

# Estimation of airflow characteristics of indoor environments in the early design stage

İlker KARADAĞ<sup>1</sup>, Nuri SERTESER<sup>2</sup>

<sup>1</sup>karadagi@itu.edu.tr • Department of Architecture, Faculty of Architecture, Istanbul Technical University, Istanbul, Turkey

<sup>2</sup>serteser@itu.edu.tr • Department of Architecture, Faculty of Architecture, Istanbul Technical University, Istanbul, Turkey

*Received: July 2018 • Final Acceptance: December 2018*

## Abstract

Estimating indoor airflow characteristics of natural ventilation systems is very significant in the early design stage. It's clear that for the early steps of design, existing numerical and experimental analysis methods are very time consuming and they require in depth knowledge of Fluid Dynamics. These methods are not efficient especially in the case that the building form changes dynamically. Besides, both wind tunnel testing and Computational Fluid Dynamics simulations are not efficient when it comes to taking output in real-time. Due to all of these reasons, a need for a fast and robust method occurs. Particle-based algorithms are efficient methods for this type of analyses however they have not been used in architectural aerodynamics. At this point, a very powerful method which doesn't require mesh (control volume) is developed. In this study, the details of the developed algorithm and the output of it are given. The algorithm was assessed in three case studies of natural ventilation systems. As a result, it is seen that the developed algorithm can be a guide in building-wind interaction analysis for architects in the early design stage. However, in our paper, we do not only present case studies, but also an analysis methodology from architectural and engineering perspectives. This is significant because the methodology and the results of this paper constitute a guide for further researches on natural ventilation with a new method and consequently contribute to improved wind quality of indoor spaces.

## Keywords

Wind efficient design, Architectural aerodynamics, Computational fluid dynamics, Particle based simulation, Natural ventilation.



## 1. Introduction

The significance of considering sustainability in the early design stage meets the need for finding long-term solutions and reducing energy consumption. If an architectural project is well planned and sustainable criteria are included in its early approach, the possibility to reduce negative impacts is greater and the cost of criteria implementation is greatly reduced. Enhancement of the building's sustainability performance should start already in the early design phase since the potential of optimisation in this phase is higher and the influence of changes of the building and the construction costs are lower (Bragança & Andrade, 2014).

There are several environmental parameters in building physics. One of the most important of them is the wind which has an important impact on the indoor comfort of buildings. The building-wind interaction can be estimated with one of three approaches or a combination of these: (1) in-situ measurements, (2) experimental analysis through wind tunnel, or (3) numerical analysis with Computational Fluid Dynamics (CFD) software. Since each approach has its own advantages and disadvantages, it is not always easy to decide which approach is the most appropriate for a given problem. A significant disadvantage of in-situ measurements and wind tunnel measurements is usually to get data only for specific points. In principle, techniques such as Particle Image Velocimetry (PIV) and Laser-Induced Fluorescence (LIF) make it possible to obtain all the data in planar or even three dimensions, but the cost of such technologies is seriously high and application for complicated geometries can be hampered through laser-light shielding by the obstructions constituting the model, e.g. in case of an urban model consisting of many buildings (Blocken & Carmeliet, 2004). In addition, recent studies of comparative studies of different wind tunnel laboratory studies have shown that in many cases there are large differences of up to 50% (NIST Technical Report, 2009). Despite these unfavourable conditions, a wind tunnel

can be very reliable if compliance with international standards such as NIST TN1655 and ASCE / SEI 49-12 and ASCE 7 is provided. These standards define the minimum requirements for conducting and interpreting wind tunnel tests to assess wind loads on buildings and other structures. It is useful for those who are preparing, conducting and commenting on wind tunnel tests for buildings, including civil engineers, architects and wind engineers (ASCE/SEI 49-12, 2012).

