DETERMINATION OF AERODYNAMIC FORCES OVER TREE MODELS USING CFD AND PARALLEL PROCESSING TECHNIQUES

Alexandre Luis Braun, allbraun@ig.com.br
João Ricardo Masuero, masuero@ufrgs.br
Armando Miguel Awruch, amawruch@ufrgs.br
Graduate Program in Civil Engineering (PPGEC) – Federal University of Rio Grande do Sul (UFRGS)
Av. Osvaldo Aranha, nº 99, 3º andar, CEP 90350-190, Porto Alegre/RS, Brasil

Abstract. *In the present work the evaluation of aerodynamic forces acting on tree models is performed using numerical simulations based on the Finite Element Method and parallel processing techniques. Damages induced by the wind action over trees in urban areas may lead to serious implications, such as interruptions in the energy supply and communication systems and risk to the safety of pedestrians. The failure mechanism of trees under wind action is ususally associated to breakage of the stem owing to excessive compression, root pullout and overturning due to excessive moments, which may be also associated to the mechanical behavior of the soil. Therefore, aerodynamic loads are a key parameter for determining the occurrence of damages in trees. In the present paper, idealized tree models are utilized to determine aerodynamic coefficents and flow fields around the tree location. The numerical model is based on the incompressible flow approach using the pseudo-compressibility hypothesis. Turbulence is taken into account considering Large Eddy Simulation (LES) with the dynamic sub-grid model. The system of governing equations is solved employing an explicit two-step Taylor-Galerkin scheme, where spatial approximations are performed using the Finite Element Method (FEM) with linear hexahedral elements and one-point quadrature. In order to solve the large system of equations associated to the flow problem efficiently, parallel processing with MPI implementation is adopted. Results obtained with the numerical simulations carried out here are compared to experimental and numerical predictions presented by different authors and the performance of the parallelization techniques utilized in the present work is also verified.*

Keywords*: Tree Aerodynamics, Computational Fluid Dynamics (CFD), Large Eddy Simulation (LES), Finite Element Method (FEM), Parallel Processing.*

1. INTRODUCTION

Tree planting is one of the most important human actions to improve the urban environment today. The influence of trees on the urban environment is mainly related to reduction of thermal effects through weakening of radiation and the impact of transpiration on the latent heat equilibrium (Liang et al., 2006). In addition, trees alter significantly the air flow in the street canyons of large cities by reducing the flow velocity and increasing the turbulence level. In response to the wind action, trees adapt their stem and roots in order to resist to overturning moments, breakage of the stem or root pullout, which may be also associated to the mechanical behavior of the soil. Some of these failure mechanisms may be induced on the tree structure by strong winds, leading to serious implications such as interruptions in the energy supply and communication systems and risk to the safety of pedestrians. Therefore, accurate reproduction of aerodynamic effects on trees is very significant for predicting wind loads on buildings and wind environment in urban areas.

Wind Engineering is a traditional and well established research field developed through extensive experimental tests performed over several structures and bodies submitted to the wind action, which are characterized with devices specially designed for wind tunnel studies. Many advances and improvements were obtained with aerodynamic and aeroelastic analyses carried out over civil structures such as bridges and buildings. In addition, wind environmental, pollutant and thermal conditions in urban areas have been also assessed by studying representative models in specialized wind tunnels. However, owing to advances observed in computers technology and numerical methods, experimental analyses in wind tunnels have been gradually replaced by numerical simulations performed with numerical models based on Computational Wind Engineering (CWE), which is a relatively new branch of Computational Fluid Dynamics (CFD) adapted for specific analysis in Wind Engineering. CWE involves some of the most challenging subjects on numerical investigation applied to aerodynamic and aeroelastic phenomena: turbulence modeling, fluid-structure interaction, boundary layer flows including separation and reattachment and, consequently, dense finite element meshes. Therefore, numerical analysis of trees under wind action can be also carried out using CWE techniques. Recent investigations may be found, for instance, in Liang et al. (2006), Mochida et al. (2008) and Endalew et al. (2009).

An important restriction related to numerical analysis of 3-D flows using the Finite Element Method (FEM) is referred to the number of finite elements utilized by the spatial discretization procedure (see [a]Braun and Awruch, 2009). It is observed that some millions of elements are usually needed in order to obtain a well defined mesh where the boundary layer and vortex trail regions can be adequately represented. Obviously, such a task can be only accomplished employing different approaches, including high performance computational systems and special programming techniques. Formerly, scalar serial codes used to be modified by using vectorization to run in supercomputers with

vector processors and shared memory. However, supercomputers are very expensive and require special maintenance procedures. Therefore, this option has been replaced by an alternative approach, where clusters of personal computers are utilized with high performance networks and parallelization techniques. Parallelization is a programming procedure where some statements are included in the code in order to permit the problem to be solved with several processors working under parallel processing. A special feature associated to clusters of personal computers is concerned to the operation and maintenance processes, which are made with very low costs when compared to other alternatives.

A parallel program is generally written using some high level language such as Fortran, where parallelization libraries are also included in order to permit the processors to work synchronously. The most common APIs (Application Programming Interface) utilized by a parallel code are OpenMP (Open Multi Processing) and MPI (Message Passing Interface). OpenMP libraries are employed by parallel systems with shared memory. On the other hand, MPI libraries are usually indicated for parallel processing with distributed memory (Masuero, 2009).

In the present work, the Navier-Stokes equations and the mass conservation equation are solved using an explicit two-step Taylor-Galerkin model (see Kawahara and Hirano, 1983; [a]Braun and Awruch, 2009). The Finite Element Method (FEM) is employed for spatial discretizations using the eight-node hexahedrical isoparametric element with one-point quadrature. Turbulent flows are analyzed using LES with the dynamic model for sub-grid scales (see Smagorinsky, 1963 and Lilly, 1992). Some simple tree models are investigated in order to determine aerodynamic coefficients and flow fields around the tree location and the respective results are compared with numerical and experimental predictions performed by other authors. All simulations are carried out using a cluster with 8 machines featuring four core processors and distributed memory architecture. The numerical algorithm was written employing the MPICH version 1.2.5, which is a MPI library developed by Argonne National Laboratory.

