33rd Annual General Conference of the Canadian Society for Civil Engineering
33e Congrès général annuel de la Société canadienne de génie civil

Toronto, Ontario, Canada
June 2-4, 2005 / *2-4 juin 2005*

GC-275

NUMERICAL EVALUATION OF WIND FLOW OVER COMPLEX TERRAIN USING AN OBJECT-ORIENTED APPROACH

G.T. Bitsuamlak[1], C. Bédard[2] and T. Stathopoulos[3]
1. RWDI Inc., Guelph, Ont, Canada
2. ETS – École de technologie superieure, Montreal, QC, Canada
3. Centre for Building Studies, Concordia University, Montreal, QC, Canada

ABSTRACT: The paper discusses the application of numerical methods to characterize turbulent wind flow over complex terrain. For this purpose, a CFD tool has been developed on the basis of finite-volume formulation using object-oriented approach and C++ programming language. This approach was chosen considering its better management, extensibility, reusability and ease to parallel programming tasks. The object-oriented features of inheritance, overloading, dynamic memory allocation, and objects have been used. Objects such as PhysicalDomain, ComputationalDomain, NeighbouringControlVolume, ControlVolume, and GridPoints have been designed based on physical relationships while following a finite volume formulation. The CFD procedure consists of solving the two-dimensional Reynolds time-averaged Navier-Stokes equations and the standard k-ε turbulence model closure using a body-fitted nearly orthogonal coordinate system. A robust grid generator tool suitable for producing nearly orthogonal grids over curved and sloping complex surfaces has also been developed using object-oriented approach. Subsequently, wind flow over complex terrain has been simulated and wind load design parameters such as speed-up ratio values have been generated for a wide spectrum of terrain geometries. Geometric configurations include single and multiple hills, as well as complex terrain. CFD results compare well with BLWT data and field measurements available in literature.

1. INTRODUCTION

Wind pressures on buildings and other structures, pedestrian level winds, and wind-induced dispersion of pollutants in urban locations depend, among other factors, on the velocity profile and turbulence characteristics of the upcoming wind. These, in turn, depend on the roughness and general configuration of the upstream terrain. Consequently, wind standards and codes of practice typically assume upstream terrain of homogeneous roughness or provide explicit corrections for specific topographies such as hills or escarpments; for more complex situations they refer the practitioner to physical simulation in a boundary layer wind tunnel. The evolution of computational wind engineering makes numerical evaluation of wind velocities over complex terrain an attractive proposition. In fact, significant progress has been made in the application of CFD for specific cases of evaluation of wind flow over escarpments, hills as well as valleys (for representative references, see Bergeles 1985, Paterson and Holmes 1993, Maurizi 2000 and Bitsuamlak et al. 2003, 2004). At present, utilization of numerical simulations to predict wind speed-up in practical applications is rather limited and designers still have to rely on physical simulations for complex terrain situations. This is mainly due to the unavailability of specialized and cost-effective numerical tools targeted to give solutions for wind flow over different topographies. Hence the present study will attempt to address this issue.

Several CFD tools such as Turbulent WInd SimulaTion-TWIST (Stathopoulos and Zhou 1993) use procedural languages where data structures are globally accessed. As a result it is difficult to modify and extend such codes. A high level of knowledge of the tool is required even for simple modifications. To address this issue, the selected paradigm should allow easy reusability (can be assembled from pre-written components with minimal efforts) as well as extensibility (the assembled system can be easily extended without modifying significantly the reused components). Therefore an object-oriented methodology has been used in the present study. Object-oriented approach abstracts the data into objects easing modification and reuse. It also eases parallel programming tasks if a need to use a cluster of processors arises.

Emphasis has been given to several numerical details in order to enhance the accuracy of the CFD simulation. This includes incorporation of influential parameters such as the ground roughness, application of appropriate boundary conditions, accurate representation of the terrain geometry using body-fitted coordinates and efficient control of grid density distribution over computational domain (CD). For this purpose a robust CFD tool has been developed on the basis of finite-volume formulation using object-oriented approach and C++ programming language. While the 2D version has been completed the work is under progress for the 3D version. A robust orthogonal grid generator suitable for the present type of study, which is characterized by curved and slopping surfaces, has also been developed. Spatial distribution of grid points is efficiently controlled to provide dense grids wherever the variation of the dependent variable is high. In the present paper the CFD tool development process will be described and the present CFD results obtained will be compared to Boundary Layer Wind Tunnel (BLWT), Field measurement (in few cases), National Building Code of Canada (NBCC), analytical and previous CFD data as well.