Wind flow characteristics can also be determined by CFD simulations. With these numerical simulations, it is possible to obtain high-resolution wind data in a very wide area around the building. In particular, precise results can be obtained with numerical analyses in which meteorological data are taken as velocity input and logarithmic wind profile is used, a suitable turbulence model is determined and a sufficient number of iterations are performed. However, the main problem with the finite element approach is the necessity of calculating a mesh that divides the simulation area. Typically, the meshing phase takes more than 80% of the time of a fluid dynamic simulation (Liu, 2002). In addition, mesh quality is a critical factor in determining the accuracy of the solution. Besides, for such technical software, the geometry must be prepared again to comply with the input requirements. But in architectural practice, complicated models are very common and to re-create these models needs too much effort.

It is known that in the preliminary design phase, existing conventional methods are very time consuming and require a deeper knowledge of fluid dynamics. These methods are not efficient, especially if the building form changes dynamically. In addition, CFD simulations are impractical in terms of evaluating the preliminary design phase when data is taken in real time. For all these reasons, a fast and reliable method is needed. At this point, it is envisaged to write an algorithm that can work in real time, does not have limitations on geometry, and most importantly does not need mesh.

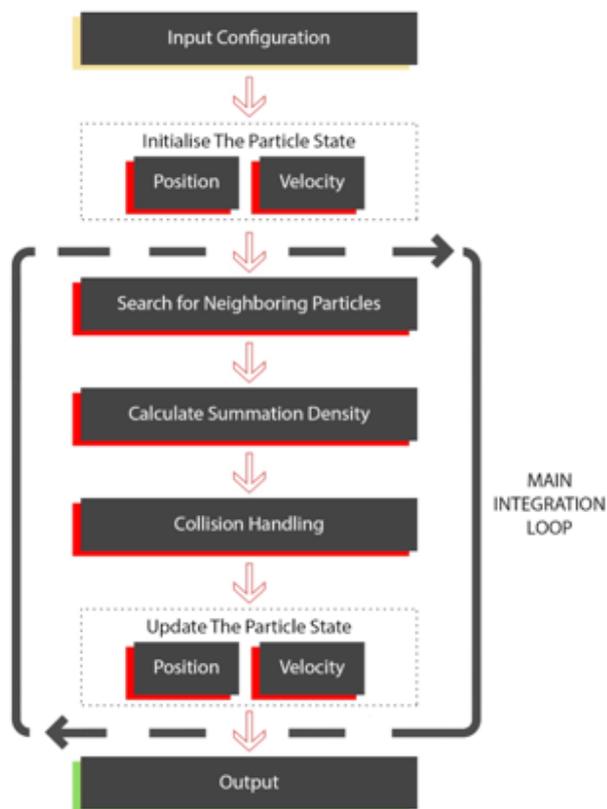


Figure 1. Flowchart.

## 2. Development of a particle-based algorithm

Three different steps have been followed in order to write an algorithm in which the wind-interaction can be solved (Figure 1). In the first step, a simulation model was decided; the second step was how to integrate physical equations; in the third and last stage, the solver (solvent) in which the solution of the equations can be carried out was determined. In the study, the wind current was reduced to particle scale. The next position and speed were estimated according to the initial position and speed of each particle. During the estimations, the self-interaction between particles and the collision between particles and geometries were also integrated into the solution.

### 2.1. Simulation model

To simulate a system, it is often necessary to try to match it as much as possible with the actual system. It should be noted that all parameters that should be taken from the actual system are selected carefully and all the com-

ponents that are not necessary for the simulation to be executed are removed. At this stage it is of the utmost importance to make the original system optimal for the analysis to be done, not copying the individual (Hensen, 2003).

In the case of building-wind interaction, it is also necessary to simulate air as a fluid. For this purpose, the airflow is reduced to particle level in this study. In order to be able to analyse a very large field flow, the number of particles is increased.

### 2.2. Integrating the equations of motion

The air current has been reduced to the particle level but it is necessary to define the equations for the interaction of the particles with each other and with the geometry. There are two methods for mathematically describing the flow in fluid dynamics. The first is to take the velocity as a function of time for each fluid particle (in other words, for each small mass in the fluid). It can be considered that a very small drop of paint is left on a stream of water and the direction and speed at which the paint moves at any time are also monitored. This corresponds to define the stream by using Lagrangian coordinates.

