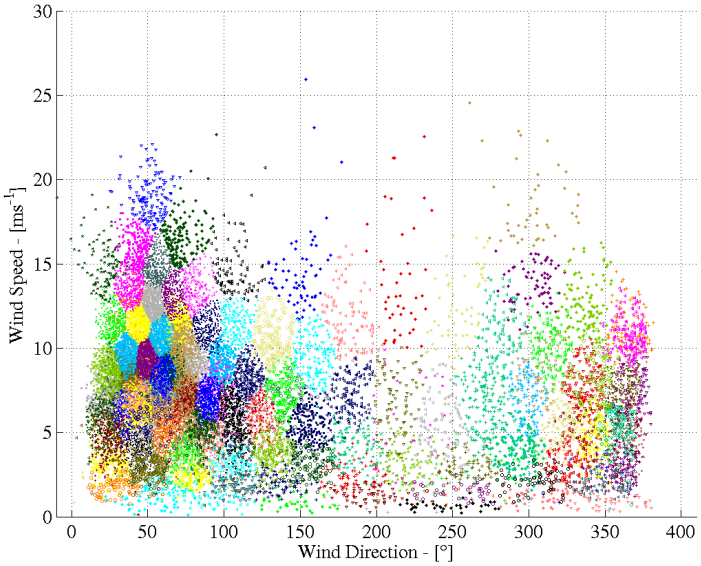

Figure C.17: The Egypt data in 126 clusters for KAMM run 2

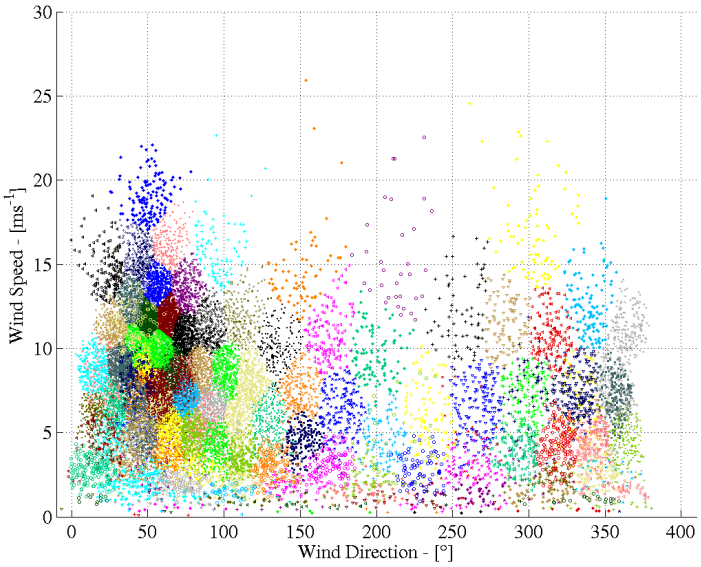

Figure C.18: The Egypt data in 126 clusters for KAMM run 3

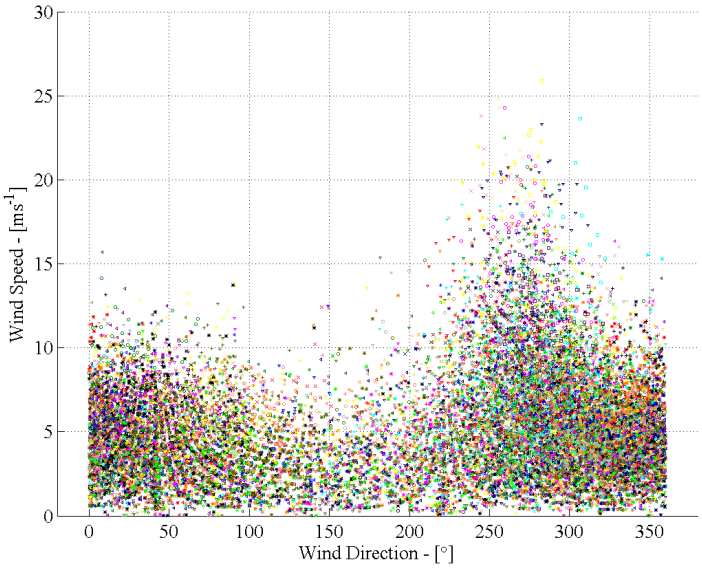

Figure C.19: The Egypt data in 126 clusters for KAMM from the existing method displayed at the second height of 1500 m

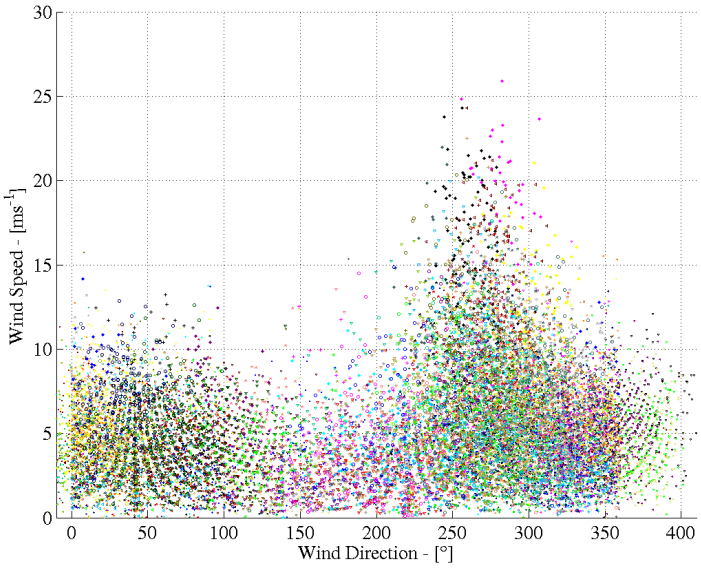

Figure C.20: The Egypt data in 126 clusters for KAMM run 1, displayed at the second height of 1500 m

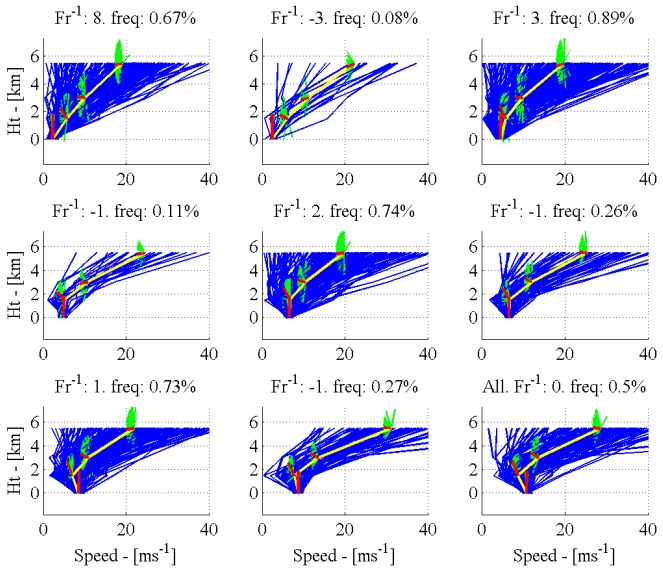

Figure C.21: The wind speed profiles for the first nine classes out of 126 from the old method for Egypt. The wind directions are also shown for each of the 4 heights where the red line is the centroid direction. The centroid inverse Froude number for the class is also shown.

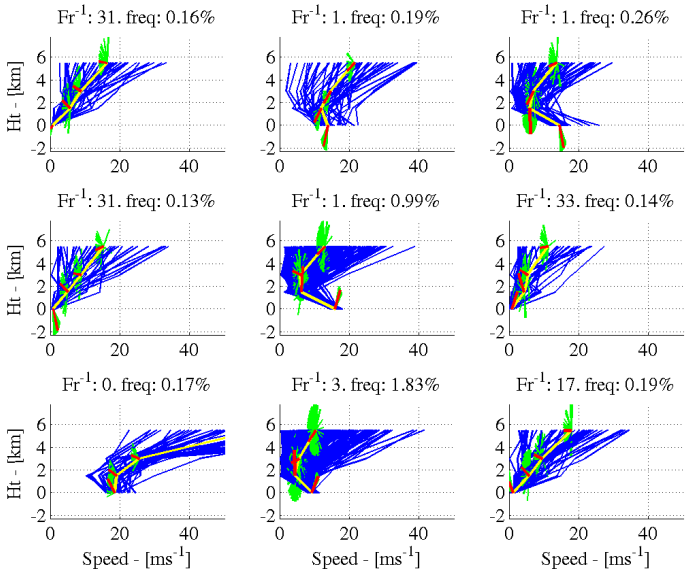

Figure C.22: The wind speed profiles for the first nine clusters out of 126 from the clustering method 2 used for Egypt.

C.3 Extra Ireland KAMM results figures

The remaining plots of the numerical wind atlas comparison results for Ireland not shown in section 8.2 are shown below. First the results for batch B, are shown for the wind energy, wind speed, wind direction and the wind roses in figures C.23 to C.26. The equivalent figures for batches C and D are shown in figures C.27 to C.33, with the exception of the wind direction rose for batch C which is in the results section of the report. Run B1 is also shown on the figures for batch D to show the trend has the number of clusters is increased since it represents the same parameter values for clustering as runs D1 and D2 except with 151 clusters. Runs D1 and D2 have 100 and 300 clusters respectively.

