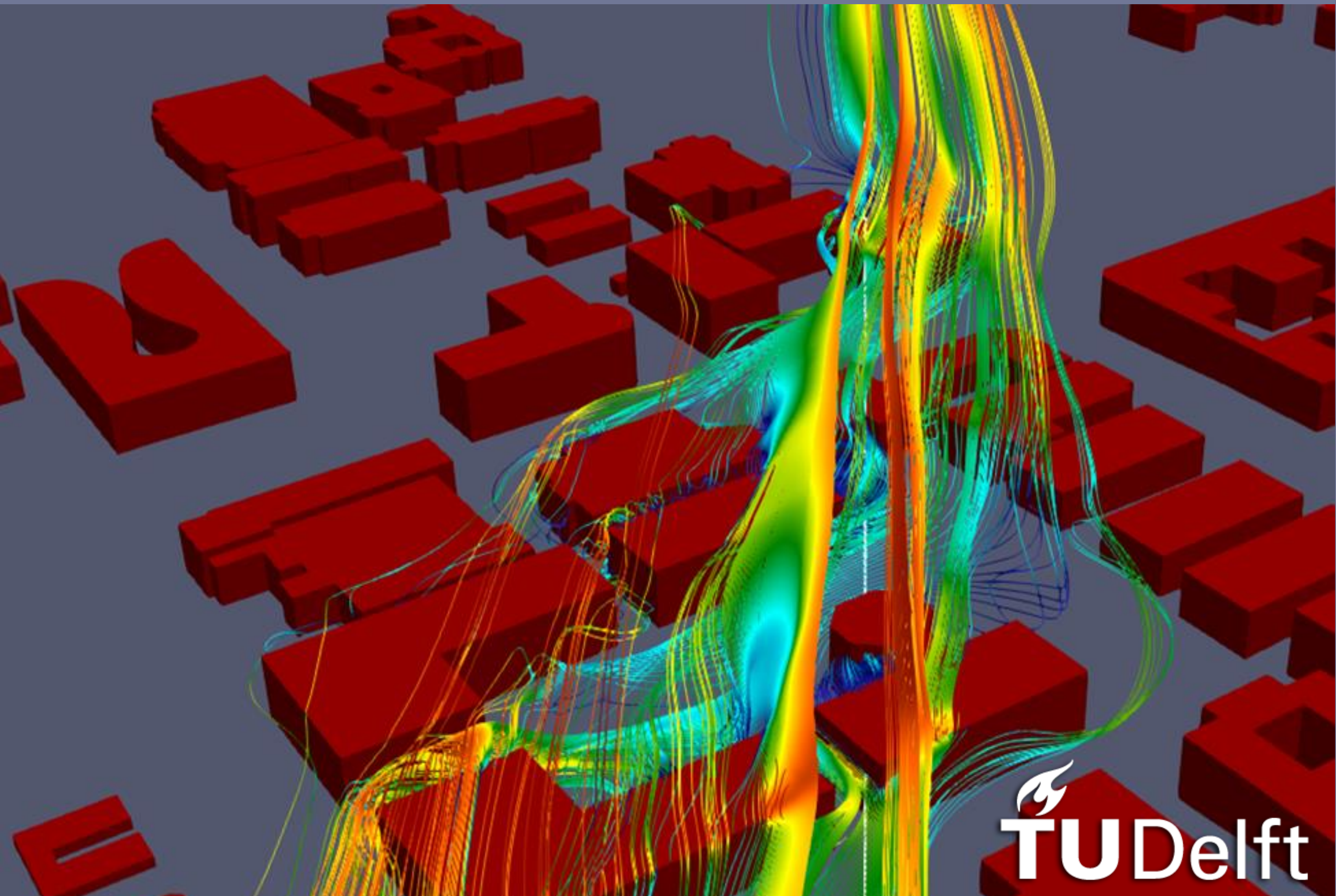
MSc in Geomatics and the built environment

Comparing the impacts of geometry level of detail in computational wind engineering with on-site urban measurements

Pinelopi Eirini Kountouri 2023



MSc thesis in Geomatics

Comparing the impacts of geometry level of detail in computational wind engineering with on-site urban measurements

Pinelopi Eirini Kountouri

January 2023

A thesis submitted to the Delft University of Technology
in partial fulfillment of the requirements for the degree of
Master of Science in Geomatics

Pinelopi Eirini Kountouri : *Comparing the impacts of geometry level of detail in computational wind engineering with on-site urban measurements* (2023)

The work in this thesis was carried out in the:

3D geoinformation group
Delft University of Technology

Supervisors:	Dr. Clara García-Sánchez
	PhD candidate. Ivan Pađen
Co-reader:	PhD candidate. Nadine Hobeika
External advisor:	Nijso Beishuizen, Bosch

ABSTRACT

Urbanization has led to more than half of the world's population living in cities. The design of sustainable and resilient urban environments is becoming increasingly critical to optimize the well-being of their inhabitants and the planet. One of the major challenges in achieving this is the complex wind flow patterns in densely built-up areas, which require accurate prediction and analysis to effectively harness the potential benefits of wind flow, such as natural ventilation and wind power generation. However, the unique features of urban landscapes, such as high-rise buildings, narrow streets, and irregular building shapes, make the analysis of wind flow within urban canopies complicated.

In recent years, Computational Fluid Dynamics (CFD) has become a vital tool for studying wind flow in urban areas. Still, the complex geometries of buildings can lead to challenges, including recirculation, reattachment, intense turbulence, and dead zones. Moreover, vegetation plays a crucial role in controlling wind flow in dense areas by acting as a physical barrier, significantly reducing the wind speed and alter the wind flow direction. Additionally, generating geometry for computational fluid dynamics (CFD) simulation in complex urban environments is a challenging and time-consuming process.

Hence, for this thesis, the City4CFD software will be used to automatically reconstruct a 3D model of Stanford University at LoD1.2, which will significantly reduce the time and effort required to generate the complex geometry necessary for computational fluid dynamics (CFD) simulations in urban environments. The results will be compared to those obtained from an already manually reconstructed model at LoD2.1 and real-world measurements conducted within the area of interest. This will allow me to determine the differences introduced by different level of detail. The research will address several sub-questions, such as the steps needed to automatically reconstruct a 3D city model, the potential improvements in simulation accuracy by increasing LoD, and the impact of complex geometries on wind flow.

The results of the thesis indicated that a more complex geometry Level of Detail (LoD) can enhance the accuracy of simulations by providing a more precise depiction of wind flow patterns. In other words, a higher LoD geometry, such as LoD2.1, can more accurately predict wind patterns in urban environments based on real-world measurements. The study observed that the LoD2.1 model, which incorporates more complex features, generated simulation outcomes that were closer to the measurements compared to the less detailed LoD1.2 model.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors, Clara García-Sánchez and Ivan Pađen, for their invaluable guidance and support throughout the entire process of completing my thesis. Their expertise and enthusiasm for my research were crucial in keeping me motivated and focused. I am incredibly appreciative for their patience, encouragement, and support to assist me achieve this important milestone in my academic career. I would also like to thank Nadine, my co-reader, for providing valuable comments on my report, and Nijso Beishuizen from Bosch for his active participation in all our meetings. Nijso's insightful questions and constructive feedback helped me to improve and enhance my work, and I am grateful for his contributions to my research.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acronyms

ABL	Atmospheric Boundary Layer
CFD	Computational Fluid Dynamics
FVM	Finite Volume Method
GIS	Geographic Information System
LES	Large Eddy Simulations
RANS	Reynolds Averaged Navier Stokes

1 INTRODUCTION

1.1 MOTIVATION

The wind flow patterns in densely built-up areas exhibit a remarkable degree of complexity, with varying speeds and directions. To achieve a sustainable and resilient urban environment, it is essential to understand and mitigate the challenges posed by the complex wind flow patterns [Kim et al., 2021]. As urban areas continue to expand, it is becoming increasingly critical to design buildings and urban areas that can effectively harness the potential benefits of wind flow, such as natural ventilation and wind power generation. However, this requires accurate prediction and analysis of wind flow within urban canopies, which is complicated by the unique features of urban landscapes, such as high-rise buildings, narrow streets, and irregular building shapes.

With more than half of the global population now living in cities, it has become increasingly important to design urban environments that are optimized for the well-being of their inhabitants and the planet [Sousa and Gorlé, 2019]. However, urban structures have become extremely complex due to economics reasons making the analysis of the wind around buildings even more difficult. Given that each city has a unique urban structure depending on regional socioeconomic status, it is critical to investigate the link between urban structures and the wind flow patterns for each city separately [Jung et al., 2019]. One crucial aspect of this process is the analysis and prediction of wind flow within urban canopies, as this information can help improve air quality, ensure pedestrian wind comfort, promote natural ventilation of buildings, and exploit available wind power [Sousa and Gorlé, 2019].

Therefore, the investigation of wind around buildings in the lower part of the atmospheric boundary layer (0 to 200 m) is crucial in order to address the aforementioned environmental issues, thus improving urban environments [Liu et al., 2017]. In the last 20 years, wind flow in urban areas have been extensively investigated. Computational Fluid Dynamics (CFD) has become a vital tool in such research due to the recent considerable advancement of computational power. The fundamental challenge in such research is that the shape of the buildings has a substantial impact on the flow patterns since complex geometries can produce recirculation, reattachment, intense turbulence, and dead zones [Akhatova et al., 2016]. Recent research in built-up areas and open terrains show that complex terrain and vegetation have a considerable impact on wind flow, pollutant dispersion and energy balance [Deininger et al., 2020]. Thus, vegetation modelling in the area of interest should not be omitted from CFD simulations.

In Computational Fluid Dynamics simulation the geometry preparation process is frequently seen as the most time-consuming and laborious operation. The greater and more complicated the geometry, the longer the effort required, hence this issue is problematic for flows around large urban regions with complex geometries [Paden et al., 2022]. To deal with this issue, a method for automatically recreating simulation-ready 3D city models will be used in the current thesis, meaning the open-source software City4CFD [Paden et al., 2022].

Specifically, the primary objective of the current MSc Thesis is to analyze the effects of the level of detail (LoD) of the buildings in computational wind engineering. This study will concentrate on testing and applying a wind flow analysis in two LoD geometries, namely LoD1.2 and LoD2.1. The study area of the research concerns the Engineering Quad in Stanford, California, where in-situ wind measurements have been conducted. Thus, the final outcome from the analysis will be compared with the real measurements to determine the most appropriate model for CFD simulations, balancing simulation time and results that are closer to reality.

1.2 RESEARCH QUESTIONS

The main research question of the current thesis is:

- *What is the impact of different geometry LoDs on wind patterns around an urban environment?*

When considering the core research question, several sub-questions arise automatically:

- *What are the needed steps to automatically reconstruct a 3D city model for the area of interest?*
- *How large can the differences introduced by the LoDs for the buildings be?*
- *Is it possible a higher LoD geometry better predict real-world measurements?*

1.3 SCOPE OF RESEARCH

The main scope of this MSc thesis is to analyze the effects of the level of detail (LoD) of buildings in computational wind engineering. Specifically, the research will focus on testing and applying a wind flow analysis in two LoD geometries, LoD1.2 and LoD2.1, in the Engineering Quad of Stanford, California, where in-situ wind measurements have been conducted. To prepare the LoD1.2 geometry for CFD simulation, the open-source software City4CFD will be used, while the second manually reconstructed model in LoD2.1 used in Sousa and Gorlé [2019] will also be used as input in CFD simulation. The final outcome of the two models will be compared with the in-situ wind measurements to determine the most appropriate model for CFD simulations, balancing simulation time and accuracy. Due to the time-consuming and laborious operation of preparing geometry and run CFD simulations, this study will limit the analysis to only two LoDs. The computational time increases as the level of detail increases, making it necessary to limit the research to these two levels. It is worth noting that this study specifically focuses on Stanford University, where in-situ wind measurements have been conducted for the purpose of validating the results. However, it is important to highlight that the methodology employed, from the automatic reconstruction of a 3D model to the execution of CFD simulations, is applicable to other similar scenarios beyond this specific case study.

1.4 THESIS STRUCTURE

The present thesis's primary chapters are organized as follows:

Chapter 2 presents a summary of the theoretical background and related work, with particular focus on the automatic reconstruction methods of semantic 3D city models as well as the common approaches employed in wind studies.

Chapter 3 outlines the methodology adopted in this study and goes over all important steps and decisions that led to a reliable result. In more detail, the process for preparing the geometry and creating the mesh are described in further depth. The setup for CFD simulations is then briefly explained, including the computational domain and boundary conditions.

Chapter 4 presents the results of the simulations and analyzes the differences found in CFD results between the two models while comparing them with the on-site measurements.

Chapter 5 highlights the thesis's conclusion. First, the key findings are outlined and the research questions are addressed. It then outlines the restrictions and proposes possible improvements in the future.

2 RELATED WORK & THEORETICAL BACKGROUND

2.1 3D CITY MODELLING FOR CFD SIMULATIONS

3D city models are increasingly used in current urban analyses, such as simulations of evacuation scenarios, energy consumption optimization for city districts and wind simulations. Their main advantage is that they can visualize complicated 3D geometry of city objects, such as buildings, vegetation and roads, as well as semantic information, such as their purpose use and construction year [Vitalis et al., 2019]. However, application-specific needs for CFD simulations have yet to be identified; 3D city models are employed only to extract geometry while creating ready geometries for CFDs require more than buildings including several extra steps [Pađen, 2021]. These steps go beyond just using 3D city models and require expertise in data processing. The process begins with pre-processing the 3D city model data to make it suitable for CFD simulations (i.e convert the data into a format that is compatible with the CFD software being used). Next, the simulation domain is defined based on the specific requirements of the analysis, and boundary conditions are assigned to simulate realistic environmental conditions. Then, a mesh is generated over the simulation domain to represent the computational grid, and the CFD simulation is set up and run. Finally, the results of the simulation are post-processed to generate useful information. Each of these steps is critical for ensuring that the CFD simulation accurately represents the real-world scenario being analyzed.

2.2 LEVEL OF DETAIL

The level of detail (LoD) is one of the most important characteristics of a 3D city model. It refers to the model's consistency with its real-world counterpart and has consequences for its usability [Biljecki et al., 2014]. The OGC standard CityGML 2.0 uses the level of detail (LoD) concept to provide a standardized way of describing the geometric detail and semantic information of 3D city models at different levels of resolution. In practise, the LoD concept is used to represent different levels of detail for buildings, trees, bridges, and other world features. This allows CityGML to provide a standardized way to exchange and share 3D city models across different applications and platforms. The Open Geospatial Consortium's CityGML 2.0 standard defines five LoDs (Figure 2.1). The transition from 2D to 3D GIS is represented by LoD0, a depiction of footprints and, potentially roof edge polygons. Extruding a LoD0 model produces a coarse prismatic model called LoD1 that only allows horizontal flat roof surfaces. LoD2 models allows for more detailed multi-pitched roof shape but simpler format than reality, and various semantic classes to model the object's parts (e.g. roof, wall). LoD3 is a highly detailed building model with windows and doors that is far more sophisticated than its predecessor. Finally, a LoD4 completes a LoD3 by adding indoor elements [Biljecki et al., 2016].

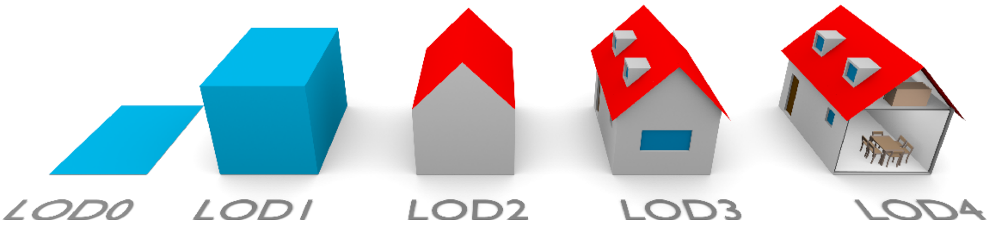


Figure 2.1: The five LODs for buildings defined by CityGML 2.0., from Biljecki et al. [2016]

However, in order to address uncertainties in the initial classification, Biljecki et al. [2016] added a set of 16 LoDs to the CityGML classification. These are now known as "TUDelft LoDs" (Figure 2.2). On the other hand, the concept of LoD is not well established in urban flow simulations. From the CWE perspective the TUDelft classification is more beneficial than the one from CityGML. This statement was supported by [Mirzaei, 2021], who suggested the introduction of TUDelft LoDs in CFD applications.

