AGILE: GIScience Series, 7, 49, 2026. https://doi.org/10.5194/agile-giss-7-49-2026
Proceedings of the 29th AGILE Conference on Geographic Information Science, 16–19 June 2026.
Eds.: Evelyn Uuemaa, Alexander Kmoch, Holger Virro, and Simon Scheider
This contribution underwent peer review based on a full paper submission.

A High-Performance Game Engine-GIS-CFD System for Interactive Urban Wind Decision Support

Xuanchen Zhou [1], Azarakhsh Rafiee [1], and Peter van Oosterom [1]

[1]Department of Architectural Engineering and Technology, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Netherlands

Correspondence: Xuanchen Zhou (X.Zhou-6@tudelft.nl)

Abstract. Wind simulation plays a critical role in urban design and planning. Most existing workflows rely on professional and general-purpose computational fluid dynamics (CFD) software. Although powerful and versatile, such tools are not designed specifically for the urban planning process, where interactivity and real-time feedback are highly emphasized. To address this gap, we propose a responsive and specialized decision support system tailored for urban wind simulation. The system enhances the decision-making process by offering an efficient workflow that seamlessly connects data preparation, simulation, and visualization steps. Its core contribution lies in the tight integration of a high-performance CFD model, a game engine, and multi-source geographical information system (GIS) data within a unified environment. The system is built in the Unity game engine to ensure efficiency, user interactivity, scene modifiability and intuitive visualization. The spatial extent and scalability of the system is expanded by incorporating data from multiple GIS sources to represent real-world urban contexts. The wind simulation is accelerated using Unity compute shaders to achieve fast, fully parallelized computation while maintaining physical fidelity. The system also includes NetCDF-based file export and a web viewer. In our testing platform, the system achieves $8.15\times$ acceleration compared to OpenFOAM. The simulation output is also validated through a case study. By combining these elements, we construct a Game Engine-GIS-CFD triplet that serves as a robust decision support system. Its efficiency and usability make it well suited for wind-related urban planning.

Submission Type. software, model, infrastructure

BoK Concepts. [GC2] Spatial simulation modelling, [GC1] Geocomputation and complex systems

Keywords. computational fluid dynamics, decision support system, wind modeling, digital twin

1 Introduction

Urban wind environment significantly impacts various aspects of human society and city life. Local wind patterns directly influence the comfort and safety of pedestrians (Blocken et al., 2012; Hågbo and Giljarhus, 2024). If utilized properly, strong urban wind can serve as a potential energy source (Juan et al., 2022), but it can also pose risks to high-rise buildings (Giangaspero et al., 2022). Moreover, the strategic management of urban wind is essential for mitigating the urban heat island effect (Jing et al., 2021; Taleghani et al., 2014) and enhancing the dispersion of atmospheric pollutants (Murena et al., 2009; Sanchez et al., 2017). Consequently, understanding these wind-environment interactions is vital for numerous applications.

Due to these significant impacts, it is crucial to consider wind effects during the early phase of urban planning (Kaseb and Rahbar, 2022; Kaseb et al., 2020). Generally, the assessment of urban wind effects can rely on multiple different modeling techniques. The diagnostic mass-consistent flow field model (Singh et al., 2008) has been used in predicting wind field using very limited computational resources. However, this acceleration comes at the cost of accuracy loss, as it cannot capture some important wind features, such as channel effect (Gowardhan et al., 2011). By comparison, computational fluid dynamics (CFD) methods provide higher-fidelity predictions by directly solving the governing fluid equations, enabling the resolution of complex flow phenomena between the buildings (Chu and Wang, 2025).

Currently, most urban wind simulations rely heavily on general-purpose CFD software packages. While offering high accuracy and versatility, they are often not specifically optimized for the iterative workflows required in early-stage urban design. They typically have steep learning curves and require substantial user expertise (Hu et al., 2022). A traditional urban CFD workflow involves a labor-intensive setup stage, including building and

terrain model acquisition, format transformation, mesh generation and simulation configuration (Deininger et al., 2020). This complex and fragmented process creates a barrier to utilizing CFD effectively within the fast-paced decision-making processes of urban design (Hu et al., 2022). Integrating CFD workflows with geographical information system (GIS) data sources offers a promising direction to address this challenge, as GIS platforms already maintain detailed and extensive structured urban datasets and provide standardized access to real-world geometry and environmental information.

In this work, we introduce a highly efficient spatial decision support system tailored for wind-aware urban planning. Our study offers three distinct advantages compared to traditional workflows. First, it is integrated with multiple geographical data sources to automatically acquire real-world geometry. Second, it establishes a unified workflow that seamlessly connects model loading, domain initialization, simulation, visualization, and data storage and retrieval, significantly reducing the manual effort required. Finally, it provides a highly efficient, GPU-accelerated wind solver that delivers near real-time performance. All these characteristics make the proposed system especially suited for the interactive exploration of wind environments during the urban planning process.