2. THE NUMERICAL MODEL

In the Computational Wind Engineering practice, wind flows are usually characterized by adopting the following assumptions ([a]Braun and Awruch, 2009):

1) Natural wind streams are considered to be within the incompressible flow range;
2) Wind is always flowing with a constant temperature (isothermal process);
3) Gravity forces are neglected in the fluid equilibrium;
4) Air is considered as a Newtonian fluid.

Considering the properties presented above and in the absence of structural motion (aerodynamic analysis), the flow governing equation are defined in a classical Eulerian kinematical description by the following expressions:

a) Momentum balance equations:

$$\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j} \qquad (i, j = 1,2,3) \quad \text{in } \Omega^f \tag{1}$$

b) Mass balance equation for pseudo-compressible flows (see Braun and Awruch, 2005 for further details) – the continuity equation:

$$\frac{\partial p}{\partial t} + v_j \frac{\partial p}{\partial x_j} + \rho c^2 \frac{\partial v_j}{\partial x_j} = 0 \qquad (j = 1,2,3) \quad \text{in } \Omega^f \tag{2}$$

c) Constitutive equation for Newtonian fluids:

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} \quad ; \quad \tau_{ij} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \lambda \frac{\partial v_k}{\partial x_k}\delta_{ij} \qquad (i, j, k = 1,2,3) \tag{3}$$

where v_i are components of the velocity vector in the i direction, x_j are components of the Cartesian coordinates vector in the j direction, t indicates the time domain, p is the thermodynamic pressure, ρ is the fluid's specific mass, c is the sound speed in the fluid field and Ω^f is the flow's spatial domain, which is bounded by Γ^{Tf}, δ_{ij} are components of the Kroenecker's delta ($\delta_{ij} = 1$ for $i = j$; $\delta_{ij} = 0$ for $i \neq j$) and μ and λ are the dynamic and volumetric viscosities of the fluid, respectively. Neumann and Dirichlet boundary conditions must be specified on Γ^{Tf} to solve the flow problem, which are given by the following expressions:

$$v_i = \overline{v}_i \qquad (i = 1,2,3) \quad \text{on } \Gamma^v \tag{4}$$

$$p = \overline{p} \qquad \text{on } \Gamma^p \tag{5}$$

$$\left[-\frac{p}{\rho}\delta_{ij} + \frac{\mu}{\rho}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \frac{\lambda}{\rho}\frac{\partial v_k}{\partial x_k}\right] n_j = \frac{\sigma_{ij} n_j}{\rho} = \overline{S}_i \qquad (i, j, k = 1,2,3) \quad \text{on } \Gamma^\sigma \tag{6}$$

where Γ^v (boundary with prescribed values $\overline{v}_i$ for the fluid velocity field), Γ^p (boundary with prescribed values $\overline{p}$ for the pressure field) and Γ^σ (boundary with prescribed values $\overline{S}_i$ for the fluid boundary tractions) are complementary subsets of boundary Γ^{Tf}, such that $\Gamma^{Tf} = \Gamma^v + \Gamma^p + \Gamma^\sigma$. In Eq. (6) n_j are components of the unit normal vector **n** at

boundary Γ^{σ}. Initial conditions for the pressure and velocity fields must be also specified at $t = 0$ to start up the flow analysis.

Atmospheric boundary layer flows may be reproduced considering the power law as inflow boundary conditions imposed on the velocity field. The power law is usually described as:

$$U(z) = \bar{U}_H \left(\frac{z}{H} \right)^p \tag{7}$$

where $\bar{U}_H$ is the reference wind speed corresponding to the reference height H, z is the height coordinate and p is an empirically derived coefficient associated to roughness characteristics of the ground (see Simiu and Scanlan, 1996).

In order to avoid excessive computational efforts to describe the smaller turbulence scales, LES is adopted in the present work (see Mochida and Lun, 2008 for detailed information on turbulence modeling). Consequently, the governing equations should be rewritten as follows:

$$\frac{\partial \overline{v}_i}{\partial t} + \overline{v}_j \frac{\partial \overline{v}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j} + \frac{1}{\rho}\left(\frac{\partial \overline{\tau}_{ij}}{\partial x_j} + \tau_{ij}^{SGS} \right) \qquad (i, j = 1,2,3) \quad \text{in } \Omega^f \tag{8}$$

$$\frac{\partial \overline{p}}{\partial t} + \overline{v}_j \frac{\partial \overline{p}}{\partial x_j} + \rho c^2 \frac{\partial \overline{v}_j}{\partial x_j} = 0 \qquad (j = 1,2,3) \quad \text{in } \Omega^f \tag{9}$$

where τ_{ij}^{SGS} are components of the Reynolds sub-grid stress tensor (which is associated to unsolved sub-grid terms that must be modeled) and overbars indicate large scale variables. The Reynolds sub-grid tensor is usually approximated according to the Boussinesq assumption:

$$\tau_{ij}^{SGS} = \rho\left(\overline{v_i' v_j'} \right) = 2\mu_t \overline{S}_{ij} \tag{10}$$

where commas indicate sub-grid scale variables, $\overline{S}_{ij}$ are components of the large scale strain rate tensor and μ_t is the eddy viscosity, which is obtained in this work employing the dynamic sub-grid scale model (see Germano et al., 1991 and Lilly, 1992) as shown below:

$$\mu_t = \rho C(\vec{x}, t)\overline{\Delta}^2 \left| \overline{S} \right| \tag{11}$$

where $C(\vec{x}, t)$ is the dynamic coefficient (with $\vec{x}$ and t indicating space and time dependencies), $\left|\overline{S}\right|$ is the filtered strain rate tensor modulus and $\overline{\Delta}$ is the characteristic dimension of the grid filter, which is associated to element volumes in 3-D FEM formulations ($\overline{\Delta}^{ele} = \sqrt[3]{vol^{ele}}$). The dynamic coefficient is updated along the time integration process taking into account the instantaneous condition of the flow field (see [a]Braun and Awruch, 2009 for additional information).