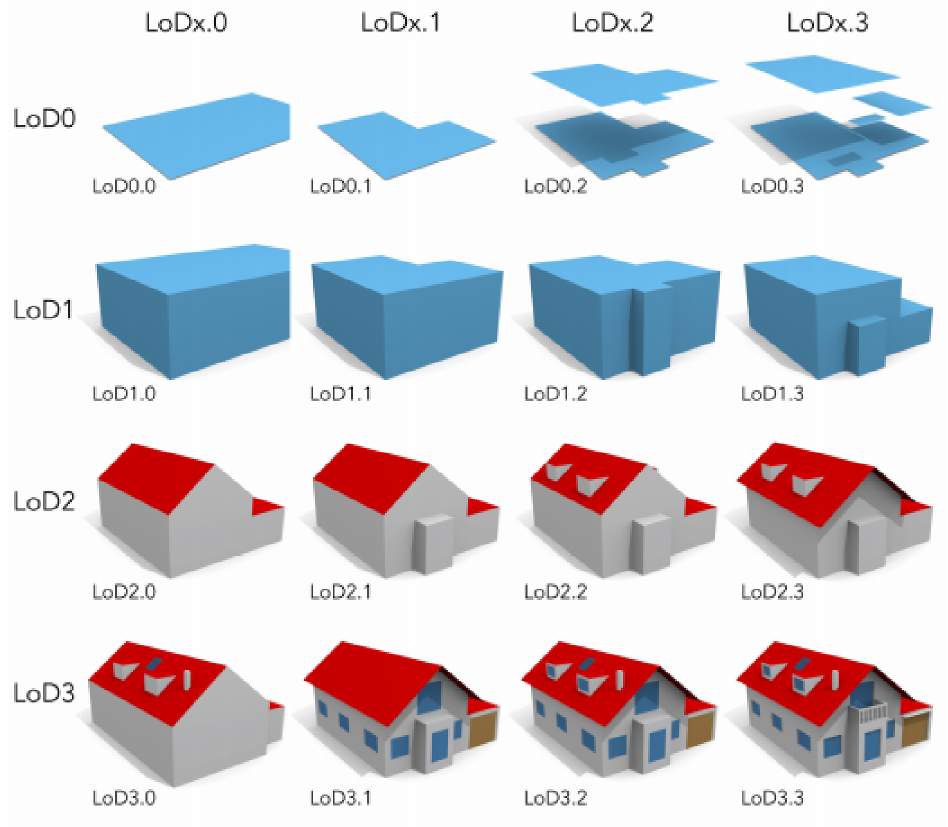


Figure 2.2: Extended LODs for buildings, from Biljecki et al. [2016]

2.2.1 Impact of buildings' LoDs in CFD simulations

As mentioned in the previous section, even if not widely used by the CFD community, 3D city models have long established the idea of levels of detail (LoDs) of the 3D geometric representation of urban regions (Open Geospatial Consortium, 2012) within the geoinformation field. García-Sánchez et al. [2021], compared wind simulation results with different LoD geometry (Figure 2.3) and distinct semantics within a part of TU Delft campus to explore the impact of the water and green surfaces as well as the oversimplified geometries on the simulation results.

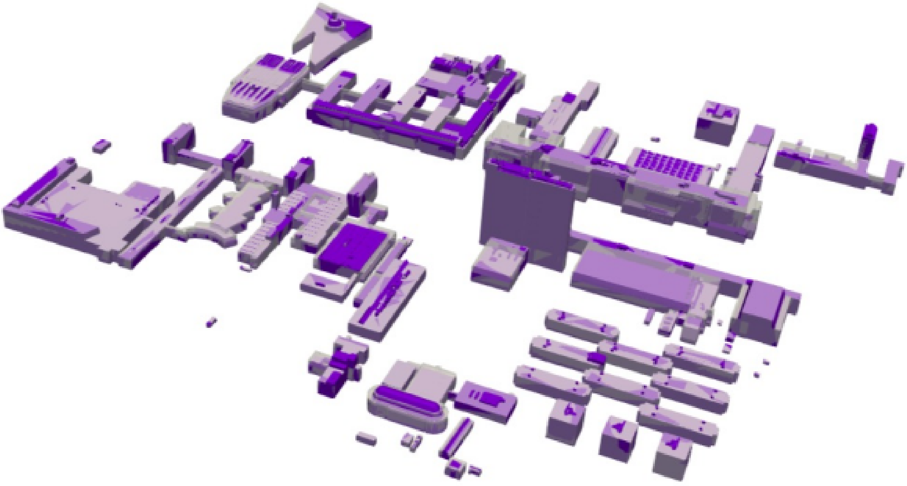


Figure 2.3: Geometry comparison between LoD1.3 with LoD2.2, from García-Sánchez et al. [2021]

The research indicated differences in the velocity between LoD2.2 and LoD1.3, with the velocity being positively correlated to the level of detail. This implies that a better surface representation which deviates from a rough sharp square edge produces more realistic results. Differences in the velocity were also observed betweenLoD2.2 with and without water, noting a lower value when the water was not included in the model. This is related with the importance of the surface roughness which is lower for water, allowing wind to accelerate close to the surface. Finally, consistent lower velocity observed between LoD2.2 with and without water and vegetation, which again explain the essential surface roughness addition. Overall, utilizing LoD2.2 instead of LoD1.3 and including water and vegetation leads to a more realistic case.

2.3 AUTOMATIC RECONSTRUCTION OF SEMANTIC 3D CITY MODELS

Automatic reconstruction of semantic 3D city models is a complex process that involves the use of advanced algorithms and techniques to create 3D models of urban areas from various data sources such as aerial imagery, LiDAR data, and street-level imagery. The models are designed to capture the geometric, topological, and semantic information of the urban environment. One example of software used for this purpose is 3dfier, a free and open-source tool that can generate 3D models from 2D datasets. Another project that focuses on the automatic reconstruction of 3D city models is the 3DBAG project, which aims to develop a comprehensive 3D model of urban areas in the Netherlands. The City4CFD project also focuses on the creation of 3D models, but with an emphasis on their use in computational fluid dynamics simulations. The automatic reconstruction of semantic 3D city models has applications in various fields such as urban planning, architecture, and environmental studies.

2.3.1 3dfier software

3D city models are essential to assess the impact of environmental factors on citizens (e.g noise, wind, air pollution and temperature) because they are used as input into variable simulations and prediction software [Ledoux et al., 2021]. However, 3D models of buildings and other man-made objects such as highways, overpasses, bridges, and trees are difficult to obtain in practice, and manually reconstructing them is time-consuming and arduous. The software **3dfier** is able to solve the aforementioned issues by automatically reconstructing 3D models in the LoD1.2. In more detail, the software takes 2D topographical datasets and extrude them to a height calculated from aerial point clouds. The final 3D dataset is semantically decomposed/labeled depending on the input polygons, and they are combined to produce one or many surface(s) without gaps, self-intersections and non-manifold edges (Figure 2.4).

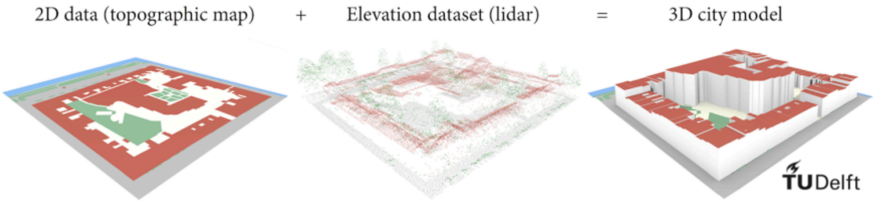


Figure 2.4: Overview of 3dfier, from Ledoux et al. [2021]

2.3.2 3DBAG project

Another relevant example is the method that was employed in the 3D BAG project, which reconstructed 10 million buildings in the Netherlands [Peters et al., 2021]. This method takes well-aligned footprints and height points as input. The input footprint is divided into roof-partition; a planar division of the footprint, each face of which corresponds to a planar piece of the roof. This 2D roof-partition is then extruded into a 3D model. In more detail, the aerial point cloud is clipped on the footprint (step1), planes and their boundaries are then detected (step2) and their intersection and boundaries lines are extracted (step3), the lines are regularized and projected into the 2D footprint (step4), the roof-partition is created (step5) and finally a 3D mesh is created by extruding the roof partition (step6).

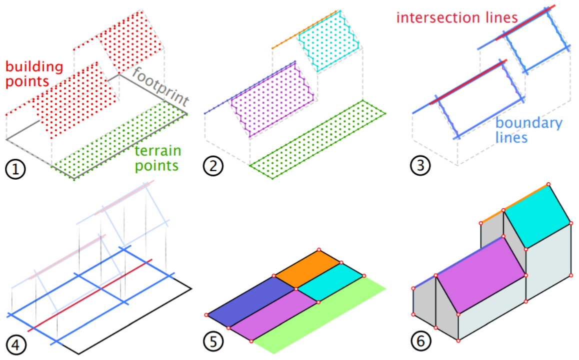


Figure 2.5: Overview of the six main steps of the reconstruction algorithm used in 3DBAG, from Peters et al. [2021]

2.3.3 City4CFD

City4CFD is an Open source software developed by the 3D Geo-information Research Group at Delft University of Technology that aims to automatically reconstruct 3D city geometries ready to be used in micro-scale computational wind engineering simulations. Particularly, from a point cloud, it can automatically reconstruct a terrain and imprint various surfaces (e.g. vegetation, water, roads). The resulting geometry is watertight; meaning that the surfaces are completely enclosed and there are no holes or gaps. The buildings and the surfaces of the model are integrated into the terrain [Paden et al., 2022].

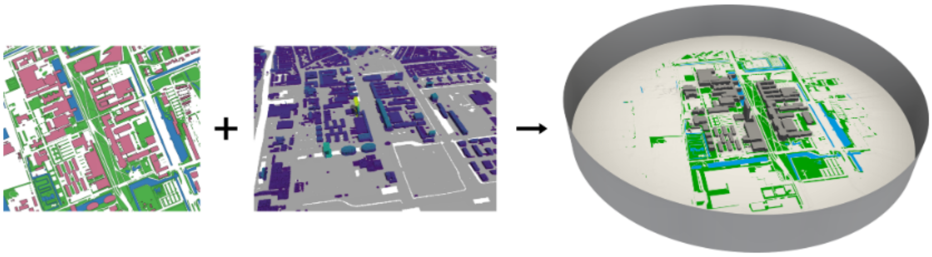


Figure 2.6: Schematic representation of the input data, 2D polygons and a point cloud and output 3D geometry, from Paden et al. [2022]

2.4 URBAN FLOW MODELLING

Studying wind flow in urban areas is necessary to find solutions to wind-environment challenges. More academics are using computational fluid dynamics (CFD) to study urban wind environments as computing capabilities and grid-generation techniques have been developed. Specifically, the Navier-Stokes (NS) equations should be solved in order to represent any urban viscous flow. These equations are a set of differential equations that explain the conservation laws of fluids. The urban wind flow occurs in the lower atmospheric boundary layers (ABL) [Blocken, 2015]. The ABL should be considered when simulating airflow while using CFD. According to Nieuwstadt and Duynkerke, the flow in the atmospheric boundary layer is turbulent, especially when it is near structures or other solid objects. The Reynolds number is used to measure the turbulence of a flow; the greater the number, the more chaos or turbulence a flow has at a certain place [Nieuwstadt and Duynkerke, 1996].

2.4.1 Conducting CFD simulations

The CFD simulation procedure begins with the creation of geometry, followed by grid generation and ends with the mathematical solution of the problem via simulation. Various software or built-in geometric modelers that can be found in several CFD programs are used to create the buildings or system geometry. After the geometry is complete, a grid/mesh will be generated using standalone meshing software or meshing tools within the CFD packages. Specifically, structured Cartesian grids, structured body-fitted grids, or unstructured grids are used to create the grids [Nielsen et al., 2007]. Then, the turbulence model, the initial and boundary conditions will be set once the grid/mesh creation is finished. The necessary boundary conditions for each field including flow velocity, pressure, turbulent kinetic energy, dissipation rate, turbulent viscosity etc. will be set. Finally, the solution monitors, which determine the convergence requirements in terms of iterations in the scaled residuals, are the second parameter after the boundary conditions that should be set. In general, iterative or repeating process is used in CFD solution techniques to continuously improve on a solution. The code will continue running until convergence is achieved. In steady state simulation, meaning that the system's properties and flow conditions are all constant in respect to time, when the slope equals zero and the lines of convergence plot cease changing, a solution is said to be converged. The results obtained from the simulation are then processed using standalone post-processing or build-in programs providing both quantitative and qualitative outputs.

2.4.2 CFD modeling approach

The direct solution of the Navier-Stokes equations for turbulent flows is extremely challenging and computationally expensive [Franke et al., 2010]. Therefore, there are several techniques to simplify the turbulence in order to solve the equations; the four major turbulence modeling approaches include Reynolds Averaged Navier Stokes (RANS), Large Eddy Simulation (LES), Detached Eddy Simulation (DES) and Direct Numerical Simulation (DNS).

- Reynolds averaged Navier Stokes (**RANS**): solves only the mean flow and small eddies and it uses a time-averaged value for the Reynolds number to model all turbulence scales. RANS has the advantages of being inexpensive, widely verified, and available in all CFD codes, but it is less accurate than LES since it does not capture smaller length scales (eddy scales).

- Large eddy simulations (**LES**): solves only large scale motions (eddies). The ability of LES modeling to solve the large eddies in the field of fluid flow makes it the most suitable method for simulating air pollutants. However, the grid in LES is more refined in contrast to RANS and hence requires more computational power for simulation.

- Detached Eddy Simulation (**DES**): combines the idea of both RANS and LES to meet the simulation challenge because neither of them can solve the simulation problem on their own. DES was developed in response to the need to solve the problem of high Reynolds number, immensely separated flows. In order to avoid RANS's well-known weaknesses for heavily separated flows, Epstein et al. (2011) employed the DES solver in their work on simulating a new method to evacuation planning.

- Direct Numerical Simulation (**DNS**): is the most precise method to solve fluid turbulence. The Navier-Stokes equations with this approach are solved using a fine mesh where all of the spatial and temporal scales included in the flow are solved. This approach only uses numerical discretization to solve the Navier-Stokes equations.

2.4.3 Turbulence model

Typically, a turbulence model's role in a CFD simulation is to close the RANS equations by computing the Reynolds stress tensor's components [García-Sánchez et al., 2017]). There are several turbulence models, including k-ϵ Turbulence model, k-ϵ realizable model, k-ϵ RNG model, k-ω, Non-linear models, the SST (Shear Stress

Transport) model, Algebraic stress model etc. The appropriate turbulence model is purely context based. To select the optimal turbulence model that can address the problem with the least amount of error, sensitivity analysis of the available turbulence models is crucial.

2.4.4 Law of the wall

Simulating turbulent flow in areas close to solid objects, such as buildings and the ground surface in a 3D model, is challenging. The air flow in urban regions is influenced by the turbulence and shear stress that take place in proximity to the walls of solid objects [Blocken et al., 2016].

Accurate modeling of turbulent flow near solid objects, such as buildings and the ground surface, requires several considerations. Firstly, it is recommended to ensure that mesh cells near a solid object are parallel to the closest wall of the object for precise modeling of near-wall flow . Secondly, the velocity of turbulent flow near a wall is estimated using the logarithmic law of the wall. This law estimates the average velocity of a turbulent flow relative to the logarithmic distance between a point and a wall in the computational domain [Mandal and Mazumdar, 2015]. However, this law is only applicable to points near a wall that are far enough from the wall. Within the viscous sub-layer, where the Reynolds shear stress is negligible, the law does not hold [Tu et al., 2018].

To check the applicability of the law of the wall for determining flow velocity in a model, we calculate a dimensionless wall unit, y+, for points in proximity to walls in the mesh. The law of the wall is valid for y+ values greater than 3 and up to 30, which is referred to as the log-layer. Points with y+ below 3 are situated in the viscous sub-layer or buffer layer, and the law of the wall is not applicable to them. The points closest to the walls must fall within the log-layer to accurately predict flow velocity using the law of the wall [Blocken et al., 2007].

3 METHODOLOGY

In this chapter, an overview of the thesis workflow is presented along with an introduction to the study area. Subsequently, a detailed description of each step of the case study is provided, leading up to the simulation results.

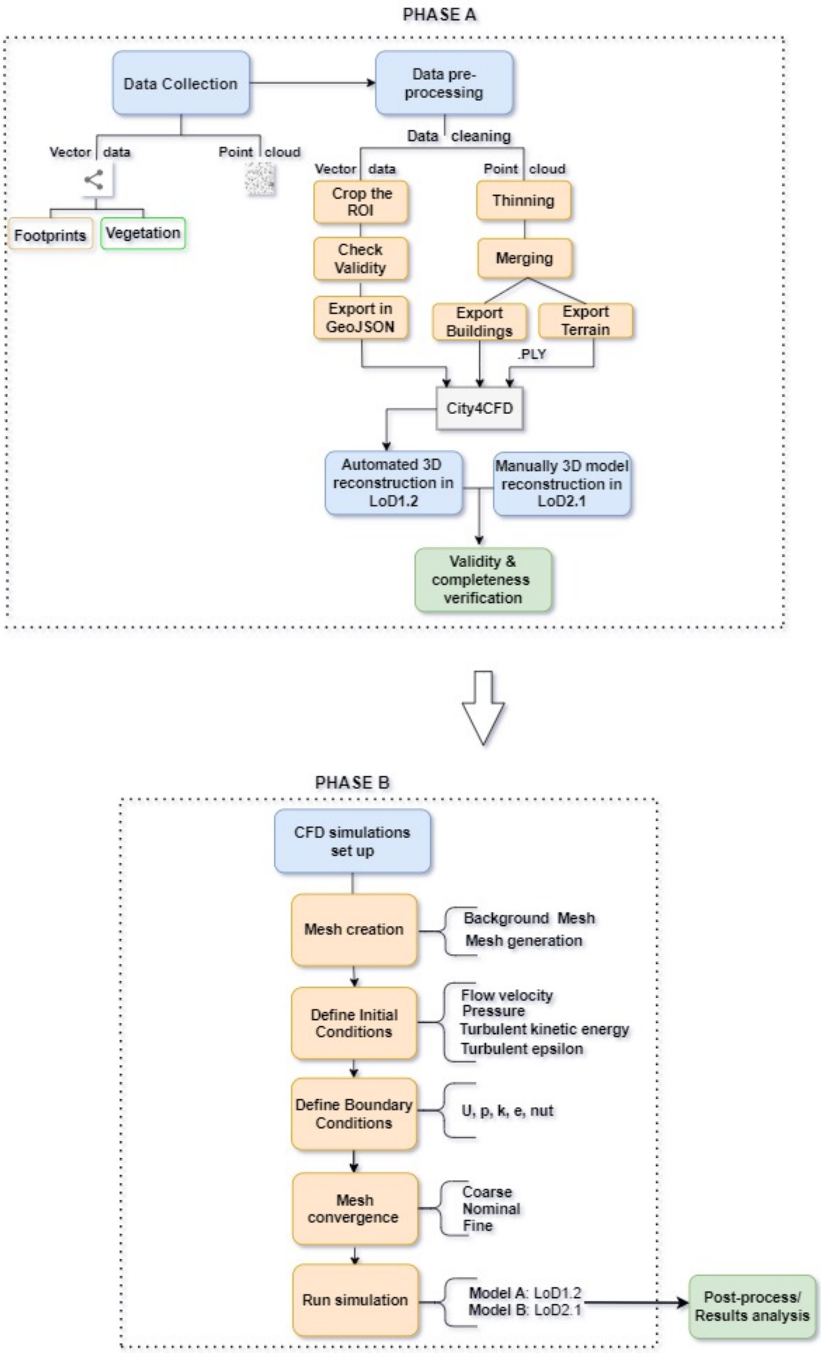


Figure 3.1: Overview of the proposed methodology

3.1 DATA PREPROCESSING

For the purpose of the current thesis diverse set of datasets was used. Based on Chapter 2, Figure 3.1 outlines the workflow followed in this study. The first step in this workflow involved the collection of diverse set of datasets. Specifically, vector data and point clouds from Microsoft Maps [1] and USGS 3D Elevation Program [2], respectively.