Another approach is to define fixed coordinates in this area by specifying a limited measurement area. Then the velocity of each particle passing through the predefined points determined in this coordinate system is examined. This time, as the small paint droplet moves, the motion is defined separately at each point, which continues in succession at each point. The instantaneous position of the paint droplet is determined on a fixed grid with reference to the coordinate system of the previously defined measuring field, not according to its local coordinate system as in the Lagrangian method. This corresponds to defining the stream using Eulerian coordinates. To achieve realistic results in the Lagrangian integration, a large number of particles must be monitored (Figure 2). The Eulerian integration, in which the flow is defined as an area and each particle is not tracked individually, but rather the velocities of the velocities

when passing through the spots in a grid system, seems more practical, but a fixed grid system will restrict the flow and will still need to define the mesh as in traditional CFD software.

In the Eulerian integration, the location, mass, and velocity of the particles must be known. At the same time, since the real world is different in terms of the environment in which the simulation is run, it is necessary to refer to each frame that is refreshed on the screen and to find the speed and position of the particles for the next frame. The greater the number of frames, the shorter the time between frames, and the estimates made converge much more to the truth.

The particles move at a speed that is initially defined, which is the wind speed in the case of wind–building interaction. In wind analyses, the velocities of 10 m above the ground level obtained by long-time measurements carried out in meteorological stations are referred to. Considering the wind speed as a force that accelerates particles instead of directly describing it as fixed particle speed leads to closer results.

It is necessary to integrate basic physical equations for particle position and velocity estimation. The equations to be used are equations of motion and equations of motion known as Newton's Second Law and give precisely how much an object will be accelerated under a net force. If the time is represented by "t", it can be indicated by dt (time difference -delta time) between both frames during the simulation. Thus, the following known physics equations are written;

$$\text{acceleration} = \text{force} / \text{mass} \quad (1)$$

$$\text{change in position} = \text{velocity} * \text{dt} \quad (2)$$

$$\text{change in velocity} = \text{acceleration} * \text{dt} \quad (3)$$

These equations should be integrated into a code. At this point, with a simple example, it can be seen how the algorithm integrates the physical equations at the basic level. The output is given in Table 1 when a fixed force of 10 Newton is applied to a stationary object weighing 1 kilogram and the iteration is performed forward with a

**Table 1.** Integrating the equations of motion.

```
double t = 0.0;
float dt = 1.0f;
float velocity = 0.0f;
float position = 0.0f;
float force = 10.0f;
float mass = 1.0f;

while (t <= 10.0)
{
    position = position + velocity * dt;
    velocity = velocity + (force / mass) * dt;
    t += dt;
}
```

**Table 2.** Outputs of Table 1 when dt = 1 s.

t=0:	position = 0	velocity = 0
t=1:	position = 0	velocity = 10
t=2:	position = 10	velocity = 20
t=3:	position = 30	velocity = 30
t=4:	position = 60	velocity = 40
t=5:	position = 100	velocity = 50
t=6:	position = 150	velocity = 60
t=7:	position = 210	velocity = 70
t=8:	position = 280	velocity = 80
t=9:	position = 360	velocity = 90
t=10:	position = 450	velocity = 100

**Table 3.** The last lines of the outputs of Table 1 when dt = 1/100 s.

t=9.90:	position = 489.552155	velocity = 98.999062
t=9.91:	position = 490.542145	velocity = 99.099060
t=9.92:	position = 491.533142	velocity = 99.199059
t=9.93:	position = 492.525146	velocity = 99.299057
t=9.94:	position = 493.518127	velocity = 99.399055
t=9.95:	position = 494.512115	velocity = 99.499054
t=9.96:	position = 495.507111	velocity = 99.599052
t=9.97:	position = 496.503113	velocity = 99.699051
t=9.98:	position = 497.500092	velocity = 99.799049
t=9.99:	position = 498.498077	velocity = 99.899048
t=10.00:	position = 499.497070	velocity = 99.999046

