

Automated CFD analysis for multiple directions of wind flow over terrain

Hervé P. Morvan[†]

School of Civil Engineering, the University of Nottingham, Nottingham, NG7 2RD, UK

Paul Stangroom[‡]

Prospect, 42 Friar Gate, Derby DE1 1DA, UK

Nigel G. Wright^{‡†}

School of Civil Engineering, the University of Nottingham, Nottingham, NG7 2RD, UK

(Received June 26, 2006, Accepted December 21, 2006)

Abstract. Estimations of wind flow over terrain are often needed for applications such as pollutant dispersion, transport safety or wind farm location. Whilst field studies offer very detailed information regarding the wind potential over a small region, the cost of instrumenting a natural fetch alone is prohibitive. Wind tunnels offer one alternative although wind tunnel simulations can suffer from scale effects and high costs as well. Computational Fluid Dynamics (CFD) offers a second alternative which is increasingly seen as a viable one by wind engineers. There are two issues associated with CFD however, that of accuracy of the predictions and set-up and simulation times. This paper aims to address the two issues by demonstrating, by way of an investigation of wind potential for the Askervein Hill, that a good level of accuracy can be obtained with CFD (10% for the speed up ratio) and that it is possible to automate the simulations in order to compute a full wind rose efficiently. The paper shows how a combination of script and session files can be written to drive and automate CFD simulations based on commercial software. It proposes a general methodology for the automation of CFD applied to the computation of wind flow over a region of interest.

Keywords: CFD; automation; wind rose.

1. Introduction

Estimations of wind flow over terrain are often needed for applications such as pollutant dispersion, transport safety or wind farm location. At full-scale a series of masts can be erected and used to collect data over a period of months, preferably long enough to cover the full seasonal

[†] Lecturer, Corresponding Author, E-mail: Herve.Morvan@Nottingham.ac.uk

[‡] Senior Engineer, E-mail: paul.stangroom@prospect-fs.co.uk

^{‡†} Professor, E-mail: nigel.wright@nottingham.ac.uk

variation. However, this is very costly and impractical for most industrial cases, especially if several locations need to be investigated. A quicker and more manageable solution is to carry out a study with a scale model in a wind tunnel, although the construction of a given geometry and the collection of a complete data set may still be time consuming and the results may be compromised by scale effects. A third method, computational simulation (either linear or nonlinear), is increasingly seen as a viable alternative by wind farm developers.

Commercial CFD codes that solve the full nonlinear Navier-Stokes equations are now often used to model wind flow over topography, to produce consistent approximations of the velocity speedups and Reynolds stresses. From an industry or engineering standpoint, accuracy is very important, as quality is imperative especially for the general flow trends. Practicality though, is also important, and if CFD is to be used where wind flow over topography is important, it must be demonstrated that CFD can be used quickly and easily without significant and ongoing user input to the process. This ease of use requires automation. Such considerations have been extremely important for engineers right through the last century and into this one, and can allow non-experts to use advanced methods. However, automation can encourage a “black-box” approach that does not give due regard to the underlying numerical procedures or physical flows. In view of this care must be taken to devise automation systems that respect best practice in CFD whilst balancing the demands of the underlying theory with those of practical design.

This paper presents a methodology for automating the CFD modelling of flow over terrain so that once a site is chosen, several wind rose directions can be modelled and a full wind map can be obtained quickly and easily, without recreating the geometry and physical conditions for each simulation. Although the aim is to minimise user actions, some user input is still necessary at the set-up stage where experience and judgement are extremely important. However the current aim here is to remove user input once the generic simulation case is set up, so that a dedicated machine can complete the necessary simulations automatically and additional wind directions can easily be computed using simple inputs.

The Askervein Hill has been selected as a test case to prove the concept and validate the model. The Askervein Hill project was part of a collaborative study of boundary layer flow over low hills conducted through the International Energy Agency Programme of Research and Development on Wind Energy Conversion Systems. The main field experiments were conducted during September and October 1983 on the Askervein Hill, which is found on the west coast of South Uist, an island towards the southern end of the Outer Hebrides off the North West coast of Scotland. Approximately 50 towers were erected and instrumented for wind measurement during the experiments. Mostly these were simple 10 m masts with cup anemometers, though there were also two 50 m towers, a 30 m tower and a 16 m tower. Thirteen 10 m towers were instrumented solely for turbulence measurement.

Full details of the experimental setups and participants in the project can be found in the main field reports (Taylor and Theunissen 1985): here only the areas relevant for a comparison with a CFD simulation of the hill are considered.

2. Methodology

2.1. CFD

Computational Fluid Dynamics (CFD) originated from two main government-led sectors:

aeronautics and energy, mainly nuclear. Anderson (1996) indicates that advances in the aeronautical industry and the Cold War were the strong drivers that led to the development of CFD in the USA. To this day the use of CFD for aerodynamics calculations in the sectors of aeronautical and automotive industries is very strong and continues to grow. The nuclear sector has also been a strong driver, especially after the 1973 oil crisis. Countries such as the USA, France and Britain developed their own nuclear research centres and dedicated Computational Fluid Dynamics services emerged very early on. Chemical and petro-chemical sectors have contributed to further development in particular in the area of chemical reaction modelling and multiphase flow. CFD has therefore undergone phenomenal development over the past 40 years. Progress has been achieved both in the technical content and the usability of CFD codes. Today all the ingredients are in place to make CFD a successful tool: companies are working towards ease of use, integrated solutions (integration between sketching, CAD, fluid and solid modelling, production under the Computer-Aided Engineering banner), and computer technology is getting cheaper and becoming available via the web (Morvan 2004).

CFD applied to environmental and civil engineering applications such as wind engineering is developing rapidly, for example in sectors such as: architecture, in particular for “one-off” buildings, e.g. the Millennium Dome; urban pollution management and control (Morvan 2004) or wind energy (Pearce and Ziesler 2000), as reported in the present paper. These new applications create a need for innovations and guidelines (Franke, *et al.* 2004).

The general process of CFD applied to wind engineering applications has been reported by several authors and the reader is referred to the existing literature on the subject (Stathopoulos 1997, Castro and Graham 1999, Wright and Easom 2003, Stangroom and Wright 2003). This paper focuses on the automation process to facilitate the use of CFD in these applications.

A commercial code, CFX-5 (ANSYS 2006), is used as part of the method deployed here. CFX-5 is a general CFD code based on the finite volume method and an algebraic multigrid coupled solver. In the present work the CFX high-resolution second order accurate discretisation scheme (ANSYS 2006) is implemented for the advection term. CFX relies on a mesher supplying discrete spatial representations of the geometry, which are then fed into a pre-processor called CFX-Post where the physics settings, such the turbulence model and the boundary conditions, are implemented.

In the simulations reported here the RNG $k-\varepsilon$ turbulence model of Yakhot and Orzag (1986) is implemented. This choice was made following simulations of the Askervein Hill carried out by the authors (Stangroom 2004) for various turbulence models, including Reynolds Stress Models (RSM), which confirmed the findings by Kim and Patel (2000) as well as Jeong, *et al.* (2002) on the suitability of this model for flow over complex terrains, with separation and recirculation, under neutral conditions. The choice of boundaries is reported in more detail in Stangroom and Wright (2003) and Stangroom (2004), but essentially involves a standard rough wall function law for the hill, for which the equivalent sand grain roughness height is computed as (Brutsaert 1982):

$$y_R = 7.5 \times z_0$$

where z_0 is the vertical roughness length, or height (from the ground), at which the zero velocity is displaced. Wieringa (1993) gives typical values of z_0 of about 0.03 m for grass and heather, a value which is consistent with Taylor and Theunissen (1983) and therefore used in the present work. Field data from a reference site located 3 km SSW of the hill, upstream of the predominant wind directions, are used to determine the parameters u^* and z_0 that would best fit the following vertical velocity profile for each flow condition (Stangroom 2004):

$$\frac{U}{u^*} = \frac{1}{\kappa} \ln\left(\frac{z}{z_0}\right)$$

where U is the velocity at elevation z , u^* is the shear velocity, κ is the von Karman constant. This profile is subsequently used as the inlet boundary condition to run the CFD models, together with a turbulence intensities of 5% to 10% used to compute k and ε at the inlet (ANSYS 2006).