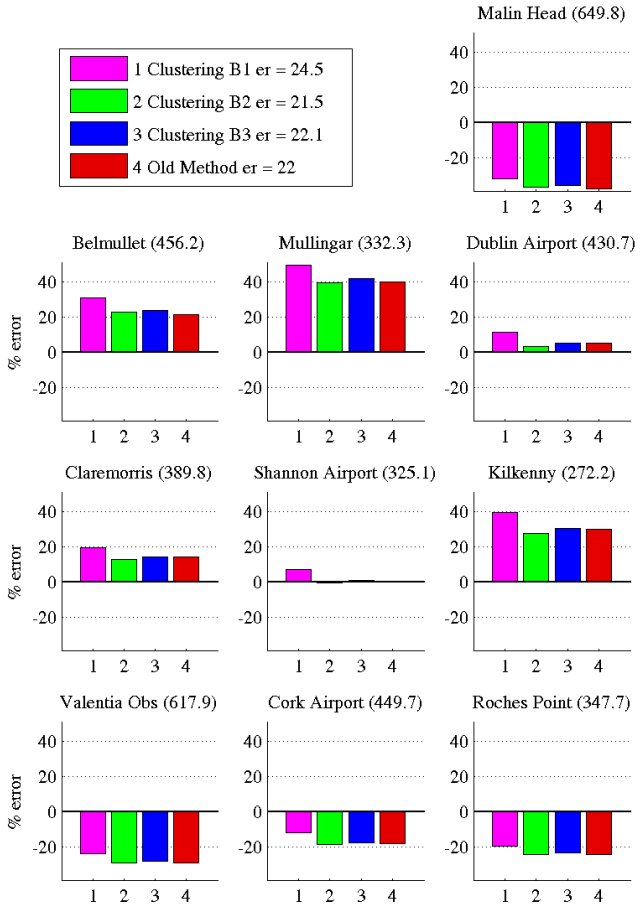

Figure C.23: Mean wind energy comparison for clustering batch B

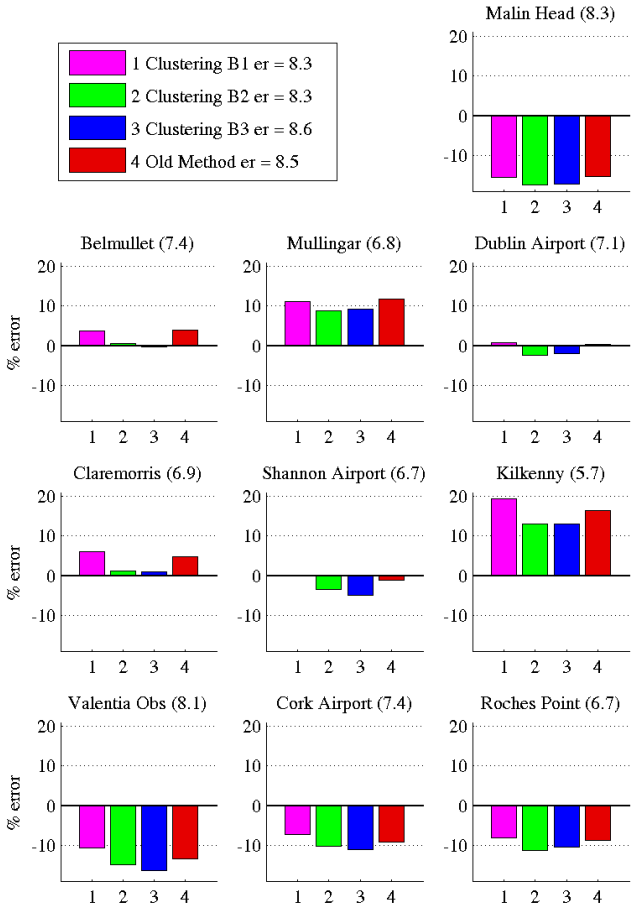

Figure C.24: Mean wind speed comparison for clustering batch B

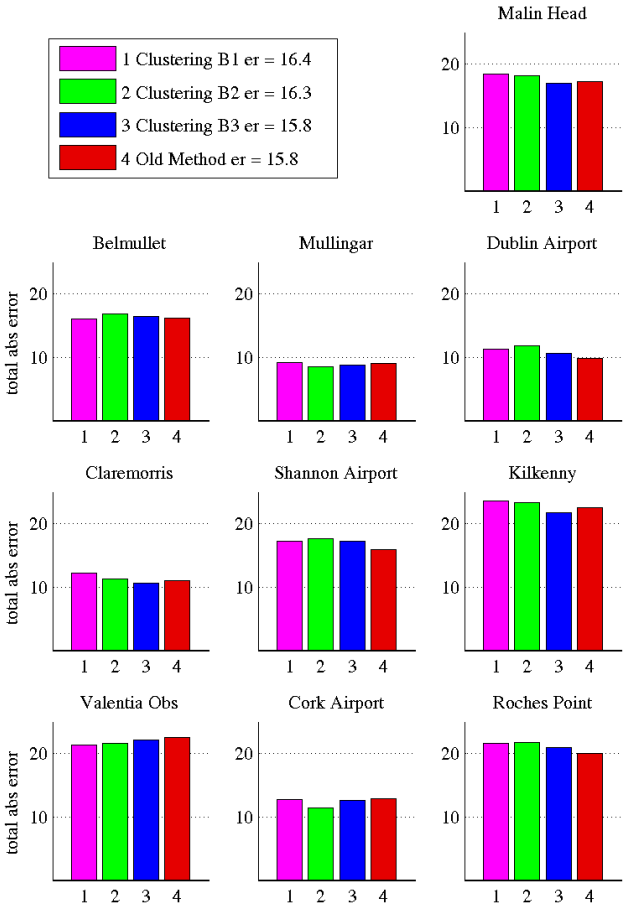

Figure C.25: Wind direction comparison for clustering batch B. The bar graph values represent the total absolute frequency error in % over the 12 sectors.