The remainder of this paper is structured as follows. Section 2 outlines the governing equations underlying the proposed wind simulator. In Section 3, we discuss the methodology used to integrate the game engine, GIS, and CFD simulation into a unified system. Section 4 presents the results, encompassing the wind simulator validation, performance evaluation, and a brief case study to demonstrate our workflow. Finally, Section 5 discusses some key design choices, and Section 6 concludes with a summary of our primary contributions.

2 Background

The numerical simulation of wind flow is fundamentally governed by the Navier-Stokes equations. However, explicitly resolving all turbulent scales (Direct Numerical Simulation) is computationally prohibitive for urban scales (Huo et al., 2021). Thus, urban wind simulations typically rely on two turbulence modeling approaches: Reynolds averaged Navier-Stokes (RANS) and large eddy simulation (LES) (Chu and Wang, 2025). RANS solves for the time-averaged flow field, modeling the effect of all turbulent fluctuations via the Reynolds stress term. In contrast, LES applies a spatial filter to the Navier-Stokes equations, explicitly resolving the large eddies while modeling the smaller, isotropic dissipative structures using a subgrid-scale model. In this work, we adopt incompressible LES as the governing equations, which is obtained by applying a spatial filter on the incompressible Navier-Stokes equation:

$$\begin{aligned} &\nabla \cdot \bar{\boldsymbol{u}} = 0, \\ &\frac{\partial \bar{\boldsymbol{u}}}{\partial t} + \nabla \cdot (\bar{\boldsymbol{u}}\bar{\boldsymbol{u}}) = -\nabla \cdot \boldsymbol{\tau} - \frac{1}{\rho}\nabla \bar{p} \\ &\qquad + \nabla \cdot \left(\nu \left(\nabla \bar{\boldsymbol{u}} + (\nabla \bar{\boldsymbol{u}})^T\right)\right), \end{aligned} \tag{1}$$

where ρ, $\bar{\boldsymbol{u}}$, and $\bar{p}$ represent constant density, spatially filtered velocity and spatially filtered pressure, respectively. ν is the kinematic viscosity, $\boldsymbol{\tau}$ is the subgrid-scale stress tensor, defined as $\boldsymbol{\tau} = \overline{\boldsymbol{u}\boldsymbol{u}} - \bar{\boldsymbol{u}}\bar{\boldsymbol{u}}$, which needs to be modelled by a chosen subgrid-scale model (Piomelli, 1999).

Moreover, directly resolving the minute turbulence structures within the viscous sublayer requires an excessively fine grid resolution near the wall and complex local grid refinement techniques. Therefore, we employed a Wall-Modeled LES (WMLES) approach to approximate the shear stress at solid boundaries without refining the mesh down to the wall.

Specifically, we adopted the standard logarithmic law of the wall, as recommended by the Architectural Institute of Japan (AIJ) guidelines for practical wind environment predictions (Tominaga et al., 2008):

$$U^+ = \frac{1}{\kappa}\ln(Ey^+) \qquad 30 < y^+ < 200, \tag{2}$$

where E is the wall roughness constant, $U^+ = \frac{U}{u_\tau}$ is the dimensionless velocity, $y^+ = \frac{y u_\tau}{\nu}$ is the dimensionless wall normal distance. u_τ is the friction velocity, and κ is the von Kármán constant.

In urban wind modeling, logarithmic wind profile is often used to estimate the vertical velocity distribution (Tieleman, 2008) as inlet boundary condition:

$$\begin{aligned} U(z) &= \frac{u^*_{ABL}}{\kappa}\ln\left(\frac{z+z_0}{z_0}\right), \\ u^*_{ABL} &= \frac{\kappa U_h}{\ln((h+z_0)/z_0)}, \end{aligned} \tag{3}$$

where $U(z)$ is the inflow value at height z, u^*_{ABL} is the atmospheric boundary layer friction velocity, z_0 is the roughness length, h is the reference height where the reference velocity U_h is obtained.

3 Methods

To establish a highly efficient system for urban wind simulation, a seamless integration of data acquisition, simulation and visualization is essential. This section details the implementation of each module within the proposed decision support system, an overview of the simulation pipeline is shown in Fig. 1.