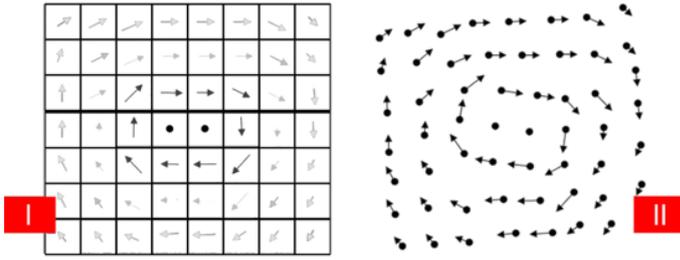


Figure 2. Eulerian integration (I) - Lagrangian integration (II).

time interval of one second (Table 1).

As seen in Table 2, at each step, both the position and the speed of the object are known. This digital integration is known as Euler Integration and is the most basic digital integration technique. Only when the rate of change is constant over the time rate (time rate) is 100% correct. If the acceleration is constant in the given example, the integration of speed is error-free. But on the other hand, the position must also be found and for this, the speed is integrated, but the speed is not constant due to acceleration and it is increasing. For this reason, it may be predicted that position integration will be not accurate. In order to be able to see the size of this error, a formula that tells how an object is moving under constant acceleration can be used, so that exact values are reached for the position:

$$\text{position} = \text{velocity} * \text{time} + \frac{1}{2} * \text{acceleration} * t^2 \quad (4)$$

When the values are substituted in Equation 4, it seems that the object should be moved to 500 meters after 10 seconds, but with Euler Integration, a result of 450 meters is obtained. This means that within a 10-second period, there will be a 50-meter faulty position difference. But  $dt = 1$  second is not an ordinary time interval. Especially in game engines, physical simulations take place at a much lower time frame than the screen frame rate (the number of frames refreshed at the moment). Because in an average shot, each frame is  $1/200$  of the time that is left for physical simulations. If the time interval had been taken as  $dt = 1/100$  seconds, i.e. if the object's position was calculated 100 times in synchronous intervals every 1 second, the results would be much closer to reality (Table 3).

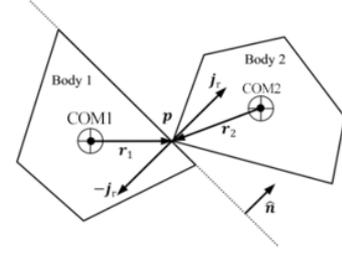


Figure 3. Impulse-based collision resolution ( $p$ : collision point, COM: centre of mass,  $j$ : impulse force) (House & Keyser, 2017)

### 2.3. Solver

It has been shown how to integrate the position and velocity of a single object up to this section, but it is known that particles must interact with each other and solid geometry in a building - wind interaction case. At this point, a force can be calculated at the contact point that occurs when the particles overlap each other and can be applied as an impulse to the particles. This force will make both particles no longer overlap with each other at the beginning of the next frame (House & Keyser, 2017). This is attempted to be performed only briefly for a single frame and is known in the literature as “impulse-based collision resolution” (Figure 3).

In architectural aerodynamics, thousands of particles are required to get high-resolution data. Until now, attempts have been made to estimate the velocity and position of a single object, and also the collision state of two different particles have been investigated. In the next step, there is a need for a solution method which should also be sufficient for many particles. For this, it can be assumed that there are two particles called A and B, each of these two particles may be thought to coincide with a particle named C. These two constraints are the two equations that need to be solved: A - C and B - C (Figure 4).

First, if the A-C interaction is supposed to be solved, this will cause both A and C to move and completely separate from each other. But this time C starts to overlap with B since the interaction between C and B is not yet calculated. (Figure 5).

This time, the interaction between B and C should be resolved. This solution allows B and C to no longer overlap

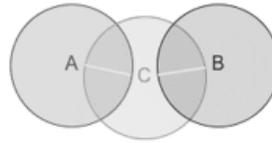
and to take their new position, but this time B and A start to overlap slightly (Figure 6).