3.1.1 Phase A: Geometry preparation

Geometry preparation to run CFD simulation involves the creation of the 3D model to be simulated. This phase is a critical step in the CFD simulation process as it directly impacts the accuracy and reliability of the simulation results. It consists of the following:

- **Data collection:** The gathering of the necessary data is the first step of the process, meaning the footprints of the buildings, the vegetation and point clouds of the region of influence. Specifically, the footprints can be found from cadastral maps or volunteered geoinformation data, vegetation from open data portals or volunteered geoinformation data and finally point clouds from U.S. Geological Survey (USGS), an official website of the United States government.

- **Data pre-processing:** After acquiring all of the necessary datasets, the quality of each one should be assessed, meaning passing the topology and geometry test in QGIS. Then the data can be clipped within a predetermined region, retaining only the data required. Finally, the datasets should be extracted into the proper format, which for 2D polygons of footprints and vegetation is GeoJSON and for point clouds of buildings and terrain is PLY.

- **Automatic 3D model reconstruction:** For the 3D model reconstruction an open source tool namely 'City for CFD' will be employed aiming to automatically reconstruct 3D city geometries suitable for micro-scale urban flow simulations. This tool can generate terrain from a point cloud and imprint different surfaces (e.g. green areas) while it enables the reconstruction of buildings from different sources such as the combination of 2D polygons and a point cloud that would be used in this case. The output geometry (OBJ or CityJSON format) will be watertight while buildings and surfaces will be seamlessly integrated into the terrain.

- **Verify the validity and completeness of the model:** Before using the two 3D models in the CFD simulation it is mandatory to ensure that the input 3D model is valid using the web application val3dity. This tool allows us to validate 3D primitives in accordance with the international standard ISO19107. In case invalid features found in the file should be corrected or replaced with valid ones. Also, it should be checked that the final result will contain zero failed reconstruction geometries meaning that all the buildings were successfully reconstructed.

[1] https://github.com/Microsoft/USBuildingFootprints

[2] https://www.usgs.gov/3d-elevation-program/what-\3dep

3.1.2 Phase B: CFD simulation

The CFD simulation phase involves the mesh generation and simulation runs necessary to obtain a numerical solution. In this phase, the geometry is converted into a discretized computational mesh, which represents the numerical domain. The mesh generation process is critical to obtain an accurate and reliable simulation, and a high-quality mesh can improve the accuracy of the solution while reducing the computational cost. It contains the following steps:

- **Create the background hexahedral mesh:** During the meshing process a blockMesh or background mesh is first produced, defining the outer boundaries and the patches for the boundary conditions. There should be at least one intersection of a cell edge with the tri-surface and cells with an aspect ratio roughly equal to 1, at least near the STL/OBJ surface. The bottom of the domain represents the real physical boundary (terrain), while the top and side faces are artificial boundaries that should be located far enough away from the area of interest to allow the flow to develop freely.

- **Create the final mesh using input geometry and the background mesh:** With the snappyHexMesh utility three-dimensional meshes are created automatically containing hexahedra (hex) and split-hexahedra (split-hex) cells. By iteratively refining the background mesh and morphing the resulting split-hex mesh to the surface, the mesh roughly conforms to the surface (i.e buildings, terrain, vegetation). Mesh Generation in OpenFOAM entails creating a structured mesh of the domain where simulations will be performed. The geometry of the domain is defined by a collection of interconnected elements, namely cells, faces, edges, and vertices that form the mesh.

- **Define initial & boundary conditions:** Defining initial and boundary conditions is a crucial step before running a simulation in OpenFOAM. Initial conditions describe the state of the flow field at the start of the simulation, while boundary conditions describe the state of the flow field at the boundaries of the simulation domain.

- **Mesh convergence:** To ensure a solution is not affected by changes in mesh resolution, a mesh independence verification process can be followed in three steps. The first step is to use a coarse mesh for the initial simulation and ensure that the residuals converge, meaning that residuals and sampled values do not significantly change between iterations and that their values of the last iteration is under a certain threshold. The second step is to refine the mesh at a constant ratio to obtain nominal and fine meshes, and ensure that residuals and sampled values converge for both. The third step is to compare the relative differences between the solutions of the nominal and fine meshes, and between the nominal and coarse meshes. If the difference between the nominal and fine meshes is within an acceptable margin and no longer significantly changes in the solution, then the nominal mesh can be used for final simulations, saving time and resources. If not, continue refining the mesh and repeating the process.

- **Run CFD simulations:** Once the mesh convergence is achieved, the next step is to prepare for the CFD simulation. This involves defining the simulation time, adding the location of probes and finally running the simulation.

- **Analysis of the results:** The final step would be to create plots (e.g residuals plots, slice at specific height for the velocity and the turbulence variables, glyph plot of the flow around the buildings, stream tracer plot for the field) in order to enhance the conclusions.

3.1.3 Footprints

Building footprints dataset was downloaded in GeoJSON format from USBuildingFootprints by Microsoft. Microsoft Maps is offering free building footprint datasets for the whole United States. This collection provides computer-generated building footprints based on satellite photos and computer vision techniques. Footprints in California are from the years 2019-2020 and include **11,542,912** buildings, with WGS84 as the Coordinate Reference System (EPSG: 4326).

Then, a mandatory step was to **clip out** a certain area using QGIS software, because working the analysis in the entire GeoJson file (3.35 GiB) can take a lot of processing time while if I cut it down to the region of interest, it can be easier and faster to deal with it. The region of interest was determined based on Sousa et al. [2018] and Sousa and Gorlé [2019], where measurements campaign was conducted on the Science and Engineering Quad from the 10th to the 12th of October 2017 (SEQ, Lat/Long: 37.438/-122.175) of Stanford University (Figure 3.2).

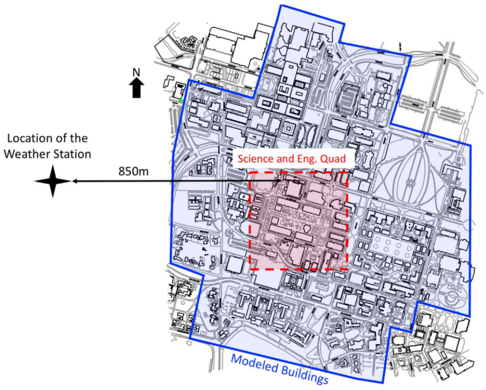
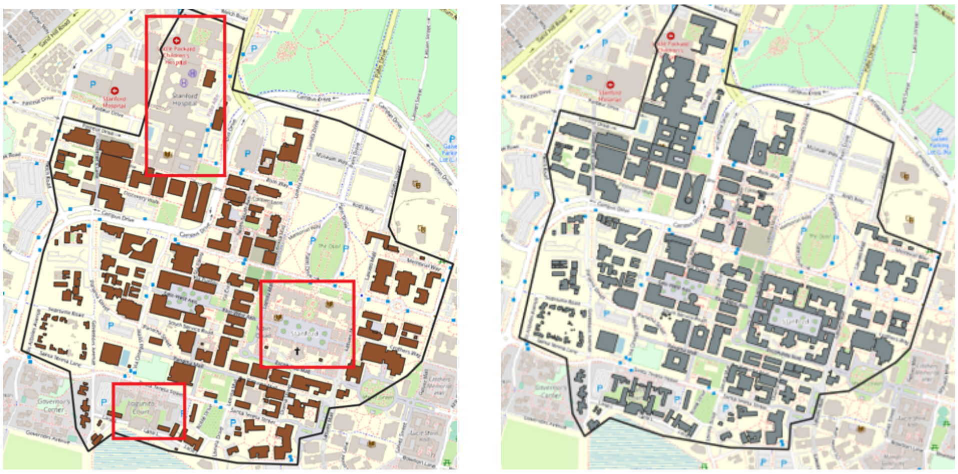


Figure 3.2: Region of Interest in Stanford's campus, indicating the location of the Science and Engineering Quad (SEQ) and the location of the weather station, from Sousa and Gorlé [2019]

During the process I observed that some important data was missing (Figure 3.3a). In order to overcome this issue the footprints of the buildings were also extracted from Open Street Map using the plugin QuickOSM [3] (Figure 3.3b). After comparing the two datasets, I discovered that the footprints retrieved from Open Street Map exhibited significant variations, indicating better precision. The conclusion was reached after comparing the buildings' point cloud with the footprints from Microsoft and the footprints from Open Street Map, showing better adaptation of the latter dataset.

The process continued with the **validity check** on the geometries of the footprint polygons. This algorithm [4] uses the strict OGC definition of polygon validity, where a polygon is marked as valid if it is closed, has a linear ring that does not self-intersect, and its boundary is composed of a set of linestrings. The running of the algorithm show that all the polygons were valid, ending up with 206 footprints of buildings in the area of interest.

[3] https://github.com/Microsoft/USBuildingFootprints
[4] https://github.com/tudelft3d/val3dity



(a) Considerable lack of footprints from Microsoft Maps

(b) Complete footprint dataset extracted from Open Street Map

Figure 3.3: Footprint datasets from different sources

Finally, the last step was to **extract** into GeoJSON format the footprint layer using the correct Coordinate Reference System, which is the NAD83(2011) / California zone 3 (ftUS) with EPSG:6420.

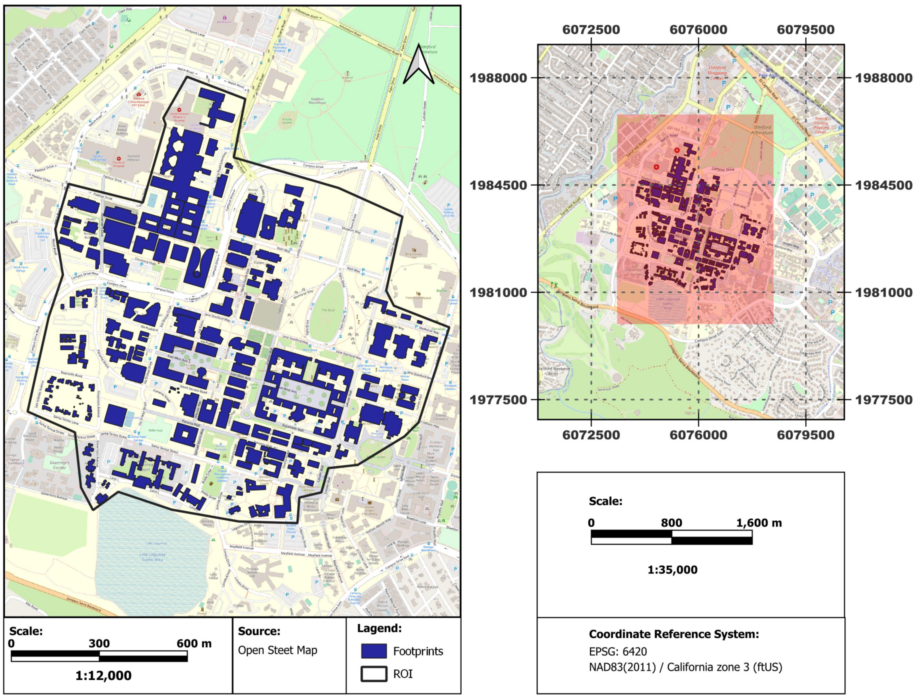


Figure 3.4: Final dataset of footprints extracted from Open Street Map

3.1.4 Vegetation

For the vegetation, a similar procedure was followed, but with the difference that the region of interest was set larger. This is because low vegetation can significantly influence the wind flow, and thus the simulation requires capturing the details of these effects. In order to accurately model the impact of the vegetation on the flow, surface roughness values were assigned based on the height and density of the vegetation. Specifically, for low vegetation, surface layer models was used to define the roughness values. These values were then incorporated into the CFD simulation to more accurately capture the effects of the vegetation on the wind flow.

However, the vegetation polygons extracted from Open Street Map were not particularly comprehensive, with large parts missing. In this case, the **manual digitizing** procedure was followed, with each new polygon receiving a unique ID.

The polygons passed the **validity check**. Finally, in the proper coordinate system, the vegetation layer was exported.

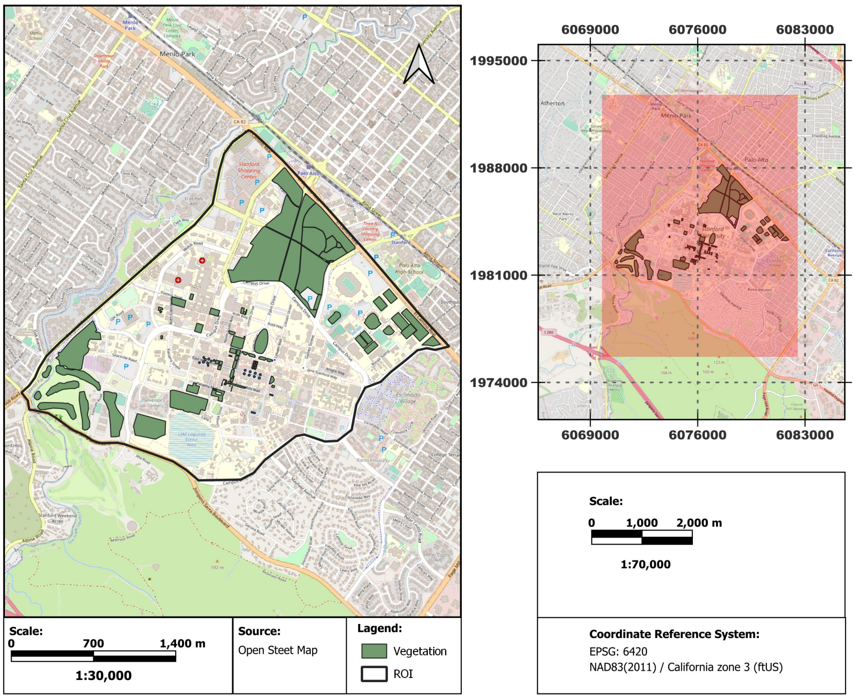


Figure 3.5: Final dataset of vegetation extracted from Open Street Map

3.1.5 Point cloud

The dataset for point cloud was downloaded from USGS. Although, given the fact that the study area did not fully fit into one tile, it was necessary to download 9 tiles to cover the whole area (Figure 3.6). The aforementioned point clouds were published on 19-05-2021 giving up-to-date information.

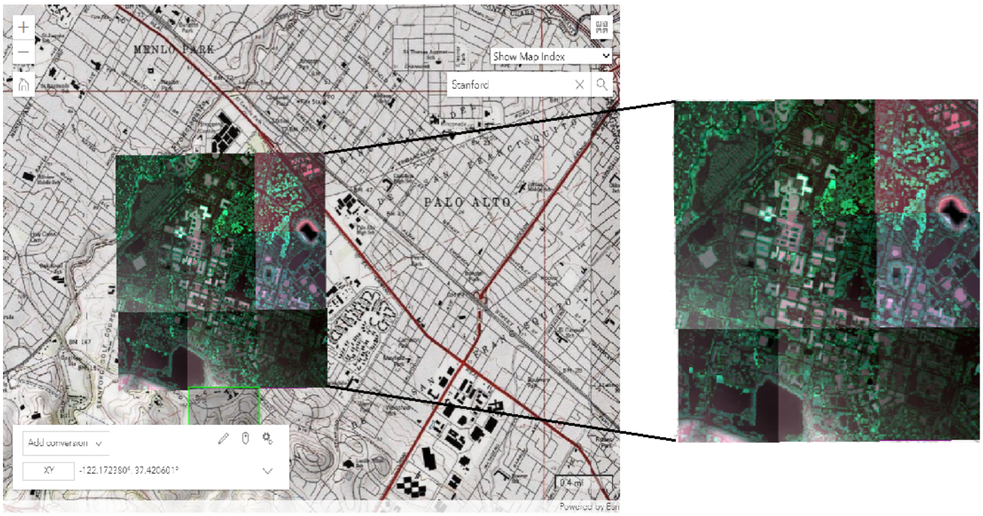


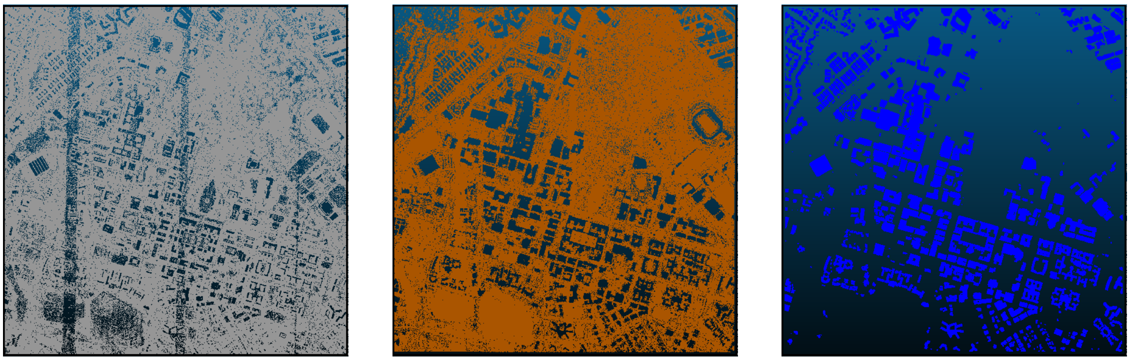
Figure 3.6: Nine point cloud tiles for the area of influence, downloaded from USGS

The process continued into a 3D point cloud processing software CloudCompare where the number of points for each point cloud tile was reduced using the **subsample** tool. They were then **merged** together producing one point cloud. Table 1 shows the size of point clouds for each processing step in greater detail. It is worth mentioning that the initial size of each tile was extremely large making the merging impossible on a laptop until they were reduced in size.