Once a solution is obtained in CFX-Solver, CFX-Post can be used to visualize the results. CFX-Pre, CFX-Post, and to a certain extent CFX-Solver, can be parameterized by way of so-called session files (see later); these can be written manually or automatically and run in batch mode to get each of the CFX tools to carry out either repetitive or conditional tasks. For example details of the wind flow boundary conditions can be written or amended “on the go” in CFX-Pre output format as a function of wind measurements taken at a mast (Morvan 2004); more simply one can write a script file that requires n occurrences of a simulation to occur with different settings using one series of meshes and one solver file only, which are amended in turn as necessary. The latter option will be used here to repeat simulations for the full wind rose and rotate the geometry with no user intervention.

PERL is a convenient “language” to use here since CFX products integrate a PERL interpreter which means that it is possible to mix CFX commands within a general coding approach based on PERL. PERL offers the added bonus that it is not compiled but interpreted, which implies that the numerical solution is portable to all operating systems at no cost.

CFX-5 is also capable of using multiple domains which can be connected via non-conformal interfaces, called GGIs or General Grid Interfaces in CFX, which the writers have exploited here to connect the region of interest (Askervein Hill) to a larger domain or fetch where the wind flow is developing. The features described here above are not unique to CFX and can be found in their main competitors as well, albeit under different names and with different languages. The framework presented here is to be viewed as a general solution strategy, highlighting the potential of CFD technology.

2.2. Automation method

The first stage of any CFD simulation is to build and discretise the domain of interest into cells. The terrain data for the lower boundary will normally be in a rectangular grid format, as this is the most commonly available Digital Terrain Model (DTM) type. Once this terrain is incorporated into the model it needs to be rotated within the domain to align with the required wind direction. One method would be to rotate the terrain area to the required angle, and re-form the whole domain with the required set-up for each simulation. This, however, is time consuming from the geometry and mesh generation point of view and also would lead to changes in grid type and grid density in the domain between each simulation.

A better method is to have an identical mesh over the hill for each wind direction and to rotate this to “face” the wind: in essence the location of the wind flow boundary condition remains unchanged, but the hill orientation relative to the inlet is changed. Interestingly, this is similar to the philosophy adopted in wind tunnel modelling. This approach gives a methodology where the grid is constructed, tested and verified only once. Unstructured meshes, based on tetrahedral elements, are used here, so the problem of orientating the mesh with the flow direction which occurs when using structured meshes for example is less of an issue. Mesh sensitivity tests in the regions of interest

could therefore be carried out for a single mesh. This becomes very important when a full wind rose needs to be simulated or when the wind flow to be modelled is not known *a priori*.

Initially, the Askervein terrain tile was placed on a virtual disc, without changing the domain geometry or mesh. Once this was done the disc could be rotated to each of the required wind directions and the modelling process performed for each one. For full wind flow analysis, all twelve wind rose directions would need modelling so that a fuller understanding of how the flow is changing over the hill can be established. Care must be taken in considering the upstream effects for each wind direction as these will change. In the case of Askervein, the majority of wind directions have a simple flat upstream fetch leading out over the sea, though for some with inland fetches there are a series of hills which would interfere with the upstream profile. While it may not be necessary to model all of the upstream topographic components, they must be considered when defining the inlet profile for the relevant simulation and when building the mesh.

With the domain setup consisting of two sections, some method must be incorporated for joining the sections and meshes together. The flow must be conserved across the boundary and the incorporation of any boundary must not affect the flow in any way.

2.3. Grid Interface

When the two domain regions come into contact, an interface must be created which conserves the fluxes across the regional boundary. In other industrial applications, these regions may have differing fluids, so allowing heat to transfer across, maybe parts of the same domain, with different mesh setups (for example, a join between a box section with a hexahedral mesh and a tubular section with a tetrahedral mesh) or maybe in rotation (this was mainly developed for turbo-machinery applications). In the work reported here the commercial code CFX has been used. In view of this the CFX approach is described here, but it is sufficiently generic and similar set-ups are available in other codes. In the CFX software used here the interface is known as a General Grid Interface, or GGI. Initially developed for the turbo machinery industry where fast rotating sections of geometry need to be linked with stationary components in relative motion, their success and reliability is well documented and now leading to their use in many other industry sectors. Tests carried out by the authors have not shown any detrimental effects linked to the use of a GGI interface for similar applications to the one introduced here (Stangroom 2004).

The methodology is based on creating the two separate meshes and then joining them by importing them into a new model. In CFX an intermediate control surface method is used between the meshes at their interface, where the flux is conserved over a 2-D region. Control volumes on each side of the interface communicate with this intermediate 2-D region and the flow is discretised in terms of fluxes across the control volume faces. This process is detailed in Fig. 1. The incoming flux F_R on the right hand side (R) across the shaded interface is equal to the weighted sum of the fluxes coming from the left hand side (L) from overlapping control volume interfaces, i.e.,

$$F_R = A_{R1}/A_1 \cdot F_{L1} + F_{L2} + A_{R3}/A_3 \cdot F_{L3}.$$

Here F_R is the flux contribution to the right mesh from the left mesh; A_1 , A_2 and A_3 are cell interface areas for the cells on the left hand side contributing to the flux on the right hand side (assuming a flux from left to right); A_{R1} , A_{R2} and A_{R3} are the fractions of cell areas A_1 , A_2 and A_3 which map onto the cell area on the right hand side for which the flux F_R is computed; the F_{Li}

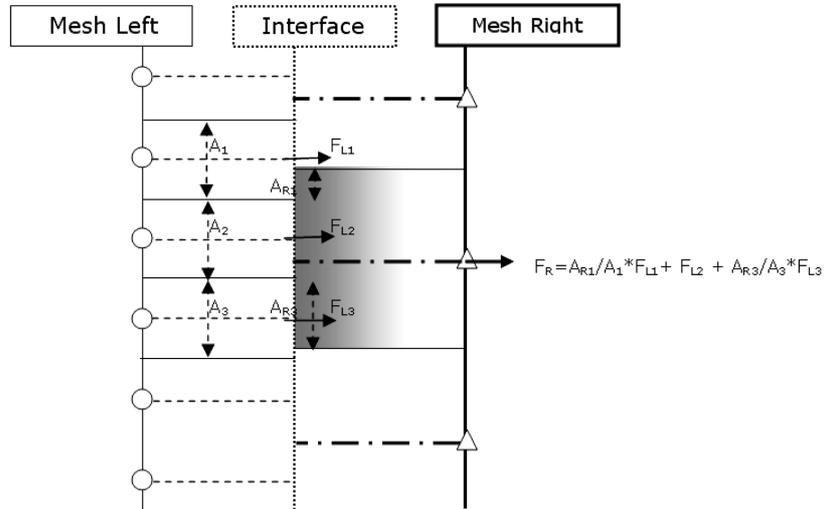


Fig. 1 Interface Principle

terms, $i = \{1, 2, 3\}$, are the corresponding left hand side fluxes which are proportionally transferred to the right cell.

The control volumes on the right hand side, linked to the control surface, will receive the flux from the left hand side in proportion to the amount of area of the control surface which they occupy. In turn, those controls volumes on the left hand side will receive the flux from the right hand side relevant to the amount of the control surface which they occupy. As all values are known at the nodes for the incoming fluxes, and a linear relationship exists across the control surface, this is a simple algebraic problem. The interface acts as some sort of internal boundary condition.

The GGI in this case allows a mesh to be created for the wider domain area (which will not be as dense as the mesh over the hill), and a second mesh to be created for the disc shape, which can be rotated to the required wind direction governed by the wider domain. Fig. 2 shows the domain setup. Effectively the virtual disc becomes a rotating cylinder with its own mesh.

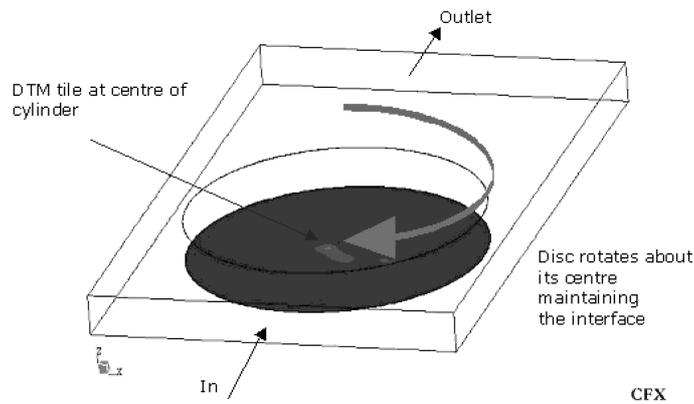


Fig. 2 Askervein setup for automation process

With this set-up, a series of script files can be created by the expert user, which:

- Rotate the cylinder to the required location,
- Update the definition file for the CFD run, as necessary, e.g., wind intensity and wind profile,
- Start and run the solver (produce a result file),
- Start the post-processor.

