Validation of Grasshopper-based Fast Fluid Dynamics for Air Flow around Buildings in Early Design Stage

Christoph Waibel[1,2], Lukas Bystricky[3], Aytaç Kubilay[1,4], Ralph Evins[2,5], Jan Carmeliet[1,4]
[1]Chair of Building Physics, ETH Zurich, Switzerland
[2]Urban Energy Systems Laboratory, Empa, Duebendorf, Switzerland
[3]Department of Scientific Computing, Florida State University, Tallahassee, USA
[4]Laboratory for Multiscale Studies in Building Physics, Empa, Duebendorf, Switzerland
[5]Energy Systems and Sustainable Cities Group, University of Victoria, Canada

Abstract

This paper presents an implementation of a custom Fast Fluid Dynamics (FFD) code in Rhinoceros and its visual programming platform Grasshopper for studying airflow around buildings and related surface pressures. The advantage over conventional CFD is significantly faster computing time. Hence, it enables the consideration of airflow during the conceptual design phase and for use with computational optimization methodologies.

We validate our implementation with different experimental and numeric studies. The results show a good match for velocity and pressure distribution with simple cases and a reasonable match for a large-scale urban case. Furthermore, we show the significant impact of an exemplary urban context on resulting surface wind pressure coefficients of a building.

The CAD integrated FFD implementation presented in this study enables the consideration of airflow on early design decisions or in optimization design methodologies in a time-effective manner.

Introduction and Background

Airflow around buildings has a multitude of effects on building energy performance, thermal comfort and occupant health. It affects ventilation through façade or roof apertures, it can influence the efficiency of building systems, and it determines the distribution of airborne pollutants caused by vehicles or plants (ASHRAE, 2001). Hence, knowledge about spatially resolved airflow patterns specific to an urban context is crucial in the design of buildings, districts and cities. In this section, we provide a brief summary of the main design aspects influenced by airflow around buildings.

Passive ventilation

Passive ventilation techniques, if designed carefully, can fulfil indoor air quality and thermal comfort requirements in an energy efficient manner and at low cost (CIBSE, 2005). Schulze & Eicker (2013) proved that naturally ventilated office buildings can save up to 44 kWh/m^2 cooling net energy per year. Surveys also show occupant's preference to control windows over "sealed" spaces. Therefore, it can also be beneficial in terms of overall satisfaction and productivity (Huizenga, Abbaszadeh, Zagreus, & Arens, 2006; Parsons, 2003). Yet, since wind fluctuates (in direction, velocity, etc.) it is more challenging to create specific conditions with passive strategies than with active ones. Simulation models that assist in the design process become especially important in this context. Khan et al. (2008) highlight the complexity of passive systems in their review on contemporary wind driven ventilation techniques.

Building systems

Active systems also need consideration of external airflow for appropriate placement and orientation of air intakes and exhausts. Bruelisauer et al. (2014) measured the significant reduction in coefficients of performance (COP) of a vertically oriented array of split type condensing units in a high-rise building in Singapore, and Gracik et al. (2015) apply Computational Fluid Dynamics (CFD) simulations to show the impact of urban neighbourhoods on the COP of cooling systems. Other systems where external airflow affects their efficiency are air-source heat pumps, micro wind turbines and solar thermal panels.

Airborne pollutants

The major contributor of urban pollutants, specifically particulate matter, is the transport sector (O'Mahoney, Gill, Broderick, English, & Ahern, 2000). Other pollution sources are from industrial plants or building heating and ventilation systems. Careful consideration of the dispersion of such pollutants is crucial in designing healthy environments.

Building Energy Modelling

Building Energy Modelling (BEM) has become mandatory in compliance with many national building regulations and green building labels such as LEED, BREEAM or DGNB. BEM programs simulate dynamic building physical processes and are used to evaluate energy consumption, peak loads and indoor environmental quality (mainly thermal comfort).

However, various parameters in BEM programs need to be adjusted in order to adequately represent the context, i.e. the surrounding microclimate, of a building model. Moonen et al. (2012) provide a thorough review on the effects of urban physics on comfort, health and energy demand – aspects often not considered in practice or research studies. Samuelson et al. (2016) present a parametric study of the urban context with the BEM program EnergyPlus; besides solar irradiation, no further microclimate effects have been considered. Pisello et al. (2016) only adjusted the wind speed in weather files to account for adjacent buildings. They found a significant impact on primary energy consumption of naturally ventilated residential buildings.

Coupling strategies for CFD and BEM

From the brief review above it becomes apparent that airflow analysis matters in the design of healthy, productive and energy efficient built environments. Hence, BEM can be coupled with CFD to better capture microclimatic effects. Zhai et al. (2002) classify coupling approaches into static and dynamic. In the former approach information is exchanged between BEM and CFD only before the execution of a simulation. In the latter approach, information is exchanged dynamically between programs, e.g. for each timestep over a time horizon, or until a convergence criterion is fulfilled.

Other studies have shown the sensitivity of coupling CFD and BEM. Djunaedy et al. (2005) showed a difference in energy consumption of a case study in the range of 10% with and without coupling. Zhai & Chen (2006) conduct a sensitivity analysis for an indoor environment and show significant differences in cooling loads and floor surface temperatures. Furthermore, they provide guidelines on the selection of the appropriate coupling approach. Recent studies present coupled design environments to account for the urban microclimate and a review of different software platforms is given in Srebric et al. (2015).

A main challenge of coupling between CFD and BEM however remains the high computational cost. Especially in early stage conceptual design, many iterations and drastic design changes are common, before one concept is finalized. Also computational optimization methodologies gain in popularity both in research and practice and again require many design iterations and evaluations (Evins, 2013). Consequently, a time efficient coupling methodology especially for this capricious phase would have enormous benefits.

One way of addressing this issue is to apply regression models. Evins et al. (2014) applied Gaussian Processing to emulate site-specific wind flow using CFD simulations as training data. However, the validity of the model is constrained to the training data, i.e. for changing urban layouts and building geometries another model would need to be trained. Ladick et al. (2015) achieved a higher generalization by directly estimating the momentum and incompressibility constraint of particles using regression forests. This way, they could achieve a speed-up of one to three times, while being independent of the specific fluid domain (geometry, velocity).

Fast Fluid Dynamics

Fast Fluid Dynamics (FFD) could offer a reasonable tradeoff between accuracy and computational cost. Proposed by Stam (1999), it was originally intended for computer visualization, but has recently also been applied for building simulation. FFD is a fully implicit projection method. Projection methods, originally introduced by Chorin (1967), are a popular method of approximating the incompressible Navier-Stokes equations because they decouple the pressure and the velocity, resulting in linear systems that are much smaller, and therefore much cheaper to solve. The major benefit of FFD over other projection methods is that it is fully implicit and hence the time step is not constrained by the Courant-Friedrichs-Lewy (CFL) condition which allows us to take large time steps without the solution becoming unstable (W. Zuo & Chen, 2009).