Table 3.1: Point clouds size

Point Cloud ID	Initial size	Subsample (3m)	Subsample (4m)	Subsample (5m)
A20_07259800	14,213,805	960,593	568,101	372,101
A20_07259825	13,750,364	902,819	533,533	350,182
A20_07259850	15,043,391	1,192,588	712,670	468,317
A20_07509800	13,644,984	953,914	564,937	370,333
A20_07509825	13,245,837	922,218	546,703	359,418
A20_07509850	13,281,408	935,342	551,961	363,194
A20_07759800	14,458,018	1,012,626	593,315	389,121
A20_07759825	14,299,969	1,010,662	597,720	393,796
A20_07759850	14,570,931	1,084,323	645,617	427,786
Merged Point Cloud	-	**8,975,085**	**5,314,557**	**3,494,248**

Finally, it was essential to **keep the appropriate classes** of the point cloud; meaning unclassified points with classification code 1 that can be used as addition points to extract buildings in case gaps existed, class 6 containing the buildings and class 2 corresponding to ground (Figure 3.7). The point clouds output in PLY format for buildings and terrain will be used as input in City4CFD. To test the algorithm and the results, extractions from merged point clouds with a 5m subsample will be utilized at first to speed up the process. The final model will employ coarser point clouds.



(a) Class 1: Unclassified

(b) Class 2: Ground

(c) Class 6: Buildings

Figure 3.7: Extraction of different classes from point cloud with subsample (5m)

3.1.6 City4CFD

For the 3D model reconstruction in LoD1.2, a **point of interest** (x=6075392.202, y=1982813.170) that falls inside the Engineering Quad was defined and then a **buffer** of 5000m was used to ensure that all the buildings of interest are inside this circle and will be explicitly reconstructed (Figure 3.8).

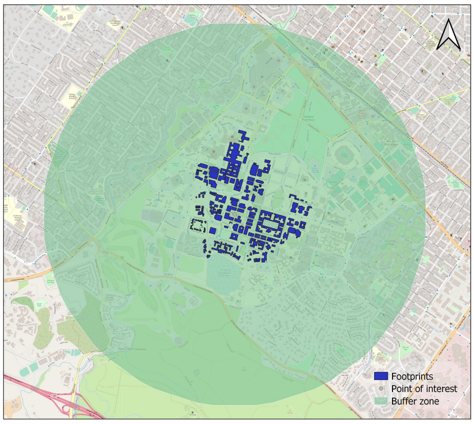
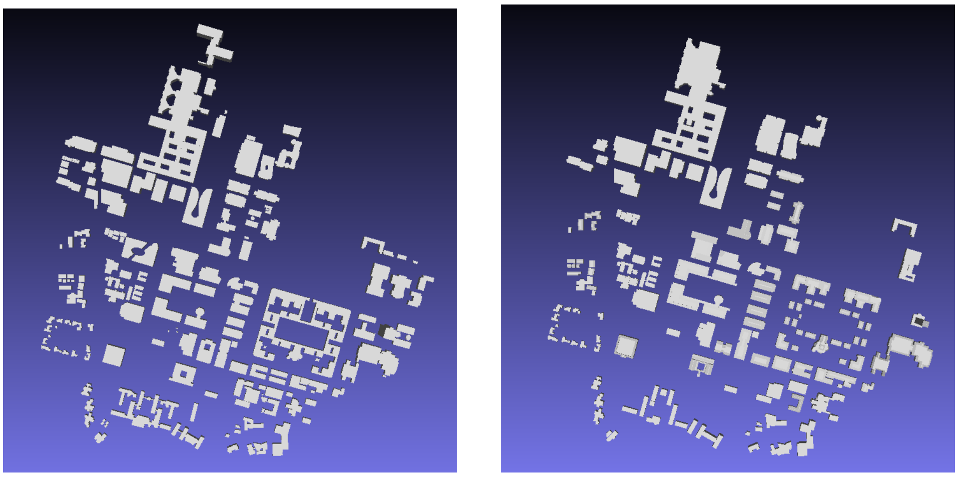


Figure 3.8: Define a point of interest and create a buffer zone of 5000m

The City4CFD executed in about 1 minute and created automatically the 3D model in LoD1.2 with zero failed reconstructions (Figure 3.9a). Comparing with the manually reconstructed 3D geometry (Figure 3.9b), some variations can be observed most probably because the input data sets that were used for the reconstruction come from another time period.


(a) Automatically reconstructed LoD1.2 3D geometry

(b) Manually reconstructed LoD2.1 3D geometry

Figure 3.9: Input 3D models for CFD simualtions

3.2 COMPARISON OF 3D MODELS

In this section the differences between the models in terms of the coordinate systems used, the time periods covered by the input data, and the level of detail of the models will be investigated.

3.2.1 Coordinate system

To be more specific, the primary difference between LoD1.2 and LoD2.1 is the different coordinate system, meaning the different way in which the points are located in space. In more detail, the coordinate system of LoD1.2 model uses US survey foot as the unit of measurement while the LoD2.1 uses meters. In addition, the two models use different reference point to locate points in space. Even though the models represent the same information, the coordinates of the points in the two models differ because they use different units of measurement and they are measured relative to different reference points (Figure 3.10).

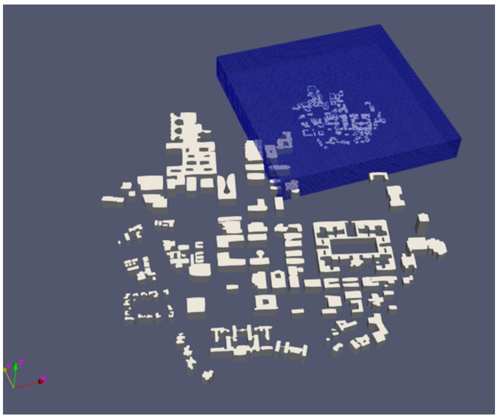
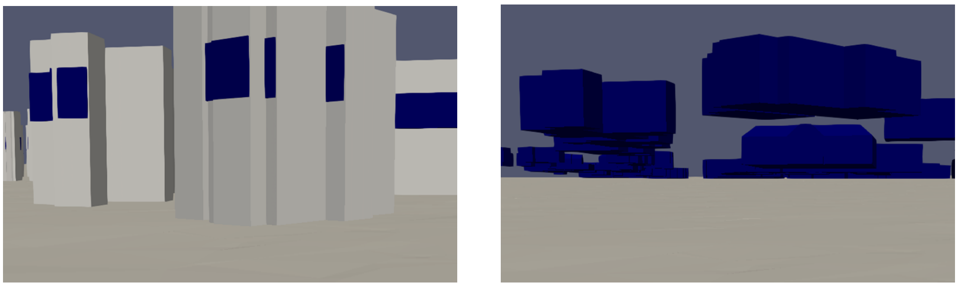


Figure 3.10: Different coordinate system between the two models

In this case because I work with two models with different coordinate system, it is important to convert the coordinates of LoD1.2 model to match the coordinate system of the LoD2.1. One way is to scale and translate the model using the *surfaceTransformPoints* command in OpenFOAM. With the scale option it is possible to specify a scaling factor for each coordinate axis (x,y,z), while with the translate option a translation vector is applied to the points in the OBJ file.

Although, after scaling and translating the model the result did not meet expectations. The buildings in the LoD2.1 model did not match the terrain, and the buildings in the LoD1.2 continue to be significantly higher. These discrepancies suggest that there may be other factors playing a role, that are affecting the alignment of the models, such as the original coordinate data, differences in the resolution of the two models, or discrepancies in the coordinate systems used by the two models can all contribute to misalignment (Figure 3.11).



(a) Higher buildings in LoD1.2

(b) Buildings in LoD2.1 do not match the terrain

Figure 3.11: Limitations due to the different coordinate system

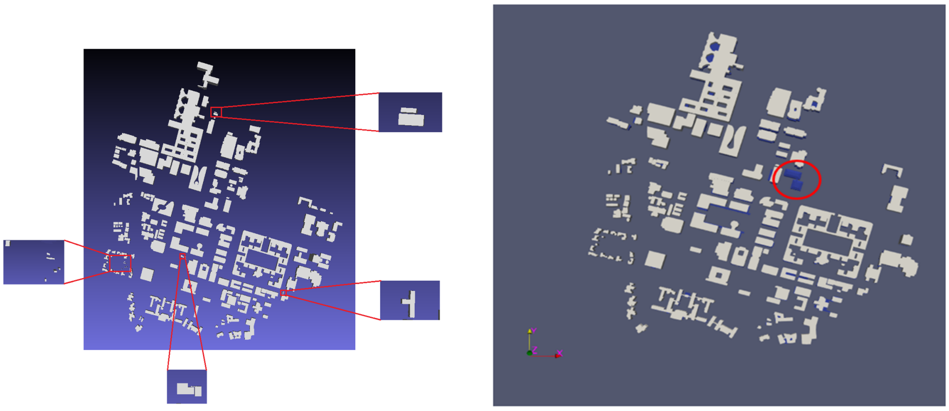
In more details, if the coordinate data for the LoD1.2 model was not accurate to begin with, scaling and translating it may not result in a perfect alignment with the LoD2.1 model. Similarly, if the LoD1.2 model has a coarser resolution than the LoD2.1 model, scaling it to match the size of the LoD2.1 model may result in a loss of detail or accuracy. Finally, if the coordinate systems used by the two models are not identical, simply scaling and translating the LoD1.2 model may not be sufficient to align it with the LoD2.1 model.

Further investigation would be needed to understand the source of these issues and identify possible solutions. One approach might be to convert initial input data (point clouds, footprints, vegetation) from survey foot to meters before used to create the 3D model. One possible way to convert the coordinate system (EPSG) of a point cloud from foot to meters was to use LAStools.

In our case, the classes 6 and 2 were selected for buildings and ground respectively. In the final step of the process, the vegetation and building polygons were transformed to the EPSG 6419 coordinate system within QGIS. After ensuring that all of the input data was in the same coordinate system (using meters as the unit of measurement), the reconstruction process was repeated in City4CFD using the updated data. This allowed for a more accurate and consistent comparison between the different models, as all of the data had the same units of measurement.

3.2.2 Time period of input data

The fact that the input data was gathered at different time periods can potentially impact the two 3D models' output. Due to changes in the built environment at the time the data was obtained, it is likely that some buildings that exist in one model may not be present in the other (Figure 3.12a). Similar to the previous example, a structure that was present in one model might have been demolished or removed between the time the data was gathered and the creation of the second model, resulting in its absence from the model (Figure 3.12b).

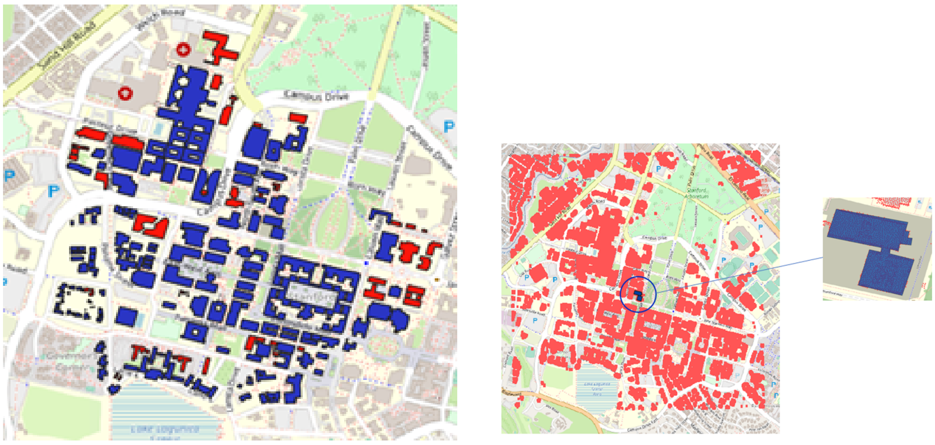


(a) Buildings that not exist in LoD2.1

(b) Buildings that not exist in LoD1.2

Figure 3.12: Limitations due to the different time period of input data

Comparing 3D models produced from input data gathered at different periods in time requires taking into account the potential changes in the built environment. In order to minimize the impact of using input data from different time periods when comparing the 3D models, certain steps were taken in QGIS to update or modify the input data. Specifically, the footprints of buildings that do not exist in LoD2.1 were removed (Figure 3.13a), and the footprints of two buildings that do not exist in LoD1.2 were manually digitized based on the point cloud of the buildings (Figure 3.13b). After updating the input data to include changes in building footprints that occurred between the two time periods, a new 3D model was created using City4CFD. This model incorporated the updated building footprint data and was intended to provide a more accurate representation of the existing situation in 2017, when the measurement campaign has taken place.



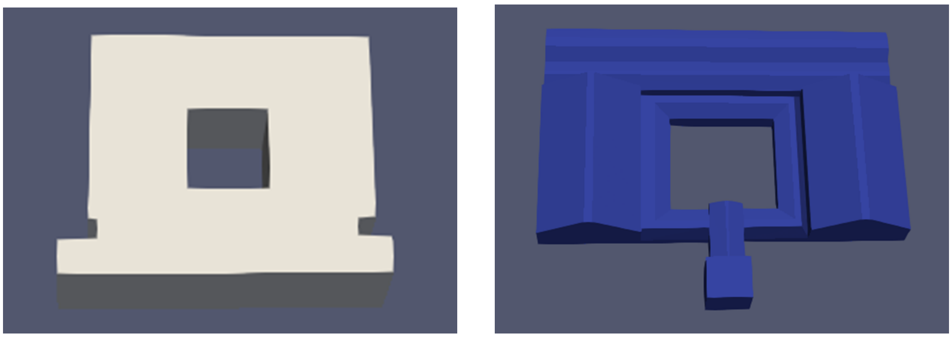
(a) Delete footprints that not exist in LoD2.1

(b) Manually digitising of two buildings footprint that not exist in LoD1.2 based on point cloud

Figure 3.13: Limitations due to the different time period of input data

3.2.3 Level of Detail

The level of detail (LoD) of a 3D model refers to the amount and complexity of 3D geometry it represents. Models with a higher LoD typically depict more intricate and precise details with greater realism, while models with a lower LoD may depict simpler, less complex details. Differences in LoD can be observed in the complexity and precision of the 3D geometry present in the model. For example, a LoD1.2 model might only include basic 3D geometry such as flat roof surfaces (Figure 3.14a), while a LoD2.1 model might include more complex and detailed features such as multi-pitched roofs (Figure 3.14b).

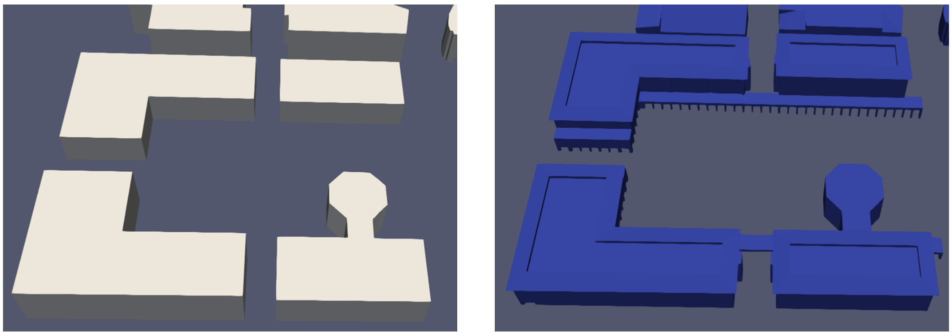


(a) Horizontal roof surface in LoD1.2

(b) More detailed multi-pitched roof shape in LoD2.1

Figure 3.14: Differences in the roof shape between the models

In addition, LoD1.2 is capable of capturing detail up to 2.5D (Figure 3.15a). The term '2.5D' is used to describe that the 3D model is not capturing the full 3D complexity of the urban environment. On the contrary, within LoD2.1 the full 3D geometry is represented including features such as open passages and columns (Figure 3.15b).



(a) LoD1.2 captures until 2.5D

(b) LoD2.1 represents full 3D geometry (open passages, columns)

Figure 3.15: Differences in the level of detail between the models

3.3 CFD SIMULATION SET UP OF CASE STUDY

The first step of the simulation setup was to input the initial wind direction and wind speed, followed by the addition of boundary conditions specified in Table 3.4. Next, the 3D geometry was prepared, and finally, the simulation was configured and ready to be executed.