2. GOVERNING EQUATIONS

The numerical evaluation of wind flow over topography uses computational fluid dynamics (CFD) technique. The governing equations employed are the Reynolds Averaged Navier-Stokes (RANS) equations together with the standard k-ε turbulence model (Launder and Spalding 1974) representing the turbulent wind flow over different cases of isolated and multiple hills. The vectorized form of the governing equations in a Cartesian coordinate system including the elliptical equations used to generate grids is as follows:

[1] $$\frac{\partial E}{\partial x}+\frac{\partial F}{\partial z}=S$$

$$\text{where}\begin{bmatrix}x-\text{coordinate}\\ z-\text{coordinate}\\ \text{Continuity}\\ x-\text{momentum}\\ z-\text{momentum}\\ k-\text{equation}\\ \varepsilon-\text{equation}\end{bmatrix} E=\begin{bmatrix}fX_x\\ fZ_x\\ \rho U\\ \rho UU-\Gamma U_x\\ \rho UW-\Gamma W_x\\ \rho Uk-\Gamma k_x/\sigma_k\\ \rho U\varepsilon-\Gamma\varepsilon_x/\sigma_z\end{bmatrix} F=\begin{bmatrix}1/f\,X_x\\ 1/f\,Z_x\\ \rho W\\ \rho WU-\Gamma U_z\\ \rho WW-\Gamma W_z\\ \rho Wk-\Gamma k_z/\sigma_k\\ \rho W\varepsilon-\Gamma\varepsilon_z/\sigma_z\end{bmatrix} S=\begin{bmatrix}Q(X)+R(X)\\ Q(Z)+R(Z)\\ 0\\ (\mu_t U_x)_x+(\mu_t W_x)_z-P_x\\ (\mu_t U_z)_x+(\mu_t W_x)_z-P_z\\ G-\rho\varepsilon\\ (C_1 G-C_2\rho\varepsilon)\varepsilon/k\end{bmatrix}$$

The subscripts x and z denote the partial derivatives in the respective coordinate directions, dependent variables U, W, P, k and ε represents mean velocity in x- and z-directions, pressure, kinetic energy and kinetic energy dissipation respectively. f represents scale factor and Q and R are functions of the scale factor (Akcelik etal. 2001). The physical quantities ρ, μ and μ_t represent density, dynamic viscosity of air, and turbulent viscosity respectively. Γ is the sum of μ and μ_t. The turbulent viscosity μ_t and turbulence kinetic energy production rate G are expressed as

[2] $$\mu_t=\rho C_\mu\frac{k^2}{\varepsilon}$$

[3] $$G=\Gamma\left\{2(U_z^2+W_x^2)+\left(U_z+W_x\right)^2\right\}$$

Model constants C_1, C_2, C_μ, σ_k, σ_ε are equal to 1.44, 1.92, 0.09, 1.0 and 1.3 respectively (Launder and Spalding 1974). Equation (1) has been transformed into the general coordinate system. For pressure correction, Rhie and Chow's (1983) interpolation technique is adopted. Further, hybrid scheme (Spalding, 1972) for convection terms and SIMPLEC iterative solution procedure of Van Doormal and Raithby (1984) for solving the dependent variables sequentially are used.

2.1. Boundary Conditions

Log-law velocity distribution is applied along the upstream boundary. Upstream boundary values of k and ε are obtained by solving the k-ε equations using inlet U and W values. In the present study, the effect of roughness has been considered throughout the ground surface of the Computational Domain (CD) in addition to incorporating its effect at the upstream boundary velocity profile. This is achieved through the computation of the shear stress, equation (4), throughout the ground surface and then incorporating it in the momentum equations for the ground control volumes.

[4] $\tau_w = U_P^* \mu y^+ / z_P \kappa \ln(z_P / z_o)$

[5] $U_P = U^* \ln(z_P / z_o) / \kappa$

where U*: friction velocity, z_p: the distance of the first grid from the ground, $y+= \rho z_p U^*/\mu$, z_o: roughness of the ground and κ: Von Karman constant (=0.4). By taking the downstream boundary far enough from the terrain feature to be simulated, downstream boundary $\partial/\partial x=0$ is assumed for all the dependent variable. Figure 1 shows the CD and summarizes the boundary conditions (BC), where H represents the height of the hill under consideration and L represents the horizontal distance from the crest to where the ground elevation is half the height of the hill.