Zuo and Chen (2009) show that FFD is well-suited for indoor airflow calculations of buildings. Consequently, Zuo et al. (2016) present a coupling of FFD in the Modelica buildings library for indoor airflow. Waibel (2012) applies FFD to optimize the shape of wind cowls to improve their ventilation performance, and Chronis et al. (2011) use it to optimize abstract surfaces. Both studies use computational optimization algorithms, the former Simulated Annealing, the latter a Genetic Algorithm.

In our present study, we focus on external airflow and surface wind pressure coefficients. We develop a custom implementation in C#[1] and integrate it into the CAD program Rhinoceros 3D[2] (Rhino) and its visual programming platform Grasshopper[3]. Implementation specifics are explained in the following section.

Implementation

The governing equations we will be interested in are the incompressible Navier-Stokes equations, given by

$$\frac{\partial \boldsymbol{u}}{\partial t} = -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{F}, \qquad (1)$$

$$\nabla \cdot \boldsymbol{u} = 0, \qquad (2)$$

where $\boldsymbol{u}(\boldsymbol{x},t) = (u(\boldsymbol{x},t), v(\boldsymbol{x},t), w(\boldsymbol{x},t))$ is a velocity vector at time t and coordinate $\boldsymbol{x} = (x,y,z)$ on a regular three dimensional Cartesian grid, $p(\boldsymbol{x},t)$ is a scalar pressure, $\boldsymbol{F}(\boldsymbol{x},t)$ is an external force such as wind or an inlet and ρ and ν are constants for the fluid density and the kinematic viscosity respectively.

The boundary conditions we will be considering are all on the velocity $\boldsymbol{u}$. In particular, we will be considering a combination of Dirichlet conditions $\boldsymbol{u} = \boldsymbol{g}$ (no-slip, inflow) and the slightly non-physical Neumann conditions $\partial \boldsymbol{u}/\partial n = h$ (outflow).

Here we present a brief overview of FFD. For more details concerning FFD see Stam (1999). For more details on general implicit projection methods, including the effects of different boundary conditions see Gresho (1990).

[1] The FFD library is available at: https://github.com/lukasbystricky/GSoC_FFD

[2] https://www.rhino3d.com/

[3] The Rhino Grasshopper plug-in is available at: https://github.com/christophwaibel/GH_Wind

Starting with an initial velocity field $\boldsymbol{u}^k = \boldsymbol{u}(t_0)$, we can compute a new velocity $\boldsymbol{u}^{k+1} = \boldsymbol{u}(t_0+\Delta t)$ in three steps: 1. Diffusion, 2. Advection, 3. Projection.

In the diffusion step we solve for an intermediate velocity $\boldsymbol{u}^*(t+\Delta t)$ from

$$\frac{\partial \boldsymbol{u}^*}{\partial t} - \nu \nabla^2 \boldsymbol{u}^* = \boldsymbol{F}, \tag{3}$$

with the boundary conditions on $\boldsymbol{u}^*$ matching those on $\boldsymbol{u}$. This can be solved using fully implicit backwards Euler:

$$\boldsymbol{u}^* - \nu \Delta t \nabla^2 \boldsymbol{u}^* = \Delta t \boldsymbol{F} + \boldsymbol{u}^{\boldsymbol{k}} \tag{4}$$

The nonlinear advection term is resolved using a semi-Lagrangian scheme. Originally proposed in Courant et al. (1952), semi-Lagrangian schemes are commonly used in weather modelling. Given the intermediate velocity field $\boldsymbol{u^*}$ at time $t = t_0 + \Delta t$, we trace back the trajectories of a fluid element at position $\boldsymbol{x} = (x,y,z)$ along a path $\gamma(\boldsymbol{x})$ to find the velocity of the fluid element at $t = t_0$. The velocity at point $\boldsymbol{x}$ is then set to the velocity of the fluid element at $t = t_0$. Doing so leads to a new intermediate velocity field $\boldsymbol{u^{**}}(x, t+\Delta t)$ which can be defined as

$$\boldsymbol{u}^{**}(\boldsymbol{x}, t_0 + \Delta t) = \boldsymbol{u}^*(\gamma(\boldsymbol{x})). \tag{5}$$

The trace path γ can be a simple first order back step, i.e.

$$\gamma(\boldsymbol{x}) = -\Delta t \boldsymbol{u}^*(\boldsymbol{x}), \tag{6}$$

or higher order tracers, e.g. midpoint where we evaluate the velocity at an intermediate point. We implemented both first and second order tracers (*Figure 1*, left) (we use second order for all simulations in this study).

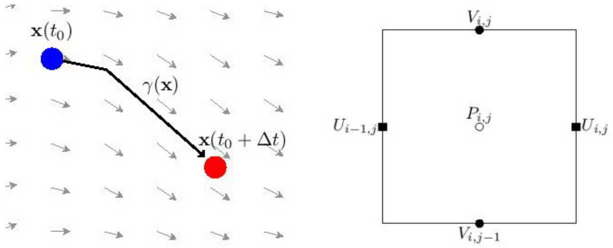


Figure 1: Left: Semi-Lagrangian scheme. Right: Cross section of the x-y plane of a single discretized cell on a staggered grid

The velocity field $\boldsymbol{u^{**}}$ is not divergence free. To obtain a divergence free field $\boldsymbol{u}$, we must perform a projection step. First, we evaluate the pressure field according to

$$\nabla^2 p^{k+1} = \nabla \cdot \boldsymbol{u}^{**}. \tag{7}$$

Following Gresho (1990) we will use homogeneous Neumann boundary conditions, i.e $\partial \mathrm{p}^{\mathrm{k}+1} / \partial \mathrm{n} = 0$ on the entire boundary.

Once we compute the pressure, we can update $\boldsymbol{u^{**}}$ to obtain the velocity field $\boldsymbol{u}(t_0+\Delta t)$ according to

$$\boldsymbol{u}^{\boldsymbol{k+1}} = \boldsymbol{u}^{**} - \nabla p^{k+1}. \tag{8}$$

To discretize in space, we use a second order finite difference scheme. In the original algorithm proposed by Stam (1999), intended for video game visualizations, Stam used a standard finite difference grid with the velocity and pressure all defined at the centres of each cell. While this implementation can recover the velocity (though not at optimal accuracy), it may not recover the pressure with any accuracy. Since we are interested in the pressure, we will use a staggered grid, where the pressure is defined at the centre of the cell and velocity components are defined at the midpoints of the edges normal to its direction (*Figure 1*, right). Since we have a regular 3-dimensional grid, we can compute $\boldsymbol{u}^*(\gamma(\boldsymbol{x}))$ by trilinear interpolation.