To perform fluid dynamics simulations I used the open-source software OpenFOAM, version 7. The simulations in this study assume that the flow is steady, incompressible and the temperature stratification is neutral.

3.3.1 Governing equations and discretisation schemes

Santiago et al. [2010] conducted a study to assess the impact of wind direction on the dispersion of flow plumes, using both the Large Eddy Simulation (LES) model and the Reynolds-Averaged Navier- Stokes (RANS) model. They observed that LES requires a higher computational cost compared to RANS. They concluded that using RANS with the k-ϵ turbulence model is a reasonable and suitable approach for simulating wind flow in urban environments. This approach has been increasingly adopted for pedestrian-level wind studies, as demonstrated in prior research by Blocken et al. [2004], Yoshie et al. [2007], and Blocken and Persoon [2009].

Thus, for this study, the Reynolds-averaged Navier-Stokes (RANS) approach with the k-ϵ turbulence model will be used. The equations that govern the flow are:

$$\frac{\partial \overline{u_j}}{\partial x_j} = 0 \tag{3.1}$$

$$\overline{u_j}\frac{\partial \overline{u_i}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + v\frac{\partial^2 \overline{u_j}}{\partial x_j \partial x_j} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j} \tag{3.2}$$

where $\overline{u_i}$ are the 3D velocity components averaged by time, ρ is the density, $\overline{p}$ the pressure and v is the kinematic viscosity. The Reynolds stress tensor $\overline{u_i' u_j'}$ is unknown and requires a turbulence model to be closed (k-ϵ).The Reynolds stress tensor in this model is computed using a linear eddy viscosity approach where the the turbulence viscosity is computed within the following equation:

$$\nu_t = 0.09\frac{k^2}{\varepsilon} \tag{3.3}$$

Lastly, there are two equations for the turbulence variables, k and ϵ:

$$u_j\frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \varepsilon = 0 \tag{3.4}$$

$$u_j\frac{\partial \epsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial \epsilon}{\partial x_j}\right] + C_{\varepsilon_1}\frac{\varepsilon}{k}P_k - C_{\varepsilon_2}\frac{\varepsilon^2}{k} \tag{3.5}$$

where P_k is the turbulent production term and σ_k, σ_ε, Cε_1 and Cε_2 are model constants values: 1.0, 1.3, 1.44 and 1.92 accordingly.

3.3.2 Initial Conditions

The initial conditions involved setting up four parameters: flow velocity, pressure, turbulence kinetic energy (TKE), and dissipation rate. The wind speed and direction used as input for the CFD simulation were obtained from the weather station closest to the study area. Specifically, the weather station is located at 10m above ground level and about 800m east of the center of the SEQ (Figure 3.16).

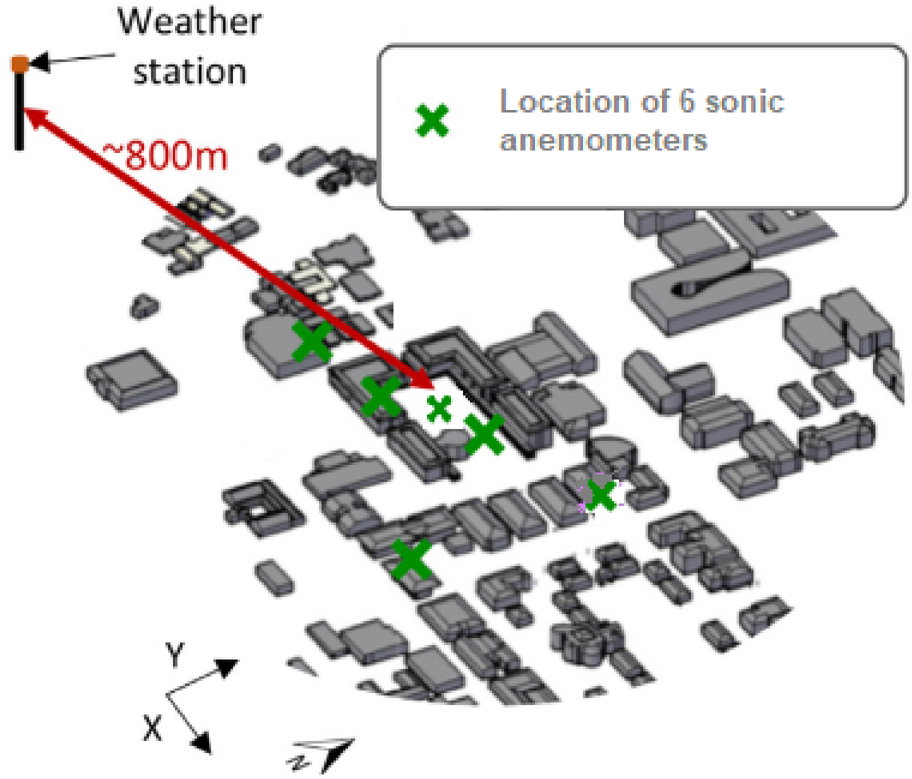


Figure 3.16: . The area of interest on Stanford's campus, along with the sensors and the weather station location

To determine the input wind direction and velocity, I analysed wind velocity and turbulent kinetic energy over a period of three days, from October 10 to October 12, 2017, as documented on the paper of Sousa and Gorlé [2019].

My goal was to identify an hour in which wind velocity and TKE remained stable, as this would allow me assume a steady flow. I observed fewer fluctuations in wind velocity and TKE during the night, likely due to the reduced solar heating and increased atmospheric stability. After analyzing the data, I found that the most stable hour was between 3-4 am on October 12th (Figure 3.17, 3.18). Therefore, I obtained the inlet velocity and the wind direction from the weather station at that time for the simulation. The wind direction corresponds to an angle of 217.55°, where the reference direction is east (0°) and angles are measured counterclockwise. Therefore, the wind is blowing from the southwest direction with a velocity magnitude of 3.06 m/s.

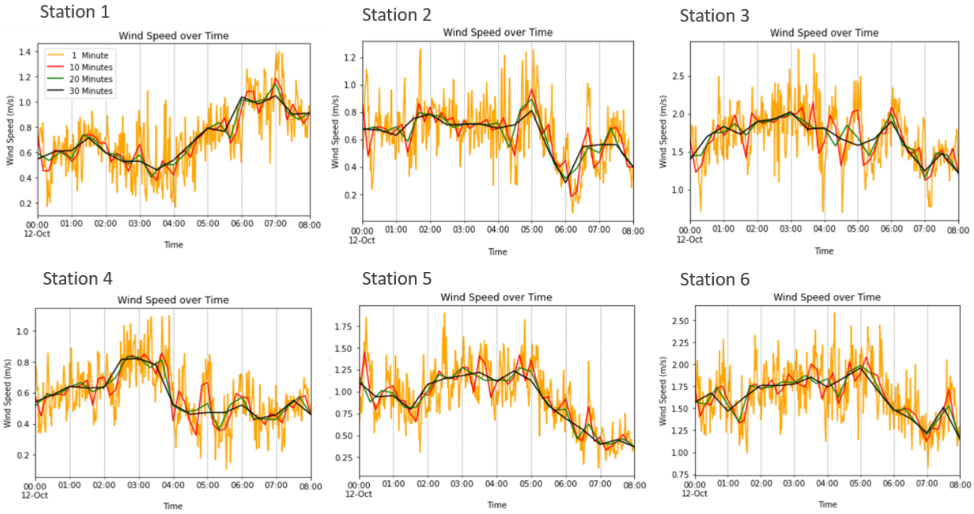


Figure 3.17: Wind speed averaged by time for October 12th, 2017

The methodology used to calculate the Turbulent Kinetic Energy (TKE) plot involves several steps. First, the raw wind speed data is processed by grouping the data into different time intervals of 1 minute, 10 minutes, 20 minutes, 30 minutes and 1 hour. For each time interval, the average wind speed is calculated. Next, the U and V components of the wind speed are calculated using trigonometric functions based on the wind direction and speed data. Then the U' and V' are the velocity fluctuations in the x and y direction, respectively. The velocity fluctuations represent the deviation of the actual velocity from the mean velocity of the fluid. Thus, to calculate the TKE the following formula was used:

$$TKE = 0.5 * (U'^2 + V'^2) \tag{3.6}$$

In this case, the W component is eliminated because I run simulation with neutral atmospheric stratification, where the vertical velocity component W is considered to be zero. Finally, the TKE values are plotted over time for each of the five time intervals.

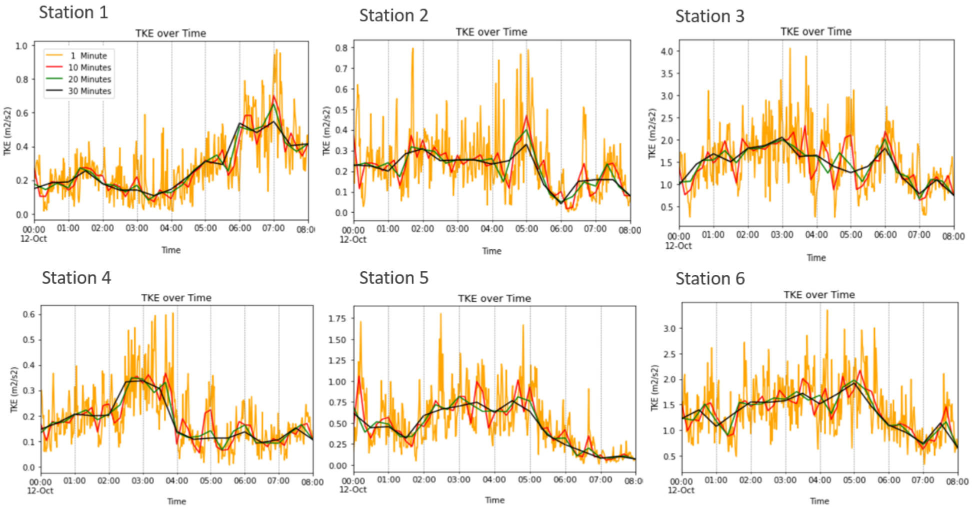


Figure 3.18: TKE averaged by time for October 12th, 2017

Thus,the data was collected during a time period that was determined to be the most steady during the night of October 12, 2017 can be found in Table 3.2.

Table 3.2: Initial wind direction and wind speed

Day	Time	Wind direction [degrees]	Wind Speed [m/s]
2017-10-12	03:00:00	220.3	3.0397854691104
2017-10-12	04:00:00	214.8	3.08448815377738
	Mean	217.55	3.06

Next, the friction velocity was calculated using the function 3.7, and for the k and ϵ variables, functions 3.8 and 3.9 respectively. These functions were previously introduced in the methodology and the results are shown in table 3.3.

Table 3.3: Initial Conditions

U [m/s]	U* [m/s]	k [m^2/s^2]	e [m^2/s^3]
3.06	0.29768	0.29538	0.00316934

3.3.3 Boundary conditions

For the inlet, a Velocity Inlet boundary condition was used where velocity was specified and pressure was calculated. For the outlet, a pressure outlet boundary condition was used where a static pressure was specified and velocity was calculated. To ensure parallel flow at the lateral and top boundaries, a symmetry boundary condition was used to enforce the velocity component normal to the boundary.

In these simulations, I made the assumption that the atmospheric boundary layer stratification is **neutral** and did not consider any temperature forcing. As a result, I used a characteristic neutral logarithmic profile at the inlet to compute the velocity:

$$U = \frac{u_*}{\kappa} \ln\left(\frac{z + z_0}{z_0}\right) \tag{3.7}$$

To initialize the xturbulence variables k and ϵ at the inlet boundary, I use the consistent formulation of the standard k-ϵ turbulence model. This involves solving the transport equations for k and ϵ, and specifying their values at the inlet based on the inlet flow conditions.

$$k = \frac{{u_*}^2}{\sqrt{C_\mu}} \tag{3.8}$$

$$\varepsilon = \frac{{u_*}^3}{\kappa(z + z_0)} \tag{3.9}$$

where U is the velocity, u_* is the friction velocity computed by the software, κ is the von Karman constant set to 0.41, the turbulence model constant C_μ is 0.09, z is the height above the surface equal to 20m , z_0 is the roughness length set to 0.3m corresponding to suburban terrain. In addition the roughness for vegetation correspond to the value of 0.03m according to the updated Davenport-Wieringa roughness classification [Blocken, 2015]. These roughness length values were used in the rough wall function based on z_0 used at the ground and vegetation patches.

It is worth mentioning that trees are not included in this study but the same low vegetation is assumed in the whole area. Lastly, two wall functions were used for solid items such as Buildings, Terrain, and Vegetation, to consider the Law of the Wall. This approach allows for a more accurate simulation of the turbulent boundary layer near these solid items. Specifically, the epsilonWallfunction and nutWallfunction were used in simulations of Buildings, while the nutkAtmRoughWallFunction and epsilonz0WallFunction was used in simulations of Terrain and Vegetation. Therefore, I set the roughness length for the boundary conditions of these walls. In more detail, the boundary condition settings used for LoD1.2 model are shown in Table 3.4.

Table 3.4: Boundary settings for Lod1.2 model in OpenFOAM

	U [m/s]	**p [m^2/s^2]**	**k [m2/s2]**	**e [m2/s3]**	**nut [m2/s]**
y0	atmBLInletVelocity	zeroGradient	atmBLInletK	atmBLInletEpsilon	calculated
y1	inletOutlet	fixedValue	uniformFixedValue	inletOutlet	calculated
x0	symmetry	symmetry	symmetry	symmetry	symmetry
x1	symmetry	symmetry	symmetry	symmetry	symmetry
Buildings	uniformFixedValue	zeroGradient	kqRWallFunction	epsilonWallFunction	nutkWallFunction
Terrain	uniformFixedValue	zeroGradient	kqRWallFunction	epsilonz0WallFunction	nutkAtmRoughWallFunction
Vegetation	uniformFixedValue	zeroGradient	kqRWallFunction	epsilonz0WallFunction	nutkAtmRoughWallFunction
Sky	symmetry	symmetry	symmetry	symmetry	symmetry

On the other hand, the LoD2.1 model implemented different boundary conditions. The mesh generated from the LoD1.2 model failed to run the simualtion when the model was rotated. Therefore, the model was not rotated and two inlet and two outlets boundaries were used for the wind instead. In more detail the Boundary settings for LoD2.1 model in OpenFOAM are show in Table 3.5.

Table 3.5: Boundary settings for Lod2.1 model in OpenFOAM

	U [m/s]	**p [m^2/s^2]**	**k [m2/s2]**	**e [m2/s3]**	**nut [m2/s]**
y0	atmBLInletVelocity	zeroGradient	atmBLInletK	atmBLInletEpsilon	calculated
y1	zeroGradient	fixedValue	fixedValue	zeroGradient	calculated
x0	atmBLInletVelocity	zeroGradient	atmBLInletK	atmBLInletEpsilon	calculated
x1	zeroGradient	fixedValue	fixedValue	zeroGradient	calculated
Buildings	uniformFixedValue	zeroGradient	kqRWallFunction	epsilonWallFunction	nutkWallFunction
Floor	uniformFixedValue	zeroGradient	kqRWallFunction	epsilonz0WallFunction	nutkAtmRoughWallFunction
Sky	symmetry	symmetry	symmetry	symmetry	symmetry

3.3.4 Solver type

The solver type pertains to the computational method employed to solve the system of equations. In this particular study, the solver used was simpleFoam, which is used for solving steady-state, incompressible, turbulent flows.

To solve for the variables U, k, and e, a smoothSolver was used, which is an iterative model that uses Gauss Seidel smoother to reach a solution within a specified tolerance. In this case, a tolerance of 1E-8 was used for the variables. Meanwhile, a GAMG solver with a tolerance of 1E-7 was employed for the pressure solver.

3.3.5 Scheme selection

The selection of scheme is crucial in a CFD simulation since it can significantly affect the solution's accuracy,stability and convergence. The governing equations in CFD are discretized and solved numerically on a computational grid (i.e Navier-Stokes equations). The numerical method's capacity to estimate the continuous equations determines the solution's accuracy, and its ability to transport information across the grid determines the solution's stability. The scheme selection relies on specific criteria, such as the required accuracy and stability, the geometry, the complexity of the flow, and the available computational resources.

3.3.5.1 *Gradient schemes*

For the gradient schemes the **cellLimited Gauss Linear 1** scheme is used.

3.3.5.2 *Divergence schemes*

For the divergence schemes the **Bounded Gauss limitedLinearV 1**, a mixed first-second order scheme was used to solve the advection term of the Navier-Stokes equations for incompressible flows.

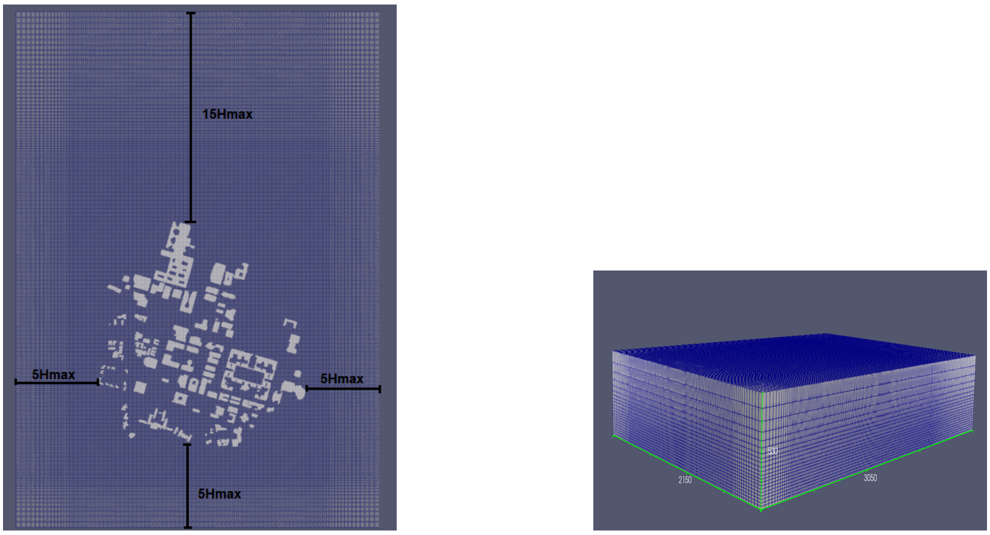
The **bounded Gauss upwind** scheme is used to simulate the transport of scalar values, k and ϵ.