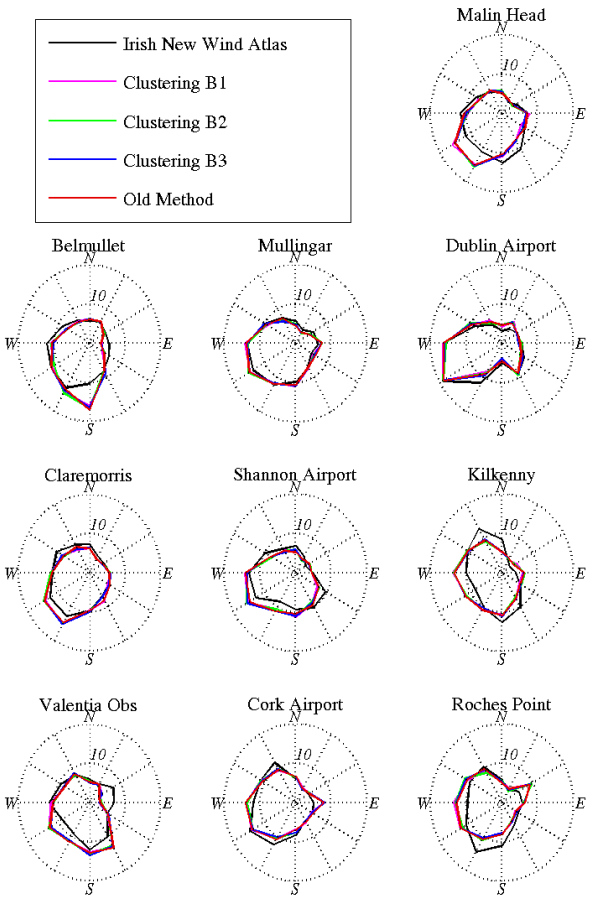

Figure C.26: Wind direction rose comparison for clustering batch B

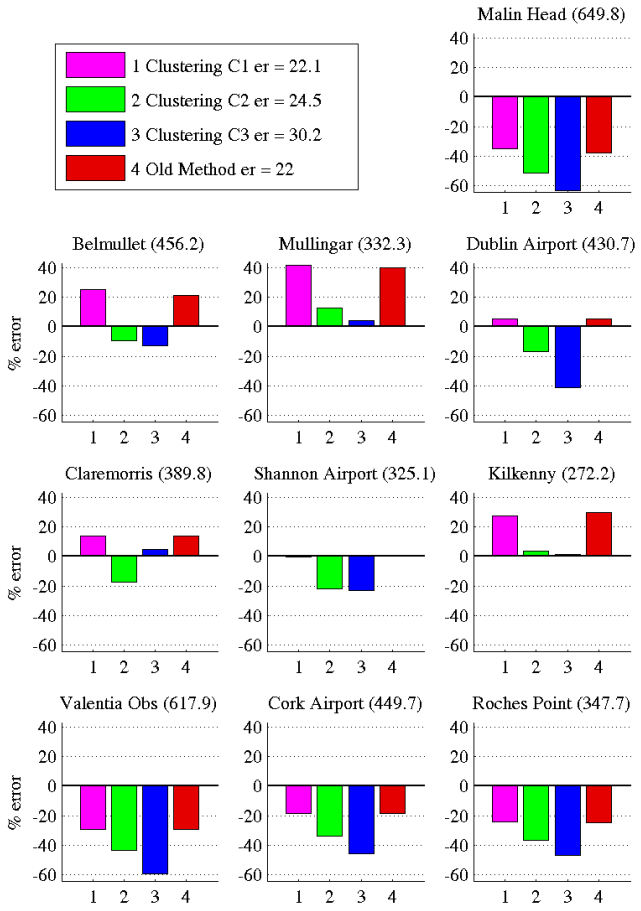

Figure C.27: Mean wind energy comparison for clustering batch C

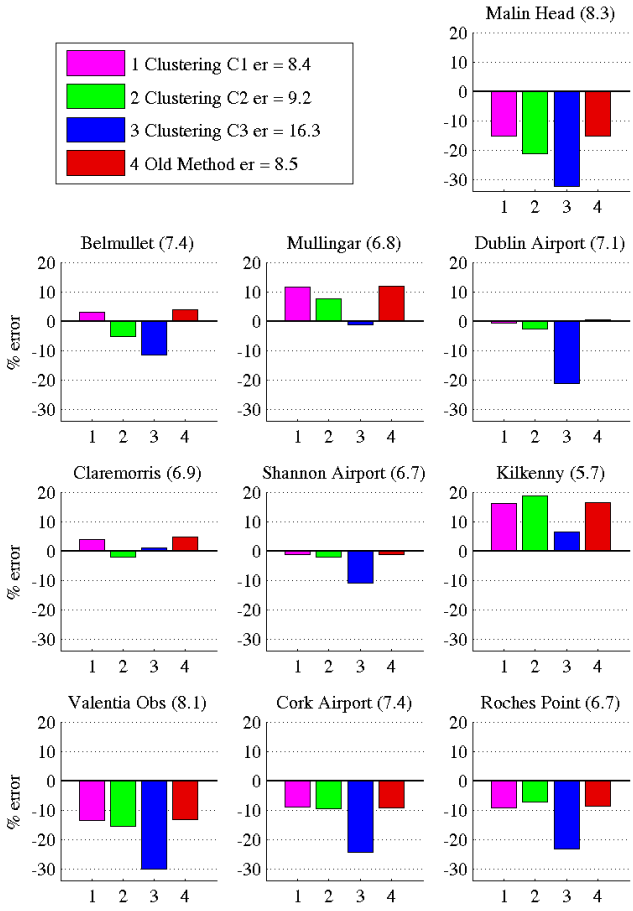

Figure C.28: Mean wind speed comparison for clustering batch C

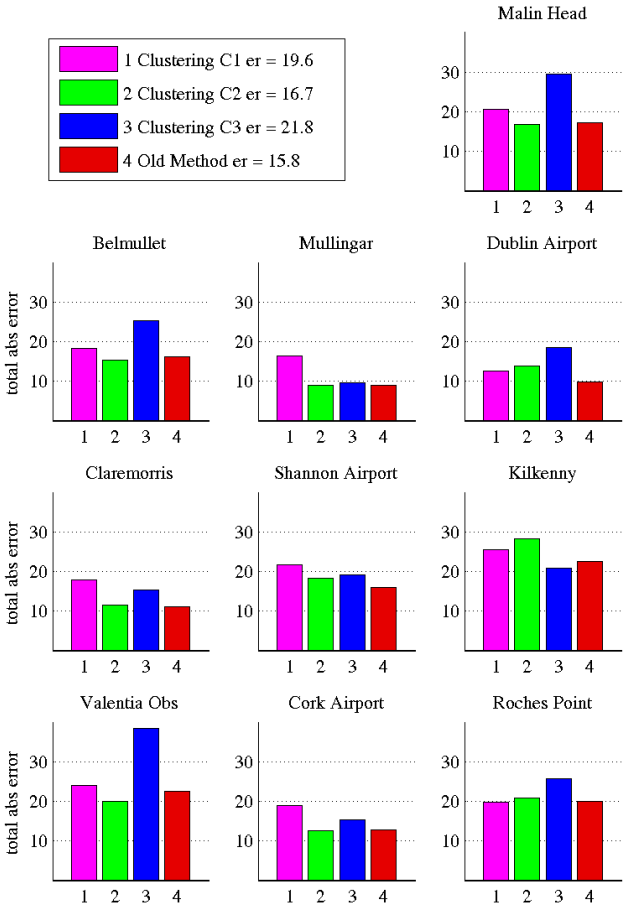

Figure C.29: Wind direction comparison for clustering batch C. The bar graph values represent the total absolute frequency error in % over the 12 sectors.

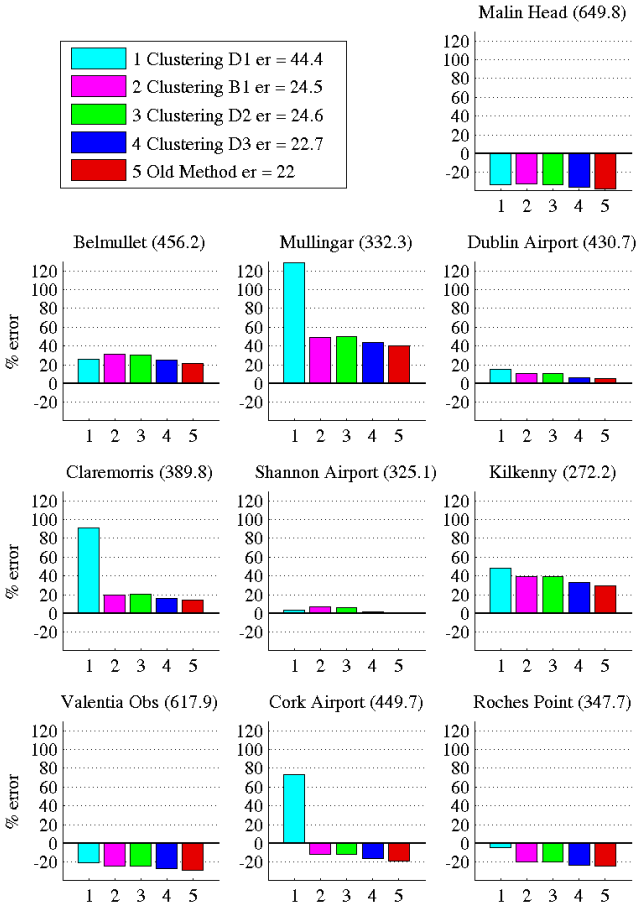

Figure C.30: Mean wind energy comparison for clustering batch D

The following figures C.33 and C.34 show the sector frequency and mean wind speeds, respectively in wind roses for batch D. For run D1 there are a small visible peak errors in the mean wind speed in the WSW sector for Cork, in the West sector for Mullingar and the WNW and WSW sectors for Claremorris. The wind frequency roses for the same stations in figure C.33 show high values in the corresponding sectors. These two factors combined give the extreme errors seen in the wind energy results for run D1 at these three stations in figure C.30 on page 159.

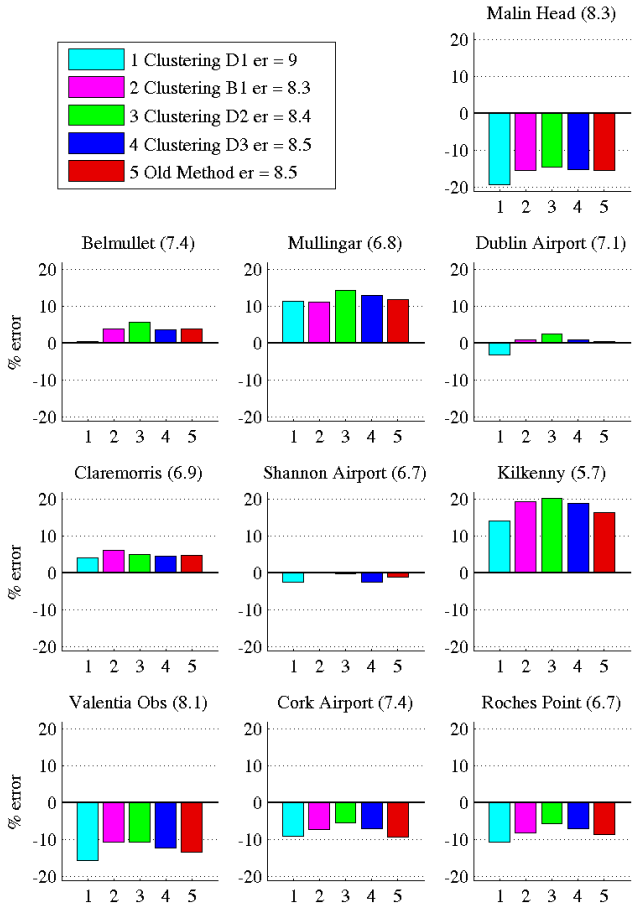

Figure C.31: Mean wind speed comparison for clustering batch D

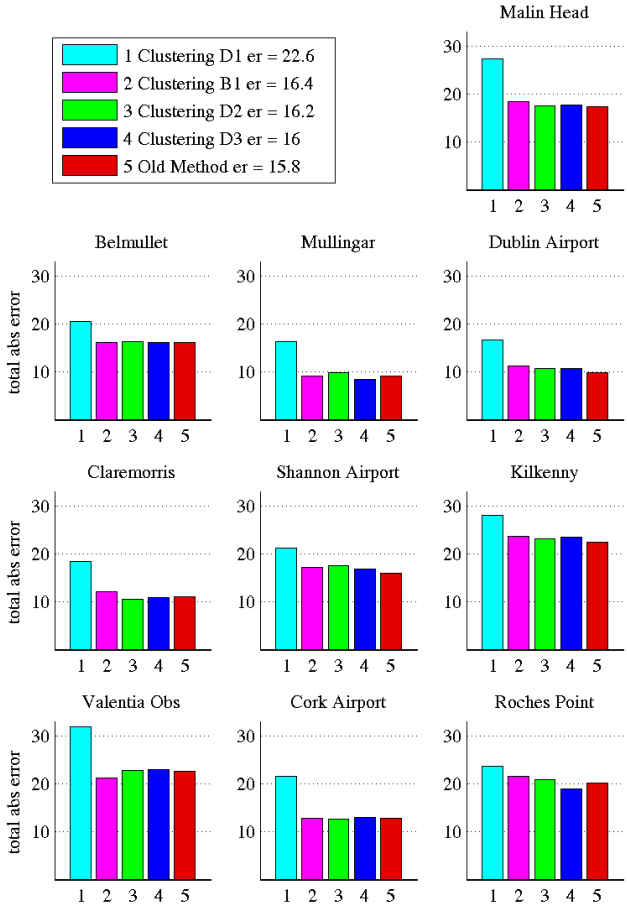

Figure C.32: Wind direction comparison for clustering batch D. The bar graph values represent the total absolute frequency error in % over the 12 sectors.