Once the results file is created, a postprocessor file, a so-called “session” file, can retrieve all the necessary data from the results, which can be left ready for analysis upon completion of each of the simulations. The cylinder is then rotated to the next direction, and the whole process continues. If the twelve wind rose directions are being monitored for example, a single session file can rotate the cylinder 30° each time, and so the set-up is even simpler. A batch file controls the whole process and can easily be expanded to include the commands to execute the solver and post-process the results (Morvan 2004).

Thus, once the user has set-up the batch file (including all session files and has created and tested a suitable mesh), no further input is necessary until the results are ready to be examined.

One of the aims of this section of work is to obtain information about the full wind map over Askervein. The wind rose obtained from the Met. Office (Stangroom 2004) gives the data for twelve wind rose directions. Seven of those are being modelled here, so a reasonably accurate wind map can be obtained. As these are the most prominent wind directions over the hill, and account for 70% of the wind flow, the results can be used to deduce reasonably accurate yearly mean values, which are of use to developers.

3. Askervein Hill

The Askervein Hill is 116 m high (126 m above sea level) and has essentially an elliptical shape, Fig. 3. The hill has a 2-km long major axis and a 1-km minor axis; the major axis is oriented along

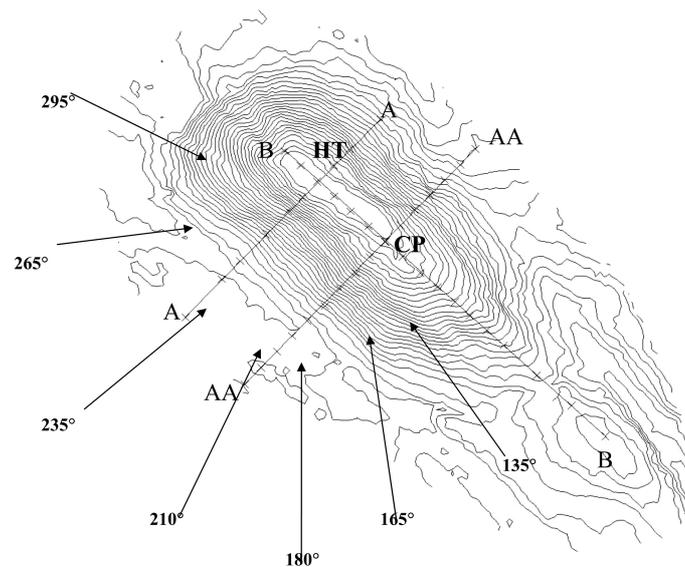


Fig. 3 Wind directions over Askervein (HT: Hill Top; CP: Central Point)

a generally NW-SE line. The distance between HT (Hill Top) and CP (Central Point) is short of 500 m. Long grass and heather covers the hill and a flat uniform fetch of 3-4 km lies to the SW with a similar roughness configuration before joining the sea.

The wind directions important to the Askervein Hill project are given in Stangroom (2004) and have been used for the automated project here. It is not appropriate to use all twelve wind rose directions due to the location of other hills in the area which will affect the incoming flow. The seven wind directions to be used here are all clear of interference from topography changes, and allow comparison with full scale and wind tunnel data. Fig. 3 shows a contour plot of the hill, indicating the wind directions.

Mesh independence is not explicitly presented in this paper, but discussion on this issue is available in associated publications relating to the same study case (Stangroom and Wright 2003, Stangroom 2004). The simulations presented here were run with a median mesh size short of 250,000 nodes, corresponding to a mean resolution of 1.25 to 1.5 m (Stangroom 2004). An inflation layer of prisms is placed in the boundary layer to allow for a much finer resolution of the flow close to the walls and curvature dependent local mesh resolution is also implemented in CFX for the hill. Tests were carried out for a simple cosine hill (Stangroom 2004) to determine the adequate mesh configuration and ensure that velocity profiles and speed-up ratios were relatively independent at the resolution used here. Whilst the speed-up ratios obtained in the present case are also satisfactory and sufficient to enable the authors to demonstrate the benefits of an automated CFD process, it is clear that it would have been desirable to use finer grids, although this was not possible at the time of the study; one particular constraint to bear in mind is that a grid interface uses up an increased amount of memory (approximately 30% with CFX-5) and so the computer limitations are greater, although this should not be a long lasting issue.

4. Set-up process

The domain size, including cylinder and surrounding box, is $10 \times 10 \text{ km}^2$ and 1km in elevation, with the hill towards the front end of the domain area inside a cylinder, Fig. 2. This arrangement is backed up by the literature (Taylor and Thenunissen 1985, Parkinson 1987) and the blockage ratio thus obtained, 2%, is within the 3% bound identified in Baetke and Werner (1990). The base disc itself is approximately 6 km in diameter and the size of the box therefore positions the boundaries at a suitable distance from the hill.

The setup process can be seen in Fig. 2 with the rotation and setup stages clearly visible. The cylinder and box are used solely to create the meshes, so only one box and cylinder are ever needed. All the boundary conditions, domain settings, flow variables, and solver considerations are performed in section 2 of the loop, Fig. 4, when creating the problem definition for the simulation.

For the purpose of automating the execution of the various simulations, a batch file is created; the terminology defined hereafter is key. This batch file is a series of script lines executing various actions, such as running the solver or postprocessor for example, using a suitable series of input files to implement suitable parameters, and calling different session files to carry out specific tasks in the process. A session file is in itself a series of commands that can be interpreted by each component of the CFD package and can lead to the automatic execution of a pre-recorded task by that software component. For example session files can be executed for CFX-Pre and CFX-Post quite easily, which increases productivity for repetitive tasks. The input files mentioned here above contain information on the meshes, the physics of the flow to be modelled, the boundary condition files or the inlet velocities

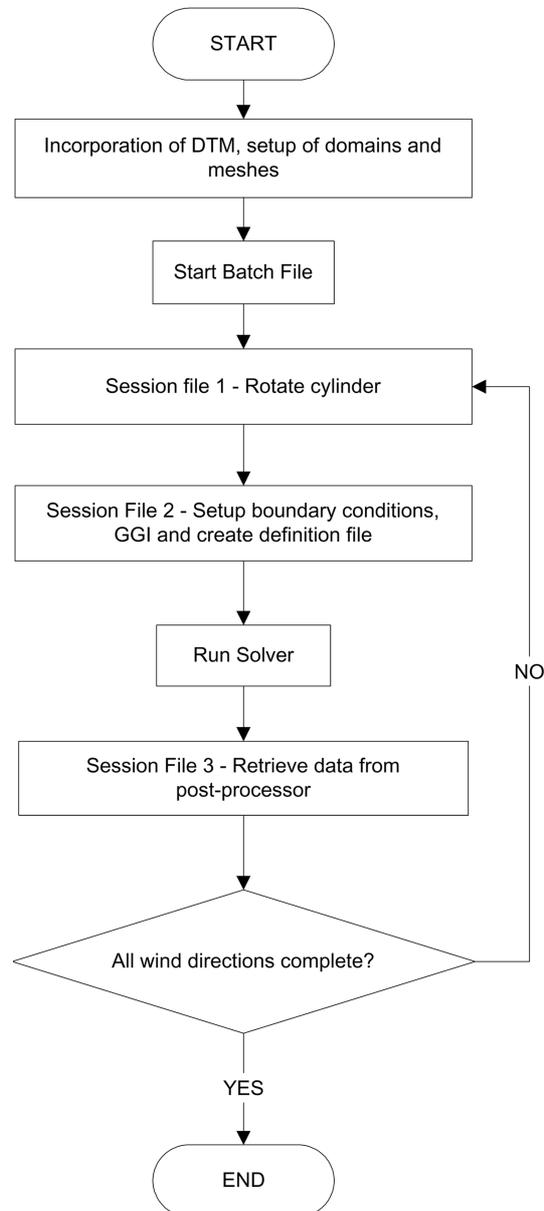


Fig. 4 Flow chart of the automation process

for example. They are usually the products of the execution of manual commands in session files by the pre-processor; this is the case for the problem definition file fed to the solver for example, which is originally written as a result of the cylinder and box assembly being produced and defined in CFX-Pre. Automating the process is therefore a two-fold task: (1) one needs to create generic session files, e.g. to rotate the cylinder inside the box or to alter the wind speed in the pre-processor or create pre-defined pictures in the post-processor and (2) one needs to write script commands to utilise these

sessions in turn and start CFX executables with the right combination of input and session files.