Each time step requires solving three diffusion equations (one for each component of $\boldsymbol{u}$) and a single Poisson equation for the pressure. To solve these systems our implementation uses a simple Jacobi solver. Currently we have parallelized our implementation on CPU, but plan to use GPU in the future, as it is known to be significantly faster than on CPU: according to Harris (2004) up to six times, according to Zuo & Chen (2010) up to 30 times. The effect of the solver has a major impact and possibly better solvers such as conjugate gradient (the systems are all positive definite) or multigrid solvers might be even faster

Wind pressure coefficients

Node-based zonal models (as typically used in BEM) or empirical ventilation models rely on the wind pressure coefficients C_p when calculating pressure differentials occurring in façade openings, stacks or cracks (CIBSE, 2005). C_p is a dimensionless number that depends on building shape, wind direction and adjacent obstructions such as from nearby buildings, vegetation or terrain. Usually BEM programs only provide C_p values for simple geometries (Cóstola, Blocken, & Hensen, 2009), e.g. EnergyPlus provides data for either low-rise, or high-rise rectangular buildings. For more complex geometry, individual C_p values should be given by the user, which can be obtained from either wind-tunnel experiments or numerical simulations.

Several authors have studied the sensitivity of C_p values in BEM: (Cóstola, Blocken, Ohba, & Hensen, 2010) show the uncertainty in airflow rates due to surface-averaged C_p values; (Ramponi, Angelotti, & Blocken, 2014) show the sensitivity of C_p values to air flow rates for night ventilation, however they found no impact on cooling demand. Furthermore, Hughes & Cheuk-Ming (2011) showed that external wind pressure provides 76% more internal ventilation compared to buoyancy, when using commercial wind towers. In conclusion, local C_p values specific to a building design and its urban context are important information to consider in BEM programs.

Generally, the wind surface pressure in [Pa] relative to static pressure is given with Bernoulli's equation:

$$p_w = C_p \rho \frac{\boldsymbol{u}_z{}^2}{2}, \tag{9}$$

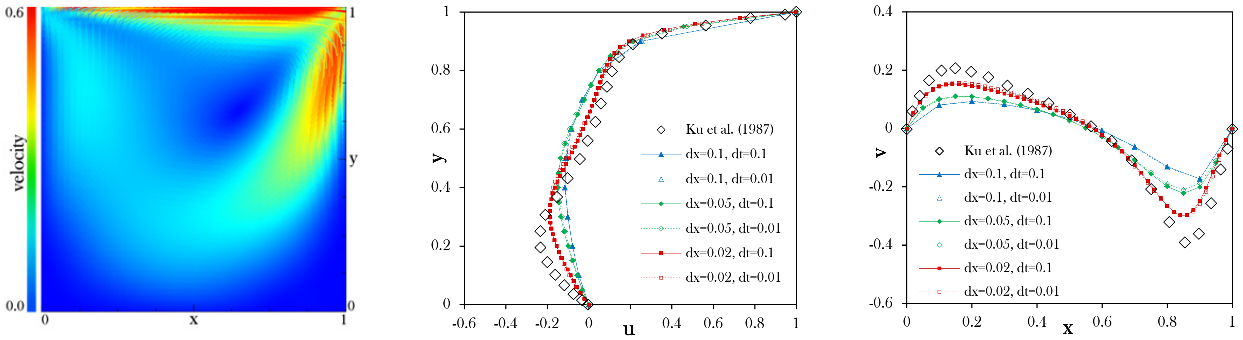


Figure 2: Cubic lid-driven cavity at Re=400, u & v velocity component at vertical & horizontal center

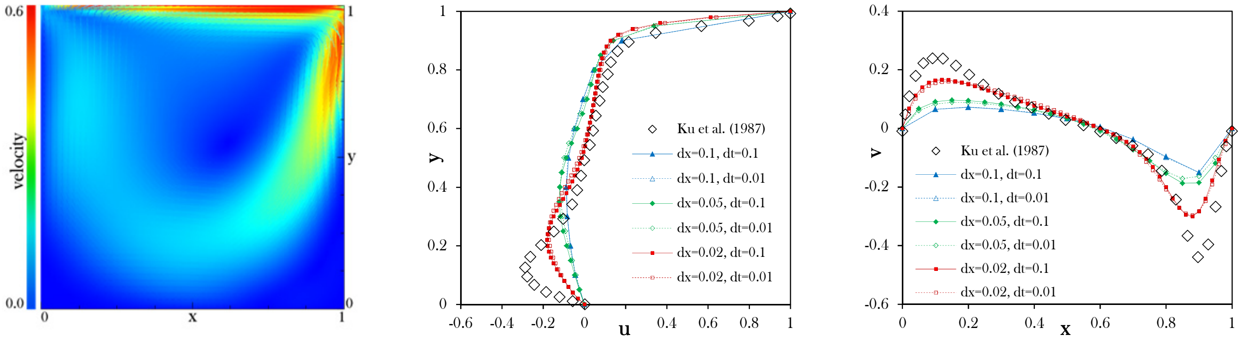


Figure 3: Cubic lid-driven cavity at Re=1000, u & v velocity component at vertical & horizontal center

where $\boldsymbol{u}_z$ is the mean wind velocity in [m/s] at height z in [m]. This is used for calculating pressure differentials between interior and external zones. Hence, we are interested in C_p values for different locations $\boldsymbol{x}$ of a building model and for different incident wind angles φ:

$$C_p(\boldsymbol{x},\varphi) = \frac{p_{\boldsymbol{x}} - p_\infty}{\frac{1}{2}\rho_\infty \boldsymbol{u}_\infty^2}, \qquad (10)$$

where $p_{\boldsymbol{x}}$ is the pressure at location $\boldsymbol{x}$, p_∞ is the pressure in the freestream, ρ_∞ is the fluid density (we use 1.2041 kg/m^3) and $\boldsymbol{u}_\infty$ is the freestream velocity. We obtain the values for pressure from the FFD simulations and the freestream velocity from a prescribed inflow profile. We measure $\boldsymbol{u}_\infty$ and p_∞ at the building height.

Geometry discretisation

We discretise mesh geometries into the three dimensional Cartesian grid of our fluid domain with a custom script, using the RhinoCommon SDK[4]. Since we have a structured grid, compared to an unstructured grid the computation for the meshing process is very fast. For a domain with 1,000,000 cells the geometry discretisation completes within one second on an 8-core Intel Core i7-4800MQ with 2.7 GHz and 16GB RAM. We will refer to this hardware set-up for any further computing times we report in this study, if not stated otherwise.

Computing time

Computing times for different grid resolutions of the validation case "Flow around high-rise building" (introduced later) are given in *Table 1*. From this, we can estimate required computing times depending on the amount of cells in the domain.