In the previous step, since A and B were positioned separately for C, there was a collision again. But as resolution continues, it is likely that better results will be obtained (Figure 7).

Now that the solution is too close to converging and it can continue to iterate again. When the number of iterations is reached, the particles will no longer overlap each other and the interactions between the particles will be solved. The error (amount of overlap) for each pair will hopefully be reduced after each iteration, and eventually, both equations will be simultaneously satisfied. Although it seems very difficult to solve all interactions independently and sequentially, it is actually an effective method and is known as the Gauss-Seidel method. In summary, if we find a method that can reach a solution by iterating only 5 times instead of 10, we would say a faster converging algorithm is obtained.

In the Gauss-Seidel method, the sequential solution is applied and each equation is solved independently of each other. The output from the previous step is used as input in the next step. However, both equations can be solved simultaneously with the same inputs and the output averages can be taken. The method emerging at this point is known as the Jacobi method (Figure 8). Although Jacobi leads to more iterative results than the Gauss-Seidel method, both equations use the same output, so they can be solved concurrently so that the need to wait for the previous step is removed. This means that the solution can be executed in parallel. For example, a system consisting of 25 equations can be solved by 25 people independently, which means 25 times faster results than Gauss-Seidel. Moreover, there is no point in which order the equations are solved, which leads to different results that can be obtained by moving in a certain order while solving the equations.

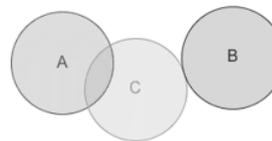
In summary, as the number of iterations increases, the solution converges more, but if a real-time solution is targeted, the optimal number of iterations must be determined. Less iteration



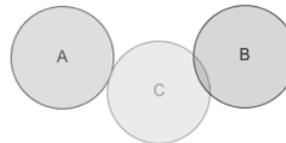
**Figure 4.** The position of C according to A and B (white lines correspond to the constraints).



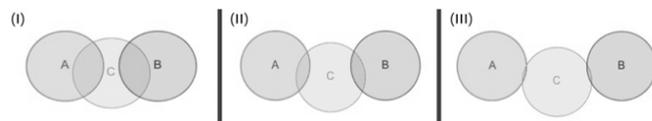
**Figure 5.** Step 1: Interaction between A and C.



**Figure 6.** Step 1: Interaction between B and C.



**Figure 7.** Step 2: Interaction between A and C.

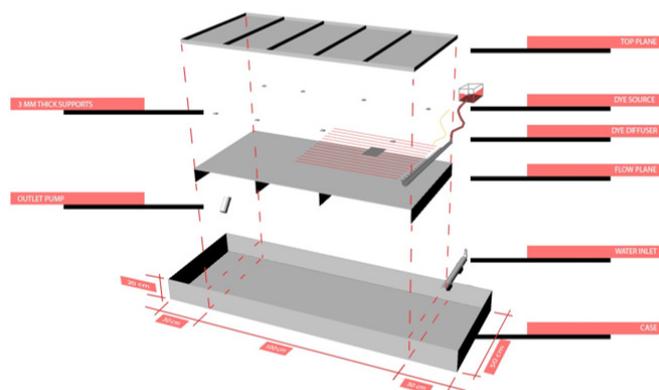


**Figure 8.** Steps of interaction resolved with Jacobi Method - I, II and III.

means faster results, but this time the accuracy is reduced.

### 3. Methodology: Validation of the algorithm

When the wind-building interaction is analysed, the flow characteristics around the basic geometries should be known. Since the algorithm gives results in a 2D plane, a similar experimental setup is needed. Hele-Shaw flow permits visualization of this kind of flow in two dimensions. The experimental setup created for Hele-Shaw flow can be seen in Figure 9.



**Figure 9.** Experimental setup created for Hele-Shaw flow.

Hele-Shaw flow is defined as Stokes Flow between two parallel flat plates separated by an infinitesimally small gap. Various problems in fluid mechanics can be approximated to Hele-Shaw flows and thus the research of these flows is of importance. An approximation to Hele-Shaw flow is specifically important to micro-flows. This is due to manufacturing techniques, which creates shallow planar configurations, and the typically low Reynolds numbers of micro-flows. The governing equation of Hele-Shaw flows is identical to that of the inviscid potential flow and to the flow of fluid through a porous medium.