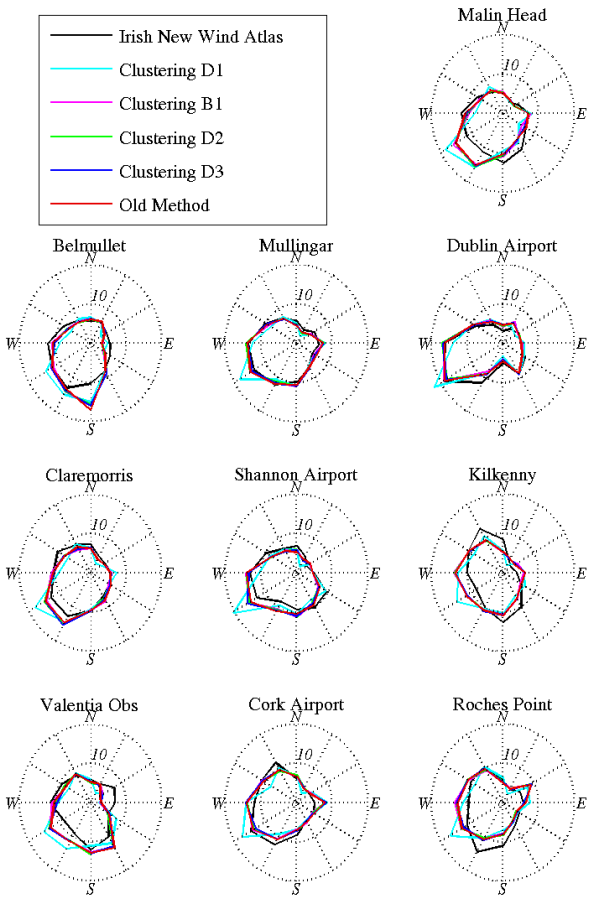

Figure C.33: Wind direction rose comparison for clustering batch D

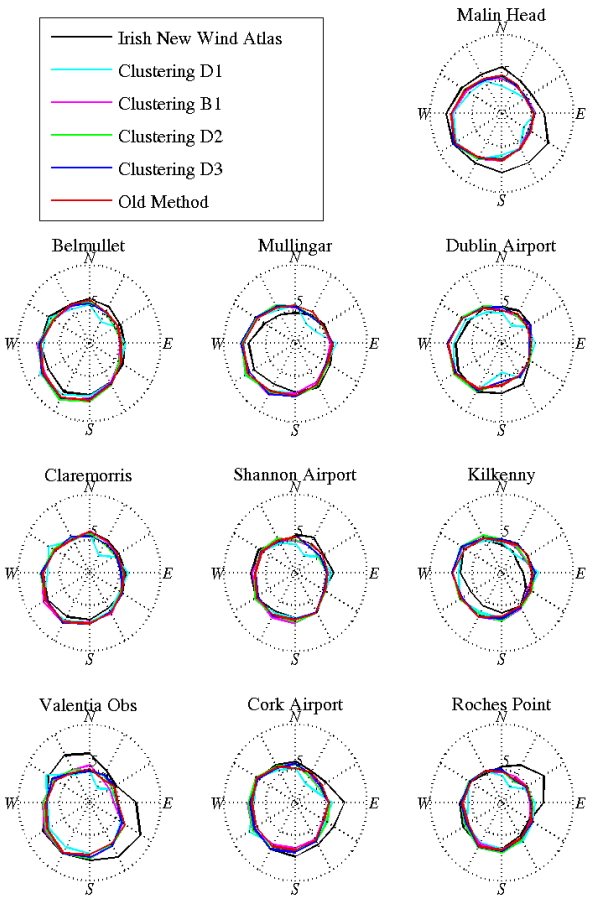

Figure C.34: Mean sector wind speed rose comparison for clustering batch D

The four absolute wind speed figures are shown in figures C.35 - C.38.

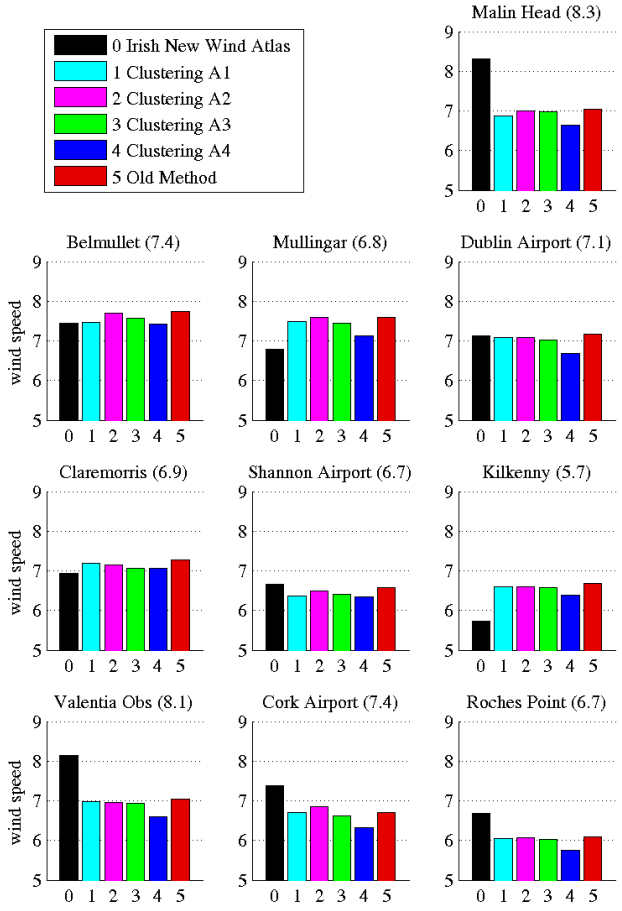

Figure C.35: Absolute wind speed comparison for clustering batch A

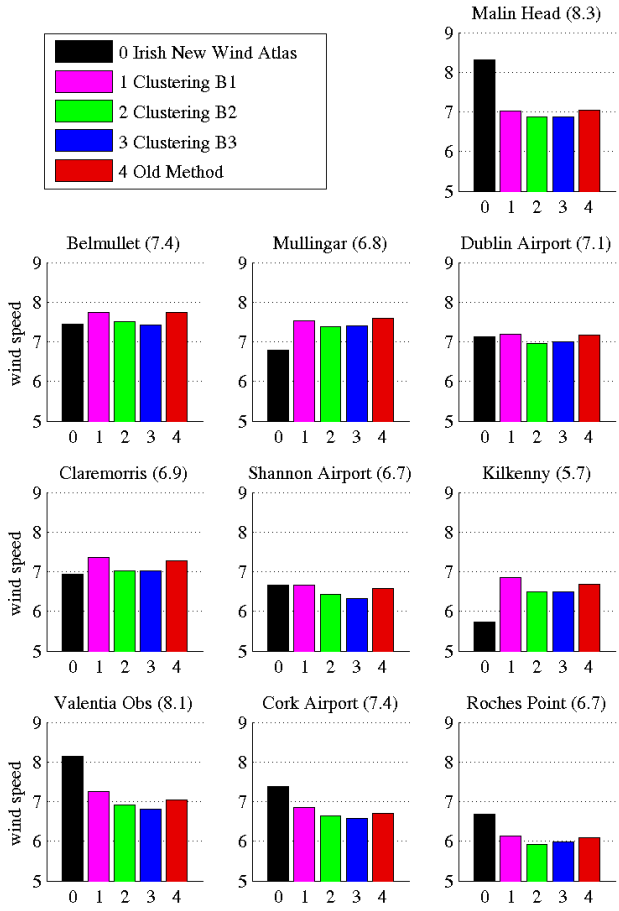

Figure C.36: Absolute wind speed comparison for clustering batch B

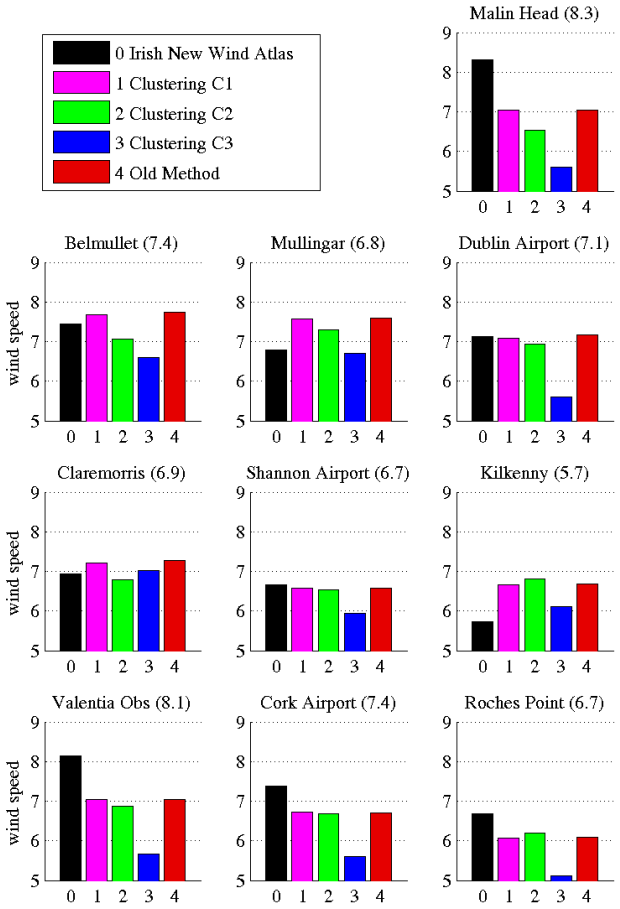

Figure C.37: Absolute wind speed comparison for clustering batch C

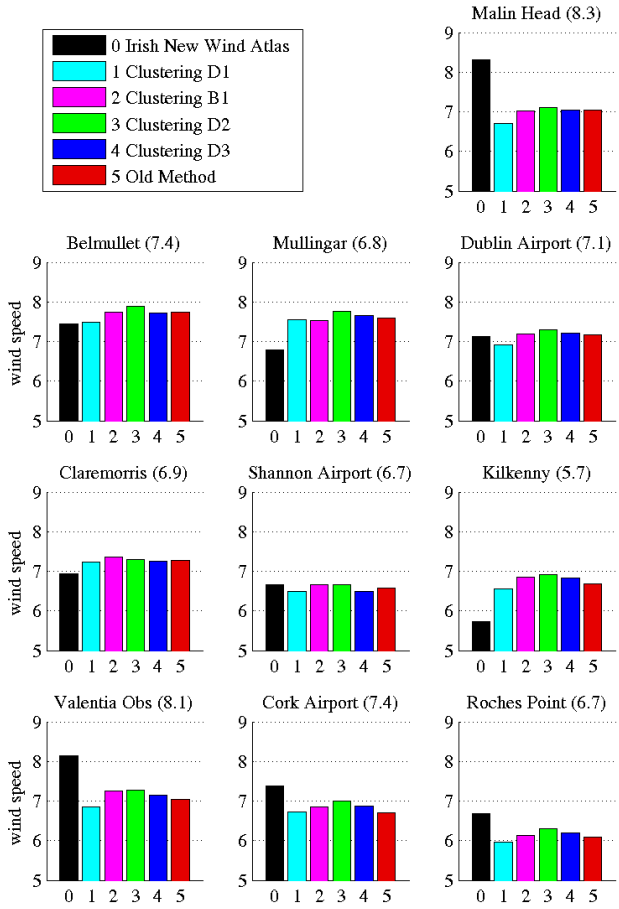

Figure C.38: Absolute wind speed comparison for clustering batch D

The mean wind speed in each sector for batches A, B and C are shown in figures C.39 - C.34.