Table 1: Computing times for flow around high-rise

dx **[m]**	**No. of cells**	**Time [min.]**
1	1,400	< 1
0.5	12,000	1
0.25	96,000	5
0.01	1,500,000	80

Interface to Rhino / Grasshopper

We implement our FFD library in Rhino and its visual programming platform Grasshopper. This has several practical advantages. It is a highly flexible 3D CAD environment used by both architectural and urban design practitioners and researchers, it offers an extensive SDK for developing custom applications and it has an active community contributing to a growing number of plug-ins for various aspects of building design, such as BEM, urban co-simulation, GIS and multi-objective optimization.

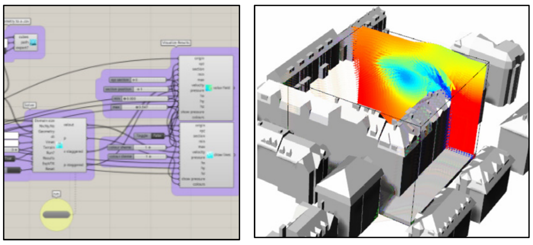


Figure 4: Implementation in Rhino

Providing an FFD plug-in in this software environment eases coupling to other simulation, optimization and design components, including BEM.

[4] http://4.rhino3d.com/5/rhinocommon/

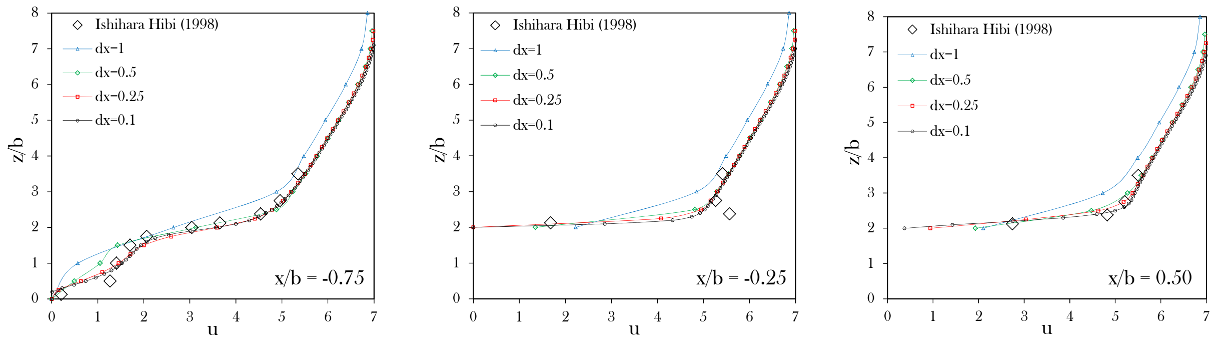


Figure 5: Flow around high-rise. u-velocity component for different locations.

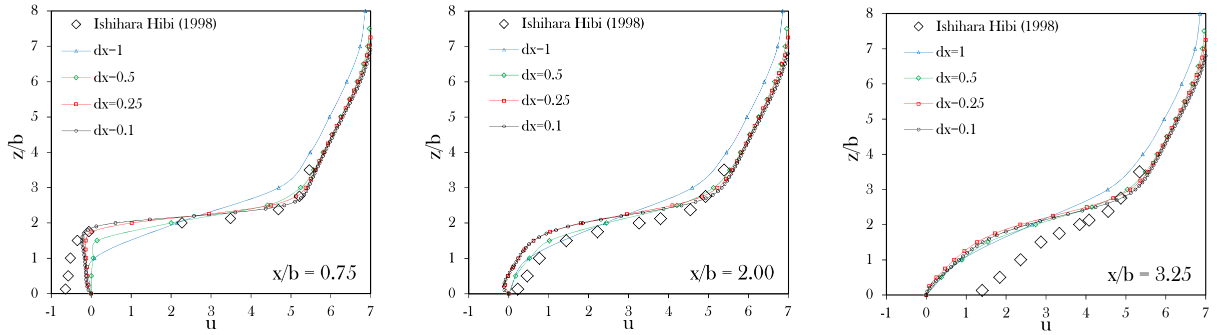


Figure 6: Flow around high-rise. u-velocity component for different locations.

We provide Grasshopper components to define the fluid domain, discretize and export geometry, visualize vector and colour fields for velocity and pressure, and to calculate and visualize C_p values on any location in the domain. *Figure 4* shows an example screenshot from the Grasshopper interface on the left and a velocity vector field on the right.

Validation

We present validation results and comparative studies for different cases, using either experimental data or CFD simulations. Currently, we run the FFD simulations until we can observe a stable flow field. For future studies, we plan to use numeric convergence indicators.

Air flow in a cubic lid-driven cavity

We use a 1 m × 1 m × 1 m cubic domain with a constant tangential velocity of 1 m/s on the lid and no-slip boundaries otherwise (*Figure 7*). We compare our model to results from Ku et al. (1987) for Reynolds numbers *Re* = 400 and *Re* = 1000. Furthermore, we test different cell sizes (*dx* = 0.1 or 10 × 10 × 10, *dx* = 0.05 or 20 × 20 × 20, *dx* = 0.02 or 50 × 50 × 50) and time steps (*dt* = 0.1 and *dt* = 0.01). The results for *Re* = 400 are given in *Figure 2* and for Re = 1000 in *Figure 3*.

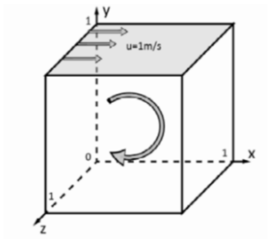


Figure 7: Cubic lid-driven cavity (from: Zuo et al. 2012)

We can observe that the results become more accurate with finer grid resolution, while general trends are already visible with the coarsest grid. However, even for the finest grid studied the resulting curves are too flat to match the inflection points of the reference data. Furthermore, for higher *Re*, FFD deviates more from the reference. Presumably, this is because turbulent effects – which we do not model – become more important. The time step does not show any clear impact on the model results. However, we observed numeric instability for large time steps in combination with fine mesh resolution. As a simplification, we use *dt* = 0.1 throughout all further simulations in this study. In practice, larger time steps can be chosen, given a sufficiently coarse mesh resolution.

Overall, the results achieved are within a reasonable accuracy even for coarse grids, especially when applying FFD as a design and optimization tool. Our results also match the findings from Zuo et al. (2012).

Flow around a high-rise building

We compare FFD simulations for the flow around a high-rise building to wind tunnel experiments from Ishihara & Hibi (1998), found in Mochida et al. (2002). The dimensions of our fluid domain are 20 m × 10 m × 7.5 m and the centre of the obstacle with the size of 1 m × 1 m × 2 m is located at 6.5 m distance to the inflow. We test different grid resolutions: *dx* = 1m, *dx* = 0.5 m, *dx* = 0.25 m, *dx* = 0.1 m, but keep *dt* = 0.1 constant.

