UrbaWind, a Computational Fluid Dynamics tool to predict wind resource in urban area

Karim FAHSSIS[a], Guillaume DUPONT[a], Pierre LEYRONNAS[a]

[a] Meteodyn, Nantes, France
Presenting Author: Karim.fahssis@meteodyn.com, Tel: +33 240 71 05 05
Corresponding Author: Guillaume.dupont@meteodyn.com, Tel: +1 267 475 4278

Abstract:
Computational Fluid Dynamics (CFD) is already a necessary tool for modeling the wind over complex rural terrains. In order to maximize energetic yield and optimize the costs, before installing the wind energy systems, a good knowledge of wind characteristics on site is required. Meteodyn has developed UrbaWind, which is an automatic CFD software for computing the wind in urban environment for small wind turbines.
Compared to rural open spaces, the geometry in urban areas is more complex and unforeseeable. The effects created by the buildings, such as vortexes at the feet of the towers, Venturi effect or Wise effect, make the modeling of urban flows more difficult. The model used in UrbaWind allows to take these effects into account by solving the equations of Fluid Mechanics with a specific model that can represent the turbulence and the wakes around buildings as well as the porosity of the trees and the effects of the ground roughness (asphalt, water or grass).
In order to validate UrbaWind's results, different study cases proposed by the Architectural Institute of Japan have been set up. The three selected cases have an ascending complexity, from the simple block to the complete rebuilding of a quarter of the Japanese city of Niigata.
The results validate UrbaWind as well for theoretical cases as for real cases by offering a minor error margin on the wind speed prediction.

Keywords: Wind modeling, CFD, urban wind energy

1.1 INTRODUCTION

In order to validate UrbaWind's results, different study cases based on practical uses proposed by the Architectural Institute of Japan [1] have been set up. The three selected cases have an ascending complexity, from the simple block to the complete rebuilding of a quarter of the Japanese city of Niigata. These published cases allowed to compare the computations results of UrbaWind with the measurements of the AIJ.

Therefore, the sites have been recreated in a file format that can be used by UrbaWind and the conditions taken for the CFD computations are as similar as possible to the experiments conditions. The values obtained by the computations are then compared with the measured values.

At first a graphical comparison is done by placing the points in a graph with the computed values in ordinate and the measured values in abscise (regression line).

Statistical indicators of the results accuracy are proposed as followed:

- The error, calculated by the formula:

$$Err = \frac{\sum_n |y - x|}{n}$$

- The determination coefficient R^2, which corresponds to the square of the Pearson product-moment correlation coefficient :

$$R^2 = \frac{(\sum (x - \bar{x})(y - \bar{y}))^2}{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}$$

This last value is the square of the covariance of the reduced centered variables. Its value is between 0 and 1. The value 0 means a total independence between both variables.

In these equations as well as in the whole document, *y* corresponds to the computed values and *x* to the measured values, both normalized.

In every case, results are considered for all points in a first time and only for consistent points in a second time. For the consistent points, we exclude all the very low speed points, which lead to the biggest error and for which a great accuracy is not required when considering the installation of wind turbines. The difference between the results obtained for both all points and consistent points is summarized in a table in the conclusion.

1.2 DESCRIPTION OF THE CFD CODE URBAWIND

1.2.1 The Equations

UrbaWind solves the equations of Fluid Mechanics, i.e. the averaged equations of mass and momentum conservations (Navier-Stokes equations). When the flow is steady and the fluid incompressible, those equations become:

$$\frac{\partial \rho \bar{u}_i}{\partial x_i} = 0 \tag{1}$$

$$-\frac{\partial \left(\rho \bar{u}_j \bar{u}_i\right)}{\partial x_j} - \frac{\partial \bar{P}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \rho\overline{u'_i u'_j}\right] + F_i = 0 \tag{2}$$

The turbulent fluxes are parameterized by using the so-called turbulent viscosity. The turbulent viscosity μ is considered as the product of a length scale by a speed scale which are both characteristic lengths of the turbulent fluctuations. The speed scale is given as the square root of the turbulent kinetic energy multiplied by the density. The turbulent kinetic energy is solved using the transport equation by including the production and dissipation terms of the turbulence where the turbulent length scale L_T varies linearly with the distance at the nearest wall (ground and buildings).

Boundary conditions are automatically generated. The vertical profile of the mean wind speed at the computation domain inlet is given by the logarithmic law in the surface layer, and by the Ekman function [2]. A 'Blasius'-type ground law is implemented to model frictions (velocity components and turbulent kinetic energy) at the surfaces (ground and buildings).