Aerodynamic forces are developed over the surface of structures immersed in a fluid stream. These forces are usually obtained by integration of pressures and shear stresses developed on the fluid-structure interface owing to the flow action. The components of the aerodynamic forces in the along-flow and cross-flow directions are referred to as drag and lift, respectively. In the present work, the aerodynamic force and moment coefficients are evaluated using the formulae presented below:

$$C_{Fx} = \frac{\sum_{i=1}^{NNI} (F_x)^i}{1/2\,\rho V_\infty^2 HW}; \quad C_{Fy} = \frac{\sum_{i=1}^{NNI} (F_y)^i}{1/2\,\rho V_\infty^2 HL}; \quad C_{Fz} = \frac{\sum_{i=1}^{NNI} (F_z)^i}{1/2\,\rho V_\infty^2 WL} \tag{12}$$

$$C_{Mz} = \frac{\sum_{i=1}^{NNI} (F_y\Delta_x - F_x\Delta_y)^i}{1/2\,\rho V_\infty^2 HLW}; \quad C_{Mx} = \frac{\sum_{i=1}^{NNI} (F_z\Delta_y - F_y\Delta_z)^i}{1/2\,\rho V_\infty^2 H^2 L}; \quad C_{My} = \frac{\sum_{i=1}^{NNI} (F_x\Delta_z - F_z\Delta_x)^i}{1/2\,\rho V_\infty^2 H^2 W} \tag{13}$$

with:

$$(\Delta_x)^i = X_i - X_g \quad ; \quad (\Delta_y)^i = Y_i - Y_g \quad ; \quad (\Delta_z)^i = Z_i - Z_g \tag{14}$$

where X_i, Y_i and Z_i are Cartesian global coordinates of a nodal point i on the fluid-structure interface, X_g, Y_g and Z_g are Cartesian global coordinates of a reference point of the body, such as the gravity center, V_∞ is the flow reference speed, NNI is the number of fluid nodal points on the body surface and L, W and H are characteristic dimensions related to length, width and height of the immersed body, respectively. The aerodynamic forces F_x, F_y and F_z at a nodal point i are obtained by numerical integration of Eq. (6) over the body surface.

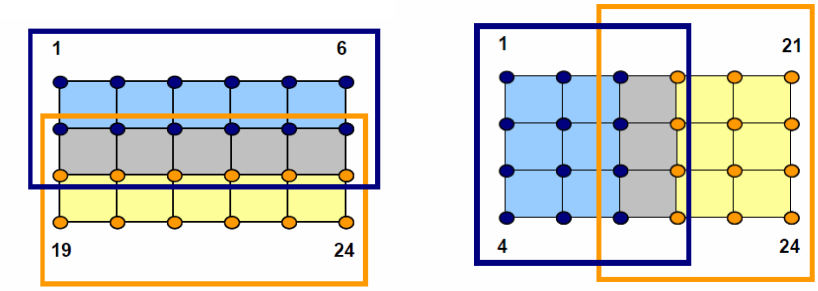
The explicit two-step Taylor-Galerkin scheme is employed in this work for the time discretization of the flow governing equations. The final form of the numerical model is obtained applying the Bubnov-Galerkin's weighted residual scheme into the FEM context on the discrete forms of the flow governing equations. Eight-node hexahedral elements are used for spatial approximations employing the one-point quadrature technique for the evaluation of

element matrices. A complete description of the numerical algorithm utilized in this work may be found in [a]Braun and Awruch (2009) and [b]Braun and Awruch (2009).

3. PARALLELIZATION METHODOLOGY

The problem associated to parallelization on distributed memory may be summarized to procedures related to work partitioning among the logical processors involved. For finite element models, the work partitioning procedure is based on the finite element mesh, where partitions may be obtained in terms of elements or nodal points. Parallel systems with distributed memory are usually characterized by powerfull processors interconected by logical networks with relatively low frequencies, where communications among processors are minimized by using redundancy of work (Masuero, 2009).

Implementation of numerical codes for parallel systems with distributed memory may be performed using data transfer procedures with or without redundancy. Redundancy is obtained when the data associated to each logical processor of the parallel system are superposed, leading to loss of efficiency. For processes running on distributed memory architecture with powerfull processors and relatively low network devices, communications are usually minimized and redundacy is consequently increased. Since finite element models are usually implemented employing element level quantities, work partitioning based on nodes leads to redundancy, which is related to elements presenting nodes associated to different logical processors. A simple example is shown in Fig. 1a, where a mesh with 24 nodes and 15 elements is equally divided into two logical processors (blue and yellow dots). Blue and yellow rectangles indicate elements belonging to the respective logical processors utilized here and grey rectangles indicate common elements, which are also referred to as ghost elements. Blue and yellow external lines are showing the nodes needed by each logical processor. By using this partitioning model, the logical processors are responsible for 10 elements, respectively, where the first logical processor is related to the blue nodes (1 to 12) and the latter is related to the yellow nodes (13 to 24). Therefore, the partitioning efficiency of the present example can be estimated considering that each processor is responsible for 10 elements, which leads to the following relation between the number of elements observed in the entire mesh and the total number of elements associated to the logical processors: 15/20 = 75%. Data referred to six nodal points are required by each logical processor in this case.



(a) (b)

Figure 1. Partitioning models considering two logical processors and a finite element mesh with 24 nodes and 15 elements: (a) nodal ordering follows horizontal lines; (b) nodal ordering follows vertical lines.

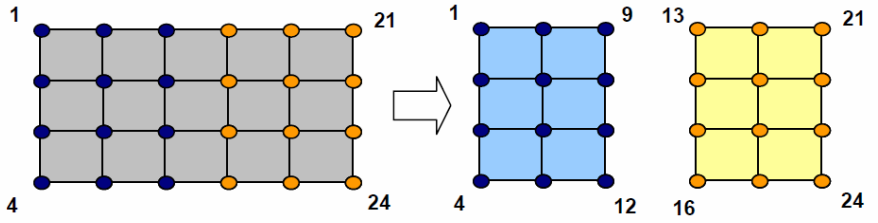
An alternative to the partitioning scheme presented above is shown in Fig. 1b, where each logical processor is associated to 9 elements. The partitioning efficiency corresponds now to the following expression: 15/18 = 83.3%. Network communications are performed considering the eight nodes associated to the ghost elements.