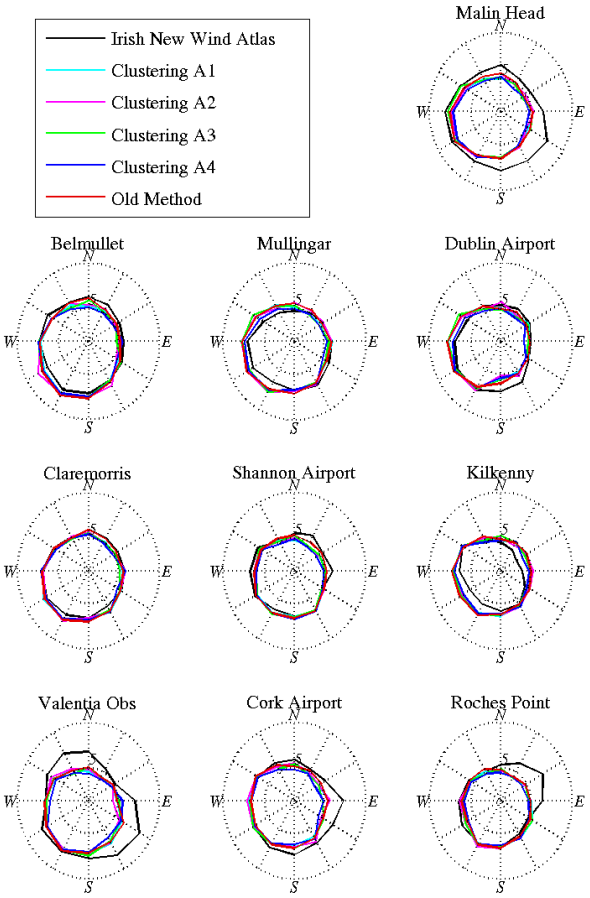

Figure C.39: Mean sector wind speed rose comparison for clustering batch A

The absolute wind speed result for Egypt is shown in figure C.42.

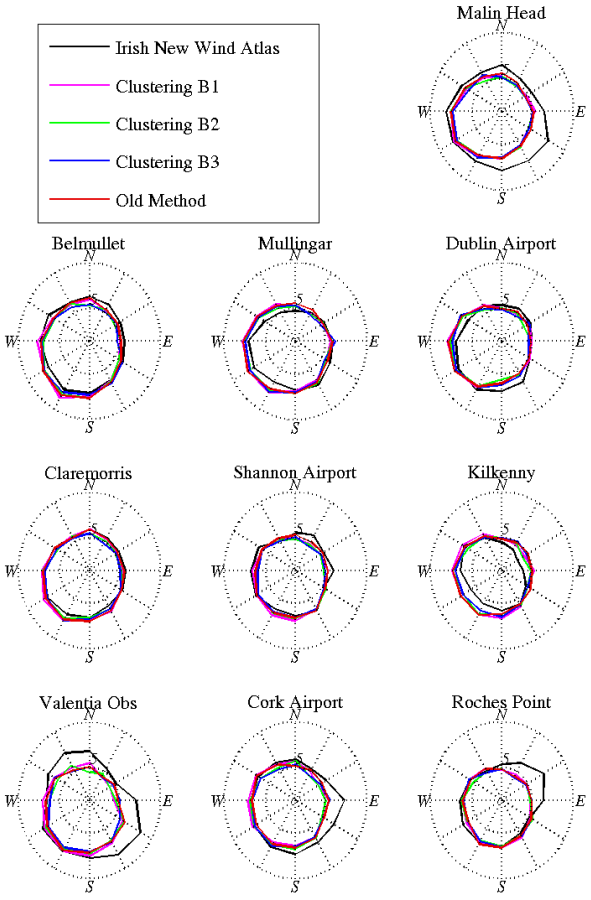

Figure C.40: Mean sector wind speed rose comparison for clustering batch B

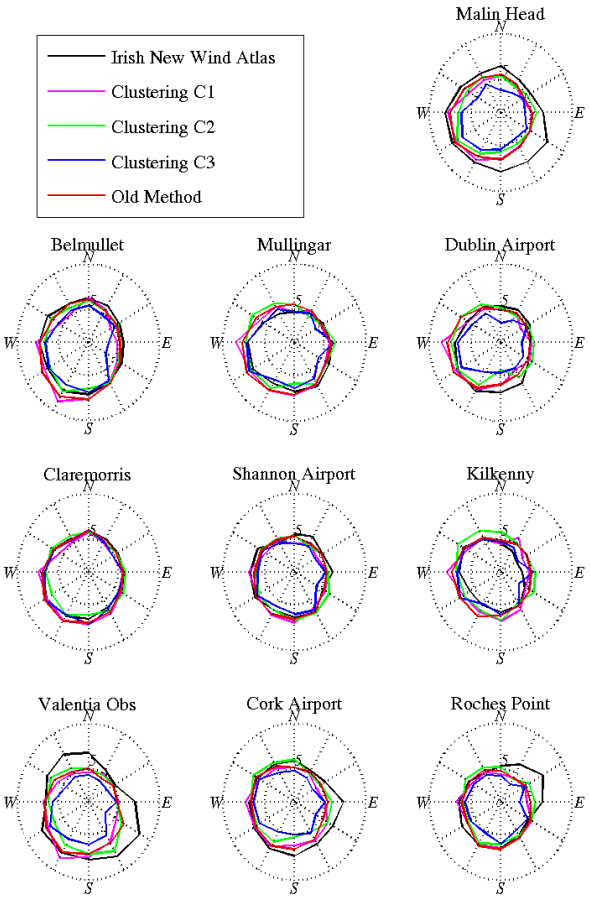

Figure C.41: Mean sector wind speed rose comparison for clustering batch C

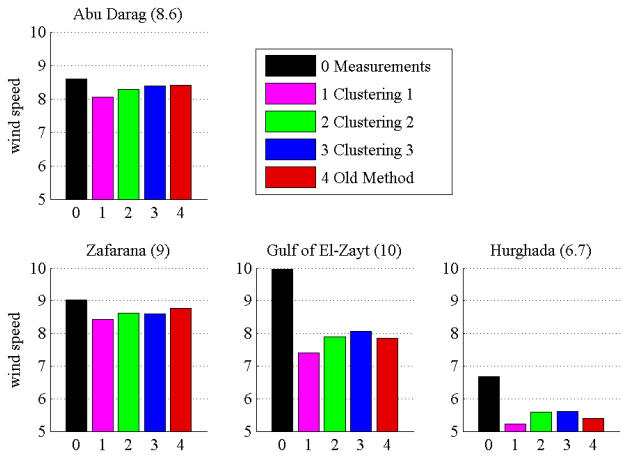

Figure C.42: Absolute wind speed comparison for the KAMM runs on Egypt

Appendix D

The parameter values used for the KAMM runs

The parameter values (as described in section 7.1) used for each clustering attempt used in KAMM for Ireland is as follows in table D.1. For each run, the top two heights inverse Froude number weights are 0.01 ($wgtFr(2) = wgtFr(3) = 0.01$). The weight on the lowest height for the inverse Frodue number and speeds and directions are both 10 ($wgtFr(1) = wgt(1) = 10$). The top two heights speed and direction weights are also 0.01 ($wgt(3) = wgt(4) = 0.01$), except run B2 where they are both 1.8 ($wgt(3) = wgt(4) = 1.8$).

Run	*R*	*sd_invFr*	*wgt(2)*	*RF*	*sd_invFr_factor*	*nCL*
A1	0·54	10	0·01	60	1·2	151
A2	0·95	12	0·01	55	1·2	151
A3	0·45	20	0·01	52	1·2	151
A4	0·5	0·8	0·01	60	1·2	151
B1	0·55	9	5	60	1·2	151
B2	0·58	50	2	60	1·2	151
B3∗	0·45	18	0·01	58	1	151
C1	10	100	0·01	80	1·2	151
C2	0·02	100	0·01	45	1·2	151
C3	0·54	0·04	0·01	60	0·8	151
D1	0·55	9	5	60	1·2	100
D2	0·55	9	5	57	1·2	300
D3	0·3	16	3	55	1·5	300

Table D.1: The parameter values for the Ireland KAMM runs. $*$ denotes a special run where the clustering is originally done for 120 clusters and then the clusters with a direction range of more than 22.15° are divided in two or three clusters. The final number of clusters in 151.

The parameter values used for each clustering attempt in the Egypt KAMM

runs are as follows in table D.2. For each run, the top two heights inverse Froude number weights are 0.01 ($wgtFr(2) = wgtFr(3) = 0.01$). The weight on the lowest height for the inverse Frodue number and speeds and directions are both 10 ($wgtFr(1) = wgt(1) = 10$). The top two heights speed and direction weights are also 0.01 ($wgt(3) = wgt(4) = 0.01$), except run B2 where they are both 1.8 ($wgt(3) = wgt(4) = 1.8$).