The effect of porous obstacles is modeled by introducing a sink term in the cells lying inside the obstacle:

$$\vec{F} = -\rho \cdot V.C_d.\left|\vec{U}\right|.\vec{U} \qquad (3)$$

1.2.2 Mesh generation

This meshing is generated by following the computed wind direction. The size of the computation domain is automatically generated so that the minimum distance between a building and a boundary condition is equal to 6 times the height of the highest building of the simulation. The generated meshing is Cartesian and unstructured (using of overlapped meshing), with automatic refinement near the ground, obstacles and of course at result points locations.

For this study, the vertical mesh dimension around the blocks in the two first cases (simple block and group of blocks) has been taken equal to 20 cm. The mesh dimension for the case of City of Niigata was 50cm in the horizontal and vertical directions for the refined cells around the buildings.

1.2.3 MIGAL-UNS Solver

1.2.3.1 DESCRIPTION

The MIGAL-UNS solver has been regularly used for some years now, and has already been fully validated on number of academic cases [3].

MIGAL is an iterative linear equations solver which updates simultaneously the wind speed components as well as the pressure on the whole computational domain, so-called “coupled resolution”. This method demands more storage capacity, but it has the advantage to dramatically increasing the convergence process.

The discretized equations linear system, is transformed via an incomplete LU decomposition ILU(0) [4]. A preconditioner of type GMRES is used in order to increase the terms in the main diagonal so that the solving robustness is improved.

Moreover, MIGAL uses a multi-grid procedure which consists in solving successively the equations on different grid levels (from the finer one to the coarser one). This method accelerates the convergence of the low frequency errors, which are known to be the limiting factor in the convergence process. For the type of problem solved here, different tests have shown a better convergence for the “V cycles” method than for the “W cycles” method. The number of grid levels is automatically computed by MIGAL.

1.2.3.2 REFINEMENT STRATEGY: ALGORITHM OF AGGLOMERATION

For the multi-grid computation, MIGAL-UNS uses an agglomeration method which joins together and agglomerates control volumes near the fine grid. This process is executed recursively until having generated a whole sequence of coarse meshes.

Once the grid hierarchy is defined, one still has to determine how the equations are created on the coarse meshes and how their solutions will be used in order to decrease the errors of long wavelengths on the finest grids.

MIGAL employs a Galerkin's projection method for generating the equations on the coarse grid. This technique, so-called "Additive Correction Multi-grid" consists in generating the equation of the coarse mesh as the sum of the equations of each corresponding fine cell. Once the solution is obtained on the coarse grid, it is introduced by correcting the values calculated previously on the fine grid with the calculated error.

1.3 THEORETICAL CASES: SIMPLE BLOCK AND GROUP OF BLOCKS

1.3.1 Description

The first study has been done on a square tower of 20 m high with a base of 10 m by 10 m. A wind with a direction of 270° is applied and the computations are done on the result points spread in two plans: a vertical plan located on x=0, which cut a tower on its middle.

The second studied site is built up of 8 blocks of 20 m of edge and an empty space in the center. The computations are done with a wind coming from the West (blue arrow) and the result points are located around the central empty space at two meters high, like shown in the figure 1.

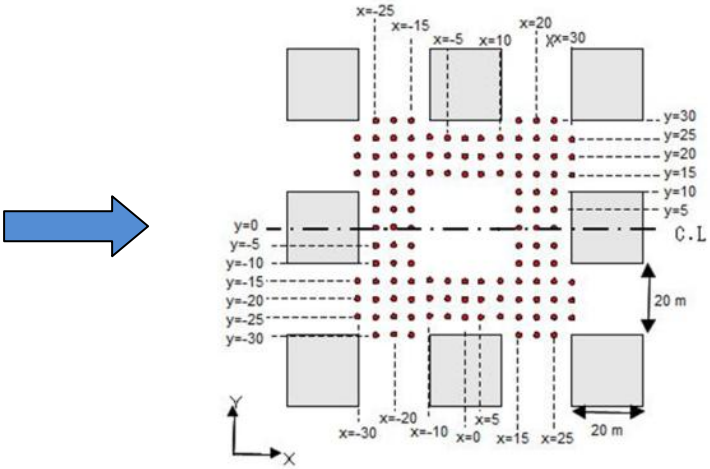

Figure 1 : Results points, z=2m

1.3.2 Results

1.3.2.1 SIMPLE BLOCK

The speed-up coefficients of the computation have been compared to the coefficients obtained by the experimental measurements. In the following figure 2, the computed values functions of the measured values for consistent points have been represented.