We choose a numerical viscosity of *ν* = 0.1 instead of the value for air, as after several test runs we could obtain the best results with this significantly increased value. Effectively, since we are lacking a turbulence model, we artificially reduce the *Re* by increasing *ν* and hence compensate for the turbulent viscosity, which we do not model. However, it would be better to use v as a function of e.g. $\boldsymbol{x}$ and $\boldsymbol{u}$, instead of a domain-uniform constant. Further work is required in this regard. The issue of numerical viscosity has been discussed and studied in Zuo et al. (2012).

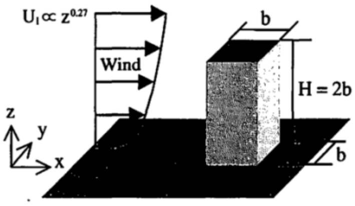


Figure 8: Flow around high-rise (from: Ishihara & Hibi 1998)

Based on the data from the wind-tunnel experiment, we fit a polynomial regression model for the inflow profile:

$$u(z) = a_1 z^7 + a_2 z^6 + a_3 z^5 + a_4 z^4 + a_5 z^3 + a_6 z^2 + a_7 z + a_8, \tag{11}$$

where $z := z/b$ with b being the building width and z the height. The coefficients are given with $a_1 =$ 1.5e-4, $a_2 =$ -4.1e-3, a_3 = 0.04558, a_4 = -0.2643, a_5 = 0.8549, a_6 = -1.56, a_7 = 2.113, a_8 = 2.671. At z = 5 m, $\boldsymbol{u}$ is around 6 m/s.

We show the results of the u-velocity component at different locations $x := x/b$ and compare it with the wind-tunnel experiment in *Figure 5* and *Figure 6*. It can be seen that for most locations FFD is close to the experiment, even for very coarse grids. Again, a finer grid resolution produces results that are more accurate. However, even for the coarsest grid with dx = 1 the results are reasonable. It is striking, that at x = 3.25 (i.e. well behind the obstacle on the leeward side) the results of the FFD simulation deviates significantly by u = 1 m/s for z between 0 and 2, even if for fine resolutions. This is as expected for FFD, since turbulent effects would primarily cause a flow reattachment in this case.

Wind pressure coefficients of a high-rise building

We simulate the pressure coefficients on the surfaces of a high-rise building and compare it with data provided by ASHRAE (2001) – which is used in EnergyPlus. In ASHRAE, the values are suggested for rectangular high-rise geometries with H > 3b, H being the building height. However, we could not acquire the original publication including specifics on the experiment set up. Hence, we suggest a qualitative rather than a precise comparison with our simulated results. For our simulation, we use an inflow profile as given in ASHRAE:

$$\boldsymbol{u}(z) = \boldsymbol{u_{met}} \left(\frac{\delta_{met}}{z_{met}}\right)^{\alpha_{met}} \left(\frac{z}{\delta}\right)^{\alpha} \tag{12}$$

where $\boldsymbol{u_{met}}$ is the wind speed, δ_{met} is the boundary layer thickness, and α_{met} an exponent from a nearby meteorological station respectively. z_{met} is the height at which the velocity is measured – typically 10 m above ground. δ and α are the boundary layer thickness and an exponent for the local building terrain. We use $\boldsymbol{u_{met}}$ = 3 m/s, terrain category (according to ASHRAE) for cities and v = 0.1. The domain is of size 300 m × 150 m × 100 m, the obstacle is 10 m × 10 m × 30 m and is centrally placed 100 m downstream. We show the results for the C_p values on the windward, leeward and side surfaces in *Figure 15* for dx = 1m, in *Figure 16* for dx = 2m and in *Figure 17* for dx = 5m, and compare it to values given in ASHRAE, shown in *Figure 14*.

The results show a good match for the windward façade with increasing accuracy at finer grids. The high-pressure spot with C_p = 0.9 is slightly off. The leeward side does not match the reference in terms of distribution. However, C_p values are similar ranging from -0.25 to -0.35. The side façades show an elongated minimum pressure field, in contrast to the reference, which shows a concentrated minimum field. It is worth noting that even a cell size of 5 m, where the obstacle is represented by only six cells over the height, produces reasonable results.

Flow field on a research campus

To provide a more complex case, we simulate the flow field of the Empa research campus in Duebendorf, Switzerland. The FFD domain has a size of 1200 m × 750 m × 100 m and we show results for dx = 5 m (0.72 million cells, ~0.75 hours running time), dx = 4 m (1.406 million cells, ~1.5 hours running time) and dx = 2 m (11.25 million cells, ~24 hours running time). Again, we use v = 0.1.

RANS CFD simulations are conducted using the realizable k-ε turbulence model using the standard wall functions with sand-grain roughness modification. The simulations have been performed with OpenFOAM® 2.4. Second-order discretization schemes are used for both the convection terms and the viscous terms. The computational grid is composed of 15 million cells. The calculation took about 30 hours in a parallel run using 48 processors.

The wind inflow profile is given by:

$$\boldsymbol{u}(z) = \frac{u_{ABL}}{\kappa} * \log\frac{z + y_0}{y_0} \tag{13}$$

where u_{ABL} = 0.9965 m/s is the atmospheric boundary layer friction velocity, κ = 0.42 is the von Karman constant, and y_0 = 0.15 m is the aerodynamic roughness length.

Figure 9 shows the flow field of the campus with the velocity magnitude on a plane 4 m above ground and *Figure 10* on a plane 10 m above ground. The left images show results of the FFD with cell size dx = 4 m (top) and 5 m (bottom) respectively, the middle images with dx = 2 m and the right images show the CFD simulations. On the 4 m plane it is striking that the FFD simulation with dx = 4 m has a significantly longer wake than the CFD, whereas the flow field in the FFD with dx = 2 m re-attaches faster than the CFD. On the 10 m plane many regions in the FFD simulation with dx = 5 m exhibit a good match with CFD, while for the FFD with dx = 2 m the wakes are significantly too small.