Nodal partitioning procedures may be also interpreted as an operation applied over the system of governing equations, where the number of nodal variables belonging to the ghost elements is related to the bandwidth of the referred system. Consequently, the most efficient partition scheme is that where the bandwidth corresponds to the smaller value. According to the work presented by Masuero (2009), the partitioning procedure can be carried out employing two alternatives related to the bandwidth minimization of the system of governing equations: (a) nodal partitioning based on a single nodal reordering performed with a bandwidth minimization algorithm; (b) nodal partitioning based on recursive nodal reordering performed with a bandwidth minimization algorithm.

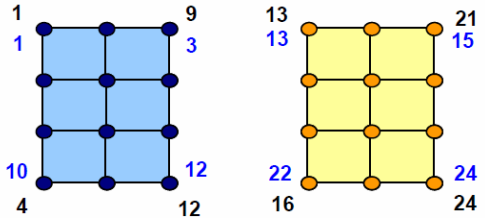
In the first approach, the finite element mesh is previously submitted to a single nodal reordering, which is obtained considering a bandwidth minimization algorithm and nodal partitioning proportional to the number of logical processors utilized in the parallel analysis. This methodology is very useful for problems characterized by a reduced number of logical processors and finite element meshes presenting a predominant coordinate direction. Moreover, clusters constituted by machines with multicore processor technology are also recommended, since communications through the network interface are only performed considering one logical processor per machine. The remaining logical processors establish communication through the local memory.

The second approach is similar to the RCB (Recursive Coordinate Bisection) scheme presented by Fox et al. (1994), where successive bisection operations based on geometrical characteristics of the finite element mesh are adopted. However, the bisection procedure is performed here considering the bandwidth of the system of governing equations in order to identify the bisection direction. The algorithm may be described as follows: (a) a nodal reordering is first

performed over the original finite element mesh, which is divided into 2 groups of nodal points; (b) the same procedure described in the first step is performed over the individual groups obtained formerly; (c) the second step is repeated until the number of groups obtained be in accordance with the number of logical processors to be utilized in the parallel analysis. An example is illustrated in Figs. 2a and 2b, where the present method is applied considering a simple finite element mesh and two logical processors. This methodology is recommended for situations where the number of logical processors available is high. In this case, the predominant coordinate direction of the finite element mesh cannot be identified anymore after successive subdivisions of the original mesh.



(a)



(b)

Figure 2. Nodal partitioning based on recursive nodal reordering for two logical processors: (a) first step; (b) second step.

Parallel processing on distributed memory architecture is carried out using the network interface for communications among the logical processors. This aspect may lead to important losses for the parallelization efficiency, where the relation between the time required for data transferring and the time required for data processing plays an important role. When the number of logical processors adopted in a parallel analysis is relatively high, communication orders must be established to avoid losses of time associated to communication queues. Masuero (2009) presented a communication scheme for a general situation where *n* logical processors have to communicate with the remaining logical processors (or some of them), which may be described as follows:

1) The logical processors considered in the parallel analysis are set in a ring arrangement;
2) A matrix containing communication steps for each logical processor is obtained considering the following expression:

$$\mathrm{INC}_{ij} = -k(-1)^{\mathrm{int}\left(\frac{j+k-1}{k}+l\right)} \qquad \text{with:} \begin{cases} k = \mathrm{int}\left(\dfrac{i+1}{2}\right) \\ l = \mathrm{mod}\left(\dfrac{i+1}{2}\right) \end{cases} \tag{15}$$

where line *i* represents the communication step for a specific logical processor denoted by column *j*, *int* is the integer result obtained from the operation $\frac{i+1}{2}$ and *mod* is the rest of that result.

3) The present procedure begins by adding the values observed in the first row of the matrix of communication steps to the number of their respective logical processors, which are identified from their column positions. The communication is established with that logical processor whose number is obtained from the previous operation.
4) If exchange of data is not expected with the logical processor indicated in the matrix of communication steps, the present logical processor goes to the next communication step (next row in the matrix of communication steps). For a specific communication step, each logical processor may be executing a different communication step (a different row in the matrix of communication steps).
5) If a reciprocal communication is not observed, the present logical processor waits for the next reciprocal communication.
6) If a reciprocal communication is observed, communication is performed and every logical processor goes to the next communication step.
7) The steps 4 to 6 are repeated until all logical processors have completed their *n*-1 communication steps.

An example showing an arrangement with 8 logical processors communicating each other is illustrated in Fig. 3. Continuous lines indicate reciprocal communications between logical processors and dashed lines stands for a logical processor initiating communication without reciprocity, which is identified by a dashed square. Logical processors with

all communications completed are represented by hexagons fulfilled with grey colour. Considering that the number of minimum communication steps for an ideal arrangement with *n* logical processor is *n*-1, the number of penalty communications is defined as the number of communication steps exceeding that minimum value. In this case, 8 communication steps and 1 penalty step are observed. It is important to notice that communications are not generally performed among all logical processors.