Table 1: Comparison Computations vs. Measurements for the simple block

First case – Simple Block	Error	R^2
For all points	5.5%	0.92
For the consistent points	4.9%	0.93

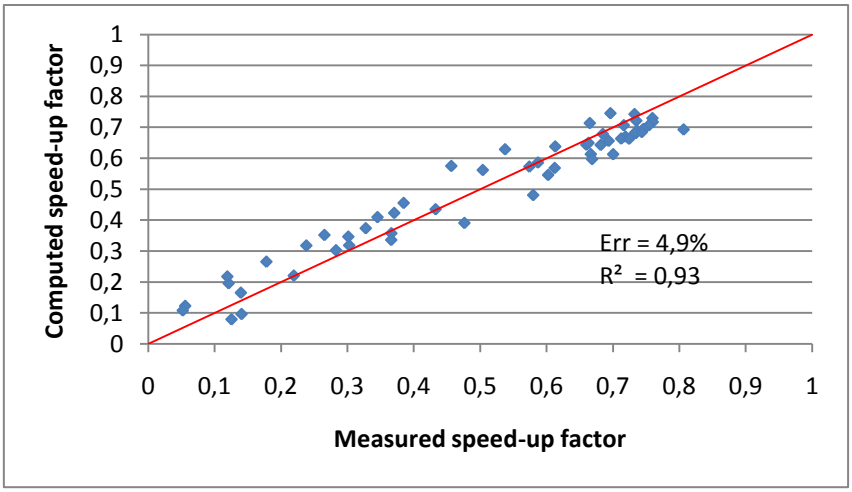

Figure 2: First case, horizontal at z=1,25m, consistent points

1.3.2.2 GROUP OF BLOCKS

The speed-up factors of the computation have been compared with the speed-up factors obtained by the experimental measures. In the following figure 3, the computed values functions of the measured values for consistent points have been represented.

Table 2: Comparison Computations vs. Measurements for the group of blocks

First case – Group of Blocks	Error	R^2
For all points	8.5%	0.70
For the consistent points	5.8%	0.71

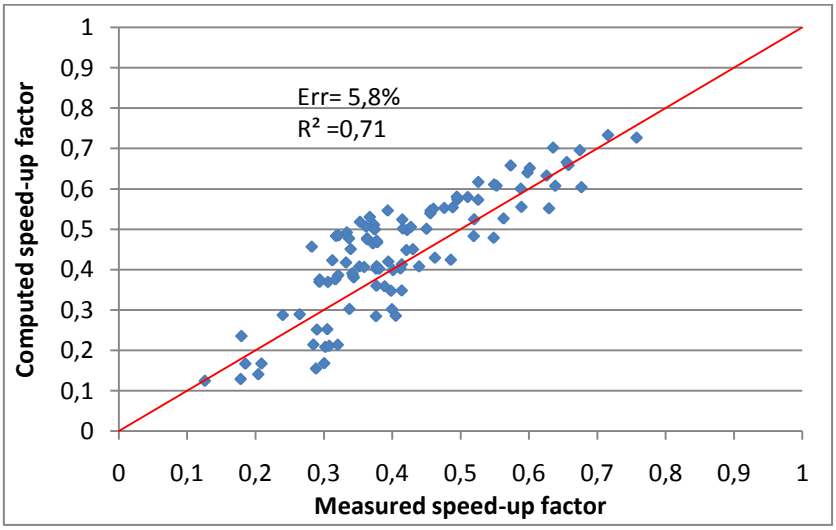

Figure 3: Second case, group of blocks, consistent points

1.3.3 Discussion

The error (5.5% in average for all points and 4.9% when selecting only the consistent points) and the determination coefficient allow to confirm the accuracy of UrbaWind's results for this first case. We can observe that the results provided by the software UrbaWind are very good

for most of the points. Only a few points are far from the linear regression curve. These differences have several explanations.

At first most of the points correspond to low wind speeds. The differences of values are thus proportionally bigger than for higher wind speeds. Furthermore, by visualizing these points, we can notice that these points are in general near the walls in areas where big speed difference can exist between points that are very near. This explains that the computation results can be different without necessarily casting doubt on the general aspect of the simulation.

A 2D map of the mean acceleration around the structure has been represented in the figure 4. The worst points are indicated in red in the figure.