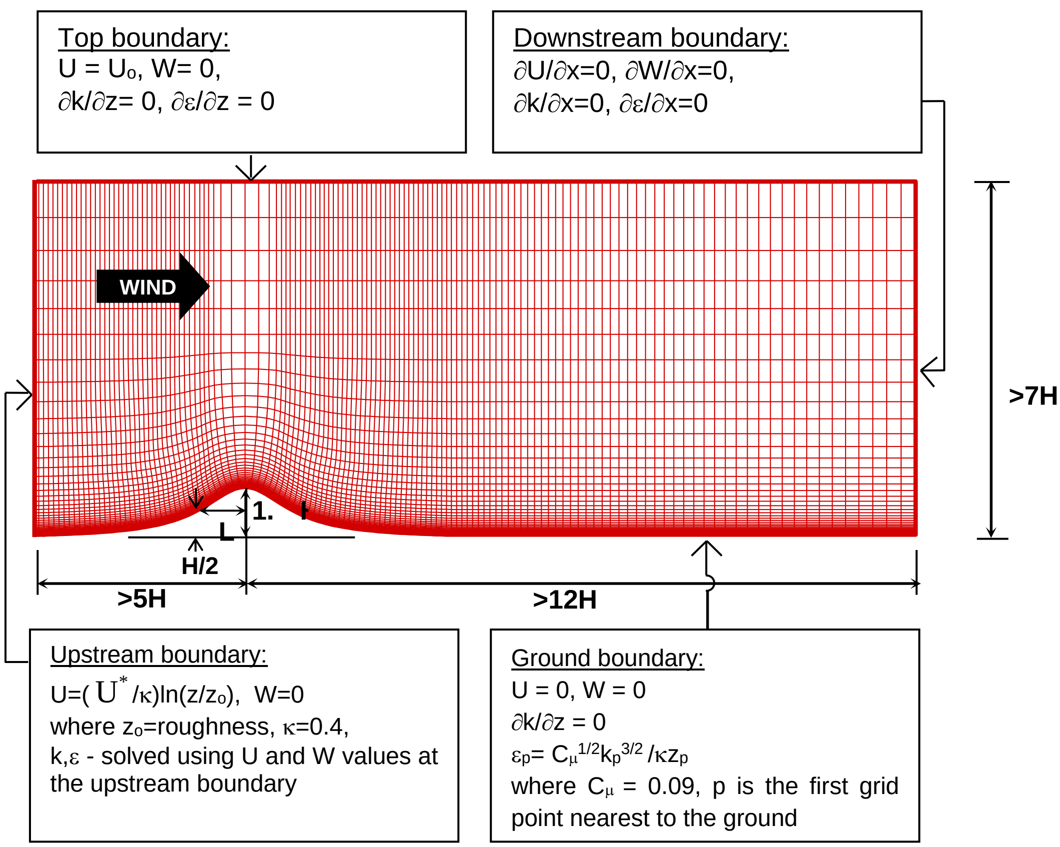


Figure 1 Computational domain dimensions and boundary conditions.

3. CFD TOOL ANALYSIS AND DESIGN

The steps in object-oriented tool development include detailed description of the problem statement (analysis), identifying potential classes from the problem statement, designing of each class and their member functions, and implementing the design with a programming language that supports object-oriented approach such as C++.

3.1. The problem statement

The main objective here is the development of a CFD tool that will use the governing equations and discretize them and iteratively solve the discretized equations for each of the dependent variables sequentially. For this purpose the following tasks are necessary while following finite volume formulation: separate CD from the complete domain; transform the CD; discretize the CD into small control volumes (CV); obtain transformation parameters; integrate partial differential equations over CV; apply finite differences and obtain algebraic equations; estimate initial conditions for each dependent variable; obtain pressure gradients; set gamma (sum of dynamic and turbulent viscosity), set convection terms; set flow terms; set diffusion terms; calculate east, west, north and south coefficients at each CV; calculate the sources at each CV; apply BCs; under-relax (adding a small percent of a variable value from the present iteration to the remaining percentage of a variable value from the previous iteration and use this combied value as a starting point for the next iteration); solve the algebraic equations for each variable sequentially; store the results; format the results for post-processing and plotting.

3.2. Objects design

Examining the problem carefully helps to identify the objects. In fact, two alternatives have been considered to define classes and their relationships. The first alternative has been to base the relationship on physical or geometrical entities that exist in the problem. The other option has been to base relationships on mathematical entities. Weller et al. (1998) used mathematical entities while developing a general purpose CFD. In their approach mathematical entities such as divergence, laplacian operator and different tensorial notations were designed as objects. This approach is suitable for developing large general-purpose application software. For a specialized application as in the present case however, the authors prefer an approach in which the designed objects expose useful geometrical relationships that exist in the problem domain as well as the physics, rather than the mathematics. Moreover, if parallel processing is required, dividing large tasks can be based on the geometrical entities, thus easing the parallel implementation.

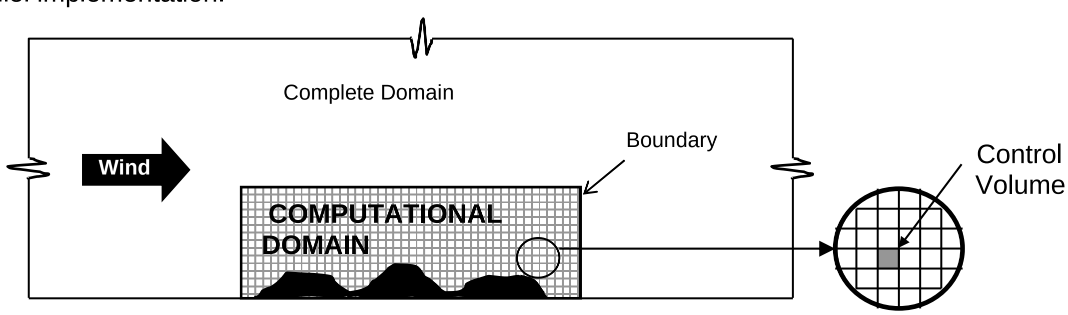


Figure 2. Geometrical definition of complete domain, CD, and CV

During the development of a 2D CFD tool, physical/geometrical entities such as CD and CV have been used as classes. The geometrical definition of these classes is shown in Figure 2. The relationship between the two classes CD and CV is “CD has a number of CVs”. An entity relationship diagram is shown in Figure 4 for the two classes. Entity-relationship diagrams show the relationships among classes in the design. Classes are shown as boxes around the class name. The links between boxes establish the relationships between classes. For example, CV is part of CD. Thus, the classes CV and CD are linked together with “part of” aggregation. Links that end in a diamond shape represent aggregation. Aggregation describes a relationship in which one object is “part of” another in an assembly. In fact, many CVs are part of a CD. A darkened circle at the end of the link represents this multiplicity. Links with no special ending represent a multiplicity of exactly one.