Firstly a session file, called session file 1, is written that assembles the cylinder (containing the hill) and the box meshes at a set angle to create the suitable problem definition files. Session file 1 is called by a script command that executes it for any one value of the angle belonging to the list {135°; 165°; 180°; 210°; 235°; 265°; 295°} as listed in Table 2. Whenever this “rotation” session file is executed via a script line a new problem definition file is created. Each of these definition files is completed by the addition of boundary conditions keywords to incorporate the matching velocity options and values at the inlet, thanks to the execution of a second session file. Script commands then call the CFD solver executable to solve the numerical problem that has been automatically set up. Finally after each individual simulation is run the post-processor is started in the background and executes session file 3, leading to the creation of pictures, graphs and other numerical outputs as requested by the user. This series of script commands is concatenated and repeated into a large batch executable which the user starts via a command line so that each of the seven wind directions and intensities are set up, run and post-processed automatically. Should any one solution fail, e.g. one wind direction does not converge, the batch process jumps to the next script line to execute it and run the next simulation. The overall process is schematised in Fig. 4.

5. Results

Results are presented for each of the seven wind directions. Comparisons for wind speedups over the hill are given for the lines AA and B, Fig. 4. The wind speed up is defined as the ratio between the wind speed at a given point on the site and the reference site velocity, U_{RS} :

$$\text{SpeedUp}_{\text{Location X}} = \frac{U_{\text{Location X}}}{U_{RS}}$$

The results produced are generally in agreement with the full scale measurements. RMS and percentage error values of the results are given in Tables 1 and 2 showing the accuracy of the CFD simulations compared to the wind tunnel and full scale data. If a more powerful machine than the

Table 1 RMS errors between data sets

Wind Direction	RMS Errors	
	CFD vs. Full Scale	Wind Tunnel vs. Full Scale
135°	0.036	
165°	0.141	0.141
180°	0.115	
210°	0.130	0.053
235°	0.189	0.095
265°	0.151	0.064*
295°	0.318	0.104*
Average	0.163	0.096

*second field data set compared, not wind tunnel data. So the wind tunnel label here refers to the second set of field survey data. Errors between Wind Tunnel and Full Scale do not include errors between the two field data sets, only the Wind Tunnel errors.

Table 2 Percentage errors between data sets

Wind Direction	Percentage Errors	
	CFD vs. Full Scale	Wind Tunnel vs. Full Scale
135°	6.65	
165°	9.32	8.9
180°	7.32	
210°	8.57	6.27
235°	15.3	6.75
265°	11.68	4.78*
295°	26.18	7.33*
Average	12.15 (9.81**)	7.31

*second field data set compared, not wind tunnel data. So the wind tunnel label here refers to the second set of field survey data. Errors between Wind Tunnel and Full Scale does not include errors between the two field data sets, only the Wind Tunnel errors.

**average values in brackets do not include the 295° wind direction.

desktop PC used here were dedicated to the process, finer meshes could be run and improved accuracy would be expected.

5.1. Speed-up ratios and errors

Figs. 5 to 11 show velocity speed-up ratios of the wind at the 10 m towers along lines AA and B, Fig. 3, for each of the seven wind directions. Comparisons along line A are not reported here since the data for lines A and AA are similar (Stangroom 2004).

With the wind approaching from 135°, Fig. 5, the flow is effectively normal to line AA and parallel to B, Fig. 3. The predictions along line B are generally good, with the CFD slightly underpredicting the flow pattern and not showing the more dramatic changes in speed-up ratio. Some discrepancy is noted between HT (Hill Top) and CP (Central Point), Table 3, where the two data sets fluctuate slightly. Along line AA the CFD predicts a much smoother set of speed-ups, which compare reasonably well with the full scale data at the hill top, but less well on each of the slopes. With the wind coming from so wide an angle, this wind direction is the only one of the seven modelled here which may be affected by the location of other hills in the region, which could account for the differences in values. The percentage error between the data sets is only 6.65% though and so this should be viewed as successful.

With the flow direction at 165°, Fig. 6, wind tunnel data is available for comparison alongside the full scale data. For both lines AA and B, the wind tunnel and CFD data are very closely matched, with the percentage error being just 4.64%. Again the full scale data values between HT and CP show some significant change along the crest of the hill which neither the CFD nor wind tunnel experience in such magnitude. The CFD underpredicts the full scale data on the upwind slopes, but improves on the lee side. Clear underprediction can be seen for the values along line B.

With the wind coming from the south, Fig. 7, excellent comparisons are found on the upwind slope of the hill between the CFD and full scale data on line AA. The CFD over predicts the full wind speed at the top of the hill, though there is a lack of full scale data at this point as the mast

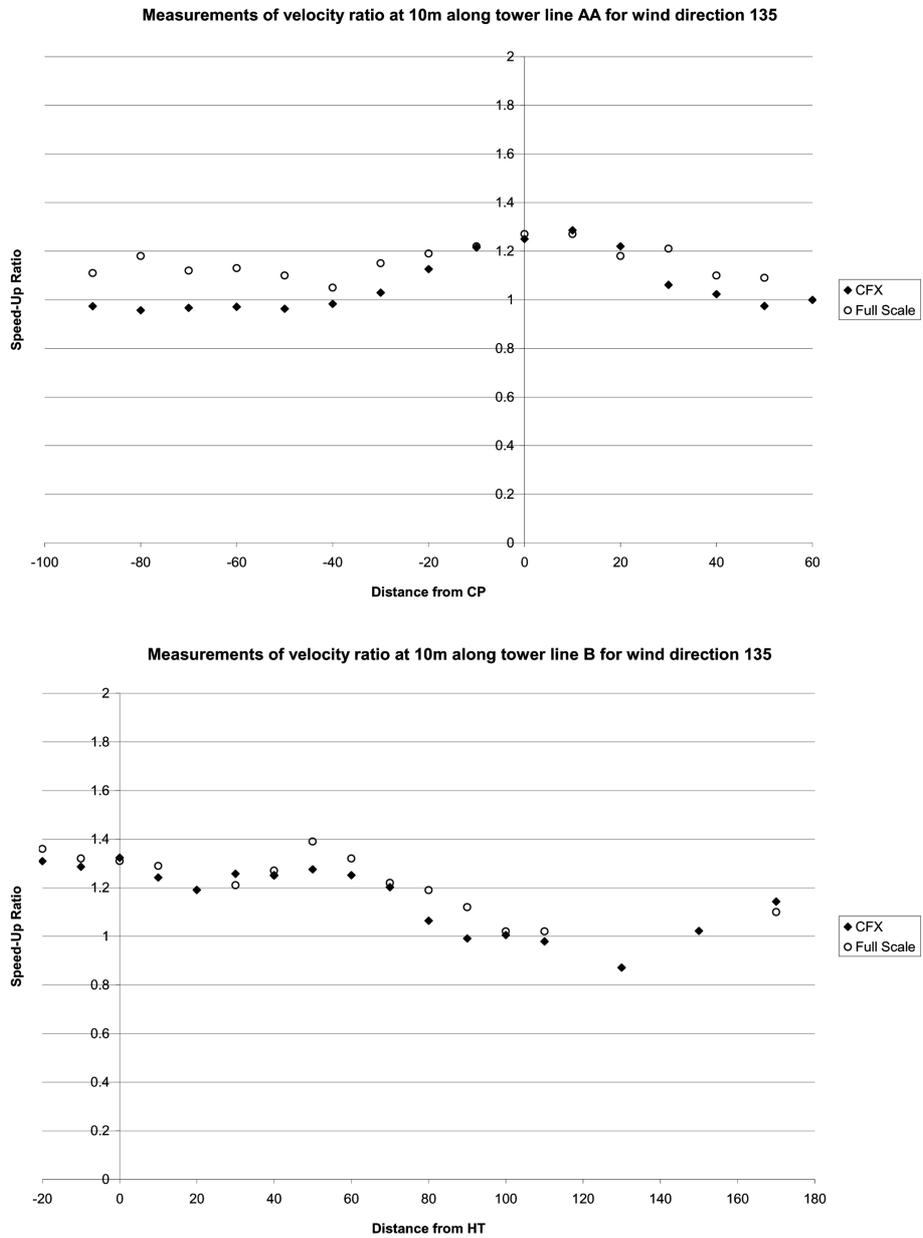


Fig. 5 Comparison between CFD simulation and full scale measurements for wind direction 135°