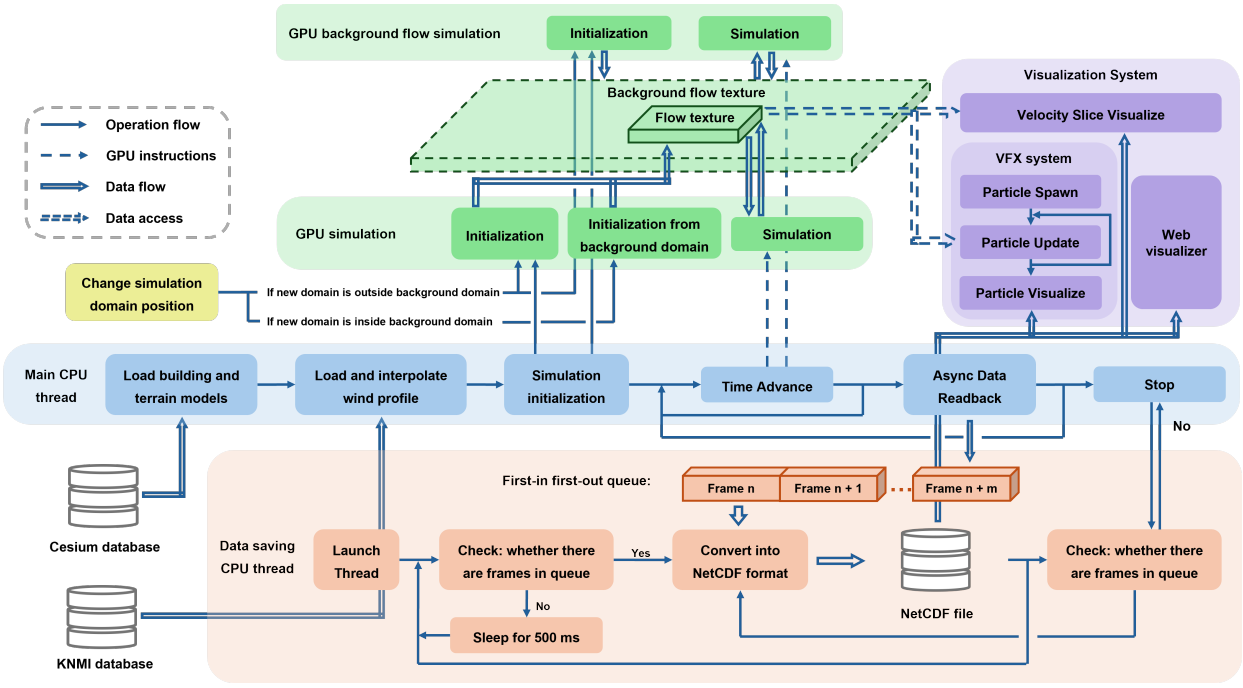

Figure 1. Architecture of the simulator.

3.1 Data Acquisition

Manual geometry acquisition represents a significant workload in traditional urban CFD workflows, which includes building and terrain model acquisition, model loading and format transformation (Deininger et al., 2020). In this work, we create an automated workflow that proceeds directly from region selection to simulation, fully bypassing the tedious model importing and experimental data loading stages.

In order to achieve this, *Cesium for Unity* is used to stream real-world 3D building and terrain models dynamically based on the current viewport (Cesium GS, Inc., 2025). By leveraging the OGC 3D Tiles standard (Open Geospatial Consortium, 2023), the system enables efficient, tile-based data transfer that minimizes memory overhead by preventing unnecessary data loading. Moreover, the wind magnitude and direction are fetched in real-time from the Royal Netherlands Meteorological Institute (KNMI) Data Platform using the Environmental Data Retrieval (EDR) API. Our system queries the k nearest weather stations (where k is a user-defined parameter) relative to the simulation site. The meteorological data from these stations are then interpolated using inverse distance weighting (IDW) to obtain the inlet boundary conditions of simulation.

3.2 Domain Discretization

Traditional CFD workflows often include time-consuming mesh generation and optimization processes, which can take several days to complete (Deininger et al., 2020). In order to adapt to rapid decision-making processes, an automated voxelization-based method that balances computational cost and accuracy was adopted.

After loading the building and terrain geometries, the simulation domain is discretized into a uniform voxel grid. A ray-casting scanline voxelization algorithm is implemented in our solver. It is accelerated using the Unity Physics `Physics.RaycastAll` API, which provides an integrated R-tree acceleration structure. An average triangle traverse time complexity of $O(\log(n))$ can be achieved, where n is the number of triangles (Chen and Belleman, 2022).

3.3 GPU Accelerated Wind Simulation

Many CFD tools oriented toward urban wind simulation integrate OpenFOAM as their core, such as SimFlow, Eddy3D, and Butterfly (Kastner and Dogan, 2022). While being robust and versatile, their reliance on CPU-based computation limits their efficiency, which is a critical factor in the urban planning process.

In contrast, the development of modern GPUs has provided a solid foundation for hardware acceleration in CFD (Dang et al., 2022). To improve the efficiency of CFD computation, we implemented a Finite Volume Method (FVM) solver entirely within the Unity game engine. Eq. (1) is solved in parallel, together with Eq. (2) to provide the near-wall stress tensor. Smagorinsky model is used to account for the subgrid-scale turbulence. To ensure the solver accuracy while maintaining numerical stability, a blended scheme consisting of 75% central differencing and 25% linear upwind differencing is

applied to the advection term. The central difference discretization scheme is applied to the diffusion term. The computationally intensive fluid dynamics calculations are written in compute shaders to leverage the massive parallelism of modern GPUs. The linear matrix systems are solved using stencil-based iterative methods, avoiding the construction of global matrices which leads to additional memory consumption. This integrated design allows for seamless synchronization with the visualization pipeline while maintaining cross-platform compatibility. The specific performance acceleration achieved by the GPU architecture is discussed in Section 4.2.