For the 3D CFD case, classes such as ComputationalDomain, ComputationalDomainTransformed, NeighbouringCV, ControlVolume and GridPoint has been designed. The geometrical defintion of these classes is shown in Figure 4. The ComptationalDomainTrasformed represents CD after transformation. Niegbouring CV consists of the central CV and its six neighbouring CVs together, so that there will be a localized and easy access of information by the central CV from its neighbouring CV's. At the boundaries, the Nieghbouring CV class has a null member class (all its arguments set to zero). For example at the top boundary, the top member CV is null. ControlVolume represents a cubic CV and stores all dependent variables and coefficient values. GridPoint represents the grid coordinate (x,y,z) at the center of the CV. The entity-relationship diagram between classes is shown in Figure 5.

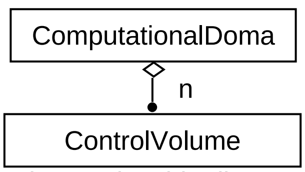
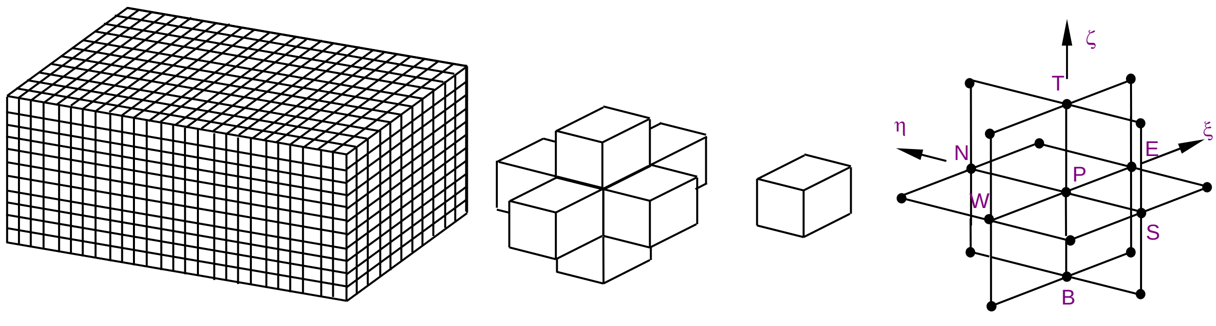


Figure 3. Entity-relationship diagram (2D CFD).



(a) (b) (c) (d)

Figure 4.Geometrical definitions of (a) ComptationalDomainTransformed, (b) NieghbouringCV, (c) ControlVolume and (d) GridPoint classes (3D CFD).

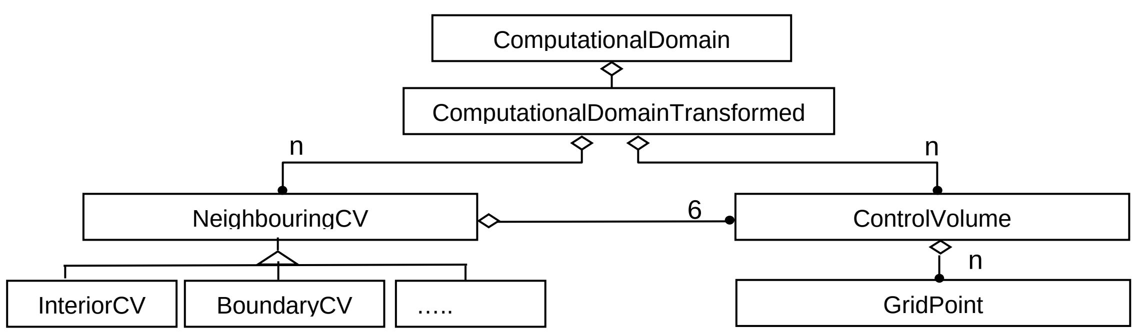


Figure 5. Entity-relationship diagram (3D CFD).

Once the relationships of each object with the rest of the system are established, the specific functions that each object must provide can be defined. The interface definition table provides a formal description of the publicly available member functions the objects provide, including arguments and return types. A sample interface definition for ComptationalDomain (for 2D CFD) is given in Table 1. The arguments and return types are given as a non-language specific description of the object type that is required. It is intended to be used directly to implement the design in a programming language.

Table 1. ComptationalDomain class interface definition (2D CFD).