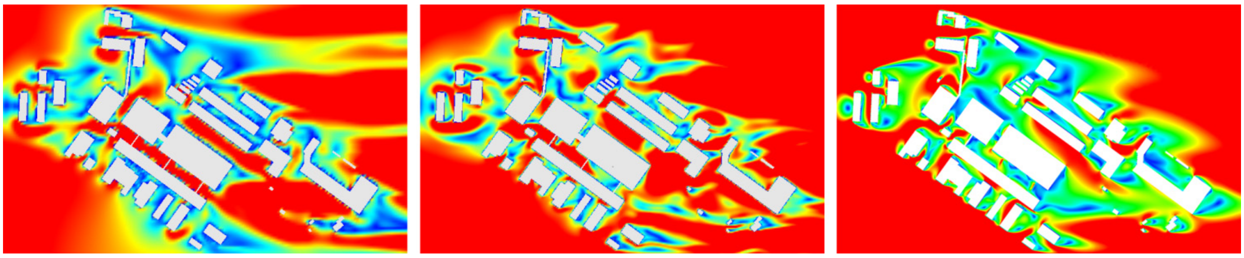


Figure 9: Empa campus, z-section in 4m height. From left to right: FFD dx=4m; FFD dx=2; CFD OpenFoam

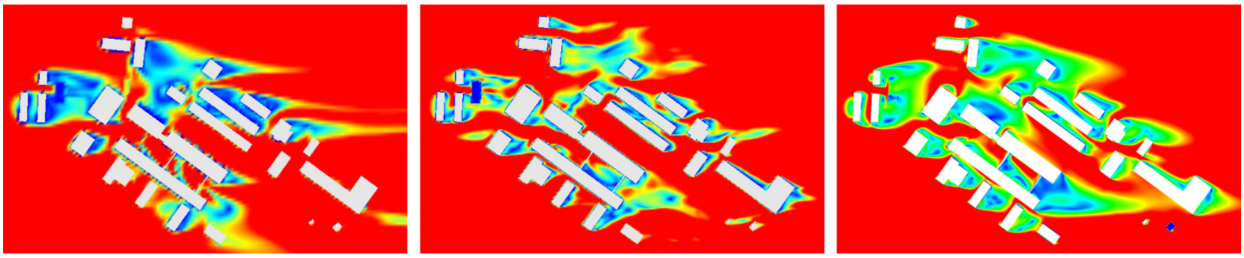
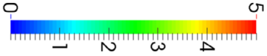


Figure 10: Empa campus, z-section in 10m height. From left to right: FFD dx=5m; FFD dx=2; CFD OpenFoam



Velocity magnitude [m/s]

Furthermore, it is noticeable that the flow field in the FFD simulation with dx = 2 m shows fluctuations. This is due to the fact, that we do not show a mean flow field over a certain time horizon, but a single time step. This should be addressed in further work.

In summary, looking closely at the figures reveals some regions of similarity between CFD and FFD, but also many discrepancies. As in the previous case studies presented here, we primarily suspect the lack of spatially resolved turbulent effects being the cause of the discrepancies. The challenge in estimating the velocity distribution especially behind buildings has also been observed by Jin et al. (2012).

Difference in C_p with urban context

In this section, we show the impact of urban context on the distribution of C_p values on building surfaces. As a case study, we use an example file from the BEM program DesignBuilder[5] V4.2., export the file as .idf (EnergyPlus format) and import the geometry as a mesh into Rhino. We use the same campus from the previous section as the urban context and place the imported building on the west part of the site. *Figure 11* shows a site plan and a rendering of the building model.

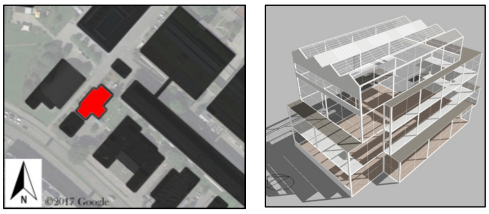


Figure 11: Site plan and building model

DesignBuilder provides the building model as a template for modelling wind and buoyancy driven natural ventilation in an atrium. It consists of 144 individual surfaces for external walls, windows, vents and roof.

We are interested in the change of C_p values for each surface, when considering the urban context compared to a stand-alone building. We conduct FFD simulations with a domain of size 300 m × 150 m × 60 m and dx = 2 m. The building is 32 m in length, 22 m in width and 16 m in height. We use the inflow profile from equation 12 with $\boldsymbol{u}_{\boldsymbol{met}}$ = 5 m/s and city terrain category. In total we conduct 16 simulations: every 45° wind direction with context and without. Each simulation requires around 20 minutes calculation time. An example distribution of C_p on the building is shown in *Figure 12* for a wind direction of 45°, considering the urban context.

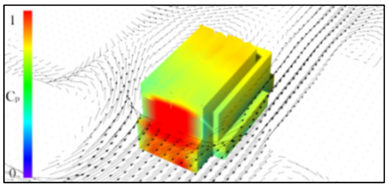


Figure 12: C_p values on the building. Adjacent buildings are hidden.

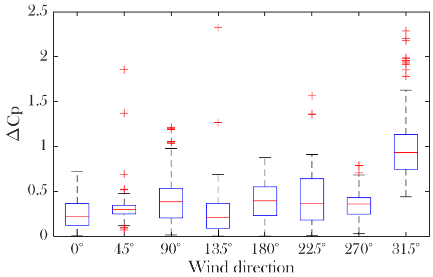


Figure 13: Absolute difference in C_p of all 144 surfaces with and without urban context for each wind direction

In *Figure 13* we show the absolute difference in C_p between a building considering the urban context and a stand-alone building for all surfaces and wind directions.

[5] https://www.designbuilder.co.uk/

From the boxplot it becomes apparent that the context has a significant impact on surface pressure distribution. Differences range up to 2.5, while the mean is around 0.2 to 0.4 depending on the wind direction. At a wind direction of 315° the difference seems to be highest. This is caused by a tall adjacent building which stands very close to the atrium building.

We also compared the overheating hours in the atrium with the adjusted C_p values for stand-alone and with context using a Zurich TMY weather file. Even though we could not find a significant difference, we suspect this is due to this specific building energy model. A more complete building model with realistic definitions of internal gains, controls, schedules, etc. (rather than a template file) should be investigated in the future, including different weather files and site contexts.

Nevertheless, this case study highlights the significant impact of adjacent buildings on C_p values and their distribution. Also, with a total computing time of around 2 hours and 40 minutes to simulate 8 wind directions, coupling FFD with BEM becomes feasible. Further study on the required grid resolution would help find the right balance between simulation accuracy and computing time.

Conclusion and Outlook

We have presented our FFD implementation in Rhino Grasshopper to study airflow around buildings and we compared the model with experimental data from the literature and with CFD simulations.

For the simple validation cases (airflow in a lid-driven cavity, airflow around a high-rise building and C_p on a high-rise building), we could achieve a good match to the references. However, our cases confirmed that FFD has low accuracy in predicting velocity distributions in the wake regions behind obstacles. Further study is required to identify a rigid methodology on setting the viscosity v, as this was found to have a significant impact especially on the wake and recirculation regions. Possibly, using v as a (simple) function of $\boldsymbol{x}$ and $\boldsymbol{u}$ could improve the accuracy, without requiring the addition of a turbulence model. This function could be trained by data from CFD, similar to the study by Ladick et al. (2015).

Replacing the iterative Jacobi solver with other methods (e.g. conjugate gradient or multigrid) for solving the Poisson and diffusion equations could improve computing times. Parallelization of the solver on a GPU is another possible way to reduce computation time (Wangda Zuo & Chen, 2010). Furthermore, we plan to add energy (Boussinesq approximation) and pollution (Fick's law) transport to our implementation.