Run	*R*	*sd_invFr*	*wgt(2)*	*RF*	*sd_invFr_factor*	*nCL*
1	0·48	5·5	2·3	52	1·2	126
2∗	0·45	2·1	0·01	51·38	1·104	126
3	0·4	1·7	0·01	56	1·2	126

Table D.2: The parameter values for the Egypt KAMM runs. ∗ denotes a special run where the inverse Froude number is multiplied by 0.1, then transformed with the inverse tan function before standardisation with the parameters. After this, the negative inverse Froude values are multiplied by two to increase the clustering importance of thermal instability.

For Risø's purposes it is noted here that the KAMM runs for Ireland were originally performed in a different order and with different numbers to how they are presented here in this report. The new numbers in this report have used a letter convention to avoid confusion. The conversion table is shown in table D.3.

Run ID for this report	**Original Run ID**
A1	1.1
A2	1.4
A3	3.2
A4	1.3
B1	1.2
B2	3.1
B3	4.1
C1	2.3
C2	2.1
C3	2.2
D1	4.2
D2	3.3
D3	3.4

Table D.3: The original KAMM run numbers as they were made at Risø matched with the numbering convention used in this report.

Appendix E

Site coordinates

The coordinates used for the sites in Ireland are shown in 3 forms in table E.1. No minutes are given for the coordinates - only degrees and seconds.

Datum	WGS84		UTM 29		Airy - as on maps	
Site	**Lat**	**Lon**	**N**	**E**	**N**	**E**
Belmullet	54° 14’	10° 0’	6010.074	434.815	332·825	69·19
Claremorris	53° 43’	08° 59’	5952.126	501.099	273·95	134·53
Cork	51° 51’	08° 29’	5744.600	535.589	65·225	166·11
Dublin	53° 26’	06° 15’	5924.125	682.689	242·85	315·9
Kilkenny	52° 40’	07° 16’	5836.723	617.219	157·39	249·495
Malin Head	55° 22’	07° 20’	6136.993	605.639	458·55	241·96
Mullingar	53° 32’	07° 21’	5932.995	609.363	254·33	242·4
Roches Point	51° 48’	08° 15’	5739.179	551.719	60·58	182·825
Shannon	52° 41’	08° 55’	5837.171	505.633	161·55	138·025
Valentia	51° 56’	10° 15’	5754.481	414.056	78·57	45·86

Table E.1: The coordinates used for the meteorological stations in Ireland. The latitude-longitude coordinates along with their UTM 29 conversions come directly from [33]. The Airy coorindate system is used for the KAMM grid.

The coordinates used for the sites in Egypt are shown in 3 forms in table E.2.

Datum	**WGS84**		**UTM 36 (km)**	
Site	**Latitude**	**Longitude**	**N**	**E**
Abu Darag	29° 16' 49.51"	32° 36' 03.03"	3239.1204	461.2272
Zafarana	29° 06' 48.9"	32° 36' 39.02"	3220.6325	462.1370
Gulf of El-Zayt	27° 47' 23.83"	33° 28' 22.97"	3074.0243	546.6014
Hurghada	27° 18' 59.5"	33° 42' 02.35"	3021.6890	569.3208
Datum	**KAMM grid x,y (km)**		**Height a.s.l. (m)**	
Abu Darag	93.9820	283·2657	11	
Zafarana	85.5259	266·7998	25	
Gulf of El-Zayt	85.3701	97·6011	14	
Hurghada	78.8781	40·9177	13	

Table E.2: The coordinates used for the meteorological stations in Egypt. The latitude-longitude coordinates along with their UTM 36 conversions come directly from [24]. The KAMM grid coorindate system used is rotated 30 degrees.

Appendix F

Perl code

The following perl program is the perl program henceforth used by Risø National Laboratory. It calls the following Fortran program to perform the Colour Quantisation and Forgy clustering algorithms.

F.1 classWithClustering.pl

```
#!/usr/bin/perl
##!/usr/unic/bin/perl5

# classWithClustering.pl

# Colour quantisation is a method for classing millions of colours in an
    image
# for displaying with only a few colours - 16 or even 256.
# The method has similar objectives to the the contruction of geostrophic
# wind classes for mesoscale modelling.
#
# This perl program is used as an input file for a fortran program of the
     same
# name, which performs the classification.
# If $forgy is set to 1, with the right parameters, the Forgy algorithm
# optimises the classes found with Colour Quantisation.
# Representative values of geostrophic wind classes are calculated for
    one
# point.
#
# The classification is done for an ASCII file which was generated by
# high_serz.f90 and surface data generated by time_vgsfc.f90 .
#
#
# Nicholas Cutler, 22.03.2005
# Mar. 21. 2005 - NJC: The program's inception.
```

```
#
#-------------------------------------------------------------------------
$HOME = $ENV{"HOME"};

# Setup coordinates.
# One file is used to make the clustering on, and the other files for
# neighbouring cooridnates are used to recalculate the frequencies for
# the frequency output file.
# Always put the main coordinate on which the clustering is to be done,
    first.
#push @ncepncar_coords, 'cdze33n28'; # Egypt - main coorindate.

#push @ncepncar_coords, 'cdze31n26'; # These are the other coordinates
    for possibly
#push @ncepncar_coords, 'cdze31n28'; # producing frequencies on in .frq
    file.
#push @ncepncar_coords, 'cdze31n31'; # These must be in the same order as
     in the file
#push @ncepncar_coords, 'cdze33n26'; # and should include the main
    coordinate (i.e.
#push @ncepncar_coords, 'cdze33n28'; # the main coorindate is repeated
    here).
#push @ncepncar_coords, 'cdze33n31';
#push @ncepncar_coords, 'cdze36n26';
#push @ncepncar_coords, 'cdze36n28';
#push @ncepncar_coords, 'cdze36n31';

push @ncepncar_coords, 'cdzw08n53'; # Ireland - main coordinate.

push @ncepncar_coords, 'cdzw02n50'; # These are the other coordinates for
     possibly
push @ncepncar_coords, 'cdzw02n52'; # producing frequencies on in .frq
    file.
push @ncepncar_coords, 'cdzw02n55'; # These must be in the same order as
    in the file
push @ncepncar_coords, 'cdzw02n57'; # and should include the main
    coordinate (i.e.
push @ncepncar_coords, 'cdzw03n51'; # the main coorindate is repreated
    here).
push @ncepncar_coords, 'cdzw03n53';
push @ncepncar_coords, 'cdzw03n56';
push @ncepncar_coords, 'cdzw05n50';
push @ncepncar_coords, 'cdzw05n52';
push @ncepncar_coords, 'cdzw05n55';
push @ncepncar_coords, 'cdzw05n57';
push @ncepncar_coords, 'cdzw06n51';
push @ncepncar_coords, 'cdzw06n53';
push @ncepncar_coords, 'cdzw06n56';
push @ncepncar_coords, 'cdzw07n50';
push @ncepncar_coords, 'cdzw07n52';
```

```
push @ncepncar_coords, 'cdzw07n55';
push @ncepncar_coords, 'cdzw07n57';
push @ncepncar_coords, 'cdzw08n51';
push @ncepncar_coords, 'cdzw08n53';
push @ncepncar_coords, 'cdzw08n56';
push @ncepncar_coords, 'cdzw10n50';
push @ncepncar_coords, 'cdzw10n52';
push @ncepncar_coords, 'cdzw10n55';
push @ncepncar_coords, 'cdzw10n57';
push @ncepncar_coords, 'cdzw11n51';
push @ncepncar_coords, 'cdzw11n53';
push @ncepncar_coords, 'cdzw11n56';
push @ncepncar_coords, 'cdzw12n50';
push @ncepncar_coords, 'cdzw12n52';
push @ncepncar_coords, 'cdzw12n55';
push @ncepncar_coords, 'cdzw12n57';

# Set the index of the main coordinate in middle the list of all
    coorindates
# starting from one with the main coorindate.
$index_main = 21; # 6 for Egypt and 21 for Ireland.
# And set the total number of coordinates used (including the main one,
    twice).
$nCoords = 33; # 10 for Egypt, 33 for Ireland.
# And the number of characters in the ncepncar_coords file names:
$lfn = 9;

# Set the path for the file to cluster on.
$file_path = "/mary-fs/home/jaba/Reanalysis/Ireland/Data/"; # Ireland.
#$file_path = "/mary-fs/home/bhoj/Reanalysis/Suez/Data/"; # Egypt.
#$file_path = "$HOME/Reanalysis/Egypt/Data/";

# And the file path for the files to make the frequencies on.
$file_path_frq = "/mary-fs/home/jaba/Reanalysis/Ireland/Data/";
#$file_path_frq = "/mary-fs/home/bhoj/Reanalysis/Suez/Data/";

$file_ext = ".d";

# Define the heights in m.
$hts0 = 0;
$hts1 = 1450;
$hts2 = 3000;
$hts3 = 5500;

# Define the outfile type.
# 1: .cl file.
# 2: .frq file.
# 3: .nsi file (nearest simulation indicies for WAsP direction
    interpolations).
```

```
# This only works if forgy == 1 as well (i.e. Forgy was used as well as
    CQ).
# 0: output for Matlab - includes the mean standard deviations on each
    variable.
$outfileType = 3;
# If .frq file is used, set whether fixed frequency (0) or wandering
    frequencies (1).
$freqType = 1;

# Setup reanalysis data files.
for $data_file (@ncepncar_coords) {
     push @data_files, $data_file;
}

# Desired number of classes.
$nCL = 120;

# Desired ratio for the speeds compared to the directions. The higher
    this number,
# the more the speed is weighted against the directions. Used in both
    Forgy and
# Colour Quantisation.
$R = 0.45;
# And the ratio factor used for the Forgy step. The higher this number,
    the more
# the speed is weighted against directions in Forgy. This number should
    be around
# 60 for a consistent improvement with Forgy after Colour Quantisation.
$rfactor = 58;

# Desired ratio between the speed/directions and the inverse Froude
    number.
# Note: the individual weightings of the height-diffs (wgtsFr) will be
    scaled so
# that this sd_invFr_ratio is the true ratio between the speeds and invFr
     numbers
# at the bottom height.
$sd_invFr_ratio = 18;
# And set the factor to change sd_invFr_ratio by when switching from CQ
    to
# Forgy. This multiplied by the sd_invFr_ratio for Forgy.
$sd_invFr_factor = 1;

# Define the weighting on the data for each height.
# wgt0 represents the lowest height. If wgt0 is 2 and the others 1, the
# bottom height will be twice as important as the other heights.
# Important to set wgt0 to 1 if nDim = 2.
$wgt0 = 10;
$wgt1 = 0.01;
$wgt2 = 0.01;
```

```
$wgt3 = 0.01;

# Define similarly the weightings for the 3 inverse Froude numbers
# based on comparing between two heights.
$wgtFr0 = 10;
$wgtFr1 = 0.01;
$wgtFr2 = 0.01;

# forgy. 1: Use the CQ seeds to run the Forgy method to make the classes.
# 0: Only use the CQ method to make the classes.
$forgy = 1;

# Define number of dimensions to classify on.
# 2: Only speed and direction at lowest height (or u and v if uv_OR_sd =
    1).
# 8: Speed and direction at all heights (or u and v if uv_OR_sd = 1).
# 11: Speed and direction at all heights plus 3 inverse Froude numbers
# (or u and v if uv_OR_sd = 1).
$nDim = 11;

# Define the maximum direction range allowed for the classes.
# This will likely increase the number of classes specified in nCL.
# If 0, no direction range limit is put on the classes.
$maxDirRange = 22.15;

# Define how separated the inverse Froude numbers are to be - this is
# multiplied to the inverse Froude numbers before the inverse tan is
# taken, and this followed by standardising the resulting numbers.
# If invFr_sep == 0, the clustering is made on the linear inverse Froude
    number.
$invFr_sep = 0;

# uvORsd = 0 means that the directions and speeds are the dimensions
# for minimising the variance.
#
# uvORsd = 1 means that the original standardisation of the speed is
# used and the u and v components are calculated from that.
#
# uvORsd = 2 means that the extra dimension is added for each height
# that such the speed and directions are used to calculate
# the variance - not u and v. In this case, the 3
# standardised dimensions for each height are:
# speed, sin(dir) and cos(dir).
# The value for R sets how the is then multiplied by the sin
# and cos terms to define how the speed is related to the
# other two dimensions in terms of distance.
# Note that uvORsd is changed to 0 for the Forgy part of the algorithm.
$uvORsd = 2;

# Define whether the stability is calculated as the inverse Froude
```

```
# number (0), or shear (1).
$invFrType = 0;

# Define whether a sin-cos conversion is required.
# If 0, no conversion is made (the normal case).
# If another number, the conversion is made.
# This is used for the *.frq file only (outfileType = 2).
$azklm = 0;

# Define if whole period is going to be used or between certain years.
# If both 0, whole period is used.
# Otherwise they are set to the start and finish years.
$start_year = 0;
$finish_year = 0;

# echo 'Max. speed of class "calms"'
# read ucalm
# $ucalm=0.1;
#if ($eqmode>=2) {
# $pcalm= $ucalm*1.0e-5; # classifying forcing
#} else {
# $pcalm= $ucalm; # classifying wind
#}

#-----------------------------------------------------------------------
# Replace the column of geostrophic wind at the surface by the data
# derived from the surface pressure
# ug_file = cdz*.d
# vgs_file = vgs4*.d?
#-----------------------------------------------------------------------
# Start of loop for each set of coordinates (ug_files)
# The fortran program classWithClustering is called.

#----------------
# set array size approx.
$nObs = 25200;

#-----------------------------------------------------------------------
@output = qx($HOME/NWA/Clustering/classWithClustering << EEND
    $nCoords
    $file_path
    $file_path_frq
    $file_ext
    @ncepncar_coords
    $index_main
    $lfn
    $freqType
    $outfileType
    $nDim
    $hts0
```

```
    $hts1
    $hts2
    $hts3
    $wgt0
    $wgt1
    $wgt2
    $wgt3
    $wgtFr0
    $wgtFr1
    $wgtFr2
    $nObs
    $nCL
    $R
    $rfactor
    $sd_invFr_ratio
    $sd_invFr_factor
    $maxDirRange
    $invFr_sep
    $uvORsd
    $invFrType
    $azklm
    $forgy
    $start_year
    $finish_year
EEND);

#-------------------------------------------------------------------------
# Grep gets the line with "Point of Analysis" in it
@pa = grep( /Point of analysis/, @output);

($text, $_) = split( ':', $pa[0]);
# d and e and just dummy variables.
( $lon, $d, $e, $lat, $rest ) = split;
@pa = grep( /classes in/, @output);
# ncl is the number of classes as obtained from the output file
( $ncl, @rest ) = split( ' ', @pa[0]);

$outbase = sprintf( "e%2.2dn%2.2d", $lon, $lat ) if ($lon>=0 && $lat>=0);
$outbase = sprintf( "w%2.2dn%2.2d",-$lon, $lat ) if ($lon< 0 && $lat>=0);
$outbase = sprintf( "e%2.2ds%2.2d", $lon,-$lat ) if ($lon>=0 && $lat< 0);
$outbase = sprintf( "w%2.2ds%2.2d",-$lon,-$lat ) if ($lon< 0 && $lat< 0);

if ($outfileType == 1)
{
   $outfile = "Class/Classes_".$outbase .".cl";
}
elsif ($outfileType == 2)
{
   $outfile = "Class/Classes.frq";
}
```

```
elsif ($outfileType == 3)
{
   $outfile = "Class/Classes.nsi";
}
else
{
   $outfile = "Class/Classes_".$outbase .".txt";
}

print "Output on file $outfile\n";

open RESULT, ">$outfile";
print RESULT @output;
close RESULT;
```