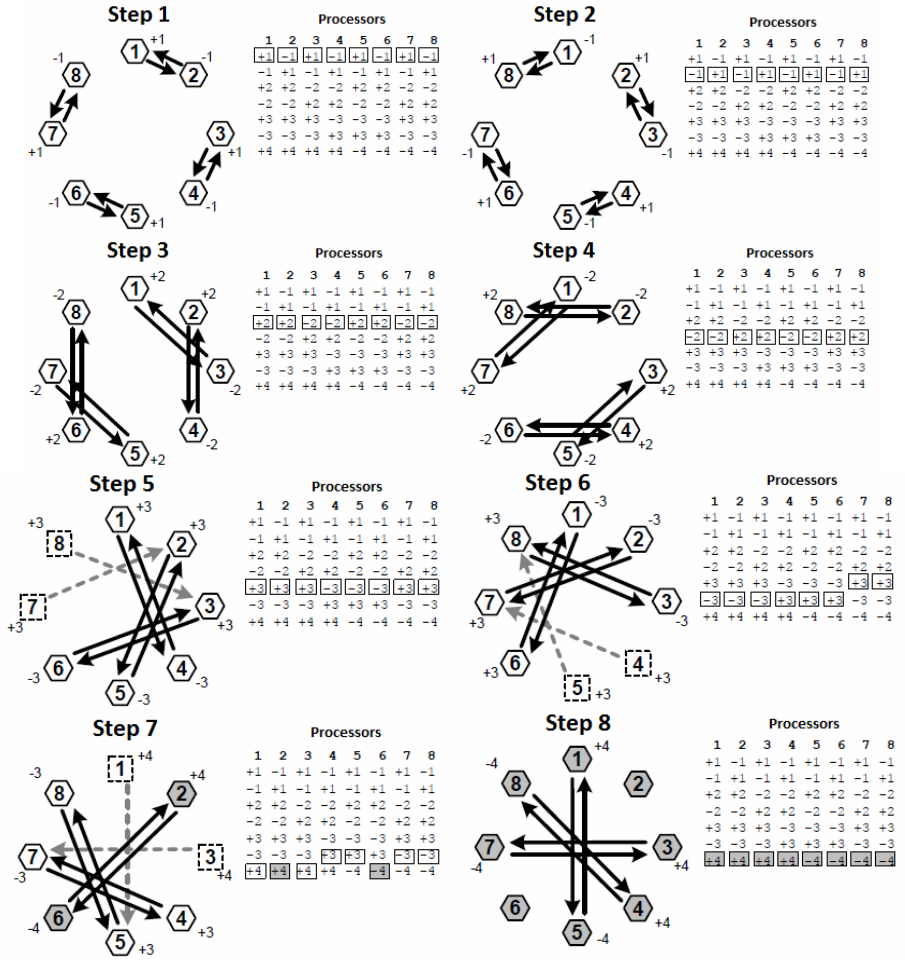


Figure 3. Communication steps considering an arrangement with 8 logical processors communicating each other.

The cluster utilized in this work was implemented using personal computers with Intel Core2 Quad 3700 Mhz processors and 8 Gb RAM. Each individual machine is connected to a Gigabit Ethernet 100/1000 Mbps switch. The opertional system is Windows XP Professional. The present numerical code was written utilizing the FORTRAN90 language and the Compaq Visual Fortran Compiler version 6.6c with the MPICH library version 1.2.5. The parallelization model adopted here is that where no master machine is employed, which is also referred as to hostless program. Consequently, all logical processors are running the same code and data are transferred among the logical processors involved in the parallel analysis. In addition, each logical processor can access all data on their local hard disc. The work charge for each logical processor is defined considering nodal partitioning and the number of logical processors utilized, where nodal reordering based on bandwidth minimization is also employed. Since the same computational characteristics are adopted by all nodes present in the cluster, the work is uniformly distributed.

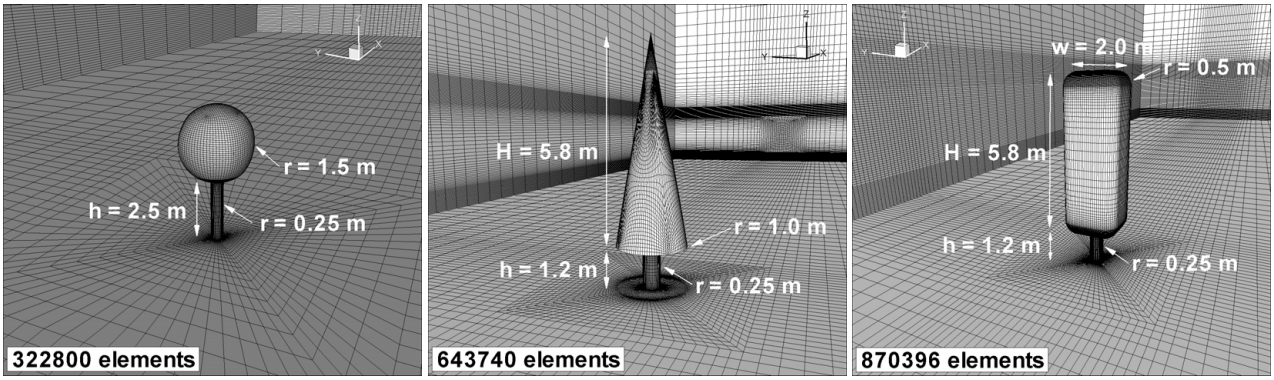
Further details about parallelization procedures and the numerical algorithm utilized in this work may be found in Braun (2007) and Masuero (2009).

4. NUMERICAL SIMULATIONS

Three simple models are adopted in this work in order to investigate the wind action over trees. Although porosity should be considered in the tree canopies owing to the foliage, the present study neglects this aspect. Consequently, the results obtained here may be useful for tree species with dense foliage or under extreme winter conditions, where the tree canopies are usually covered with snow.

Figure 4 shows the finite element meshes and some geometrical properties utilized in the present simulations. The physical constants representing the fluid and time steps employed in the time discretization are shown in Table 1, leading to a Reynolds number ($Re = \rho VD/\mu$) = 10^5 for all cases analyzed here. Velocity boundary conditions are imposed on the front wall of the computational domains in order to reproduce an atmospheric boundary layer flow on

the incident wind, which is considered according to a velocity profile given by the power law equation for smooth flows, where velocity fluctuations are disregarded (see Eq. 7). In addition, non-slip conditions are imposed on the tree surfaces and on the ground. Symmetry boundary conditions are applied on the side walls and constant pressure is considered on the back walls.