3.3.6 Set up of the computational model

The computational model follows the recommended best practices guidelines by Blocken [2015] and Franke et al. [2004] in order to guarantee precise and reliable CFD simulations.

3.3.6.1 *Computational domain dimensions*

According to guidelines, the inlet, lateral, and top boundaries are set at a distance of at least 5 times the height of the highest building, the Hoover Tower (Hmax = 75m), while the outflow boundary is set 15 times Hmax away from the urban building model to allow the wake flow behind the buildings to fully develop (Figure 3.19a). Thus the domain extends for approximately 2 x 3 km^2 in the horizontal direction and 530m in the vertical direction (Figure 3.19b).



(a) Domain design rules

(b) Domain size

Figure 3.19: Numerical domain

3.3.6.2 Computational mesh

In order to model the airflow in a specific area, the space is divided into smaller sections using a mesh generator called snappyHexMesh. An example of this type of computational mesh is shown in Figure 3.20, which is designed for studying the airflow around the Engineering QUad. The figure illustrates that the density of cells in the mesh increases closer to the ground and to the area of interest for in x and y direction. Three refinementBoxes were cteated, one around the Engineering Quad, one around the all buildings and one at a larger area to ensure a smooth transition between the large background cells and the small cells around the area of interest.

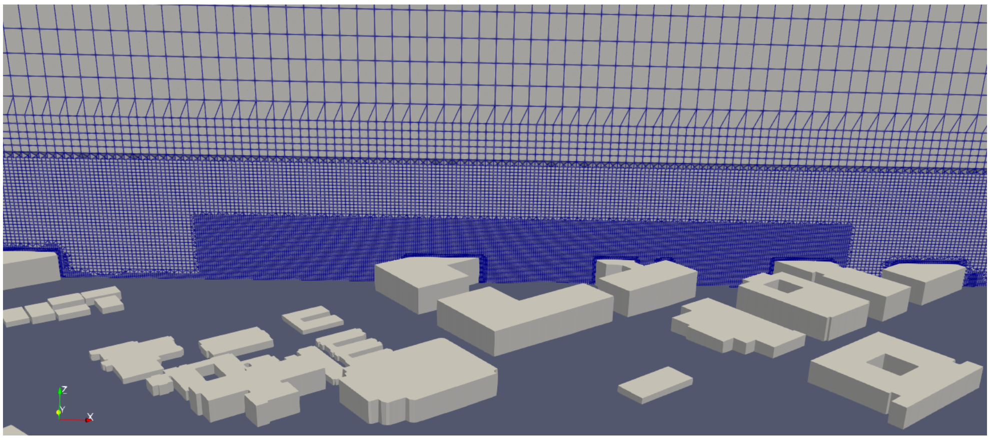


Figure 3.20: Cross-section view of mesh showing the three cell sizes and the extra refinement for the terrain close to area of interest

The resulting meshes contain mostly hexahedras and some tetrahedras, with approximately 12, 23 and 32 million cells for coarse, nominal and fine mesh accordingly. The cell density increases close to buildings and semantic surfaces (Figure 3.21).

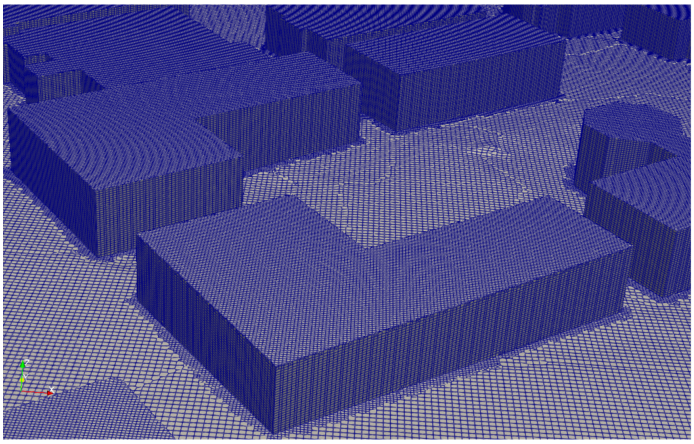


Figure 3.21: LoD1.2 mesh snapshot

3.4 TOOLS

For the purposes of the current thesis a combination of software, programming languages and plugins & libraries used or planed to be used. More details can be shown in Table 3.6.

Table 3.6: Tools & Technology

Technology	Purpose
Software	
QGIS	Pre-processing of data
Cloud Compare	Point cloud processing
Meshlab	Visualization
OpenFOAM	CFD solver
Paraview	Post processing & visualization
Programming languages	
C++	CFD geometry preparation, OpenFOAM implementations
Python	Scripting
Plugins\Libraries	
QuickOSM	Extracting data from OpenStreetMap
LAStools	Lidar processing
CGAL	Geometry processing

4 RESULTS AND DISCUSSION

This chapter focuses on the results obtained from the simulation, with a particular emphasis on the convergence time. A comparison of the convergence time for the coarse nominal and fine mesh between 5002 to 8002 iterations will also be presented. Results from selected stations will be discussed and the mean value will be used for further analysis. Additionally, a comparison will be made between the results obtained from the LoD1.2 and LoD2.1 models with the measurement results.

4.1 SIMULATION TIME

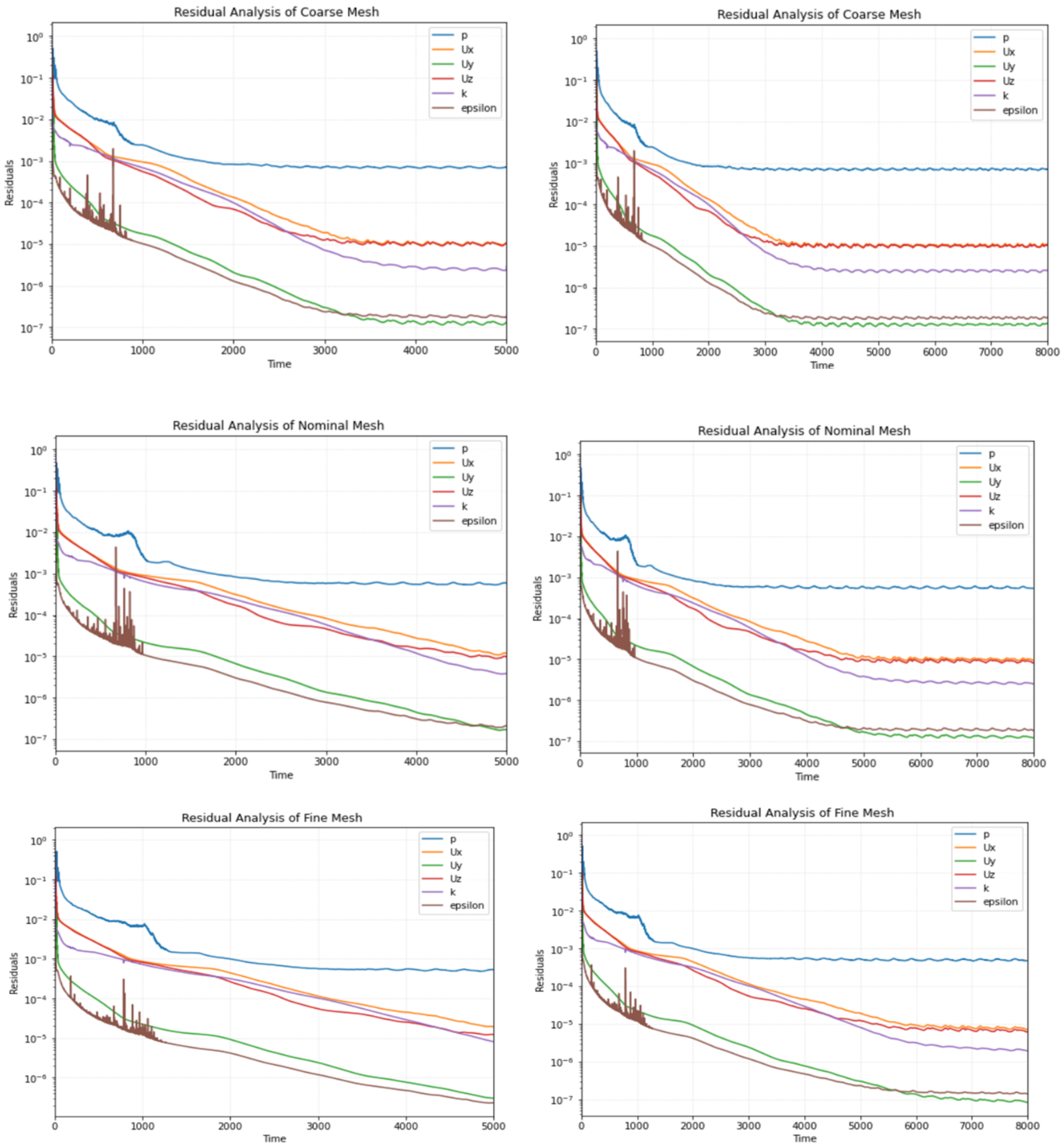
To generate three different meshes, namely coarse, nominal, and fine, with 12, 23, and 32 million cells, respectively, it took approximately 20, 40 minutes and 1 hour. However, it took around 3-4 hours to generate the coarse mesh if it was not run in parallel. Therefore, the hierarchical method of decomposition was used to divide the domain into **32 subdomains**, with each subdomain refined in the x, y, and z directions using a coefficient of 0.001. Specifically, the domain is divided into 4 subdomains in the x-direction, 4 subdomains in the y-direction, and 2 subdomains in the z-direction. In this way, the execution time reduced to 20-25 minutes. The execution time for simulation increased as expected with the increase in mesh resolution. To achieve convergence in all three meshes, I increased the number of iterations from 5002 to 8002, which almost doubled the execution time. Further details about the execution time can be found in Table 4.1.

Table 4.1: Execution time for simulation running in parallel with **32 subdomains**

Iterations	Coarse	Nominal	Fine
5002	5h	8h	13h
8002	10h	14h	23h

4.2 MESH CONVERGENCE

In this study, I started with a coarse mesh to ensure that the residuals converge. I ran the simulation and found that all fields converged with 5002 iterations. However, when I progressed to the nominal and fine mesh, I found that the fields did not converge. I continued the procedure with more iterations and ended up with a total of 8002 iterations to ensure that the case fully converged (Figure 4.1).

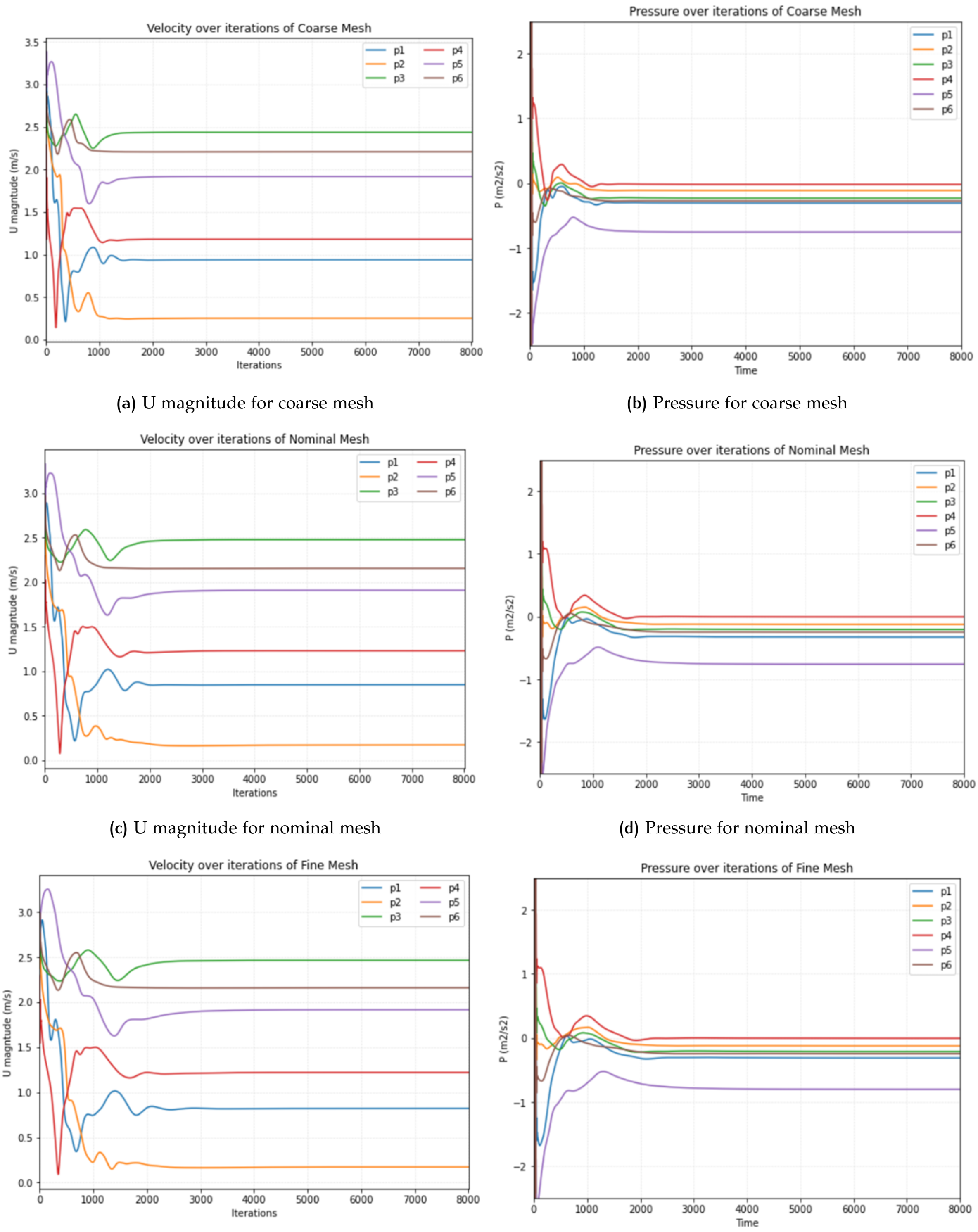


(a) Case did not converge with 5002 iterations

(b) Residuals convergence with 8002 iterations

Figure 4.1: Residuals Convergence after Increased Iterations

After analyzing the residual plots for each variable, I noticed small fluctuations that suggest the solution may not be fully converged. However, In cases where the fluctuations are small and the residuals have decreased to a low level, it may be reasonable to assume that the solution is converged. I then proceeded to conduct a residual convergence test by plotting U magnitude and pressure values throughout the iterations to ensure that their values remained stable with no significant changes (Figure 4.2). After 2000 iterations, there is no further change in the U magnitude and pressure values across all three meshes.



(a) U magnitude for coarse mesh

(b) Pressure for coarse mesh

(c) U magnitude for nominal mesh

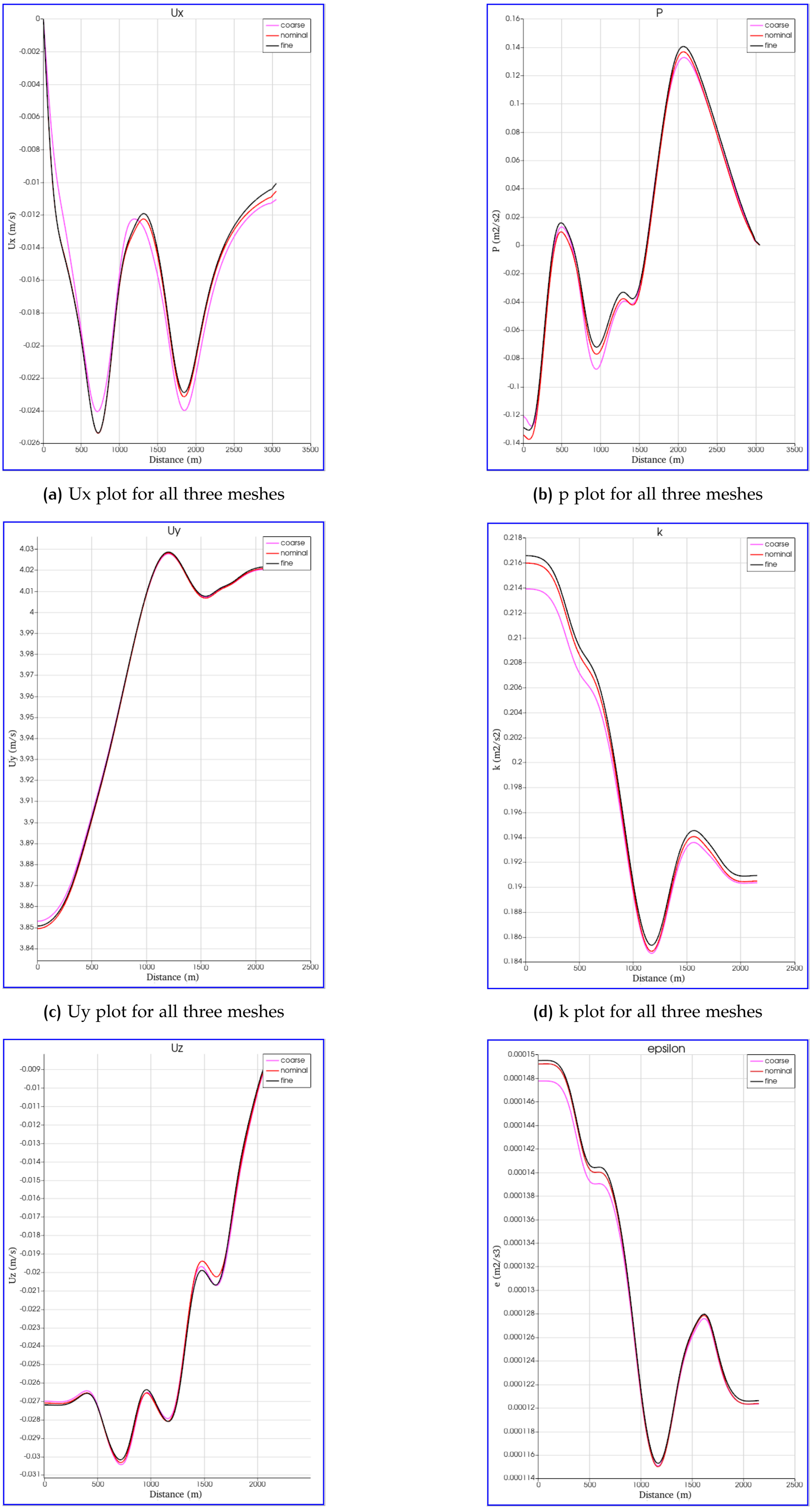
(d) Pressure for nominal mesh

(e) U magnitude for fine mesh

(f) Pressure for fine mesh

Figure 4.2: Comparing U magnitude and pressure values over iterations

I also created field plots over a line in Paraview to compare the three meshes, and found that the nominal mesh was closer to the fine mesh, while the differences between the coarse and nominal meshes were minimal (Figure 4.3).