Appendix G

Fortran 90 code

The following fortran 90 program is called by the above Perl program. It performs the Colour Quantisation and Forgy algorithms and henceforth being used by Risø National Laboratory. Only the header comments are shown.

G.1 classWithClustering.f090

```
program classWithClustering
!-----------------------------------------------------------------------
! classWithClustering performs a 2-stage clustering algorithm on a set
! of NCEP/NCAR goestrophic wind data. This is the classification stage
! of the mesoscale modelling method, as run by RisNational Laboratory
! in Roskilde, Denmark.
!
! This fortran program is called from a perl program of the same
! name, classWithClustering.pl, which initialises some parameters for
! the classification (see the perl program for the details).
! This program executes the colour quantisation (CQ)
! algorithm for making wind classes on NCEP/NCAR data.
! If the setting is made from the perl program, the classifcation is
! optimised on a second stage of clustering using the Forgy method.
!
! Colour quantisation is a method for classing millions of colours in an
! image for displaying with only a few colours - 16 or even 256.
! The method has similar objectives to the the contruction of geostrophic
! wind classes for mesoscale modelling.
!
! The Forgy method is a non-hierarchical clustering method. It uses the
! centroids from the CQ result as seeds to build the clusters around.
! The data is classed with the "closest" seed (the distance defined by
! the parameters set in the perl program) and new centroids are
    calculated.
! The new centroids are used as new seeds for the process to be iterated
```

```
! until the total sum of squared error stops decreasing and the optimum
    is
! reached.
!
! Representative values of geostrophic wind classes are calculated as
! the final centroids, from the original data read in. The output is
! written to a .txt, .cl or .frq file, depending on the parameter set in
! the perl program. The .txt file contains the statistics of the results.
! The other two files are the files required for Ris's current mesoscale
! modelling process with KAMM.
!
! The program reads ascii-files with data in columns. The data consists
    of
! geostrophic speed [m/s], direction [deg], virtual temperature [K],
! pressure [hPa]. The virtual temperature is converted to virtual
    potential
! temperature. From this, the stratification can be calculated.
!
! Nicholas Cutler, VEA Risoe, 22.03.2005
! Mar. 21. 2005 - NJC: The program's inception.
! May 24. 2005 - NJC: Program neatened up for Report.
!
!-------------------------------------------------------------------------
```

Appendix H

MatLab code examples

The header comments for a couple of representative MatLab programs are shown below.

H.1 plotOldClasses.m

```
% Risoe National Laboratory and the Technical University of Denmark.

% Numerical Wind Atlas thesis

% Nick Cutler s000144

% March 2005

%-----------------------------------------------------------------------
%
% Matlab program to read in a ".d" data text file with 24 columns
% starting at the 6th row, containing the data for:
% - yr, month, day, hour,
% - wind speed at 4 heights
% - wind direction at 4 heights
% - potential temperature at 4 heights
% - pressure at 4 heights, and
% - relative humidity at 4 heights.
%
% Pressure and relative humidity are not needed.
%
% The program also reads in a ".lim" file as produced from the old
% wind classification method containing the limits for the classes.
% There are 9 columns:
% - First 2 are the minimum and maximum direction,
% - next two are the minimum and maximum wind speed, and
% - next two are minimum and maximum inverser froude number
```

```
% - last 3 are a '#', class number and class frequency. Last 3 are not
% needed by this method.
%
% Then, this script assigns the data to classes defined by the limits.
% It produces a scatter plot showing the classes in the u,v plane.
%
%-------------------------------------------------------------------
```

H.2 plotClustersAllH.m

```
% Risoe National Laboratory and the Technical University of Denmark.

% Numerical Wind Atlas thesis

% Nick Cutler s000144

% March 2005

%-------------------------------------------------------------------
%
% Matlab program to read in a text file with 15 columns containing
% the data for: u and v at 4 different heights, the stability based
% inverse Froude number at 3 height differences, 3 shear based
% inverse Froude numbers at 3 height differences and a cluster ID
% assigning the wind data to a cluster.
%
% Then, this script assigns the data to clusters and produces plots
% showing selected clusters with their profile members and the
% centroid.
%-------------------------------------------------------------------
```