Our Grasshopper-integrated FFD implementation allows an efficient coupling to other aspects of architecture and urban design, including energy modelling. It will allow the study of wind flow patterns depending on building geometry and urban configurations to be linked to the impact on natural ventilation, building systems performance or pollution distribution. Improved predictions are expected from building energy models, as site conditions can be considered more precisely.

In conclusion, we hope that this work will stimulate researchers and practitioners to include micro-climate airflow effects in a time-effective manner during design and optimization.

Acknowledgements

The work is related to the Competence Center – Energy and Mobility "Synergistic Energy and Comfort through Urban Resource Effectiveness" project and the Swiss Competence Centers for Energy Research "Future Energy Efficient Buildings and Districts" project. Furthermore, this work has been partially supported by a stipend within the context of the Google Summer of Code (GSoC) 2016 as part of the HUES platform (https://hues-platform.github.io).

References

ASHRAE. (2001). Airflow Around Bulildings. In *Handbook "Fundamentals"* (111th ed.). Atlanta: American Society of Heating, Refrigerating and Air Conditioning Engineers.

Bruelisauer, M., Meggers, F., Saber, E., Li, C., & Leibundgut, H. (2014). Stuck in a stack—Temperature measurements of the microclimate around split type condensing units in a high rise building in Singapore. *Energy and Buildings*, *71*, 28–37.

Chorin, A. J. (1967). A Numerical Method for Solving Incompressible Viscous Flow Problems. *Journal of Computational Physics*.

Chronis, A., Turner, A., & Tsigkari, M. (2011). Generative Fluid Dynamics: Integration of Fast Fluid Dynamics and Genetic Algorithms for Wind Loading Optimization of a Free Form Surface. In *Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design* (pp. 29–36). San Diego, CA, USA

CIBSE. (2005). *AM10: Natural Ventilation in Non-Domestic Buildings*. London.

Cóstola, D., Blocken, B., & Hensen, J. L. M. (2009). Overview of pressure coefficient data in building energy simulation and airflow network programs. *Building and Environment*, *44*(10), 2027–2036.

Cóstola, D., Blocken, B., Ohba, M., & Hensen, J. L. M. (2010). Uncertainty in airflow rate calculations due to the use of surface-averaged pressure coefficients. *Energy and Buildings*, *42*(6), 881–888.

Courant, R., Isaacson, E., & Rees, M. (1952). On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, *5*(3), 243–255.

Djunaedy, E., Hensen, J. L. M., & Loomans, M. (2005). External coupling between CFD and energy simulation: implementation and validation. *ASHRAE Transactions*, *111*(1), 612–624.

Evins, R. (2013). A review of computational optimisation methods applied to sustainable building design. *Renewable and Sustainable Energy Reviews*, *22*, 230–245.

Evins, R., Allegrini, J., & Moonen, P. (2014). Emulating Site-Specific Wind Flow Information for use in Building Energy Simulations. *Building Simulation and Optimization (BSO 2014), London, UK.*

Gracik, S., Heidarinejad, M., Liu, J., & Srebric, J. (2015). Effect of urban neighborhoods on the performance of building cooling systems. *Building and Environment*, *90*, 15–29.

Gresho, P. M. (1990). On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory. *International Journal for Numerical Methods in Fluids*, *11*(5), 587–620.

Harris, M. J. (2004). Fast Fluid Dynamics Simulation on the GPU. In *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics* (First ed.). Addison-Wesley Professional.

Hughes, B. R., & Cheuk-Ming, M. (2011). A study of wind and buoyancy driven flows through commercial wind towers. *Energy and Buildings*, *43*(7), 1784–1791.

Huizenga, C., Abbaszadeh, S., Zagreus, L., & Arens, E. A. (2006). Air quality and thermal comfort in office buildings: Results of a large indoor environmental quality survey.

Ishihara, T., & Hibi, K. (1998). An experimental study of turbulent boundary layer over a steep hill. In *Proceedings of the 15th National Symposium on Wind Engineering* (pp. 61–66).

Jin, M., Zuo, W., & Chen, Q. (2012). Improvements of Fast Fluid Dynamics for Simulating Air Flow in Buildings. *Numerical Heat Transfer, Part B: Fundamentals*, *62*(6), 419–438.

Khan, N., Su, Y., & Riffat, S. B. (2008). A review on wind driven ventilation techniques. *Energy and Buildings*, *40*(8), 1586–1604.

Ku, H. C., Hirsh, R. S., & Taylor, T. D. (1987). A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations. *Journal of Computational Physics*, *70*(2), 439–462.

Ladick, L., Jeong, S., Solenthaler, B., Pollefeys, M., & Gross, M. (2015). Data-driven Fluid Simulations using Regression Forests. *ACM Trans. Graph. Article*, *34*(9), 2–5.

Mochida, A., Tominaga, Y., Murakami, S., Yoshie, R., Ishihara, T., & Ooka, R. (2002). Comparison of various k-E models and DSM applied to flow around a high-rise building - report on AIJ cooperative project for CFD prediction of wind environment -. *Wind and Structures*, *5*(2~4), 227–244.

Moonen, P., Defraeye, T., Dorer, V., Blocken, B., & Carmeliet, J. (2012). Urban Physics: Effect of the micro-climate on comfort, health and energy demand. *Frontiers of Architectural Research*, *1*(3), 197–228.

O'Mahoney, M., Gill, L. W., Broderick, B. M., English, L., & Ahern, A. (2000). *Scope of Transport Impacts On The Environment. ERTDI report 9.*

Parsons, K. (2003). *Human Thermal Environments: The Effects of Hot, Moderate, and Cold Environments on Human Health, Comfort, and Performance* (2nd Ed.). London: Taylor & Francis Inc.

Pisello, A. L., Castaldo, V. L., Taylor, J. E., & Cotana, F. (2016). The impact of natural ventilation on building energy requirement at inter-building scale. *Energy and Buildings*, *127*, 870–883.

Ramponi, R., Angelotti, A., & Blocken, B. (2014). Energy saving potential of night ventilation: Sensitivity to pressure coefficients for different European climates. *Applied Energy*, *123*, 185–195.

Samuelson, H., Claussnitzer, S., Goyal, A., Chen, Y., & Romo-Castillo, A. (2016). Parametric energy simulation in early design: High-rise residential buildings in urban contexts. *Building and Environment*, *101*, 19–31.

Schulze, T., & Eicker, U. (2013). Controlled natural ventilation for energy efficient buildings. *Energy and Buildings*, *56*, 221–232.

Srebric, J., Heidarinejad, M., & Liu, J. (2015). Building neighborhood emerging properties and their impacts on multi-scale modeling of building energy and airflows. *Building and Environment*, *91*, 246–262.