(a) Ux plot for all three meshes

(b) p plot for all three meshes

(c) Uy plot for all three meshes

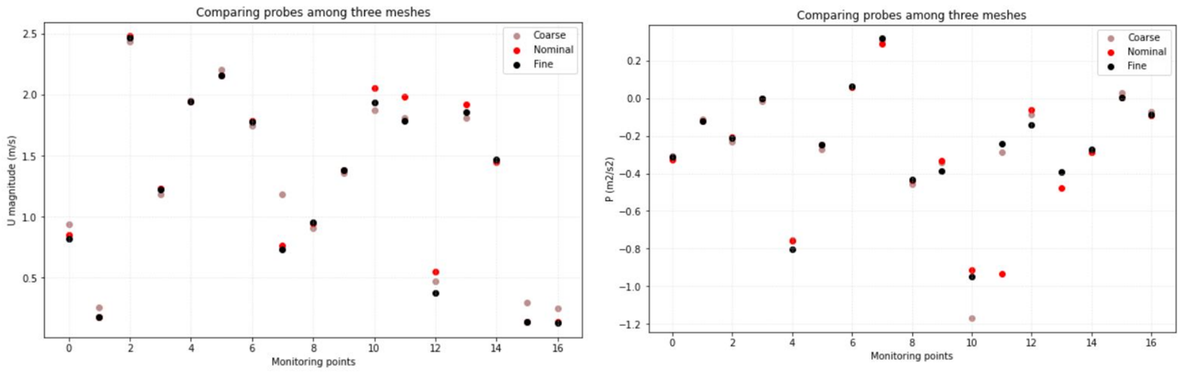
(d) k plot for all three meshes

(e) Uz plot for all three meshes

(f) e plot for all three meshes

Figure 4.3: Mesh independence plots

Lastly, I tested the U magnitude and pressure for all three meshes in the last iteration and observed that the nominal and fine meshes were very close (Figure 4.4). Based on these results, I successfully passed the mesh convergence test and decided to continue with the nominal mesh to save time and resources.



(a) U magnitude among all meshes

(b) Pressure among all meshes

Figure 4.4: Comparison of Velocity Magnitude and Pressure for Three Different Meshes: Final Iteration

4.3 RESIDUALS CONVERGENCE OF CASE STUDY

The plot of residuals in Figure 4.5 shows that initially the residuals varied noticeably, but after 7000 iterations, all lines tended to become parallel to the x-axis with a slope tending to zero. The pressure values had higher residuals than the velocity (Uy) values, which is typical for an urban environment. Achieving small residuals in such complex geometries can be difficult, although in our case study, convergence was achieved at around 10^{-5} for Ux and Uz, 10^{-6} for k, and 10^{-7} for k and Uy.

Interestingly, despite using the same nominal mesh as for the mesh convergence study, the residuals in the current case did not show fluctuations. This could be due to different boundary conditions and numerical schemes used in this case study, which resulted in better convergence. It is also possible that the mesh used for mesh convergence may not be suitable for accurately capturing the flow behavior in different wind directions and speeds, which were used as examples and not actual values from the case study.

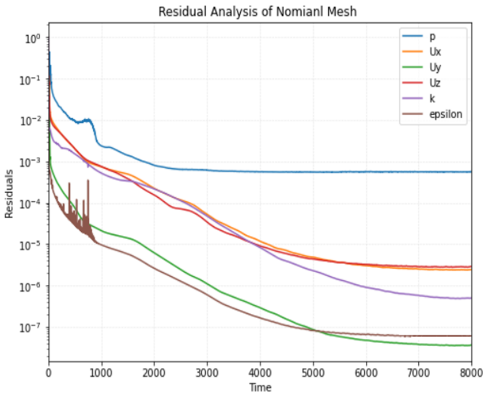


Figure 4.5: Convergence Analysis of Residuals: Case Study Achieves Convergence After 7000 iterations

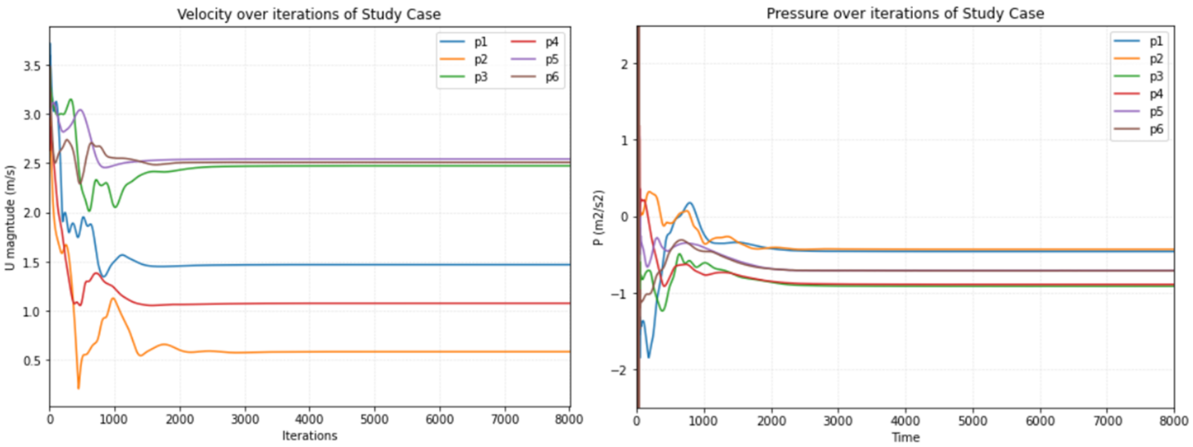


Figure 4.6: Convergence analysis of U magnitude and pressure values over iterations in Case Study: stable values achieved after 2500 iterations

4.4 MEASUREMNTS

For the purpose of comparing simulation results with actual measurements, wind velocity over six different stations was analyzed (Figure 4.7). The analysis focused on a specific time period with the most stable conditions, which was from 3:00 to 4:00 on October 12th, 2017. From the plots of wind velocity for each station, the mean velocity averaged over the one-hour and 45 minutes period was calculated from Figure 4.8. I chose to analyze the 45-minute interval because the plotted velocity appeared to be more steady during this period as compared to the one-hour interval. In particular, during the last 5 minutes of some stations, such as Station 3 and 6, the velocity showed a rapid decrease. These mean values was considered as the wind velocity of measurements from the stations and was used for further comparison with the simulation results.

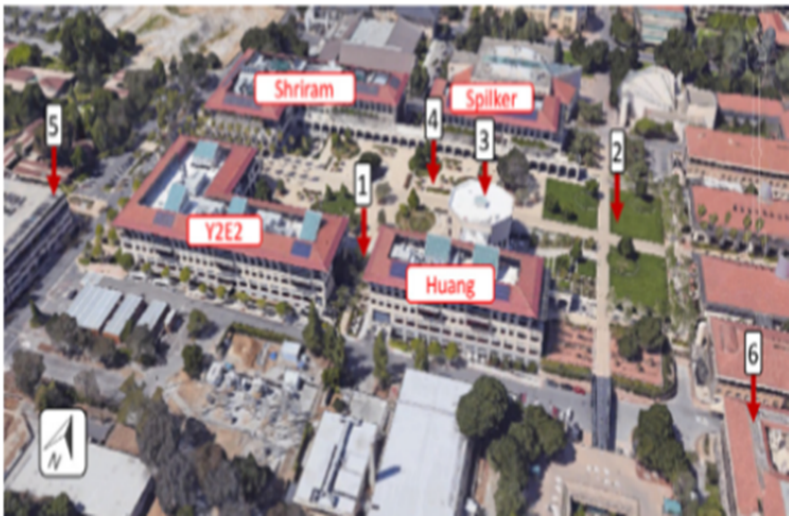
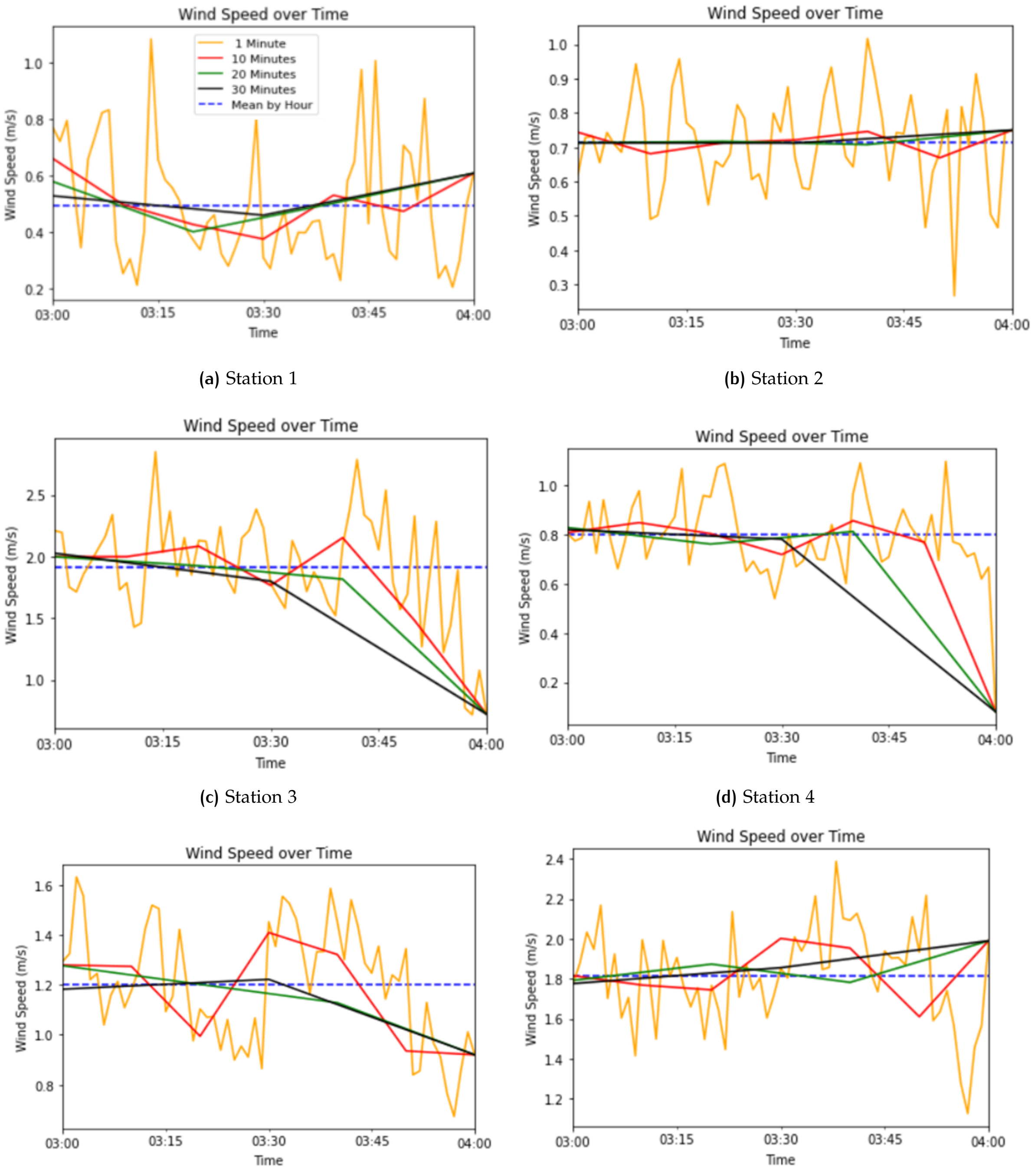


Figure 4.7: Location of Stations

Upon analyzing the data, small differences in the mean values of velocity were observed between measurements taken over 1 hour interval and those taken over 45-minute interval, particularly at stations 3 and 6. The difference in mean velocity at station 3 was observed to be 0.1 m/s between the two time intervals, with a higher mean velocity observed over the 45-minute interval. Similarly, station 6 showed a difference of 0.12 m/s in mean velocity, with a higher mean velocity observed over the 45-minute interval. Further details about the observed differences in mean velocity between the 45-minute and one-hour intervals can be found in the table 4.2.

Table 4.2: Wind velocity measurements from the stations

	Station 1	Station 2	Station 3	Station 4	Station 5	Station 6
Umean 1hour [m/s]	0.49	0.71	1.91	0.80	1.20	1.82
Umean 45min [m/s]	0.50	0.72	2.01	0.80	1.25	1.94



(a) Station 1

(b) Station 2

(c) Station 3

(d) Station 4

(e) Station 5

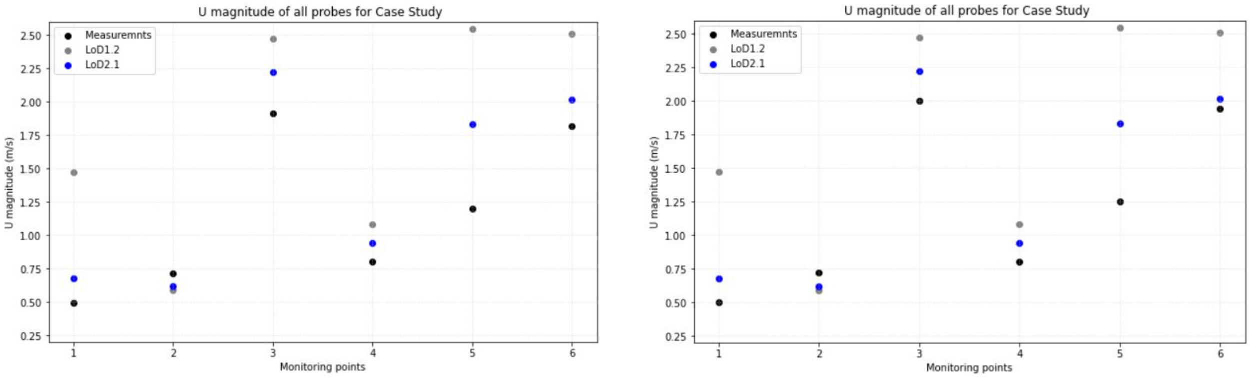
(f) Station 6

Figure 4.8: Wind speed averaged by time for October 12th at the selected time period

4.5 COMPARISON BETWEEN MODELS AND MEASUREMENTS

To evaluate the performance of the two models, error plots were created, and RMSE and MAPE values were calculated as statistical analysis metrics. The RMSE (Root Mean Squared Error) measures the differences between the predicted and actual values, indicating how well the model fits the data. The MAPE (Mean Absolute Percentage Error), on the other hand, measures the accuracy of the model in forecasting by indicating the average percentage difference between predicted and actual values. Using these metrics, I can compare the two models and determine which one has a better fit to the measurements.

The plot reveals that the LoD2.1 simulation results are more closely aligned with the actual measurements, exhibiting the greatest difference in Station 3 and 5. In contrast, the LoD1.2 model demonstrates higher differences with the measurements, particularly in Stations 1, 5, and 6. This phenomenon can be attributed to the fact that these stations experienced the most fluctuations in wind velocity during the time interval selected for the analysis. Station 2, on the other hand, demonstrated the most accurate results, with both models approximating the actual value very closely (Figure 4.9).



(a) U magnitude averaged over 1 hour

(b) U magnitude averaged over 45 minutes

Figure 4.9: Comparison of LoD1.2 and LoD2.1 Simulations with 1-Hour and 45-Minute Average Measurements

The error plots were created to further analyze the differences between the LoD1.2 and LoD2.1 models and the measurements. The results showed that for the LoD1.2, two values at stations 1 and 5 exceeded 60% of the error. In contrast, the LoD2.1 model had most values within a 20% error range, with station 5 being slightly higher at around 40%. The LoD2.1model performed better overall, with RMSE and MAPE values of 0.31 m/s and 51.77% correspondingly. For LoD1.2 these values correspond to 0.78 m/s and 84.94% proving the LoD1.2 is more prone to errors (Figure 4.10).

