Exemplifying the importance of simulation parameters in building applications by validating airflow models in Grasshopper

Ran Zhang¹, Christoph Waibel², Xiaodong Xu¹, Ke Liu¹, Thomas Wortmann³
¹ School of Architecture, Southeast University, Nanjing, China
² Chair of Architecture and Building Systems, ETH Zurich, Zurich, Switzerland
³ Institute for Computational Design and Building Systems, University of Stuttgart, Stuttgart, Germany

Abstract

Computational Fluid Dynamic (CFD) programs provide access to affordable and validated airflow simulations. Several digital tools have been developed to make these expert programs more accessible also to architectural and urban designers. This paper considers three airflow simulation plug-ins for Rhino Grasshopper: Butterfly, Eddy, and GH Wind. The first two use OpenFOAM as an engine, and the latter implements a custom Fast Fluid Dynamics (FFD) solver. FFD is, in principle, much faster than CFD, but also less accurate, especially for turbulent flows. Accuracy is a critical consideration in using these simulators, given the impact on design decisions such tools might have. By validating these simulators with experimental and numerical results, this paper studies parameter settings that affect their accuracy and provide recommendations for achieving more accurate and consistent results. We find that, especially with default parameters in OpenFOAM, simulations may result in significant errors, bearing the risk of leading to wrong design choices. Thus, our findings highlight the need for a critical stance when using airflow simulation tools for architectural design and research.

Highlights

- Designers and researchers who apply airflow simulations should not only rely on default settings.
- Accuracy is important for future research and will help architectural and urban designers to achieve more accurate results.
- Designers can achieve more accurate results, especially when using FFD in early design stages.

Introduction

Airflow is a critical consideration in architectural and urban design. Computational Fluid Dynamic (CFD) simulation tools, an example of Building Performance Simulations (BPS), were developed in the past decades to support outdoor environment control, indoor environmental comfort, ventilation analysis, façade pressure analysis, structural analysis, and wind-performance-based urban form design. One of ten challenges in using BPS for sustainable design identified by Hong et al. (2018) is simulation time, especially for large projects. Another critical consideration in using these simulators is their accuracy. More specifically, we are interested in the robustness of results by these programs depending on settings and simulation parameters used. This is particularly relevant for building sciences, considering these tools are likely to be used by architectural and urban designers that may not have a thorough engineering background and expert CFD training. This paper explores the effects of simulation settings such as the inflow profile, compares them with experimental results from the literature, and gives recommendations on achieving more consistent and accurate simulation results. Furthermore, we explore the trade-off between simulation time and accuracy inherent to CFD and Fast Fluid Dynamics (FFD) and provide guidance for more robust settings while maintaining reasonable simulation speed.

Background

Computational Fluid Dynamics (CFD) Simulation

Navier-Stokes equations, combined with turbulence models. This paper considers two CFD tools integrated into Rhinoceros/Grasshopper: Butterfly (Ganj et al. 2019) and Eddy3D (Kastner and Dogan 2021). Both tools provide interfaces to OpenFOAM, a well-validated, open-source CFD engine (OpenFOAM and The OpenFOAM Foundation n.d.). Chronis et al. (2017) use RhinoCFD and Butterfly to integrate CFD into the computational design process of a 3D-printed wall. combine CFD with an energy model to study the effect of solar-induced wall temperature on urban wind flow. Zhang et al. (2020) use CFD and Fast FFD to optimize the shape of a high-rise in terms of aerodynamics. Shirzadi (2020) develops a method to improve the accuracy of urban airflow simulations in OpenFOAM with calibrated k-ε turbulence models. Kumar et al. (2021) use CFD to analyse exterior and passive interior ventilation at the scale of individual rooms. Kabošová et al. (2021) compare CFD results from Procedural Compute, an OpenFOAM plug-in in Grasshopper, with InFraRed, a machine-learning model for generalized CFD predictions and note several discrepancies. Long simulation times are a challenge for such studies, especially in fast-paced conceptual design phases. Less accurate FFD simulations can address this challenge.