Table 3 Yearly mean values

Location	Yearly Mean Value (10 m)
HT	12.65 m/s
CP	12.05 m/s

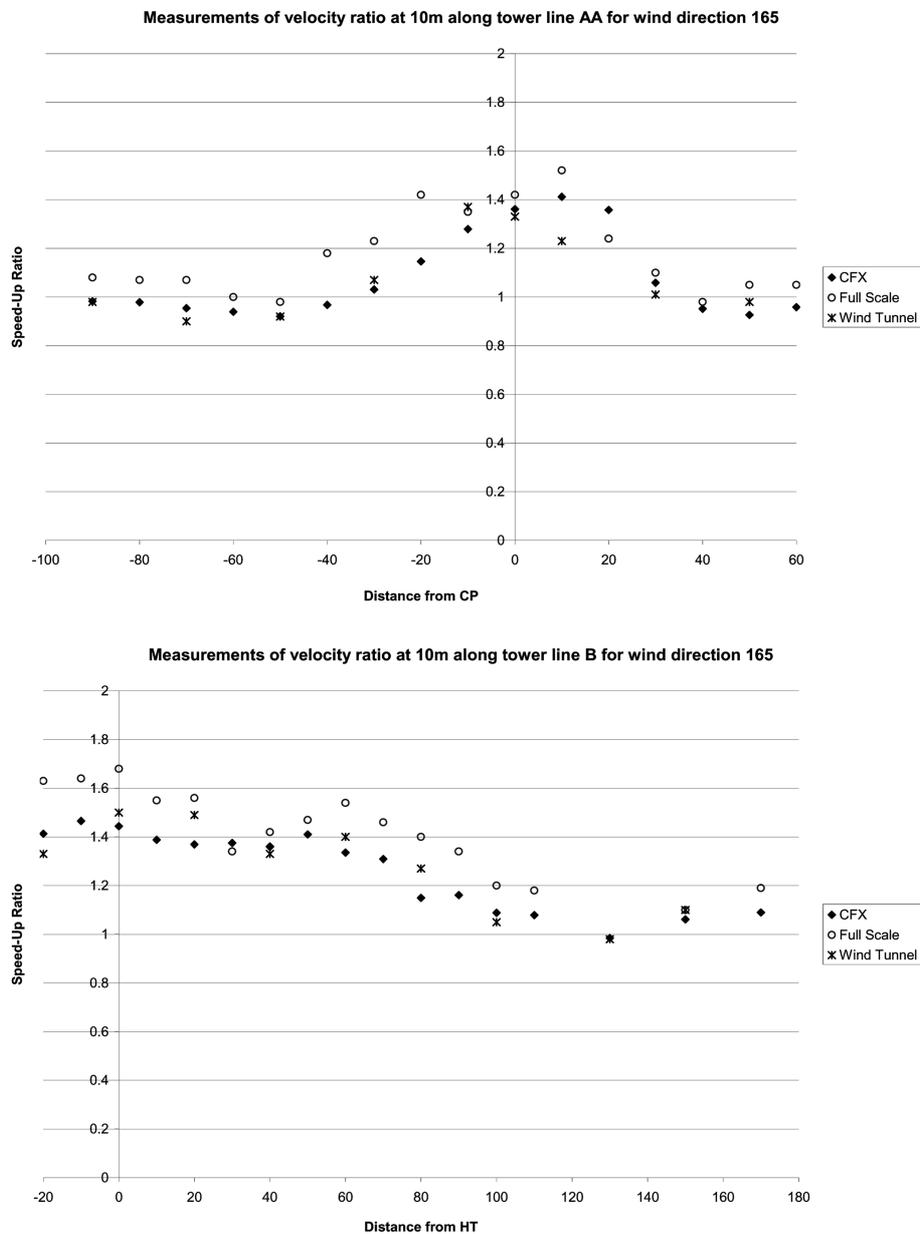


Fig. 6 Comparison between CFD simulation and full scale measurements for wind direction 165°

AANE10 has no reading for that set. Along line B, for which the flow is close to normal, the CFD predicts well the general trend of velocity speed-ups, with a 7.32% error on average, though this time anomalies along the hill crests are seen in both data sets. The full scale predicts higher than the CFD around CP, but further along the crest at HT, the data sets have inverted.

At 210°, Fig. 8, the wind is almost parallel to AA and normal to line B. Excellent predictions are again noticed along line AA where the CFD and wind tunnel results both agree well with the full

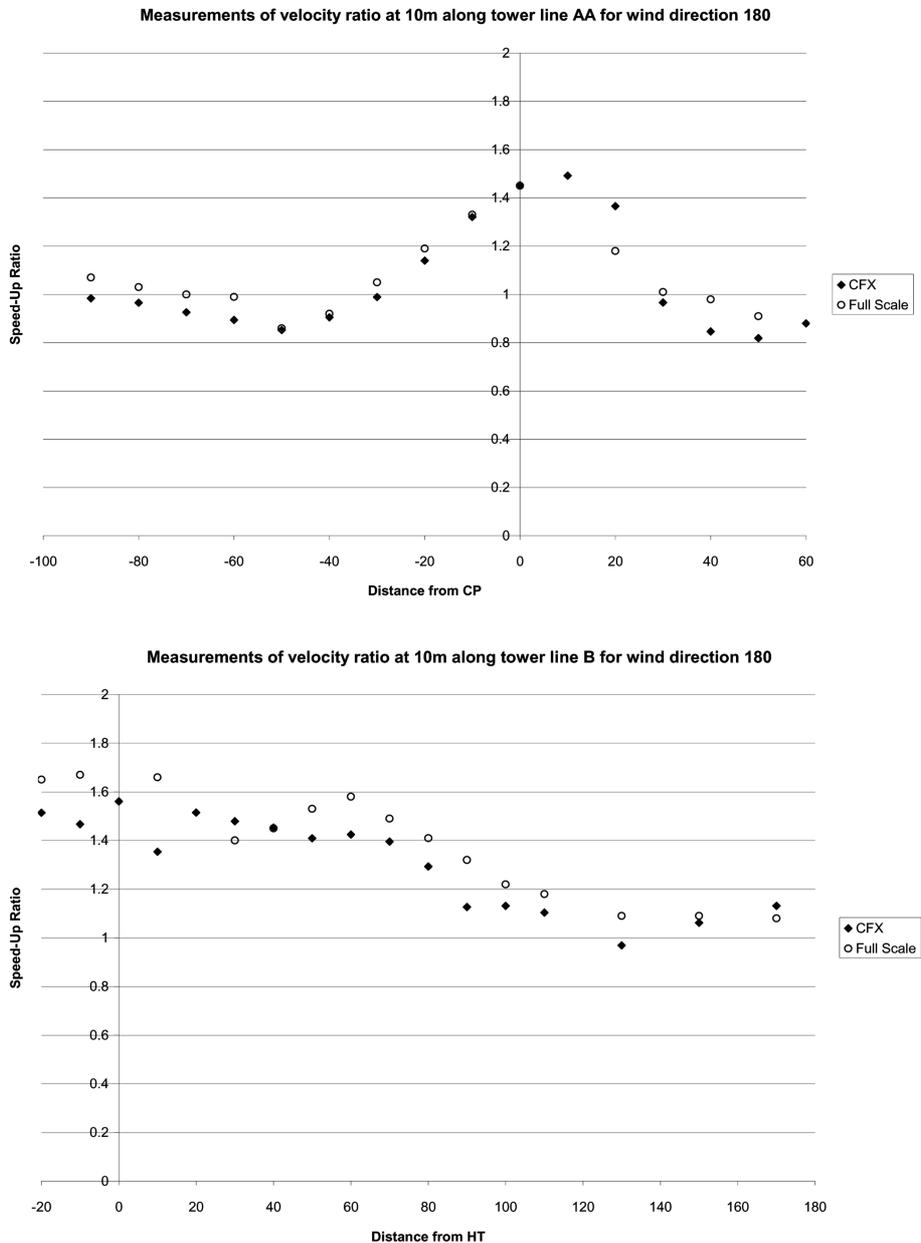


Fig. 7 Comparison between CFD simulation and full scale measurements for wind direction 180°

scale data. In this case, the CFD slightly underpredicts the speed-up ratio for the top of the hill, but stays within 8.6% of the full scale values on average. Along line B the wind tunnel has better agreement than the CFD results with the full scale, though again the discrepancies along the hill crest are present. The CFD predicts a generally smoother velocity speedup, though this is understandable as the CFD simulation is set-up as a simplified flow situation with few boundary layer effects presents and a less complicated environment.