Member functions	Arguments	Returns	Purpose
Construction	CD size and number of grid points		Creates an instance of a specific type of CD
SetGrid		Coordinates of CD grids	Discretizes the CD into CVs
SetGridTransform Parameters		Parameters used to transform the PH to CD	Provides geometric transformation parameters
InitializeDepVar	Variable type	Guessed U, W, P, k and ε	Initializes each variable with guessed values
SetPressCorrParas		Pressure correction parameters	Provides geometric transformation parameters
SetPressureGradients		Pressure gradient at grid point and face of CV	Provides the pressure gradients
SetConvectiveTerms			Provides convection terms
SetFlow		Flow parameters	Provides flow terms
SetDiffusion		Diffusion parameters	Provides diffusion terms
SetSources	Variable type	Source terms	Sets the sources at each CV for each variable
SetTurbulentVisc	Variable type	Turbulent viscosity	Provides turbulent Viscosity
SetGamma	Variable type	Sum of dynamic viscosity and turbulent viscosity	Provides gamma
SetCoefficients	Variable type	East, west, north and south coefficients at each CV for U, W, P, k and ε	Sets the east, west, north and south coefficients at each CV for each variable
ApplyBound			Applies boundary conditions
UnderRelax			Under relaxes variables
PressureCont	Variable type	Pressure contribution	Provides pressure contribution
SolverManager	Variable type		Selects 1D, 2D or 3D solvers
Solve			Solves algebraic equations
CorrectUandV		Corrected U and V	Corrects velocity field
CorrectConvCoef		Corrected Convective terms after each iteration	Corrects convective coefficient terms
Plot		Inputs to TECPLOT	Formats the output for the graphical tool (TECPLOT)
Output			Setup output of desired results
CalculateSpeedUp		Speed up	Extracts speed up values from the velocity field
ConvergCri			Provides the convergence criteria
IntializeFromFile	File names		Initializes variables from files

3.3. Implementation in C++

Once the object design is complete, the next task is implementation in a computer language. The natural choice for an object-oriented design is an object-oriented programming language, which is why C++ has been used. The objects described in the previous section lend themselves to direct implementation as classes in C++. The Entity-relationship Diagram gives the dependencies the class has with other classes. The Interface definition table defines the member functions the class must provide. What remains for the implementation are identifying the specific C++ data types used as the arguments and return values of the member functions, and the data types used for the class' instance variables. The classes in the present study are designed from scratch. Intrinsic C++ types, such as Boolean, ints (integers) and doubles (real numbers) have been used. A data structure that can store objects or any other data types has been designed and implemented as well by using dynamic memory allocation employing pointers and “new” commands available in C++ programming language. Documentation at each level of computer code development has also been given emphasis for sake of better clarity/efficiency either during the present research, future research or operation stage.

4. CFD RESULTS AND DISCUSSION

Wind flow over different types of topography has been simulated using the developed CFD tool over representative geometries selected to cover a wide spectrum of terrain conditions. These include two-dimensional isolated, multiple hills, and complex terrain. To validate and highlight the advantages (or disadvantages) of the currently developed numerical method, detailed comparisons are made with results obtained from five other methods: field measurements, BLWT experiments, analytical methods, NBCC provisions (1995), and existing CFD simulations. For this purpose, in most cases the specific dimensions of the terrain geometries used are chosen to be similar to those used in the BLWT experiments found in literature. This is due to the fact that in BLWT experiments, boundary conditions are better controlled, thus producing reliable and widely acceptable results by the engineering community at large. Preliminary numerical experimentation has been conducted to select the size of the CD so that the location of the boundaries has no influence on the numerical results. Suggested minimum CD for a hill has a height, upstream length and downstream length equal to 7H, 5H and 12 H respectively, where H is the hill height. It may be noted that for high wind speed flows, larger CD size could be necessary.

4.1. Isolated Hill

Under this category, wind flow over steep single cosine hills is presented. The geometry and roughness parameters used are similar to those used in Carpenter and Locke's (1999) BLWT experiments, i.e. a steep hill with H=200 mm and L=200 mm is used. Scaling up these dimensions by 1000 gives the actual hill dimensions. A roughness of 0.02 m (at full scale) equivalent to open terrain category is considered. A 115x36 mesh covering 3.4m x 1.2 m is used. Numerical results are shown and compared in Figure 6. The gradual increases in velocity as the wind approaches the hill and the recirculation region behind the hill are clearly predicted. For reasons of clarity, in all velocity field vector figures, the velocity vectors are plotted skipping at every other grid point in the x-direction. For validation purposes, Fractional Speed-Up Ratio (FSUR) defined in terms of $U(z)/U_0(z)$ are compared at the crest of the hill where they are critical. The agreement between the present CFD and BLWT results is excellent as shown in Figures 6(c). The present CFD simulation results agree better with the BLWT data than Carpenter and Locke's (1999) CFD data. Amongst other factors, the close agreement is believed to be primarily due to the incorporation of roughness in the present study through the application of log law throughout the ground boundary that depicts exactly the roughness conditions used by the Carpenter and Locke BLWT experiments. Moreover, the body-fitted grid used in the present study, which allows representation of the true geometry of the BLWT model together with effective spatial distribution control of the grid points over the CD that allow allocation of denser grids where they are most required, may have also contributed to the improvement. FSUR values at the crest of the hill are also generated using NBCC provisions and Weng et al. (2000) guidelines, which were derived from the Multi Spectral Finite Difference (MSFD) model of Beljaars et al. (1987) and its nonlinear extension NLMSFD by Xu et al. (1994). There is sizable discrepancy among Weng et al. (2000) values and the BLWT data near the steep hilltop as shown in Figure 6(c) where Weng et al. guidelines overpredict the FSUR values. This is attributed to the inadequacy of the underlying linear theory used by the guidelines, especially for wind flows over steep hills that are characterized by the presence of flow separation and large recirculation regions.