Fast Fluid Dynamics (FFD) Simulation

Compared to CFD, FFD is a simplified fluid dynamics solver, initially developed for fast physical visualisation in computer games (Pardyjak & Brown 2007). Originally, FFD was intended to "imitate the right visual" rather than "perform the right values" (Waibel et al. 2021). Zuo & Chen (2009) extended FFD with the addition of turbulence models, but found that FFD remains best suited for laminar flow. Kaushik & Janssen (2015)
investigate the accuracy of a smoke solver in SideFX Houdini using the same wind tunnel results as in this paper (Figure 1, modified after (Ishihara & Hibi 1998)). They conclude that the simulation results of the best animation frame are within 10% error but that it is unclear how to identify this frame. Waibel et al. (Waibel et al. 2017) present GH_Wind, an open-source FFD plug-in for Grasshopper. Their implementation exhibits accuracy on simple validation cases but struggles with “predicting velocity distributions in the wake regions behind obstacles.” Their findings were consistent with previous exploration and development (Jin et al. 2012). We use GH_Wind for the FFD simulations in this paper. Due to its speed, FFD is a valuable tool for conceptual wind flow design. However, compared with the slower CFD, the accuracy of FFD is limited, chiefly by the lack of a turbulence model. This paper thus investigates the accuracy of FFD relative to CFD and physical wind tunnel results and the impact of simulation parameters on this accuracy.

Methodology

Parametric Building Geometry

We use a building volume with width to depth to height ratio of 1:1:2 as a testing geometry (Figure 1). This geometry was used in wind tunnel experiments (Ishihara and Hibi 1998), and for validating simulations (Kaushik and Janssen 2015 and Waibel et al. 2017). The wind speed is 3 m/s at 10-meter height. According to the inflow profile, this velocity increases with height and decreases toward the ground.

In the wind-tunnel experiments, there were 66 fixed measurement points along nine tracks at the z-axis (Ishihara & Hibi 1998). However, they only validate within one single FFD model’s wind tunnel tools. In our situation, it is necessary to fix the gap among three different tools with this geometry by providing other series of tested points.

Different with the previous validation study on 66 stable tested points, the tested points in this study are divided as three series for GH_Wind: Δz/b = 0.1, 0.25, and 0.5, till z/b = 3.5 to provide responding dx (i.e. cell size) setting results when dx = 0.1, 0.25, and 0.5. The time of calculation will change when we switch between the different groups. It will provide relative-convincing results in three different groups rather than in one. However, simulation time will be too expensive and necessary if we use an over-detailed cell size in the CFD model (e.g., dx = 0.1 in Butterfly with 1000 iteration will cost about 27 hours). Therefore, we only export one set of the results when dx = 1, containing the maximum tested positions and sufficient sample variation. Compared with previous research, the overall tested point positions could build more adaptive results variations and decrease the limitation of differences among three tools in 66 tested points.

CFD Simulation

The size of the wind tunnel is 20×10×7.5 meters, with cells of 1×1×1 meter (i.e., dx = 1). We set a wind speed of 3 m/s at 10 meters above the ground as a basic setting. But Butterfly and Eddy3D generate the wind tunnel geometry differently: In Eddy3D, the "box-shaped domain" component allows direct control of the wind tunnels’ length, width, and height, irrespective of the bounding box the analytic geometry. In Butterfly, the wind tunnel is generated around the bounding box of the analysed geometry, with parameters for extensions along the X-, Y-, and Z-axis (Figure 2). We manually corrected the wind tunnel in two different wind tunnel generation logic. Also, the mesh generation logics are different among these three tools: Butterfly apply a relatively detailed "snappyhexmesh" script to transfer the original mesh or "brep" in Grasshopper into muti-hex-mesh shape; Eddy3D contain same OpenFOAM meshing-scripts by default-settings with cube mesh; GH_Wind apply domain-uniform cube-mesh scripts by dividing the tested geometry in the wind tunnel with x, y and z distance and use the cube to switch the original solid space. With "snappyhexmesh" script in Butterfly settings, it occurs huge time-cost for OpenFOAM engine to transfer the original mesh or "brep" in Grasshopper into an integrated mesh, which will increase the time of deleting repeatable and unreadable hexagon surfaces by mending the...
potential space but will decrease the difference between mesh shape with original geometry shape (Figure 3).