3.4 Data Export

For subsequent visualization, validation, and analysis, the simulation's flow field can be exported to the NetCDF-4 format (NSF Unidata Program Center, 2023; Humphries, 2024). This format is well-suited for high-resolution fluid dynamics data due to its self-describing nature and native support for partial file reading, which is highly useful in post processing. Furthermore, to maintain a responsive user experience and a stable frame rate, the data export process is offloaded to a background worker thread. Whenever a save request is triggered, this dedicated asynchronous thread handles the serialization and disk I/O, preventing these expensive operations from blocking the Unity main thread and stalling the real-time simulation.

3.5 User Interaction

To enable urban building configuration, several building models are pre-configured in the framework. During the simulation, these assets can be instantiated, transformed (translated, rotated, scaled), or removed to assess a proposed structure's aerodynamic influence.

Moreover, the user may want to relocate the simulation domain during decision-making. We accelerate this process by maintaining a concurrent large size, low-resolution background flow field. When the primary domain is relocated, its initial state is interpolated from this background field rather than starting from scratch. Consequently, the foreground flow does not need to be reset to $t = 0$, which significantly accelerates the decision-making process.

3.6 Fluid Visualization

Real-time, comprehensive visualization is essential for intuitively conveying wind direction, magnitude, and overall flow dynamics. To support the analysis of simulation results, our framework employs three visualization modules within Unity: planar slices, vector glyphs (arrows), and particle tracing. By integrating the CFD solver directly within the game engine, the visualization modules can directly read data from the simulation textures. This enables a zero-copy rendering pipeline, avoiding unnecessary data transfer between the simulation and rendering stages. Specifically, the particle tracing module leverages the Unity Visual Effect Graph (VFX Graph) to directly sample the simulated velocity field textures, updating particle positions and orientations on the fly for highly efficient 3D flow visualization. An example is given in Fig. 6. Additionally, the simulator includes a web-based visualization tool for replaying and analysing saved NetCDF datasets, for example the wind field change after moving a building location.

3.7 Data and Software Availability

The developed Game Engine-GIS-CFD system is open-source and freely accessible to facilitate reproducibility. The core framework is available via GitHub at: https://github.com/puppy2333/UnityWindSimulation. Additionally, the interactive web visualizer has been deployed on Hugging Face Spaces (https://xuanchenzhou-windfieldvisualizer.hf.space/), with its corresponding source code provided at: https://huggingface.co/spaces/xuanchenzhou/WindFieldVisualizer/tree/main.

4 Results

This section provides an evaluation of the proposed GPU-accelerated urban wind simulator. Section 4.1 validates the accuracy of our solver using a single high-rise building test case. Section 4.2 compares the efficiency of our solver with OpenFOAM using MPI. Section 4.3 further demonstrates a typical workflow of our solver.

4.1 Validation

The proposed solver is validated using a simplified version of the airflow around an isolated high-rise building benchmark (Liu and Niu, 2016). This configuration is widely recognized as a standard validation experiment within the Architectural Institute of Japan (AIJ) (Tominaga et al., 2008).

Fig. 2 illustrates the computational domain. The building dimensions are $b \times b \times 2b$, where $b = 4m$ is the building width and depth. The computational domain was discretized using stretched grids, with a minimal grid size of $\Delta x = 0.2m$ close to the building. The grid size increases by a constant factor of 1.08 away from the building. The final domain boundaries terminate at the first cell face that strictly meets or exceeds the minimal distance constraints demonstrated in Fig. 2. At the inlet ($x = 0$), a logarithmic wind profile is applied, as detailed in Section 2. The aerodynamic roughness length, z_0, is set to $0.1m$ to represent typical urban ground roughness. At the outlet, a zero-gradient (Neumann) boundary condition is applied for velocity, alongside a fixed pressure boundary condition. A no-slip boundary condition is enforced at

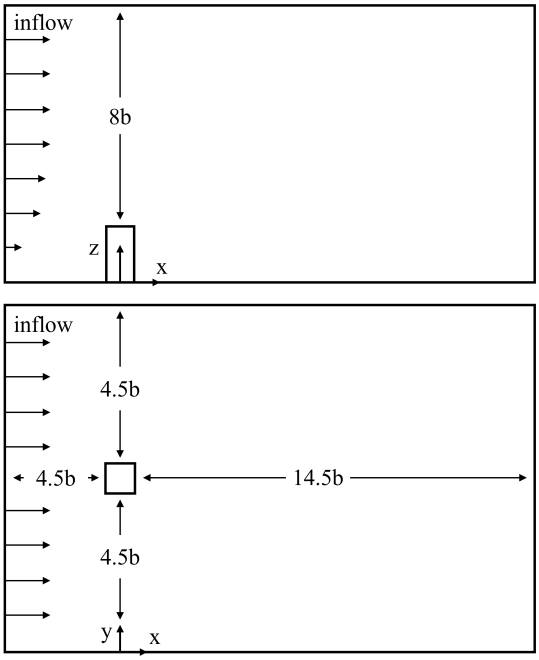

Figure 2. Approximate simulation domain size (up: side view, down: top view).