4.2. 2D multiple hills

The geometry and roughness parameters used under this category are similar to those used in Carpenter and Locke's (1999) BLWT experiments. Thus a roughness of 0.02 m (at full scale) is used. The hills have H=200 mm and L=200 mm. Distance between the crests of the hills is 8H as shown in Figures 7(a). A 179x36 mesh covering a CD of 5.2 m x 1.2 m is used. Figures 7(a) and (b) show the geometery of the hills and mean wind velocity vectors, respectively. The mean flow characteristics for multiple hill case fundamentally differ from that of an isolated hill case especially for the downstream hill. This is because the wind flow separation from the upstream hill (Hill-1) results in increased turbulence on the downstream hill. This in turn causes the flow to attain more uniformity resulting in a reduction of the peak FSUR values just above the crest of the downstream hill (Hill-2). Hence the FSUR values for the downstream hill (Hill-2) are smaller than the values for the upstream hill (Hill-1), as shown in Figure 7(c). At the crest of the

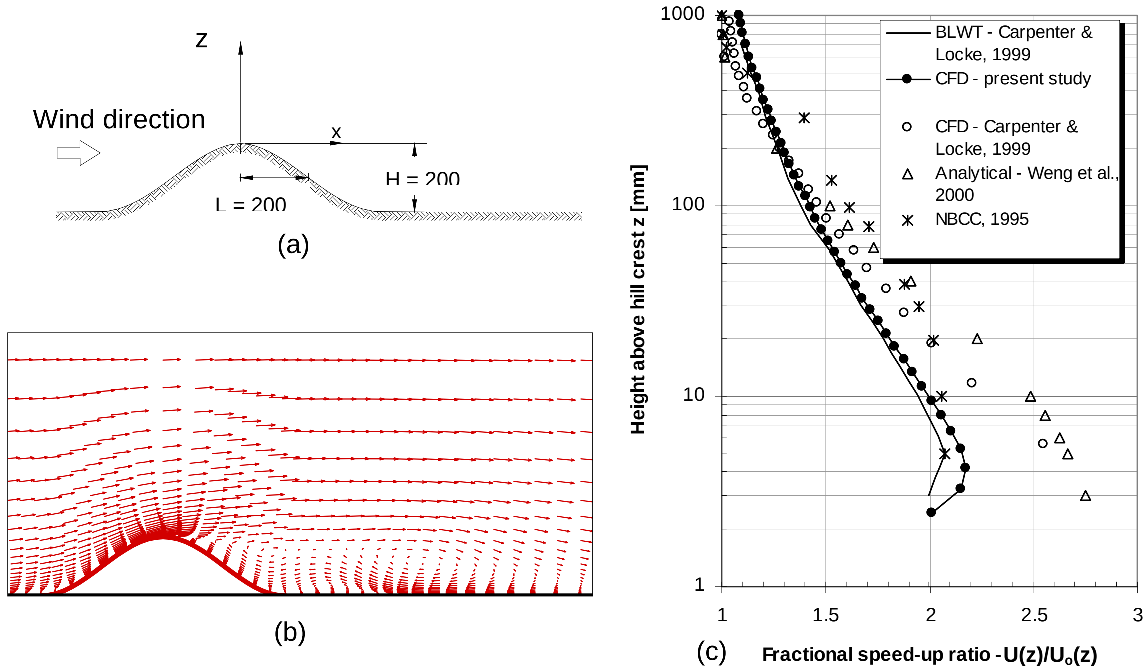


Figure 6. Single Cosine hill. (a) geometry used by the present as well as Carpenter and Locke's (1999) studies, (b) velocity field, and (c) FSUR value comparisons above the crest among BLWT, CFD, analytical and NBCC data. All dimensions are in [mm].

respective hills and for the height range of most engineering interest (5-100 m), upstream hill FSUR values are 30 % larger than the downstream FSUR values. This percentage in terms of FSUR values can translate to approximately 70 % increase in wind load. To date no NBCC provision (1995) exists for the case of double hills. In practice NBCC provisions for single hills apply to the double hill case. However, when NBCC provisions are used for the downstream hill case, FSUR values are overpredicted approximately by 44% for the height range of 5-100 m. The wind load is therefore over predicted by 100%. The economical impact of accurate wind load prediction has been studied by Horesfield et al. (2002). It was shown that with the use of more accurate loads a 19% of the concrete lateral load-resisting system (around 6500 m3) and 2.1% of usable floor area have been saved. The practice of using similar FSUR values for both the upstream and downstream hills usually obtained from the single hill investigations is indeed a conservative approach. Hence, in areas characterized by chains of mountains, the wind load has to be evaluated based on studies of multiple hills, rather than the case of a single hill. In addition, the detailed wind field evaluations carried out in the present study can be of great importance in environmental planning. In an industrialized area for example, it is apparent from Figure 7(b) that the recirculation between the hills could entrap pollutants. Furthermore considering the increasing trend of using wind energy alternatives in Canada and around the world, the present numerical simulation tool can also be useful in wind farm planning.