Figure 2: Comparison between Wind Tunnel Logics between Butterfly and Eddy3D: boundaries are built towards the envelope of the tested geometries with values of "top, sides, and leewards" in scripts of Butterfly, but built towards the central point projected on XY plane with values of "height, width, and length" in scripts of Eddy3D, which leads a hard-code difference to build for the same geometry.

FFD Simulation
The size of the wind tunnel is $20 \times 10 \times 7.5$ meters, with cell sizes of $1 \times 1 \times 1$ meter (i.e., $dx=1$). We set a wind speed of 3m/s at 10 meters above the ground as a basic setting. The tested points in this study are further divided as three series for GH_Wind: $\Delta z/b = 0.1, 0.25, \text{and} 0.5$, till $z/b=3.5$ to provide responding $dx$ (i.e. cell size) setting results when $dx= 0.1, 0.25, \text{and} 0.5$ (Figure 4, Figure 5, Figure 6). With cube-mesh script in Butterfly settings, FFD solvers can build the tested geometry in the wind tunnel with x, y, and z distance and apply the cubes to switch the original solid space with a fast response. Compared with "snappyhexmesh" scripts in Butterfly, it raises the speed of meshes' transferring but with a relatively low similarity, especially when the tested geometry's envelope is not perpendicular or parallel to the x-, y-, and z-axis (Figure 3).

Figure 3: Comparison of mesh-generating logic between Snappyhexmesh scripts and Cube-mesh scripts.

Data Validation and Correlation Study
Wind velocities are collected from the three tools in Grasshopper and listed in nine sections drawing with correlation calculation, whose section line in $y=0$ and $x/b=-0.75, -0.50, -0.25, 0.00, 0.50, 0.75, 1.25, 2.00, \text{and} 3.25$ (Figure 1). The validation study studies the accuracy of CFD and FFD tools for architectural design and research by validating these three tools with default settings (i), first manual setting (ii), and the final settings (iii). The detailed settings are listed in Table 1.

Table 1: Detailed settings of three tools with default settings (i), first manual setting (ii), and the final settings (iii).

<table>
<thead>
<tr>
<th>Sample Name</th>
<th>Tunnel Size</th>
<th>Cell size</th>
<th>Kinematic Velocity</th>
<th>Iterations</th>
<th>Inflow Setting</th>
<th>Time (mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Butterfly – (i)</td>
<td>$22^*11^*7.5$</td>
<td>Default (1)</td>
<td>NA</td>
<td>600</td>
<td>OpenFOAM</td>
<td>40</td>
</tr>
<tr>
<td>Eddy3D – (i)</td>
<td>$20^*10^*7.5$</td>
<td>Default (1)</td>
<td>NA</td>
<td>1000</td>
<td>OpenFOAM</td>
<td>5</td>
</tr>
<tr>
<td>GHwind (i)</td>
<td>$20^*10^*7.5$</td>
<td>0.1, 0.25, \text{and} 0.5</td>
<td>$1e-1$</td>
<td>dt=1 mean dt=10</td>
<td>ABL</td>
<td>10,6,5</td>
</tr>
<tr>
<td>(slover_wentao)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Butterfly – (ii)</td>
<td>$22^*11^*7.5$</td>
<td>1</td>
<td>NA</td>
<td>1000</td>
<td>OpenFOAM</td>
<td>120</td>
</tr>
<tr>
<td>Eddy3D – (ii)</td>
<td>$20^*10^*7.5$</td>
<td>1</td>
<td>NA</td>
<td>1000</td>
<td>OpenFOAM</td>
<td>5</td>
</tr>
<tr>
<td>GHwind (ii)</td>
<td>$20^*10^*7.5$</td>
<td>0.1, 0.25, \text{and} 0.5</td>
<td>$1.51e-5$</td>
<td>dt=1 mean dt=10</td>
<td>OpenFOAM</td>
<td>10,7,5</td>
</tr>
<tr>
<td>(slover_V2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Butterfly – (iii)</td>
<td>$22^*11^*7.5$</td>
<td>1</td>
<td>NA</td>
<td>600</td>
<td>OpenFOAM</td>
<td>120</td>
</tr>
<tr>
<td>Eddy3D – (iii)</td>
<td>$20^*10^*7.5$</td>
<td>1</td>
<td>NA</td>
<td>1000</td>
<td>OpenFOAM</td>
<td>5</td>
</tr>
<tr>
<td>GHwind (iii)</td>
<td>$20^*10^*7.5$</td>
<td>0.1, 0.25, \text{and} 0.5</td>
<td>$1e-1$</td>
<td>dt=1 mean dt=10</td>
<td>OpenFOAM</td>
<td>10,7,5</td>
</tr>
<tr>
<td>(slover_V2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results
The paper includes three series of comparisons among three tools in experiments: (1) with default settings, (2) with first setting settings, and (3) with the final settings (Table 1).