the ground ($z = 0$). For the top and lateral boundaries, symmetry conditions are applied for both velocity and pressure. The Reynolds number, calculated based on the building height and the reference inlet velocity at a height of $10m$, is 5.333×10^6. The simulation is conducted using a time step Δt of $0.005s$. It ran for a total of $200s$, yielding a maximum Courant number of 0.6324, which ensures the Courant-Friedrichs-Lewy (CFL) condition is satisfied throughout.

The results from our solver are compared against a baseline simulation performed in OpenFOAM v2312 (OpenCFD Ltd, 2023). The OpenFOAM setup employs the PISO algorithm, large eddy simulation with the Smagorinsky subgrid-scale model and the log-law wall function. A residual tolerance of 10^{-5} was consistently applied to both the velocity prediction and pressure correction steps across both solvers. However, the native relative residual formulation in OpenFOAM necessitates frequent global reduction operations, which are highly inefficient on GPU architectures. To mitigate this computational bottleneck, we adopted an alternative solution:

$$r = \frac{\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2}{N_x N_y N_z}. \tag{4}$$

where r is the calculated residual, N_x, N_y and N_z are the number of cells in x, y and z directions, respectively.

To compare our solver's results with OpenFOAM, mean and root-mean-square (RMS) velocity profiles are extracted along two sample lines, $x/b = 3.5$, $y/b = 5$ and $x/b = 3.5$, $z/b = 0.375$. The sample rate is $20Hz$, with data collected over a time window from $50s$ to $200s$. This collection window is selected to begin after more than five flow-through times, satisfying the criteria outlined in (Okaze et al., 2026). As shown in Fig. 3 and Fig. 4, the validation result demonstrates good overall agreement between the two solvers, confirming the usability of our solver.

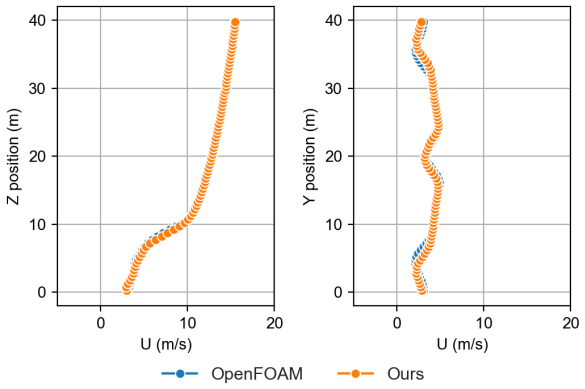

Figure 3. The mean wind velocity distribution around a single building. Left: $x/b = 3.5$, $y/b = 5$, right: x/b=3.5, $z/b = 0.375$.

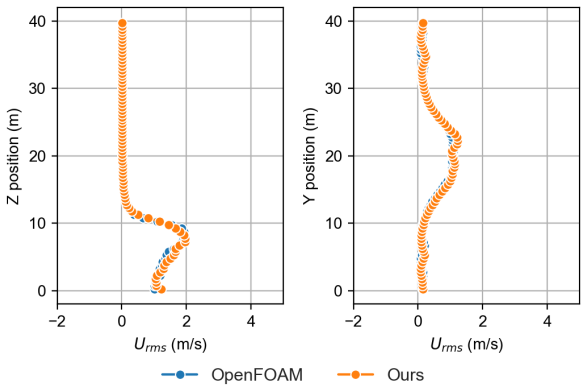

Figure 4. The RMS wind velocity distribution around a single building. Left: $x/b = 3.5$, $y/b = 5$, right: x/b=3.5, $z/b = 0.375$.

4.2 Performance Evaluation

The computational efficiency of our solver is evaluated against OpenFOAM v2312 (OpenCFD Ltd, 2023). The benchmark simulations were conducted on a mobile workstation equipped with an Intel Core Ultra 9 185H CPU and an Nvidia GeForce RTX 4070 Laptop GPU. Due to the hybrid performance-efficiency core architecture of the target CPU and the communication overhead of inter-process messaging, the optimal number of MPI processes for OpenFOAM does not necessarily align with the total physical CPU core count. Consequently, multiple OpenFOAM benchmark configurations were tested and compared with our GPU-based simulator. It was found that 14 processes led to the best performance in our testing platform.