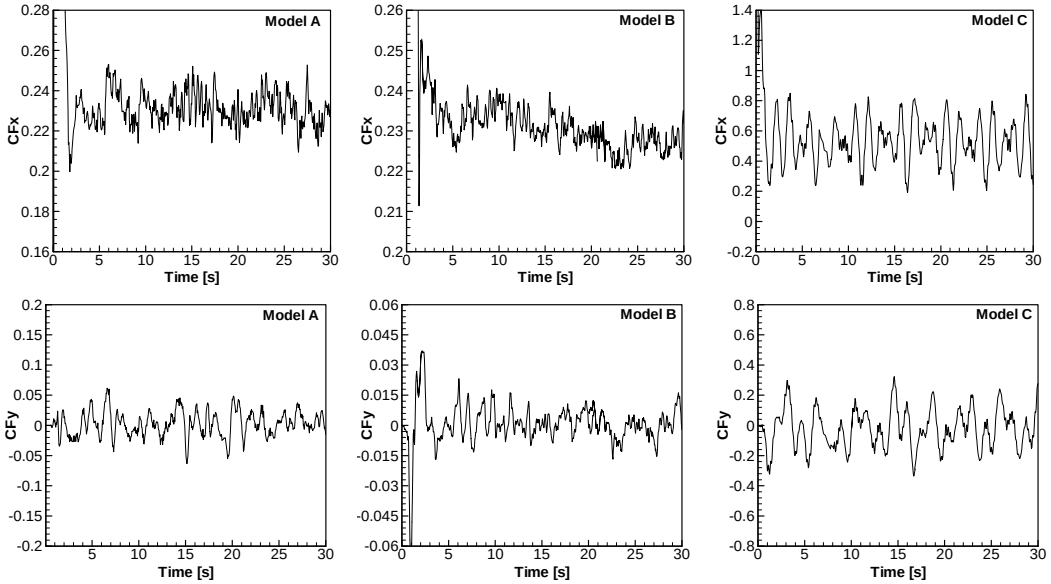
Model A Model B Model C

Figure 4. Finite element meshes and geometrical configurations utilized in the numerical simulations.

Table 1. Physical constants and additional parameters adopted in the present simulations.

Parameter	Model		
	A	B	C
Specific mass (ρ) [Kg/m^3]	1.25	1.25	1.25
Dynamic viscosity (μ) [N.s/m^2]	$6.7\text{x}10^{-4}$	$2\text{x}10^{-4}$	$2\text{x}10^{-4}$
Characteristic dimension (D) [m]	3.0	2.0	2.0
Characteristic wind speed (V) [m/s]	22.2	9.46	9.46
Model height (h) [m]	5.5	7.0	7.0
Reference Height (H) [m]	5.5	9.0	9.0
Reference wind speed (U_H) [m/s]	22.2	10.0	10.0
Exponent p (see Eq. 7)	0.22	0.22	0.22
Time step (Δt) [s]	$5\text{x}10^{-5}$	$2.5\text{x}10^{-5}$	$4\text{x}10^{-5}$

Time histories of some of the aerodynamic coefficients measured during the present simulations are shown in Fig. 5. Comparisons are then performed in Table 2, where time-average drag values calculated from the results presented in Fig. 5 are listed together with experimental and numerical predictions obtained by other authors. The results obtained with the numerical model proposed in this work agree reasonably with those obtained by the remaining references. It is important to notice that the present results are observed in the range of values presented by Mayhead (1973), who carried out extensive experimental analyses over several tree models.



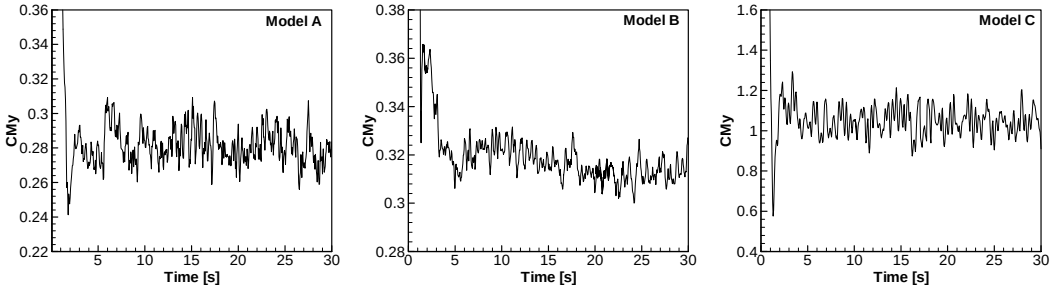


Figure 5. Aerodynamic coefficient records obtained during the numerical simulations.

Table 2. Drag coefficients for tree models.

Source	Drag coefficient
Present work – Model A	0.24
Present work – Model B	0.23
Present work – Model C	0.55
Mayhead (1973) – various models	0.17 – 0.56
Koizumi (2010) – various models	0.17 – 0.33
Gardiner et al (2000) – similar to Model B	0.29
Yoshida et al. (2006) – similar to Model A	0.20

In order to evaluate the flow characteristics near the model region, time-average velocity profiles along the center line of the computational domain are shown in Fig. 6. Results obtained here are presented in the longitudinal direction, which is parallel to the wind stream. Every profile position is referred to the tree position by using the relative coordinate x/h, where h is the model height (see Table 1). In addition, time-average pressure fields obtained with the present numerical model are also shown in Fig. 7, which are represented on the symmetry plane of the computational domain.