#### 4. Selected cases for experimental validation

Flat-faced and sharp-edged geometries are often referred to as “bluff bodies” and appear in many building forms. Streamlines around these geometries do not follow the surface of the geometry continuously throughout the flow, from the windward region to the leeward region. Instead, the flow separates from the building surface at sharp corners, where the momentum of the fluid passes through the weak cohesive viscous forces holding the fluid together. Along the separation line, a shear layer is formed and a turbulent wake area is developed in the leeward region, which is surrounded by the diverging stream from both sides. The predictable separation state of the stream always has similar characteristics at sharp edges and corners when the bluff bodies are concerned, and similar flow

characteristics are observed even at very different wind speeds (Stathopoulos & Blocken, 2016).

Natural ventilation systems, which are known for reducing the dependence on mechanical ventilation and reduce energy consumption, are effective strategies to achieve sustainable performance. The ventilation principle indicates how the exterior and interior airflows are linked, and hence how the natural driving forces are utilised to ventilate a building. Furthermore, the ventilation principle gives an indication of how the air is introduced into the building, and how it is exhausted out of it.

Cross-ventilation is the case when air flows between two sides of a building envelope by means of wind-induced pressure differentials between the two sides. The ventilation air enters and leaves commonly through windows, hatches or grills integrated into the façades. The ventilation air moves from the windward side to the leeward side. A typical example is an open-plan office landscape where the space stretches across the whole depth of the building. The airflow can also pass through several rooms through open doors or overflow grills. The term cross ventilation is also referred to when considering a single space where air enters one side of the space and leaves from the opposite side. In this case, the ventilation principle on the system level can be either cross- or stack ventilation. As the air moves across an occupied space, it picks up heat and pollutants. Consequently, there is a limit to the depth of a space that can be effectively cross-ventilated. In this context, a room with exactly the same apertures on both windward and leeward side walls was analysed and it was seen that the flow accelerated by the Venturi effect as expected. The Venturi effect is defined as the reduction in fluid pressure that occurs when a fluid flows through a constricted section. This effect is clearly visible in both streamline view (algorithm) and the Hele-Shaw experiment (Figure 10).

As a further alternative, a simple living space was addressed and the openings with different sizes were arranged on windward and leeward

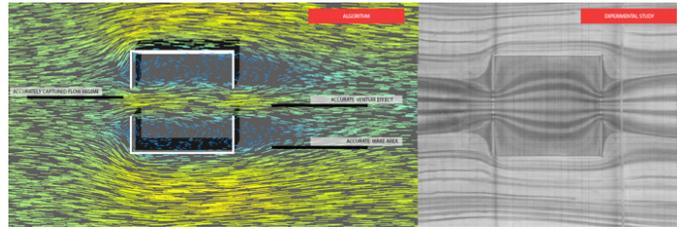
sides. The expected state of cross ventilation during the analysis was clearly observed. Flow development stages are given in Figure 11. In particular, it has been observed that the areas where the flow is separated and the wakes in the leeward region can be caught in a significant accuracy. The direction of the airflow passing through the openings and the acceleration of flow also show the accuracy of the results to estimate the flow characteristics. As in the case of experimental verification, it was observed that the algorithm could produce sufficient output for the preliminary design stage.

As another case, a simple room was addressed and the openings of the same sizes were arranged on windward and along sides. The expected state of cross ventilation during the analysis was clearly observed. Flow characteristics are given in Figure 12. In particular, it has been observed that the areas where the flow is separated and the wakes in the leeward region can be caught in a significant accuracy. The direction of the airflow passing through the openings and the acceleration of flow also show the accuracy of the results to estimate the flow characteristics. As in the case of experimental verification, it was observed that the algorithm could produce sufficient output for the early design stage.