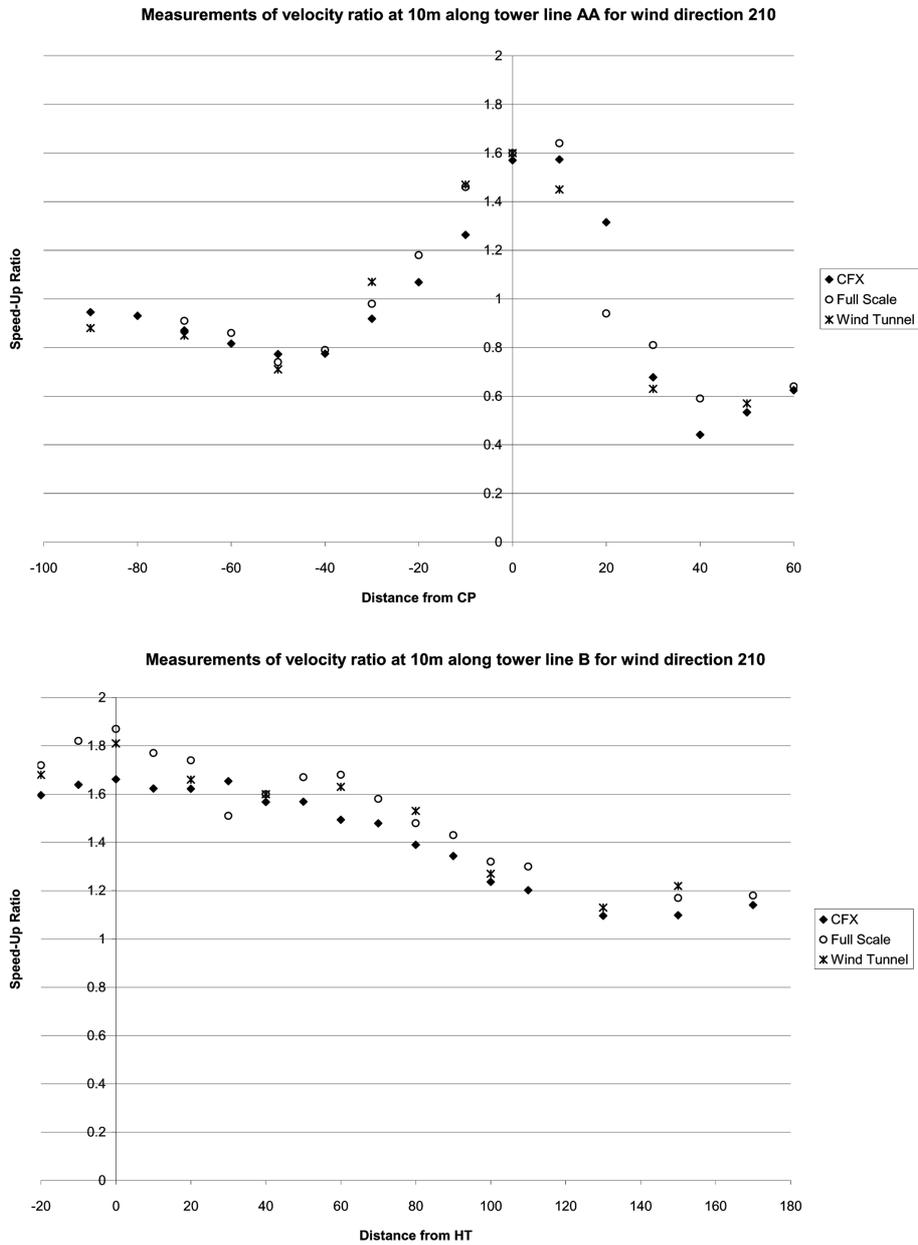


Fig. 8 Comparison between CFD simulation and full scale measurements for wind direction 210°

At 235°, Fig. 9, again the flow is normal to B and parallel to AA. Excellent predictions are found on the upwind slope of the hill for both the wind tunnel and CFD. On the lee slope for line AA, both the full scale and wind tunnel show a steep drop off in the velocity, which the CFD is unable to capture with its current setup. The steep drop off is found, but it is predicted to occur slightly further down the slope than was found during the experiments. It is from this direction that flow separation is most likely to occur and what was indeed noticed by the participants during the field survey, and

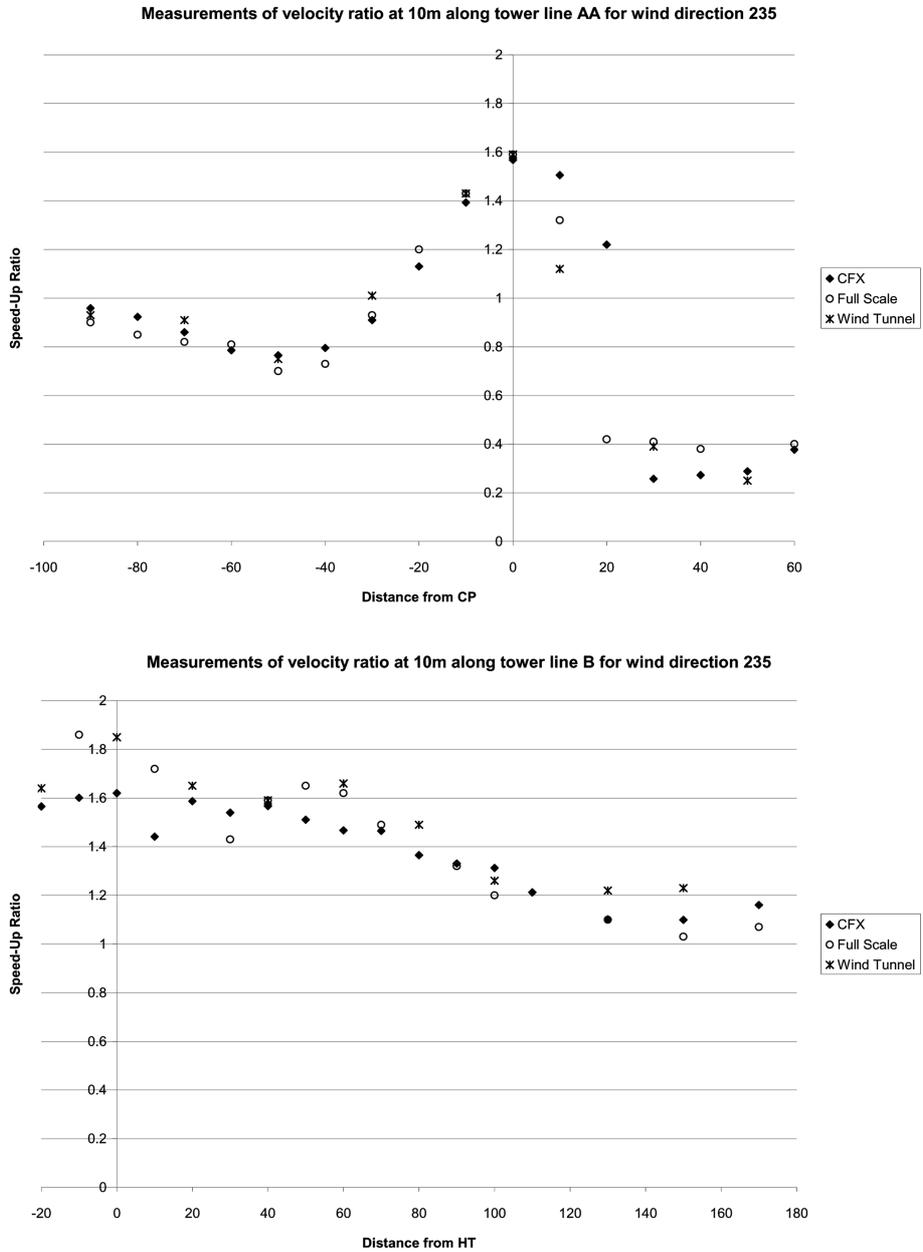


Fig. 9 Comparison between CFD simulation and full scale measurements for wind direction 235°

this goes some way to explaining the steep change in velocity, and some of the velocity differences along the hill crest. Line B shows the CFD predicting well on the lower slopes of the hill (right hand side of the graph) but again shows discrepancies between all three data sets along the hill crest.