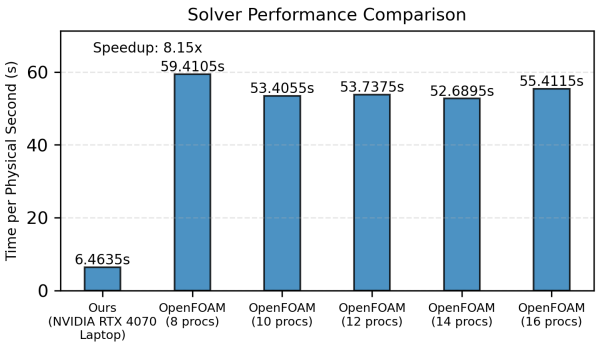

Figure 5. Performance comparison of our simulator and multiple OpenFOAM settings.

As shown in Fig. 5, our simulator achieves approximately $8.15\times$ speedup compared to the most efficient OpenFOAM configuration we tested. This significant speedup is primarily attributed to the efficient parallelization of the CFD algorithms on the GPU.

4.3 Test Case

In order to demonstrate the usability and working pipeline of our simulator, a typical scene including a high-rise building and several low-rise structures is simulated (Fig. 6). The simulation is initialized by defining the domain centroid, dimensions, and grid resolution. For this test case, the scene is chosen at a region in Delft, the Netherlands. The geographical coordinates are (51.9904°N, 4.3766°E), the simulated domain has dimensions of $x = 300m$, $y = 160m$, $z = 80m$, $\Delta x = 1m$. The time step is set to $\Delta t = 0.025s$, yielding a maximum Courant number of 1.02. The inlet boundary condition can be established via two modalities: manual specification and automatic data retrieval. In the manual mode, the user can prescribe specific wind magnitude and direction parameters to test hypothetical scenarios. In the automatic mode, the system retrieves near real-time wind speed and direction from KNMI. The inflow conditions are derived by interpolating data from the four nearest meteorological stations (in this case, two in Rotterdam, one in The Hague, and one in Leiden). The simulation domain is automatically rotated to align with the interpolated wind direction.

Once the boundary conditions are defined, the simulation can be advanced in time. The simulated velocity field can be rendered through several visualization systems, as shown in Fig. 6 (top left, bottom left and right). Users can also instantiate and manipulate several built-in building geometries to observe their aerodynamic influence on the wind field, as shown in Fig. 6 (top right). Furthermore, by coupling the simulation with a nested, coarse-grid background domain (in this case, the domain size is $x = 800m$, $y = 800m$, $z = 80m$, $\Delta x = 4m$), the domain can be relocated during the simulation by interpolating from the large background domain. This architecture significantly accelerates the flow development compared to a "cold start" for each simulation region relocation.

5 Discussion

In this work, we present a Game Engine-GIS-CFD triplet that enables efficient wind-building interaction simulation. An important design choice was the implementation of a custom CFD solver via Unity compute shaders, rather than coupling our system with external software such as OpenFOAM. While general-purpose CFD software offers broader physical versatility, our compute shader-based approach prioritizes the interactivity essential for early-stage design. As demonstrated in Sections 4.1 and 4.2, our approach achieves better performance while maintaining similar accuracy. Furthermore, zero-copy visualization is achieved by keeping the simulation and visualization pipeline entirely within GPU memory, which allows high-fidelity particle tracing via the VFX Graph with minimal overhead.

Our framework also demonstrates that the triplet architecture is a scalable blueprint for urban digital twins. Beyond wind modeling, this approach can also be adapted to other decision making processes, such as flood or noise modeling, providing powerful tools for early-stage urban development.

6 Conclusion

In this paper, we presented a near real-time, interactive urban wind simulation framework designed to bridge the gap between CFD simulation and iterative urban design. By leveraging GPU-accelerated compute shaders within Unity, we demonstrated the feasibility of obtaining efficient feedback on wind flow patterns in response to building modifications.

The practical workflow of the system was validated through a case study. Our results show that by integrating GIS data acquisition, user manipulation, simulation and visualization into a unified environment, the decision-making process can be executed effectively within a real-time loop, thereby offering critical support during the early stages of urban planning.

Future work will focus on four primary directions. First, we plan to integrate quantitative wind comfort metrics, such as the NEN 8100 and Lawson criteria. These additions can provide planners with actionable, regulatory-compliant insights. Second, we aim to further enhance the solver's efficiency by implementing adaptive mesh refinement within compute shaders, allowing for the simulation of much larger domains. Third, we plan to extend the solver's physical models to include scalar transport equations. This will enable the simulation of pollutant dispersion and urban heat island effects, further evolving the framework into a comprehensive