Appendix I

SAS code

I.1 clusterTestCyl.sas

```
/* clusterTestCyl.sas */
/* */
/* Nicholas Cutler, s000144 */
/* March, 2005 */
/* */
/* Get data from example file for Egypt: Egypt_e33n28.d */
/* Make some clustering analyses on the data. */

/* The data gives wind speeds and directions and these */
/* must be converted to u and v. Note the directions */
/* are where the wind comes from. */

/* Clustering variances based on translation to a */
/* cylinder. */

/* Get data first */
data egypt0;

infile 'Data/Egypt_e33n28_noOut.d' firstobs=6 obs=24840;
/*infile 'Data/Ireland_w08n53.d' firstobs=6 obs=24841; */

input yr mo day hr sp0 sp1 sp2 sp3 D0 D1 D2 D3;

sinD0 = cos((270 - D0)*3.1415926536/180);
sinD1 = cos((270 - D1)*3.1415926536/180);
sinD2 = cos((270 - D2)*3.1415926536/180);
sinD3 = cos((270 - D3)*3.1415926536/180);
cosD0 = sin((270 - D0)*3.1415926536/180);
cosD1 = sin((270 - D1)*3.1415926536/180);
cosD2 = sin((270 - D2)*3.1415926536/180);
cosD3 = sin((270 - D3)*3.1415926536/180);
```

```
proc means data=egypt0 ;
var sp0-sp3 sinD0-sinD3 cosD0-cosD3;

/*proc standard mean=0 std=1 data=egypt0 out=egypt1;*/
/*var sp0-sp3 sinD0-sinD3 cosD0-cosD3;*/

/*proc means data=egypt1;*/
/*var sp0-sp3 sinD0-sinD3 cosD0-cosD3;*/

data egypt;
  set egypt0;

  /* Standardise the speeds to mean 0 and stdev;*/
  /*sp0 = (sp0 - 8.0071648) / 3.6113620;*/
  /*sp1 = (sp1 - 5.0504440) / 3.0509798;*/
  /*sp2 = (sp2 - 7.4186093) / 5.0334971;*/
  /*sp3 = (sp3 - 14.8664588)/ 10.5055759;*/

  /* Standardise the speeds to stdev 1;*/
  sp0 = sp0 / 3.6113620;
/* sp1 = sp1 / 3.0509798; */
/* sp2 = sp2 / 5.0334971;*/
/* sp3 = sp3 / 10.5055759;*/

  /* Set fixed factor for how speed and direction are related in distance
      ;
  sp0 = sp0 * 0.5;
/* sp1 = sp1 * 0.5;*/
/* sp2 = sp2 * 0.5;*/
/* sp3 = sp3 * 0.5;*/

  /* Set weightings between the heights for the wind.*/
/* sp0 = sqrt(25) * sp0;*/
/* sinD0 = sqrt(25) * sinD0;*/
/* cosD0 = sqrt(25) * cosD0;*/
/* sp1 = sqrt(4) * sp1;*/
/* sinD1 = sqrt(4) * sinD1;*/
/* cosD1 = sqrt(4) * cosD1;*/
/* sp2 = sqrt(1) * sp2;*/
/* sinD2 = sqrt(1) * sinD2;*/
/* cosD2 = sqrt(1) * cosD2;*/
/* sp3 = sqrt(1) * sp3;*/
/* sinD3 = sqrt(1) * sinD3;*/
/* cosD3 = sqrt(1) * cosD3;*/


/* Keep only the variables required for the clustering;*/
/*keep sp0-sp3 sinD0-sinD3 cosD0-cosD3;*/
keep sp0 sinD0 cosD0;
```

```
/*----------------------------------------*/

/* Make data set with random selection of data;*/
data egypt2;
   set egypt;
   ran = ranuni(47);
   if ran < 0.5; /* This number defines the rough percentage of data to
       get;*/
                 /* It gives the same random numbers each time;*/

/* Do a clustering analysis on the kept Egypt data;*/
proc cluster data=egypt outtree=tree method=twostage K=27 pseudo p=300;
   /*var sp0-sp3 sinD0-sinD3 cosD0-cosD3;*/
   var sp0 sinD0 cosD0;

/* Use 86 clusters since that is the same as the what the old method;*/
/* made at this stage. The tree method prepares the data for being;*/
/* printed at the next stage;*/
proc tree data=tree noprint out=out ncl=86;
   /*copy sp0-sp3 sinD0-sinD3 cosD0-cosD3;*/
   copy sp0 sinD0 cosD0;

/* Do a fast clustering analysis on the kept Egypt data;*/
/*proc fastclus data=egypt maxclusters=90 seed=seeds out=out;*/
/* input sp0 sinD0 cosD0;*/

/* Print file for checking;*/
/*data _NULL_;*/
/* set egypt;*/
/* file 'infile.txt';*/

/* put sp0-sp3 sinD0-sinD3 cosD0-cosD3 CLUSNAME;*/
   /*put sp0 sinD0 cosD0;*/

/* Print only the u0, v0 and CLUSNAME column in;*/
/* a file ready for Matlab to read in;*/
data _NULL_;
   set out;
   file 'outfile.txt';

 /* put sp0-sp3 sinD0-sinD3 cosD0-cosD3 CLUSNAME; */
   put sp0 sinD0 cosD0 CLUSNAME;

run;
```

I.2 fastclusTest.sas

```
/* fastclusTest.sas */
/* */
```

```
/* Nicholas Cutler, s000144 */
/* March, 2005 */
/* */
/* Get data from example file for Egypt: Egypt_e33n28.d */
/* Make some clustering analyses using FASTCLUS */

/* The data gives wind speeds and directions and these */
/* must be converted to u and v. Note the directions */
/* are where the wind comes from. */

/* Get data first */
data egypt0;

infile 'Data/Egypt_e33n28_noOut.d' firstobs=6 obs=24840;
/*infile 'Data/Ireland_w08n53.d' firstobs=6 obs=24841;*/

input yr mo day hr sp0 sp1 sp2 sp3 D0 D1 D2 D3;

sinD0 = cos((270 - D0)*3.1415926536/180);
sinD1 = cos((270 - D1)*3.1415926536/180);
sinD2 = cos((270 - D2)*3.1415926536/180);
sinD3 = cos((270 - D3)*3.1415926536/180);
cosD0 = sin((270 - D0)*3.1415926536/180);
cosD1 = sin((270 - D1)*3.1415926536/180);
cosD2 = sin((270 - D2)*3.1415926536/180);
cosD3 = sin((270 - D3)*3.1415926536/180);

proc means data=egypt0;
var sp0-sp3;

/*proc standard std=1 data=egypt0 out=egypt1;*/
/*var sp0 sp1 sp2 sp3;*/

/*proc means data=egypt1;*/
/*var sp0-sp3;*/

data egypt;
set egypt0;
/* Standardise the speeds to mean 0 and stdev;*/
  /*sp0 = (sp0 - 8.0071648) / 3.6113620;*/
  /*sp1 = (sp1 - 5.0504440) / 3.0509798;*/
  /*sp2 = (sp2 - 7.4186093) / 5.0334971;*/
  /*sp3 = (sp3 - 14.8664588)/ 10.5055759;*/

  /* Standardise the speeds to stdev 1;*/
  sp0 = sp0 / 3.6113620;
/* sp1 = sp1 / 3.0509798;*/
/* sp2 = sp2 / 5.0334971;*/
/* sp3 = sp3 / 10.5055759;*/
```

```
  /* Set fixed factor for how speed and direction are related in distance
      ;*/
  sp0 = sp0 * 0.5;
/* sp1 = sp1 * 0.5;*/
/* sp2 = sp2 * 0.5;*/
/* sp3 = sp3 * 0.5;*/

  /* Set weightings between the heights for the wind.*/
/* sp0 = sqrt(25) * sp0;*/
/* sinD0 = sqrt(25) * sinD0;*/
/* cosD0 = sqrt(25) * cosD0;*/
/* sp1 = sqrt(4) * sp1;*/
/* sinD1 = sqrt(4) * sinD1;*/
/* cosD1 = sqrt(4) * cosD1;*/
/* sp2 = sqrt(1) * sp2;*/
/* sinD2 = sqrt(1) * sinD2;*/
/* cosD2 = sqrt(1) * cosD2;*/
/* sp3 = sqrt(1) * sp3;*/
/* sinD3 = sqrt(1) * sinD3;*/
/* cosD3 = sqrt(1) * cosD3;*/


/* Keep only the variables required for the clustering;*/
/*keep sp0-sp3 sinD0-sinD3 cosD0-cosD3;*/
keep sp0 sinD0 cosD0;

/*----------------------------------------*/

/* Get seed points from old classes;*/
data seeds;
  infile 'Matlab/clus86Results/cs_2stgK28.txt';
  /*input u0 v0;*/
  input sp0 D0;
  /*input u0 v0 u1 v1 u2 v2 u3 v3;*/

  D0 = D0*55;
  sinD0 = cos((270-D0)*3.1415926536/180);
  cosD0 = sin((270-D0)*3.1415926536/180);

  keep sp0 sinD0 cosD0;

/* Do a fast clustering analysis on the kept Egypt data;*/
proc fastclus data=egypt maxclusters=90 seed=seeds out=out;
   /*var u0 v0 u1 v1 u2 v2 u3 v3;*/
   /*var u0 v0;*/
   var sp0 sinD0 cosD0;

/* Print only the u0, v0 and CLUSNAME column in;*/
/* a file ready for Matlab to read in;*/
data _NULL_;
```

```
   set out;
   file 'fastclus_outfile.txt';
   /*put u0 v0 u1 v1 u2 v2 u3 v3 CLUSTER;*/
   /*put u0 v0 CLUSTER;*/
   put sp0 sinD0 cosD0 CLUSTER;

run;
```