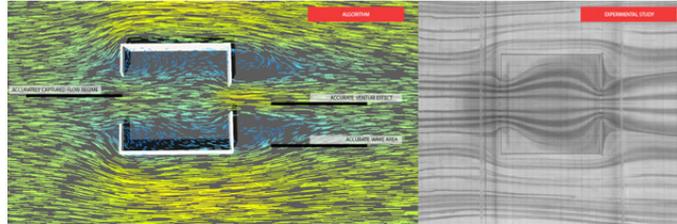
## 5. Conclusion

Details of a new algorithm for the assessment of indoor airflow characteristics at early design stage were explained. This algorithm works in real-time and does not require meshing (finite control volume). Three different steps have been taken in the algorithm development process: the design of the simulation model, the integration of physical equations and the design of the solver.

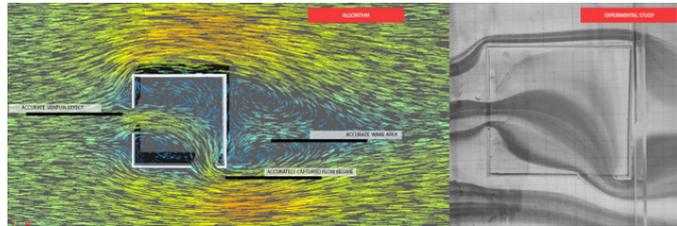
Architectural aerodynamics analysis mostly consist of external and internal cases, so the context of this study was limited with internal cases. In the scope of the study, three common natural ventilation cases have been analysed by both the developed algorithm and the experimental setup. As a result of validation studies, it was seen that the developed algorithm can be a guide



**Figure 10.** A simple cross ventilation case tested with both the algorithm (1) and the Hele-Shaw setup (2).



**Figure 11.** A simple cross ventilation case of converging flow tested with both the algorithm (left) and the Hele-Shaw experiment (right).



**Figure 12.** A simple cross ventilation case tested with both the algorithm (left) and the Hele-Shaw setup (right).

for the building-wind interaction analysis for architects in the early design phase. In particular, real-time analyses will enable architects to get real-time data. So that architects will be able to change the building form according to the results. The real-time output of the algorithm provides a guide for them to determine the optimum building form regarding the wind - building interaction. Besides, this particle-based algorithm, in which the parameters having limited effect to the results are determined in advance, is developed considering architectural practice.

The algorithm needs to be verified numerically by computational fluid dynamics software and in the wind tunnel experimentally as well. Furthermore, the algorithm will be improved to simulate small details in the building envelope by means of allowing simulation with a more significant number of particles.

## References

ASCE/SEI 49-12. (2012). Wind tunnel testing for buildings and other

structures: Reston, VA: American Society of Civil Engineers.

Blocken, B., & Carmeliet, J. (2004). Pedestrian Wind Environment around Buildings: Literature Review and Practical Examples. *Journal of Thermal Envelope and Building Science*, 28(2), 107-159. doi:10.1177/1097196304044396.

Bragança, L., Vieira, S. M., & Andrade, J. B. (2014). Early Stage Design Decisions: The Way to Achieve Sustainable Buildings at Lower Costs. *The Scientific World Journal*, 2014, 1-8. doi:10.1155/2014/365364.

Hensen, J.L.M. (2003). Simulating building performance: just how useful is it? *REHVA Journal*, nr. 4, Federation of European Heating, Ventilating and

Air-conditioning Associations - REHVA, Brussels.

House, D., & Keyser, J. C. (2017). *Foundations of physically based modelling and animation*. Boca Raton: CRC Press, Taylor & Francis Group.

Liu, G. (2002). *Mesh Free Methods*. doi:10.1201/9781420040586.

NIST Technical Report. (2009). "Toward a standard on the wind tunnel method".

Stathopoulos, T., & Blocken, B. (2016). Pedestrian Wind Environment Around Tall Buildings. *Advanced Environmental Wind Engineering*, 101-127. doi:10.1007/978-4-431-55912-2\_6.