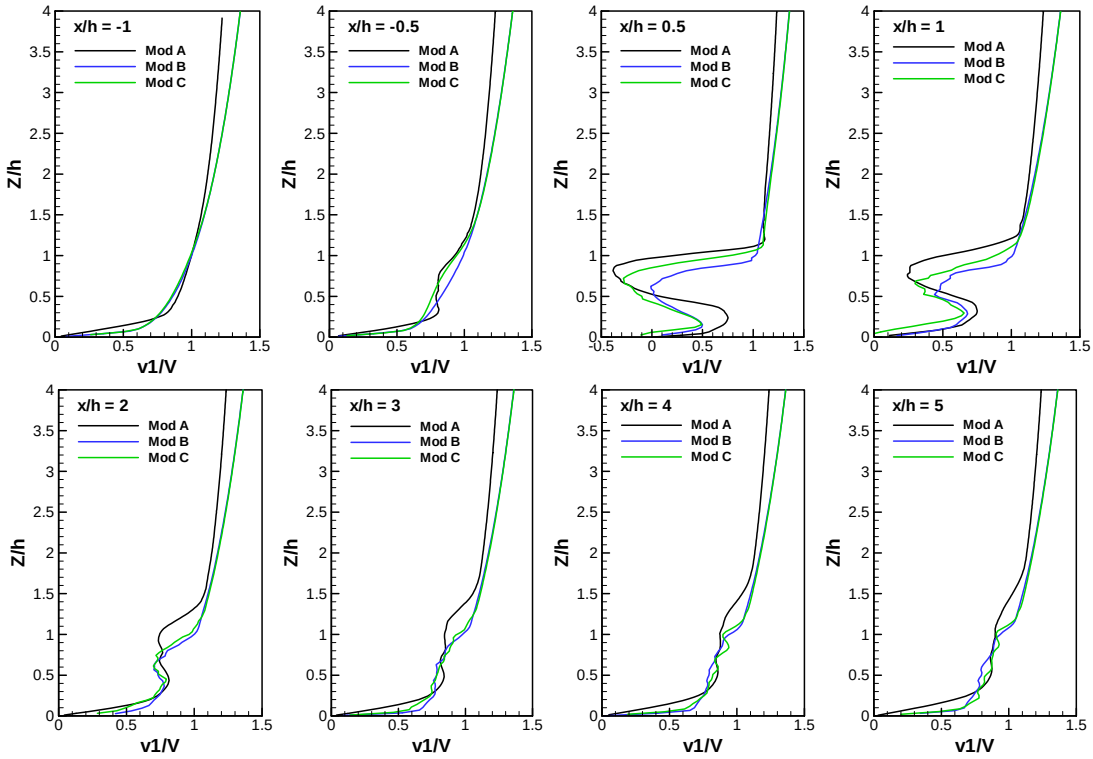


Figure 6. Velocity profiles obtained at different positions along the center line of the computational domain.

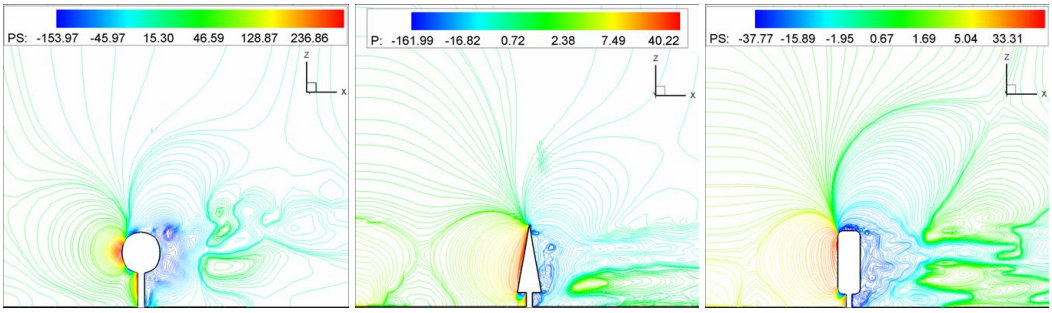


Figure 7. Time-average pressure fields obtained on the symmetry plane of the computational domain.

The wind environmental conditions around the tree models may be evaluated from Fig. 8, where instantaneous streamlines obtained at t = 30s are shown. It is observed that the present formulation was able to reproduce most of the circulation patterns typically developed in the surroundings of bluff bodies immersed in wind streams with atmospheric boundary layer characteristics. Aerodynamic phenomena such as recirculation zones behind and in front of the bodies and separation/attachment zones were reproduced in accordance with experimental observations (see Simiu and Scanlan, 1996).

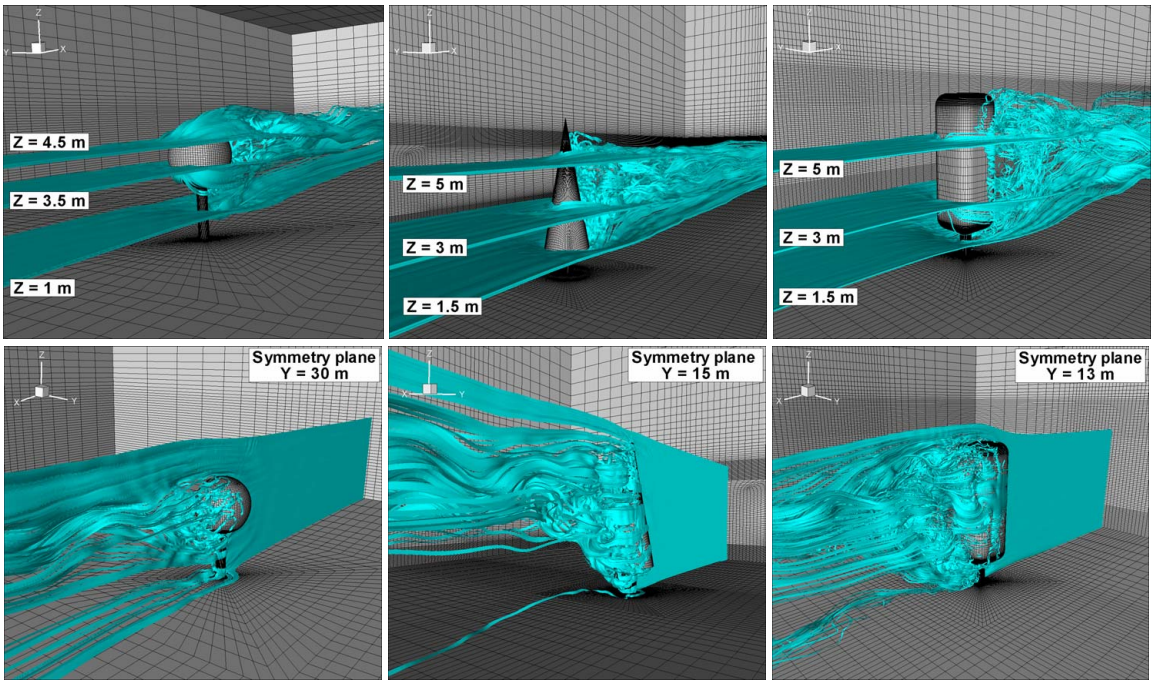


Figure 8. Instantaneous streamlines obtained at t = 30s.

The computational performance of the parallelization scheme adopted in this work may be verified using some parameters such as Speed-up (Sp) and Efficiency (Ep), which is evaluated as function of the number of logical processors (p) and number of finite elements. In Fig. 9, the Speed-up and Efficiency curves obtained in this study are presented.