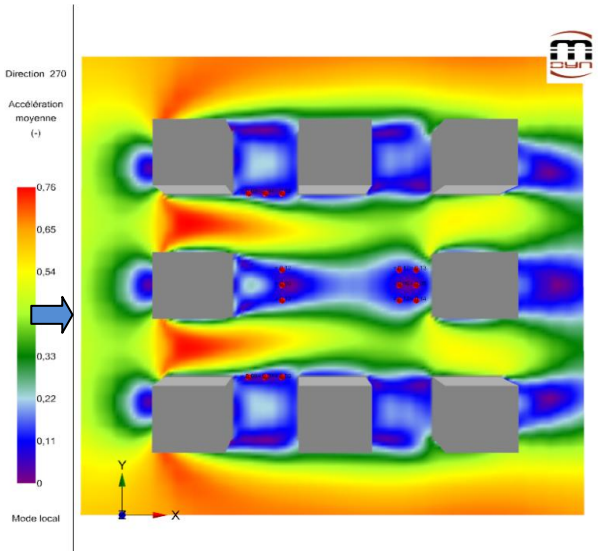

Figure 4: Second case, visualization

We can notice that the less consistent points are the points located in the areas where the mean acceleration is near zero. As all the CFD models, it is in the area near the joining distance that the results are the most difficult to reproduce. Thus, for both theoretical cases, the underestimation of these low values does not cast doubt on the relevance of the computation.
Moreover one can observe a value of the error which is equal to 4.9% for the first case and 5.8% for the second case when selecting only the consistent points, which validates partly the results provided by UrbaWind in both cases.

1.4 PRACTICAL CASE: QUARTER OF NIIGATA (JAPAN)

1.4.1 Description

The last study has been done in the quarter of the Japanese city of Niigata. The wind is coming in the direction of 225° (blue arrow) and results points are placed in several random locations as indicated in the figure 5, at two meters high.

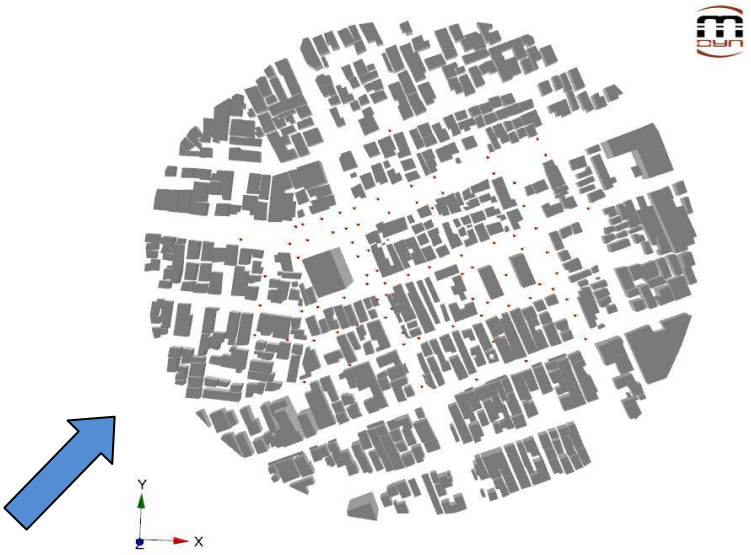

Figure 5: Third case, result points, z=2m

1.4.2 Results

The speed-up factors of the computation have been compared to the speed-up factors obtained by the experimental measurements. In the following figure 6, the computed values functions of the measured values for consistent points have been represented.

Table 3: Comparison Computations vs. Measurements for the city of Niigata

Third case – City of Niigata	Error	R^2
For all points	5.9%	0.75
For the consistent points	5.4%	0.79