With the wind coming from 265°, Fig. 10, almost due west, the graphs show the CFD compared to two different field survey results, which have the wind coming from 263° and 268° respectively. These slight changes in wind direction should not have a large effect on the flow over the hill, and

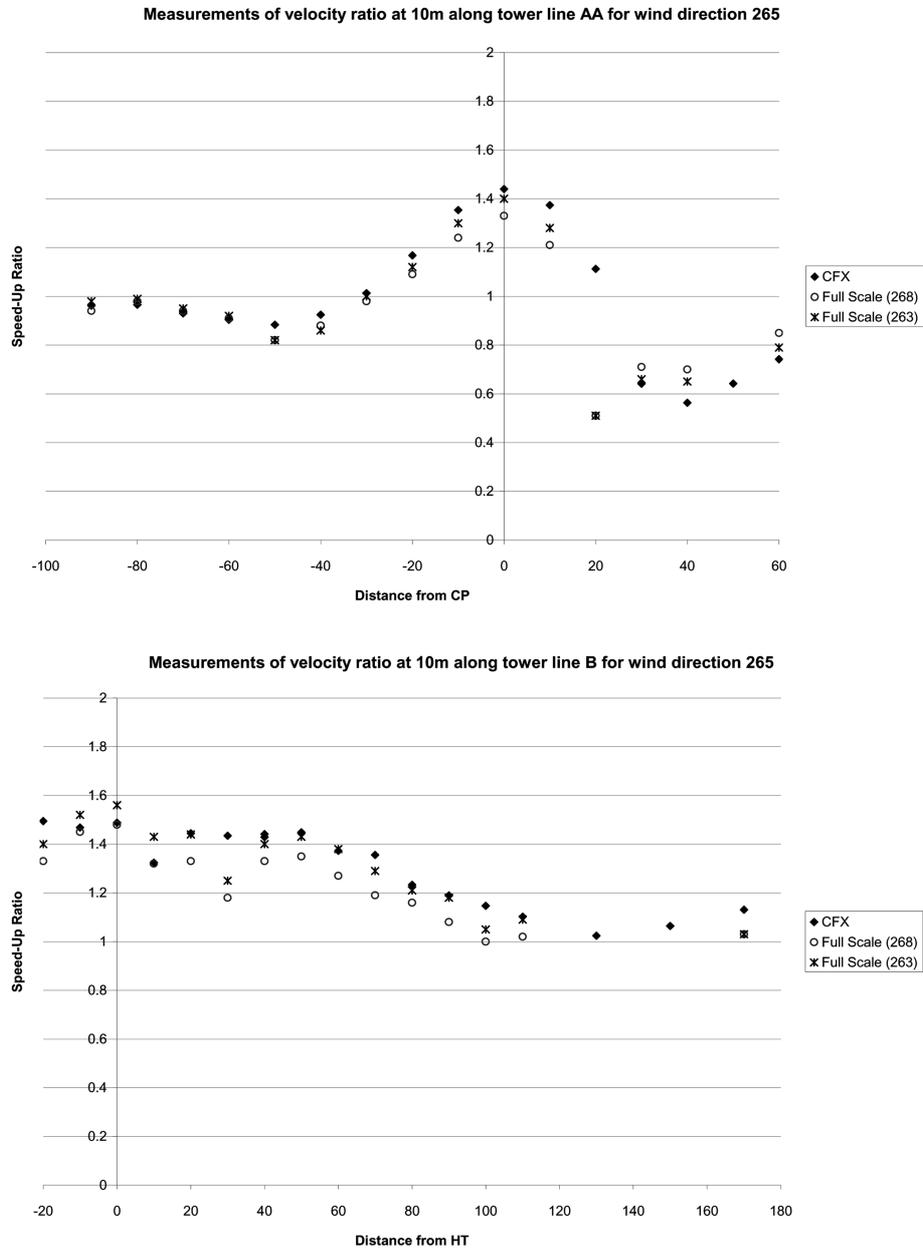


Fig. 10 Comparison between CFD simulation and full scale measurements for wind direction 265°

indeed on the upwind slope, both field data sets and the CFD are very close in their predictions. On the lee slope (along line AA) both full scale data sets show extreme drops in wind speed, which is not predicted by the CFD. Numerical models in general are known not to predict large changes in velocity, and again this could be a flow separation bubble which is found in the full scale, but not well predicted by the numerical model.

Results along line B are more interesting as the two field data sets are well separated, showing

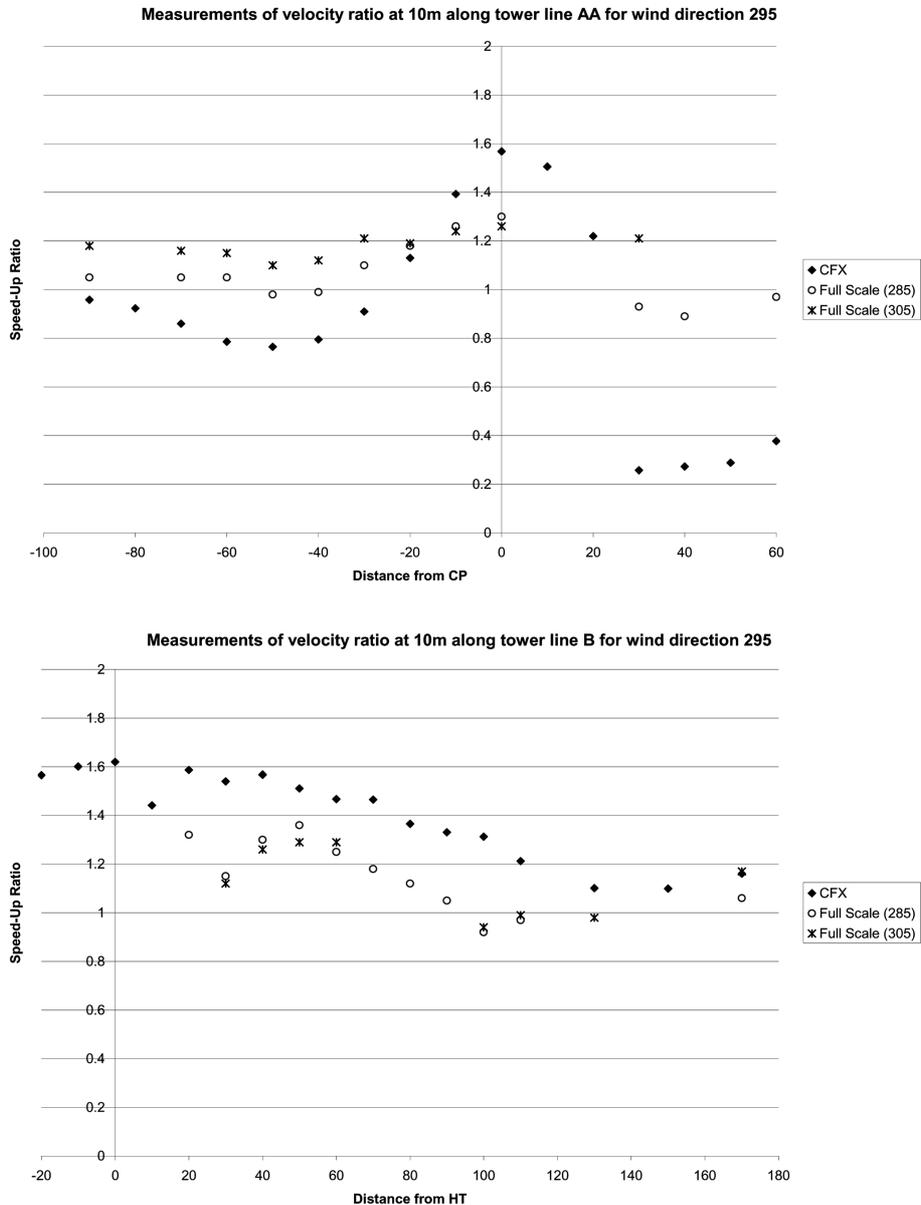


Fig. 11 Comparison between CFD simulation and full scale measurements for wind direction 295°

that the field results can change dramatically depending on the time of day of the survey with the weather conditions at that time. The CFD predictions are much closer to the 263° data set than to the 268°. Again there are discrepancies along the hill crest, but the interesting point occurring from these graphs is the differences between the full scale data sets which are on average, 8.43% apart. The CFD is within 5% of one, yet almost 12% from the second.