Stam, J. (1999). Stable Fluids. *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 121–128.

Waibel, C. (2012). Non-deterministic Shape Optimization of Wind Cowls by Applying Simulated Annealing and Fast Fluid Dynamics. In *2nd Conference: People and Buildings*. London

Zhai, Z., Chen, Q., Haves, P., & Klems, J. H. (2002). On approaches to couple energy simulation and computational fluid dynamics programs. *Building and Environment*, *37*(8–9), 857–864.

Zhai, Z. J., & Chen, Q. Y. (2006). Sensitivity analysis and application guides for integrated building energy and CFD simulation. *Energy and Buildings*, *38*(9), 1060–1068.

Zuo, W., & Chen, Q. (2009). Real-time or faster-than-real-time simulation of airflow in buildings. *Indoor Air*, *19*(1), 33–44.

Zuo, W., & Chen, Q. (2010). Fast and informative flow simulations in a building by using fast fluid dynamics model on graphics processing unit. *Building and Environment*, *45*(3), 747–757.

Zuo, W., Jin, M., & Chen, Q. (2012). Reduction of numerical diffusion in FFD model. *Engineering Applications of Computational Fluid Mechanics*, *6*(2), 234–247.

Zuo, W., Wetter, M., Tian, W., Li, D., Jin, M., & Chen, Q. (2016). Coupling indoor airflow, HVAC, control and building envelope heat transfer in the Modelica Buildings library. *Journal of Building Performance Simulation*, *9*(4), 366–381.

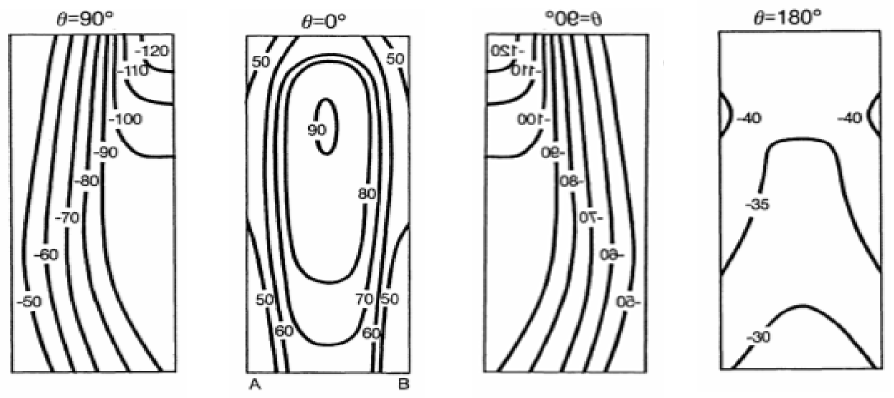


Figure 14: C_p distribution for high-rise buildings with $H > 3b$ (ASHRAE, 2001)

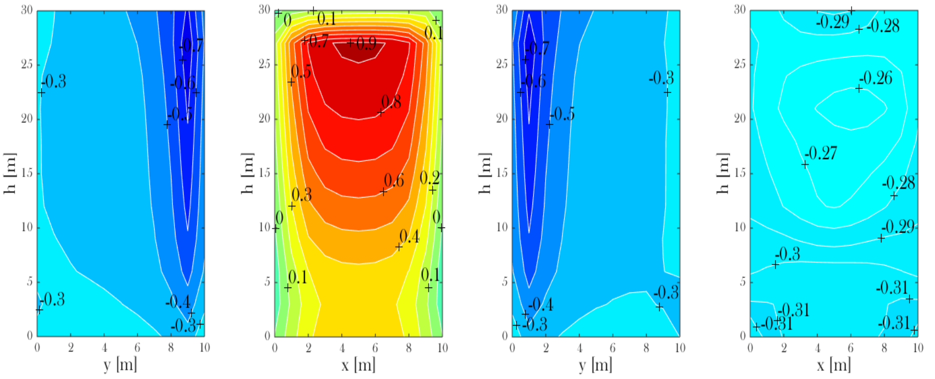


Figure 15: C_p distribution on high-rise building, FFD simulation with dx=1m

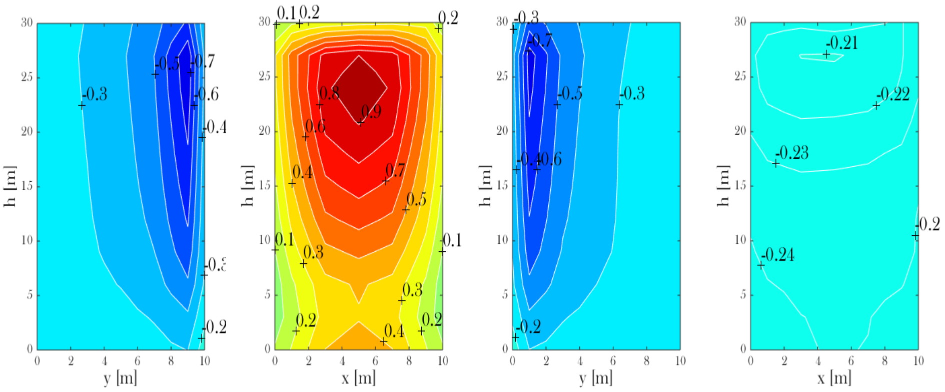


Figure 16: C_p distribution on high-rise building, FFD simulation with dx=2m

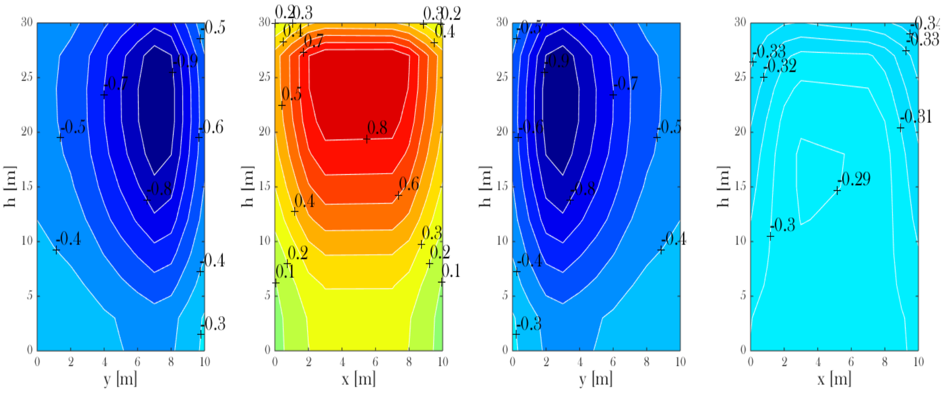
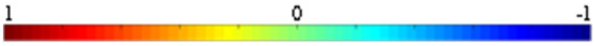


Figure 17: C_p distribution on high-rise building, FFD simulation with dx=5m



Wind pressure coefficient Cp