4.3. 2D Complex terrain

Under this category, a numerical simulation of wind flow over a complex terrain is carried out. The terrain has a maximum height of 500 m and a roughness of 0.7 m (at full scale). A 167x150 mesh covering 3000 m x 8000 m is used. The gradual increases in velocity as the wind approaches the hill and the recirculation region behind the hill is clearly predicted as shown in Figure 8. Currently, the authors are working on the comparison of this numerical result with a boundary layer wind tunnel data set for validation purposes.

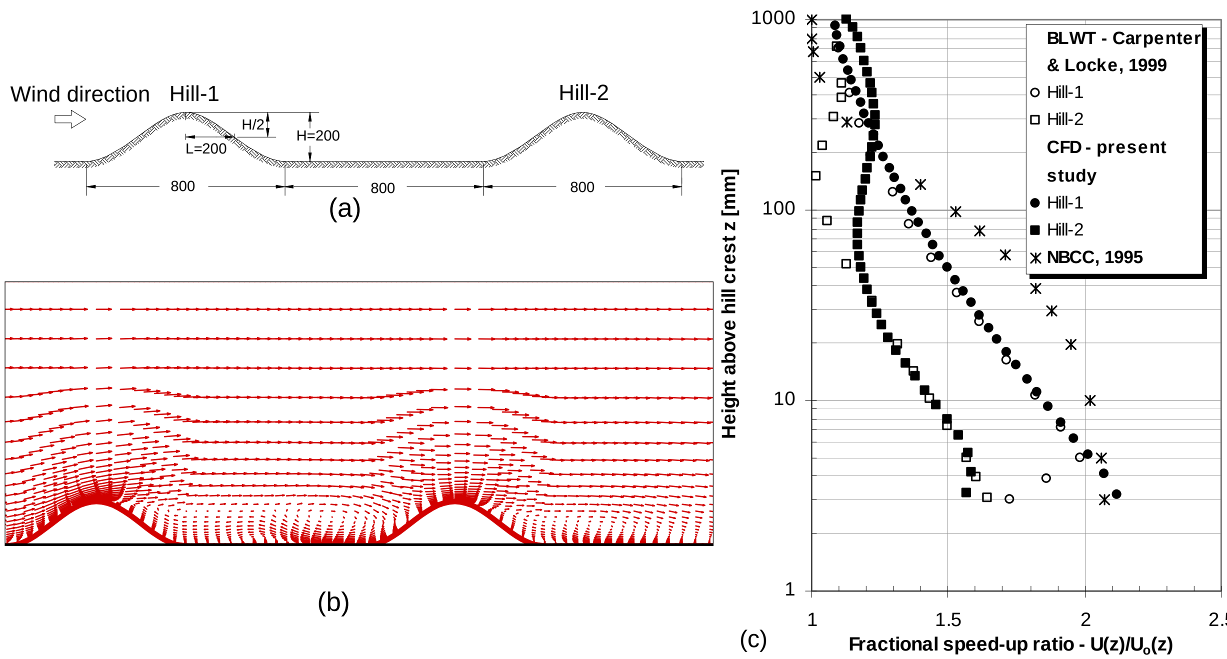


Figure 7. Numerical simulation of wind flow over multiple hills: (a) geometry used by the present as well as Carpenter and Locke's (1999) studies, (b) velocity field, and (c) FSUR value comparisons above the corner crest among BLWT, Field, CFD and NBCC data. All dimensions are in [mm].

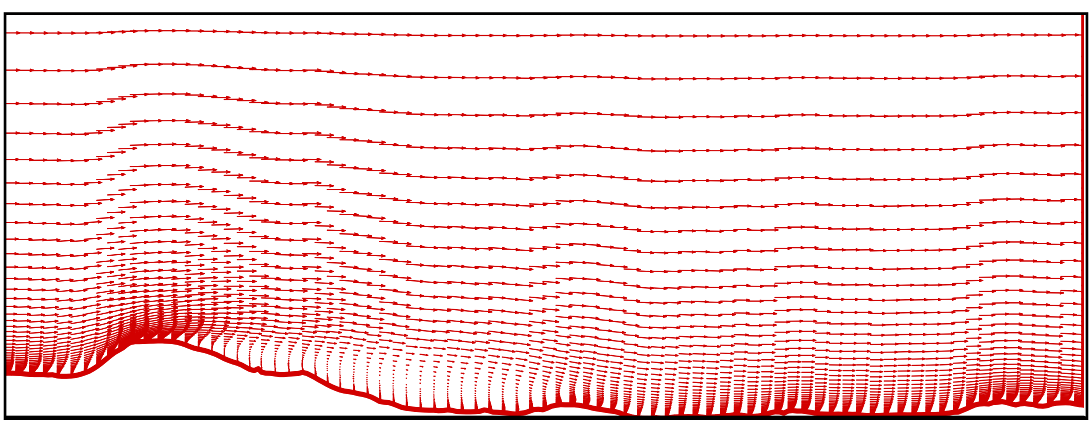


Figure 8. Numerical simulation of wind flow over a complex terrain (velocity field).