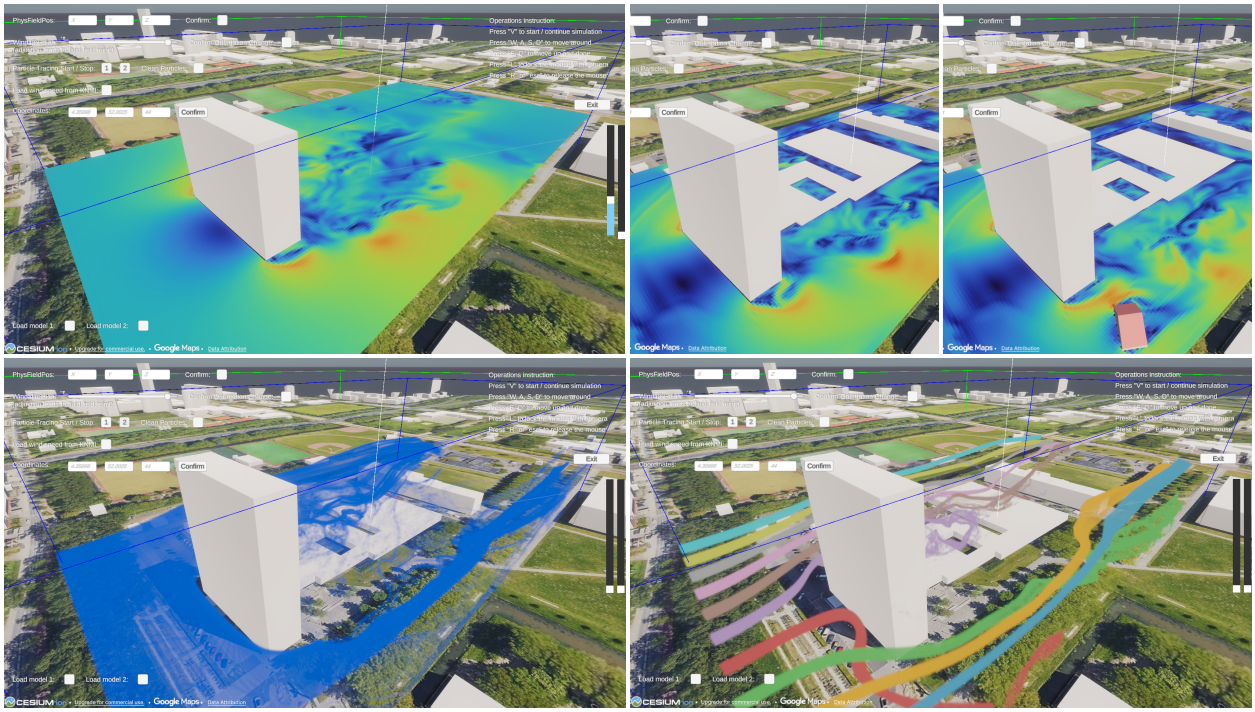

Figure 6. (Top left) The wake generated by a high-rise building. (Top right) Comparison of the fluid field with and without the addition of a small house. (Bottom left and right) Demonstrations of two particle tracing visualization systems.

and versatile tool for climate responsive urban planning. Finally, we aim to further validate the solver by comparing the simulation results against experimental wind tunnel data, and by performing grid convergence tests to evaluate the impact of spatial resolution.

7 Declaration of Generative AI in writing

The authors declare that they have used Generative AI tools in the preparation of this manuscript. Specifically, the AI tools were utilized for language editing, improving grammar, and sentence structure, but not for generating scientific content, research data, or substantive conclusions. All intellectual and creative work, including the analysis and interpretation of data, is original and has been conducted by the authors without AI assistance.

References

Blocken, B., Janssen, W., and van Hooff, T.: CFD simulation for pedestrian wind comfort and wind safety in urban areas: General decision framework and case study for the Eindhoven University campus, Environmental Modelling & Software, 30, 15–34, 2012.

Cesium GS, Inc.: Cesium for Unity, https://github.com/CesiumGS/cesium-unity, accessed: 2026-02-18, 2025.

Chen, J. and Belleman, R. G.: Measvre: Measurement tools for unity vr applications, in: 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), pp. 760–761, IEEE, 2022.

Chu, R. and Wang, K.: CFD in Urban Wind Resource Assessments: A Review, Energies, 18, 2626, 2025.

Dang, G., Liu, S., Guo, T., Duan, J., and Li, X.: Direct numerical simulation of compressible turbulence accelerated by graphics processing unit: An open-source high accuracy accelerated computational fluid dynamic software, Physics of Fluids, 34, 2022.

Deininger, M. E., von der Grün, M., Piepereit, R., Schneider, S., Santhanavanich, T., Coors, V., and Voß, U.: A continuous, semi-automated workflow: From 3D city models with geometric optimization and CFD simulations to visualization of wind in an urban environment, ISPRS International Journal of Geo-Information, 9, 657, 2020.

Giangaspero, G., Amerio, L., Downie, S., Zasso, A., and Vincent, P.: High-order scale-resolving simulations of extreme wind loads on a model high-rise building, Journal of Wind Engineering and Industrial Aerodynamics, 230, 105 169, 2022.

Gowardhan, A. A., Pardyjak, E. R., Senocak, I., and Brown, M. J.: A CFD-based wind solver for an urban fast response transport and dispersion model, Environmental fluid mechanics, 11, 439–464, 2011.