With the wind coming from 295°, Fig. 11, almost parallel to line B, normal to line AA, and almost opposite to 135°, the CFD results are again compared to two field survey data sets, from

285° and 305°. Along line B the field data sets are very close to each other, and the CFD dramatically over predicts the speed-up ratios. Even the general trend of results is not well picked up, as the field results show a sharp drop just before the wind reaches CP, which is barely noticeable in the CFD data, though a drop is noticed just after the wind leaves HT, for which no field data is available. It is possible then that the CFD predicts this drop to occur 20 m earlier than found at full scale, but without further full scale data, it is impossible to say.

Along line AA, the two field data sets are even less well matched though neither shows the large changes in speed-up ratio predicted by the CFD. Indeed this is by far the worst dataset, with the CFD predicting speedup ratios as low as 0.25 and none of the field data falls below 0.9. There is a lack of field data, but this still cannot explain the CFD result.

The consistent swapping over of trend lines along the hill crests between HT and CP is unexpected. It occurs for all seven wind directions and could be due to a number of factors including topography effects. The DTM only has data points every 50 m and there may well be some other topography not captured by the DTM which is affecting the flow.

5.2. Yearly mean values

Table 3 shows yearly predicted values of velocity at Hill Top (HT) and Central Point (CP). The wind rose obtained for Benbecula, the nearest Met. Office measuring station, showed the prominent wind directions and speeds (given as percentages per speed intervals; e.g. 5.4% of the Southward wind was in the interval 1 to 10 knots for the period January 1986 to December 1995, 3.2% in the interval 11 to 16 knots etc.) over the Askervein Hill, and so a yearly mean is calculated based on these intervals. All twelve directions could be computed, albeit with some adjustments due to the presence of neighbouring hills, if required, but this gives a very good indication of the values available and the type of results that can be obtained with this approach. Very quickly one can identify, within a reasonable level of accuracy, whether the site is suitable for a wind farm. Full yearly mean profiles could be determined if necessary to allow various turbine heights to be considered. This again demonstrates the advantages of using CFD.

Wind roses are available for a large number of areas of the country, and yearly average values and a full wind map of a region could be created using CFD. In full scale experiments, measuring masts would have to be erected and monitored for the full year to create the same data, which would be expensive and time consuming. CFD could therefore provide valuable information to planners, engineers and architects about the wind conditions of a region at relatively low cost.

5.3. CPU time

The time taken for the process on a dual processor PC (2 Pentium III 1 GHz chips) equipped with 1 GB of RAM was approximately 24 hours for set-up and simulation of each wind direction, so in this case, for seven wind directions, a total of one week dedicated computer time. The grid interface increases the number of iterations required to reach steady state and in most cases this number was around 230 per simulation, which explains the extra computation time relative to previous work done by the authors without recourse to the use of a grid interface (Stangroom 2004). Newer machines and alternative OS would be significantly faster.

The time necessary to create the generic files necessary to run this job is more difficult to measure. It is a function of the complexity of the terrain map, flow conditions and experience of the

user with CFD and when dealing with the automation of the CFD software using external scripts. For the present work and because of the experience of the authors with the software the process can be detailed as follows:

- Incorporation of DTM
- Pre-processing of model, including initial simulation to confirm domain and mesh suitability
- Set-up of batch file and creation of all necessary session files

could be completed in about a week, allowing for errors and problem solving.

5.4. Computational requirements

Care must be given to ensure that enough computational power and memory is available to cope with the grid interface within the domain and as a ‘rule of thumb’ the following points should be noted if the process is to be undertaken:

- The geometry surfaces either side of the grid interface must match identically.
- File names and locations should be obvious with obvious locations to simplify the batch file creation (e.g. relative paths etc.), and to ease problem solving if any errors are found.
- While no user input is required during the simulation process, it is useful to check on the solvers regularly to ensure that the residuals are decreasing. In any case this verification should be carried out at the end of the simulations before exploiting the numerical results.

6. Conclusions

The automation of the simulation process has been successfully completed and offers a means of more straightforwardly applying CFD for wind flows with varying direction. This is likely to be of use in wind power and pedestrian-level wind environment analysis.

Advantages have been seen over field surveys and wind tunnel experiments. The versatility of numerical models is highlighted in the manipulation of the geometry and the retrieval of data. While the wind tunnel geometry can also be manipulated, it is much more arduous to obtain such amounts of data so quickly, and full scale experiments would require significant lengths of time spent on site, over several seasons.

Within a relatively short time period, yearly mean values have been deduced with a model that is accurate (on average) to about 10% for the speed up ratio computed here for wind potential studies. Improved meshes would lead to improvements in accuracy. Further data could be produced as necessary with items such as local wind roses easily deduced, based on wind roses of the surrounding areas.

CFD is a useful tool to aid the wind analysis process. Apparent anomalies (e.g. local recirculation, wind reduction or acceleration) that are encountered during a CFD simulation can then be tested either at full scale (on-site) or in a wind tunnel, to investigate their causes.

Acknowledgements

The authors would like to acknowledge the support of Powergen Renewables and the Engineering and Physical Sciences Research Council.

References

- Anderson, J. D. (2006), *CFD: The Basics with Applications*, McGraw Hill.
- ANSYS, www.ansys.com/cfx.
- Baetke, F. and Werner, H. (1990), "Numerical simulation of turbulent flow over surface-mounted obstacles with sharp edges and corners", *J. Wind Eng. Ind. Aerodyn.*, **35**, 129-147.
- Brutsaert, W. H. (1982), *Evaporation in the Atmosphere*, D. Reidel publisher.
- Castro, I. P. and Graham, J. M. R. (1999), "Numerical wind engineering: the way ahead?", *Proc. Instn Civ Engrs Structs & Bldgs*, **134**, 275-277.
- J. Franke, C. Hirsch, A. G. Jensen, H. W. Krüs, M. Schatzmann, P. S. Westbury and S. D. Miles, J. A. Wisse, N. G. Wright (2004), *Recommendations on the Use of CFD in Wind Engineering*, COST Action C14, Impact of Wind and Storm on City Life and Urban Environment, Brussels.
- Jeong, U. Y., H. M. Koh and H. S. Lee (2002), "Finite element formulation for the analysis of turbulent wind flow passing bluff structures using the RNG $k-\epsilon$ model", *J. Wind Eng. Ind. Aerodyn.*, **90**, 151-169.
- Kim, H. G. and V. C. Patel (2000), "Tests of turbulence models for wind flow over terrain with separation and recirculation", *Boundary Layer Meteorology*, **94**, 5-21.
- Morvan, H. P. (2004), "Automating CFD for non-experts", *J. Hydroinformatics*, **7**(1), 17-29.
- Parkinson, H. G. (1987), "Measurements of Wind Flow Over Models of a Hill", Oxford, Ph D Dissertation University of Oxford.
- Pearce, D. L. and C. D. Ziesler (2000), "Using CFD modelling to improve the estimation of offshore wind resource", *IMechE Seminar - Power Generation by Renewables*, 207-219.
- Stangroom, P. (2004), "CFD Modelling of Wind Flow over Terrain", Nottingham, Ph D Dissertation University of Nottingham.
- Stangroom, P. and N. G. Wright (2003), "CFD modelling of the Askervein Hill", *International Conference on Wind Engineering, Lubbock, Texas*.
- Stathopoulos, T. (1997), "Computational wind engineering: Past achievements and future challenges", *J. Wind Eng. Ind. Aerodyn.*, **68**, 509-532.
- Taylor, P. A. and Teunissen, H. W. (1987), "The Askervein Hill project: Overview and background data", *Boundary Layer Meteorology*, **39**, 15-39.
- Taylor, P. A. and Teunissen, H. W. (1985), *The Askervein Hill Project: Report on the Sept./Oct. 1983, Main Field Experiment*, Ontario, Canada, Meteorological Services Research Branch.
- Taylor, P. A. and Teunissen, H. W. (1983), *The Askervein Hill Project: Report on the Sept./Oct. 1982, Experiment to Study Boundary Layer Flow over Askervein, South Uist, Ontario, Canada*, Meteorological Services Research Branch.
- Wieringa, J. (1993), "Representative roughness parameters for homogeneous terrains", *Boundary Layer Meteorology*, **63**, 323-364.
- Wright, N. G. and G. J. Easom (2003), "Non-linear $k-\epsilon$ turbulence model results for flow over a building at full-scale", *Applied Mathematical Modelling*, **27**(12), 1013-1033.
- Yakhot, V. and Orzag, S. A. (1986), "Renormalization group analysis of turbulence: Basic theory", *J. Science and Computing*, **1**, 3-51.