5. CONCLUSIONS

A robust CFD tool for numerical evaluation of wind flow over different types of topographies have been developed, using object-oriented approach, and validated with the objective of producing flow parameters useful for estimating wind loads on buildings in the vicinity of complex terrain. Consistent FSUR values have been generated using the developed CFD code and their comparison with the experimental values depicted a good agreement over different terrain geometries including two-dimensional single and multiple hills. The CFD proves useful by generating speed-up ratios for geometries such as multiple hills that are not addressed in the NBCC provisions, and incorporates as well influential parameters such as roughness of the surface.

6. REFERENCE

Akcelik V., Jaramaz, B., Ghattas G. (2001), "Nearly orthogonal two-dimensional grid generation with aspect ratio control", *Jnl. of Comput*. Physics, Vol. 171, 805–821.

Bergeles, G.C. (1985), "Numerical computation of turbulent flow around two-dimensional hills*", Jnl. of Wind Eng. Ind. Aero*, Vol. 21, 307-321.

Beljaars, A.C., Walmsley, J.L., Taylor, P.A. (1987), "A mixed spectral finite-difference model for neutrally stratified boundary-layer flow over roughness changes and topography", Boundary-Layer Meteorology, Vol. 38, 273-303.

Bitsuamlak, G.T., Stathopoulos, T., Bedard, C. (2003) "Numerical evaluation and neural net predictions of wind flow over complex terrain", *11th International Conference on Wind Engineering*, June2-5, Lubbock, Texas, USA, Vol. 2, 2673-2679.

Bitsuamlak, G.T., Stathopoulos, T., Bedard, C. (2004) "Numerical Evaluation of Wind Flow over Complex Terrain: A Review", *Jnl. of Aero. Eng*., Vol. 17, No. 4, pp. 135-145

Carpenter, P., Locke, N. (1999), "Investigation of wind speed over multiple two-dimensional hills*", Jnl. of Wind Eng. Ind. Aero*, Vol. 83, 109-120.

Horsfield, J.N., Chan C.M. and Denoon R.O. (2002) "Towards sustainable development through innovative engineering", *Housing Conference*, Wanchai, Hong Kong.

Launder, B.E. and Spalding, D.B. (1974), "The numerical computation of turbulent flows", *Comput. Methods Appl. Mech. Eng.*, Vol. 3, 269-289.

Maurizi, A., (2000), "Numerical simulation of turbulent flows over 2D valleys using three versions of the k-ε closure model", *Jnl. Wind Eng. Ind. Aero*, Vol. 85, 59-73.

NBCC, National Building Code User's Guide – Structural Commentaries (Part 4), Canadian Commission on Building and Fire Codes, National Research Council of Canada, Ottawa (1995).

Paterson, D.A., Holmes, J.D. (1993), "Computation of wind flow over topography", *Jnl. of Wind Eng. Ind. Aero,* Vol. 46&47, 471-478.

Rhie, C.M., Chow, W.L. (1983), "Numerical study of the turbulent flows past an airfoil with trailing edge separation", *AIAA Journal*, 21, 1525-1532.

Spalding D. B. (1972), "A novel finite-difference formulation for differential expressions involving both first and second derivatives", *Int. Jnl. Num. Methods Eng*., Vol 4, 551-559.

Stathopoulos, T. and Zhou, Y.S. (1993), "Numerical simulation of wind-induced pressures on buildings of various geometries" *Jnl. of Wind Eng. Ind. Aero*, Vol 46-47, 419-430.

Van Doormal, J.P and Raithby, G.D. (1984), "Enhancements of the SIMPLE method for predicting incompressible fluid flows", *Numer. Heat Transfer*, Vol 7, 147-163.

Weller, H.G., Tabor, G., Jasak H., Fureby, C. (1998), "A tensorial approach to computational continuum mechanics using object orientated techniques", *Computers in Physics* Vol 12(6), 620 – 631.

Weng, W., Taylor, P.A., Walmsley, J.L. (2000), "Guidelines for airflow over complex terrain: model developments", *Jnl. of Wind Eng. Ind. Aero*, Vol. 86, 169-186.

Xu, D., Ayotte, A.W., Taylor, P.A. (1994), "Development of a non-linear mixed spectral finite difference model for turbulent boundary-layer flow over topography", *Boundary-Layer Meteorology*. Vol. 70, 341–367.