Result1: default settings
Figure 4 shows the validation of default settings. From $x/b=-0.75$, the data is low-correlated between CFD and FFD tools. The cause is that GH_Wind's default settings load a different inflow file with the OpenFOAM engine. We set the same inflow with Butterfly and Eddy3D in the following two groups.

Results2: the first manual settings
Figure 5 shows the validation with the first manual (i.e., based on 1000-times iterations in Butterfly and Eddy3D and with 1.51e-1 kinematic velocity). Also, we discover
the hard-code problems among these three tools' wind-tunnel logic and improve the wind-tunnel size (Table 1) to build a similar wind tunnel to ensure their same wind air inlet positions. Their correlation coefficients are: (1) Butterfly’s with Eddy3D’s are 0.99 in all section; (2) Butterfly’s with GH_Wind’s are 0.94 in the section of \( x/b = -0.75, 0.81 \) in \( x/b = 0.00 \), and 0.88 in \( x/b = 0.75 \).

**Results3: the final manual settings**

Figure 6 shows the validation with improved (the final) settings, aiming to achieve a high correlation among the three tools’ data. Compared with the first manual settings’ data, the improved correlation are: (1) Butterfly’s with Eddy3D’s are 0.99 in all section; (2) Butterfly’s with GH_Wind’s is around 0.96 (i.e. the minimum is 0.94 in \( x/b = 2.0 \)).

**Discussion and Conclusion**

With default settings, different inflow profiles causes low correlations between the results of the CFD tools (Butterfly and Eddy3D) and the FFD tool (GH_Wind). Using the same inflow profile for all makes the results more comparable. But with the first manual settings, the correlations between the three tools still is relatively low. This low correlation is caused by the settings for kinematic velocity and the different wind-tunnel generation logic of the three tools.

Combining the initial manual settings with an adjusted parameter for kinematic velocity, we observed a substantial increase in the correlation between the CFD tools and the FFD tool, from 0.81 to 0.96. This enhanced accuracy is important for future research endeavors and empowers architectural and urban designers to achieve more accurate results, especially when using FFD in early design stages.

It is important to acknowledge two limitations within this study. Firstly, our findings are primarily based on simulations involving simple building geometries. Therefore, future research should focus on validating our approach using more intricate geometries. Secondly, the distinctive wind-tunnel generation logic utilized by the CFD tools directly impacts the results obtained. Consequently, manual corrections to the digital wind tunnels were necessary. By meticulously adjusting simulation parameters in Butterfly and Eddy3D, we achieved a remarkable reduction in the mismatch between the simulated data and the experimental wind tunnel data, from an initial discrepancy of 29% (when using default parameters) to a mere 4% to 6% mismatch (with calibrated parameters), depending on \( x/b \) (refer to Figure 6). Given the current surge in surrogate modeling, where machine learning models are employed as replacements for certain simulators, our study underscores the criticality of utilizing exclusively validated simulation data as the benchmark for comparison.

To summarize, designers and researchers engaged in airflow simulations should: (1) refrain from relying solely on default settings; (2) possess a comprehensive understanding of the mesh generation process; and (3) when in doubt, opt for more precise simulation settings, even at the expense of higher computational costs.

**Acknowledgement**

This research was partially supported by the National Natural Science Foundation of China (NSFC) under NSFC 51978144, grant 3902-2020-3598 from the initial project funding scheme (“Anschubfinanzierung”) of the Department of Architecture at ETH Zurich, and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC 2120/1 - 3908311618.

**References**


Appendix

Figure 4: Comparison of airflow tools using default settings (i) with the data of (Ishihara & Hibi 1998).
Figure 5: Comparison airflow tools using first manual settings (ii)
Figure 6: Comparison of airflow simulation tools using final manual settings (iii).