Hågbo, T.-O. and Giljarhus, K. E. T.: Sensitivity of urban morphology and the number of CFD simulated wind directions on pedestrian wind comfort and safety assessments, Building and Environment, 253, 111 310, 2024.

Hu, Y., Xu, F., and Gao, Z.: A comparative study of the simulation accuracy and efficiency for the urban wind environment based on CFD plug-ins integrated into architectural design platforms, Buildings, 12, 1487, 2022.

Humphries, N.: C-sharp-Interface-to-netCDF, https://github.com/NickHumphries/C-sharp-Interface-to-netCDF, 2024.

Huo, H., Chen, F., Geng, X., Tao, J., Liu, Z., Zhang, W., and Leng, P.: Simulation of the urban space thermal environment based on computational fluid dynamics: A comprehensive review, Sensors, 21, 6898, 2021.

Jing, Y., Zhong, H.-Y., Wang, W.-W., He, Y., Zhao, F.-Y., and Li, Y.: Quantitative city ventilation evaluation for urban canopy under heat island circulation without geostrophic winds: Multi-scale CFD model and parametric investigations, Building and environment, 196, 107 793, 2021.

Juan, Y.-H., Rezaeiha, A., Montazeri, H., Blocken, B., Wen, C.-Y., and Yang, A.-S.: CFD assessment of wind energy potential for generic high-rise buildings in close proximity: Impact of building arrangement and height, Applied Energy, 321, 119 328, 2022.

Kaseb, Z. and Rahbar, M.: Towards CFD-based optimization of urban wind conditions: Comparison of Genetic algorithm, Particle Swarm Optimization, and a hybrid algorithm, Sustainable Cities and Society, 77, 103 565, 2022.

Kaseb, Z., Hafezi, M., Tahbaz, M., and Delfani, S.: A framework for pedestrian-level wind conditions improvement in urban areas: CFD simulation and optimization, Building and Environment, 184, 107 191, 2020.

Kastner, P. and Dogan, T.: Eddy3D: A toolkit for decoupled outdoor thermal comfort simulations in urban areas, Building and Environment, 212, 108 639, 2022.

Liu, J. and Niu, J.: CFD simulation of the wind environment around an isolated high-rise building: An evaluation of SRANS, LES and DES models, Building and Environment, 96, 91–106, 2016.

Murena, F., Favale, G., Vardoulakis, S., and Solazzo, E.: Modelling dispersion of traffic pollution in a deep street canyon: Application of CFD and operational models, Atmospheric Environment, 43, 2303–2311, 2009.

NSF Unidata Program Center: Network Common Data Form (NetCDF) Version 4.9.3, https://doi.org/10.5065/D6H70CW6, 2023.

Okaze, T., Kikumoto, H., Ikegaya, N., Nakao, K., Ono, H., Nakajima, K., Imano, M., Hasama, T., Tabata, Y., Kishida, T., et al.: AIJ guidelines on the applications of large-eddy simulation to pedestrian wind environment: Recommendations and validation benchmarks, Journal of Wind Engineering and Industrial Aerodynamics, 269, 106 321, 2026.

Open Geospatial Consortium: 3D Tiles Specification 1.1, OGC Standard 22-025r4, Open Geospatial Consortium, http://www.opengis.net/doc/cs/3D-Tiles/1.1, accessed: 2026-02-18, 2023.

OpenCFD Ltd: OpenFOAM Version v2312, ESI Group, Bracknell, UK, https://www.openfoam.com/, 2023.

Piomelli, U.: Large-eddy simulation: achievements and challenges, Progress in aerospace sciences, 35, 335–362, 1999.

Sanchez, B., Santiago, J. L., Martilli, A., Martin, F., Borge, R., Quaassdorff, C., and de la Paz, D.: Modelling NOX concentrations through CFD-RANS in an urban hot-spot using high resolution traffic emissions and meteorology from a mesoscale model, Atmospheric Environment, 163, 155–165, 2017.

Singh, B., Hansen, B. S., Brown, M. J., and Pardyjak, E. R.: Evaluation of the QUIC-URB fast response urban wind model for a cubical building array and wide building street canyon, Environmental fluid mechanics, 8, 281–312, 2008.

Taleghani, M., Sailor, D. J., Tenpierik, M., and van den Dobbelsteen, A.: Thermal assessment of heat mitigation strategies: The case of Portland State University, Oregon, USA, Building and Environment, 73, 138–150, 2014.

Tieleman, H. W.: Strong wind observations in the atmospheric surface layer, Journal of wind engineering and industrial aerodynamics, 96, 41–77, 2008.

Tominaga, Y., Mochida, A., Yoshie, R., Kataoka, H., Nozu, T., Yoshikawa, M., and Shirasawa, T.: AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings, Journal of wind engineering and industrial aerodynamics, 96, 1749–1761, 2008.