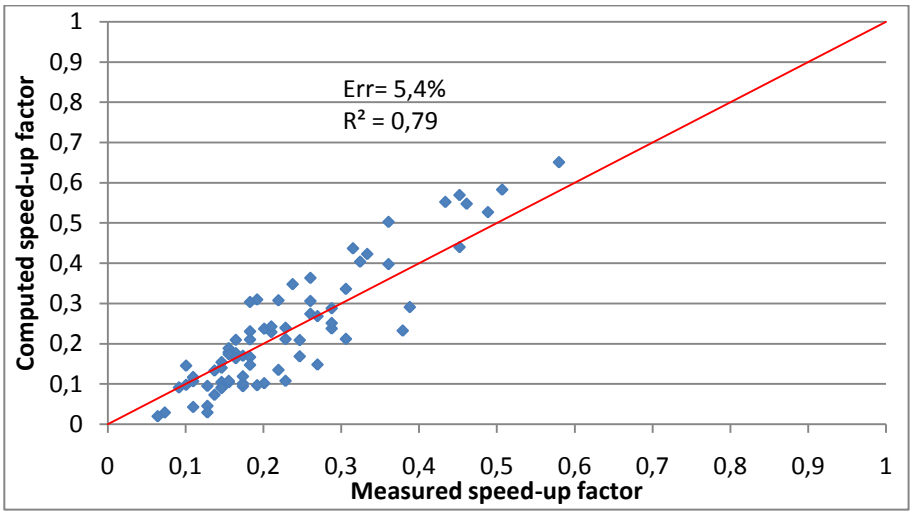

Figure 6: Third case, quarter of Niigata, consistent points

1.4.3 Discussion

Like in the previous cases, the points where the computed value of the speed-up factor is far from the measured value are represented in the software UrbaWind.

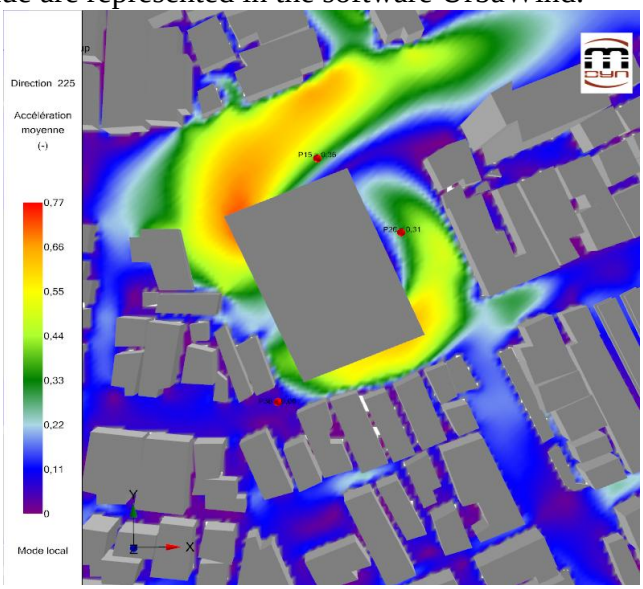

Figure 7: Third case, quarter of Niigata, visualization

The situation is similar to the cases 1 and 2. Indeed, one can observe in the figure 7 that the worst points correspond either to low-speed areas, or to areas where a small change of location can lead to an important speed variation.

Moreover, the error and the determination coefficient allow again to validate the results of UrbaWind. Indeed 90% of the points have an error value lower than 12% and the mean error is lower than 5.9% for all points and 5.4% when selecting only the consistent points.

1.5 CONCLUSION

Finally, the table 1 shows that the results provided by UrbaWind are much closed to the results of the experimental measurements of the Architectural Institute of Japan. Indeed the typical error of the computations when considering all points is at most 8.5% and even decrease to 5.5% in the first case.

Table 4: Summary table of the comparison Computations vs. Measurements

	Error for all points	Error for the consistent points	R^2
First case – Simple Block	5.5%	4.9%	0.93
Second case – Group of Blocks	8.5%	5.8%	0.71
Third case – City of Niigata	5.9%	5.4%	0.79

To conclude, this study allows validating the software UrbaWind as well for theoretical cases (first and second cases) as for real cases (quarter of Niigata) by offering a minor error margin.

Abbreviations:

$\overline{u}_i$: Components of the mean wind vector in a Cartesian frame

u'_i : Components of the turbulent fluctuations

$\overline{P}$: Mean pressure value

μ : Air dynamic viscosity

ρ : Air density

C_d : Volume coefficient of frictions, which is proportional to the porous obstacle density

V : Volume of the considered cell.

References:

[1] Architectural Institute of Japan, 2008, *Guidebook for Practical Applications of CFD to Pedestrian Wind Environment around Buildings*. (http://www.aij.or.jp/jpn/publish/cfdguide/index_e.htm)

[2] Garratt J.R., 1992, *The atmospheric boundary layer*, Cambridge Atmospheric and space sciences series.

[3] Ferry M., 2000, "The MIGAL solver", Proc. Of the Phoenics Users Int. Conf., Luxembourg, 2000

[4] Ferry M., 2002, *New features of the MIGAL solver*, Proc. Of the Phoenics Users Int. Conf., Moscow, Sept. 2002