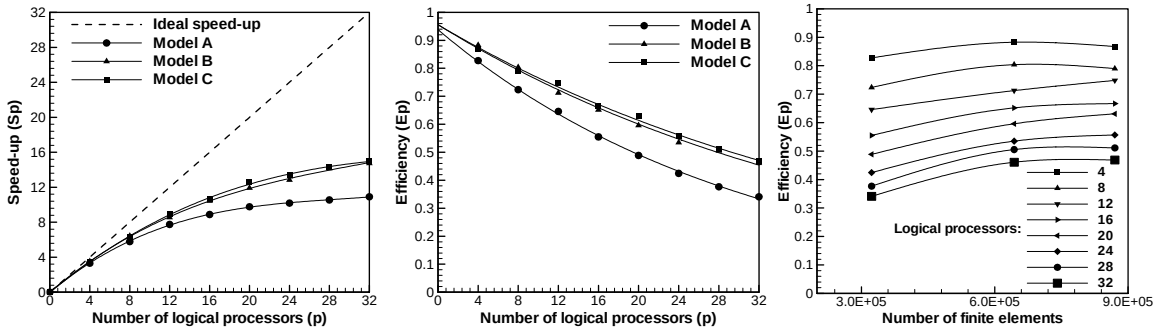


Figure 9. Computational performance obtained by the parallelization scheme utilized in this work. Speed-up: Sp = T_1/T_p, T_1 – execution time corresponding to the sequential algorithm; T_P – execution time corresponding to the parallel algorithm with p logical processors; Efficiency: Ep = Sp/p.

5. CONCLUSIONS

In the present work, numerical simulations to investigate the wind action over tree models were carried out using a numerical model to analyze wind engineering problems. A parallelization methodology was also introduced in order to improve the algorithm performance for applications characterized by dense finite element meshes. The results obtained here demonstrate that the present model can be applied to tree aerodynamics, although some additional features must be implemented in the numerical code to take into account important aspects associated to porosity of the tree canopy. Moreover, the parallelization methodology utilized in this work reduced significantly the execution time required by the sequential algorithm. Nevertheless, Speed-up (Sp) and Efficiency (Ep) evaluations performed here indicate that some modifications are necessary to improve the computational performance when a large number of logical processors are employed in the parallel analysis.

6. ACKNOWLEDGEMENTS

The authors wish to thank CNPq (Brazilian council of research) for the financial support.

7. REFERENCES

Braun, A.L., 2007, "Simulação Numérica na Engenharia do Vento Incluindo Efeitos de Interação Fluido-Estrutura", Tese de Doutorado – PPGEC/UFRGS, Porto Alegre, Brasil.

Braun, A.L. and Awruch, A.M., 2005, "Aerodynamic and aeroelastic analysis of bundled cables by numerical simulation", Journal of Sound and Vibration, Vol. 284, pp. 51-73.

[a]Braun, A.L. and Awruch, A.M., 2009. "Aerodynamic and aeroelastic analyses on the CAARC standard tall building model using numerical simulation", Computers and Structures, Vol. 87, pp. 564-581.

[b]Braun, A.L. and Awruch, A.M., 2009. "A partitioned model for fluid-structure interaction problems using hexahedral finite elements with one-point quadrature", International Journal for Numerical Methods in Engineering, Vol. 79, pp. 505-549.

Endalew, A.M., Hertog, M., Delele, M.A., Baetens, K., Persoons, T., Baelmans, M., Ramon, H., Nicolaï, B.M. and Verboven, P., 2009. "CFD modelling and wind tunnel validation of airflow through plant canopies using 3D canopy architecture", International Journal of Heat and Fluid Flow, Vol. 30, pp. 356-368.

Fox, G.C., Williams, R.D. and Messina, P.C., 1994, "Parallel Computing Works", Morgan Kaufmann Publishers, San Francisco, USA.

Gardiner, B., Peltola, H. and Kellomäki, S., 2000, "Comparison of two models for predicting the critical wind speeds required to damage coniferous trees", Ecological Modelling, Vol. 129, pp. 1-23.

Germano, M., Piomelli, U., Moin, P. and Cabot, W.H., 1991, "A dynamic subgrid-scale eddy viscosity model", Physics of Fluids, Vol. A3, No. 7, pp. 1760-1765.

Kawahara, M. and Hirano, H., 1983, "A finite element method for high Reynolds number viscous fluid flow using two step explicit scheme", International Journal for Numerical Methods in Fluids, Vol. 3, pp. 137-163.

Koizumi, A., Motoyama, J. and Sawata, K., 2010, "Evaluation of drag coefficients of poplar-tree crowns by a field test method", Journal of Wood Science, Vol. 56, pp. 189-193.

Liang, L., Xiaofeng, L., Borong, L. and Yingxin, Z., 2006. "Improved k-ε two-equation turbulence model for canopy flow", Atmospheric Environment, Vol. 40, pp. 762-770.

Lilly, D.K., 1992, "A proposed modification of the Germano subgrid-scale closure method", Physics of Fluids, Vol. A4, No. 3, pp. 633-635.

Masuero, J.R., 2009, "Computação Paralela na Análise de Problemas de Engenharia Utilizando o Método dos Elementos Finitos", Tese de Doutorado – PPGEC/UFRGS, Porto Alegre, Brasil.

Mayhead, G.J., 1973, "Some drag coefficients for British forest trees derived from wind tunnel studies", Agricultural Meteorology, Vol. 12, pp. 123-130.

Mochida, A. and Lun, I.Y.F., 2008. "Prediction of wind environment and thermal comfort at pedestrian level in urban area", Journal of Wind Engineering and Industrial Aerodynamics, Vol. 96, pp. 1498-1527.

Mochida, A. and Tabata, Y., Iwata, T. and Yoshino, H., 2008. "Examining tree canopy models for CFD prediction of wind environment at pedestrian level", Journal of Wind Engineering and Industrial Aerodynamics, Vol. 96, pp. 1667-1677.

Simiu, E. and Scanlan, R., 1996, "Wind Effects on Structures", Wiley, New York, USA.

Smagorinsky, J., 1963, "General circulation experiments with the primitive equations, I, the basic experiment", Monthly Weather Review, Vol. 91, pp. 99-135.

Yoshida, S., Ooka, R., Mochida, A., Murakami, S. and Tominaga, Y., 2006, "Development of the three dimensional plant canopy model for numerical simulation of outdoor thermal environmental", In: proceeding of the ICUC6, Sweeden.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.