Error plots were created for both models using 45-minute mean measurements. Comparing these plots with the previous ones, it can be concluded that the performance with 45-minute mean measurements is slightly better. Specifically, the RMSE values for both models decreased by approximately 0.4 m/s, while the MAPE values decreased by 1-2 percent. The improvement is not significant as the difference between the mean values of 1 hour and 45 minutes was not very large. This explains why the results are only slightly better, as the measurements were closer to the 45-minute mean (Figure 4.11).

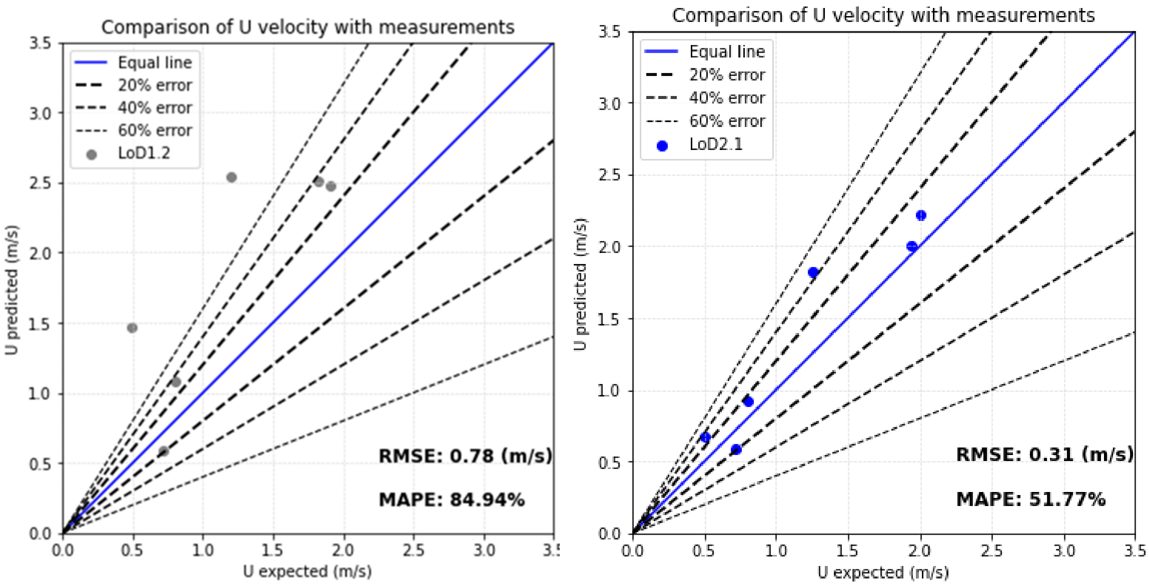


Figure 4.10: Error Analysis of LoD1.2 and LoD2.1 Models Using Hourly Mean

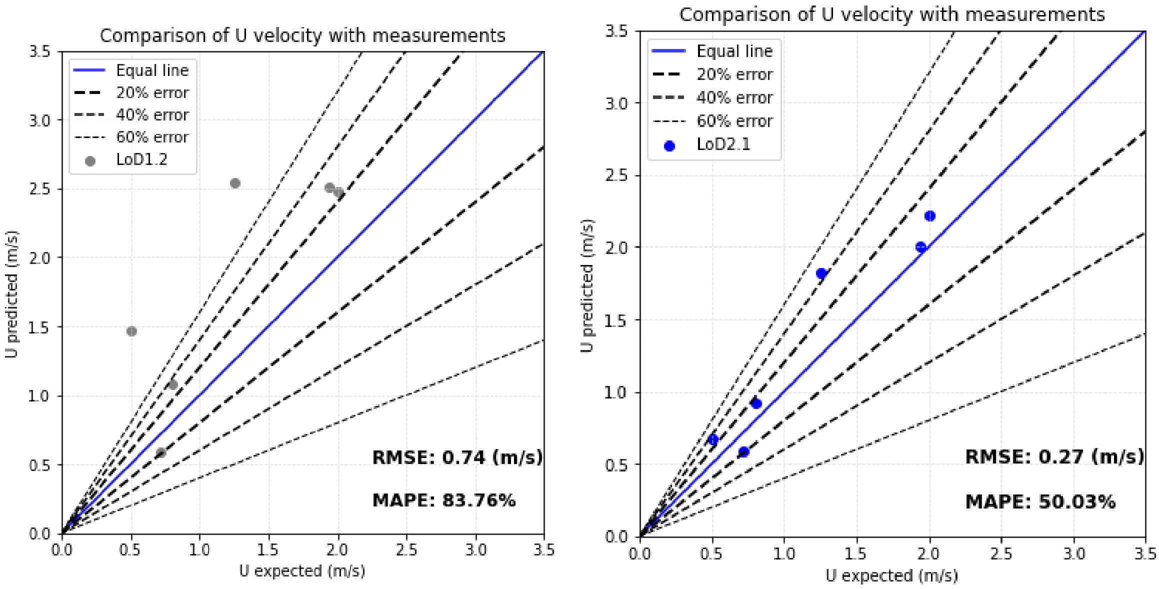
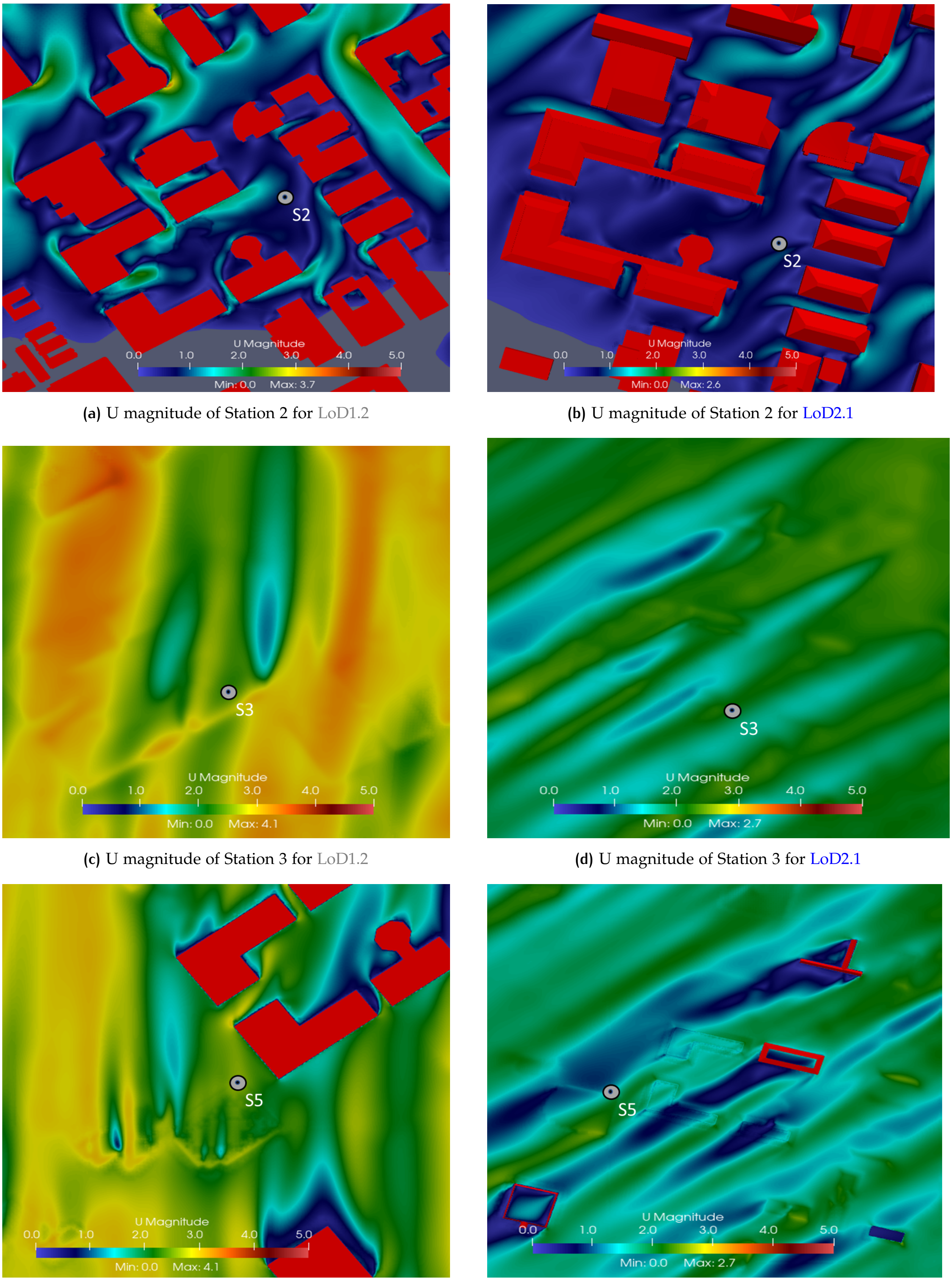


Figure 4.11: Error Analysis of LoD1.2 and LoD2.1 Models Using 45 minutes Mean

To further analyze the results of the wind simulation, contour plots of velocity magnitude and turbulent kinetic energy (TKE) were created for both models at the height of each station. Based on the plots, it is observed that the velocity magnitude undergoes a greater change at higher elevations between the two models, particularly in stations 3 and 5 at heights of 50 and 43.6, respectively. Also, The LoD1.2 model showed higher values (Figure 4.12). The TKE plots revealed that both models had low TKE values at Station 2, where the simulation results were more accurate. Conversely, Station 3 and 5 had higher TKE values, which may explain the reason for the larger differences observed between the models and measurements. Interestingly,LoD1.2 had higher TKE values than LoD2.1 at both Station 3 and 5, which may further explain the greater differences between the models and measurements (Figure 4.13).



(a) U magnitude of Station 2 for LoD1.2

(b) U magnitude of Station 2 for LoD2.1

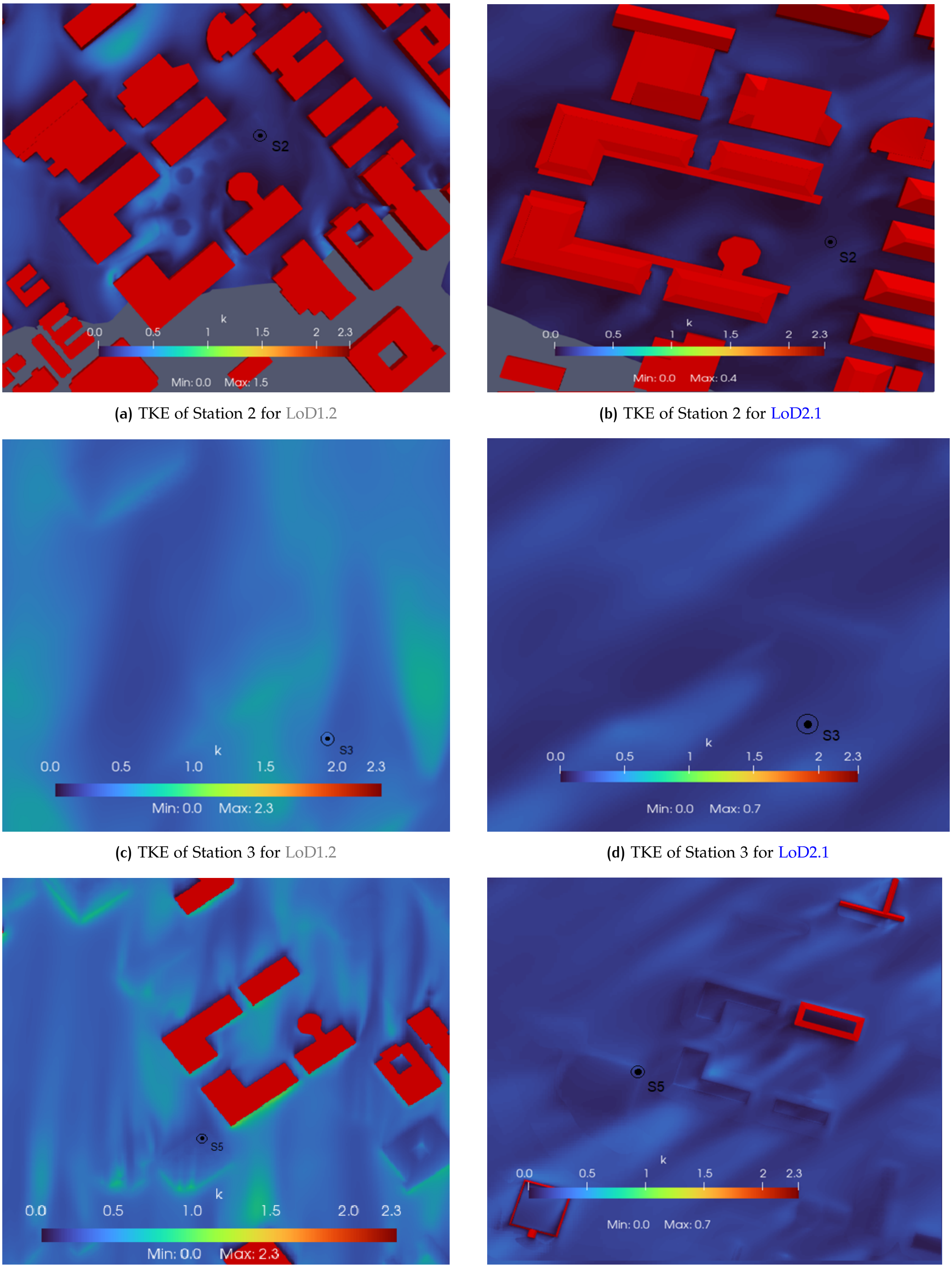
(c) U magnitude of Station 3 for LoD1.2

(d) U magnitude of Station 3 for LoD2.1

(e) U magnitude of Station 5 for LoD1.2

(f) U magnitude of Station 5 for LoD2.1

Figure 4.12: Contour plots of U magnitude at the height of each station for both models



(a) TKE of Station 2 for LoD1.2

(b) TKE of Station 2 for LoD2.1

(c) TKE of Station 3 for LoD1.2

(d) TKE of Station 3 for LoD2.1

(e) TKE of Station 5 for LoD1.2

(f) TKE of Station 5 for LoD2.1

Figure 4.13: Contour plots of Turbulent Kinetic Energy (TKE) at the height of each station for both models

5 CONCLUSIONS

This chapter provides a brief summary of the key results and recommendations derived from the completed study. The initial part of the section focuses on addressing the research questions posed in introduction using the results presented in the previous chapter. Subsequently, it highlights the constraints of the current study, and finally, concludes by presenting suggestions for future research work.

5.1 ADDRESSING THE RESEARCH QUESTIONS

The main research question of the current thesis is:

- *What is the impact of different geometry LoDs for the wind around an urban environment?*

  The impact of different geometry Levels of Detail (LoDs) on wind simulation accuracy in urban environments is an important consideration. The current analysis limited to a period of steady velocity, during the night of the 12th of October, showed that LoD2.1 performed better than LoD1.2 for those specific conditions. Further verification should be included with other periods of time and conditions to generalize the superiority of higher detailed models. In more detail, the LoD2.1 model, which includes more complex and detailed features such as multi-pitched roofs and captures the full 3D geometry of the urban environment, showed better performance in simulating wind velocities compared to the LoD1.2 model during the observed time period. These results suggest that including more precise and accurate geometry can lead to more accurate wind simulation results. However, it is important to note that the results obtained may not necessarily hold for other time periods or contexts, and further research is required to validate these results for other scenarios.

When considering the core research question, several sub questions arise automatically:

- *What are the needed steps to automatically reconstruct a 3D city model for the area of interest?*

  Geometry preparation is a crucial but laborious task in the computational fluid dynamics (CFD) simulation workflow. To automate this process, 2D geographical data and point cloud-based elevation data are combined to reconstruct terrain, buildings, and other surface layers such as vegetation. The first step involves selecting the area of interest and extracting the footprints using software like QGIS and the QuickOSM plugin, which can be exported into the GeoJSON format. This procedure is repeated for other features like vegetation, water, and roads. Next, point cloud data is downloaded from a source like the USGS site and pre-processed to extract the relevant data for buildings and terrain. It is important to ensure that the point cloud and GeoJSON files are in the same coordinate system. These datasets are then fed into the City4CFD software, which allows for the introduction of a point of interest and a buffer layer to cover the desired area for reconstruction. After running City4CFD, a 3D model with LoD1.2 geometry is generated, which aims to be error-free and ready for further analysis such as input in CFD simulations.

- *How large can be the differences introduced by geometry discrepancies, meaning the different LoDs for the buildings?*

  The results of my study suggest that the choice of LoD can impact wind simulation outcomes. Specifically, in the observed time period and under the prevailing conditions, I found that the LoD2.1 model outperformed the LoD1.2 model in capturing wind patterns around an urban environment. However, it is important to note that my results are limited to this specific time period and set of conditions, and may not be applicable in other scenarios. Specifically, in my research the differences introduced by geometry discrepancies between LoD1.2 and LoD2.1 was as high as 30%. The comparison between the two models showed that the LoD2.1 model had most results falling within a 20% error range, while the LoD1.2 model had most results within a 40% error range, with two of them exceeding 60% error. It is important to note that the degree of discrepancy introduced by different LoDs may vary depending on the specific urban environment being studied.

- *Is it possible a higher LoD geometry better predict real-world measurements?*

  Based on current research that was limited to a specific day and time with steady velocity, showed that a higher LoD geometry, such as LoD2.1, better predicted real-world measurements of wind patterns in urban environment. The study found that the more detailed LoD2.1 model, which includes more complex and detailed features of the urban environment, produced simulation results that were closer to actual measurements compared to the less detailed LoD1.2 model. This indicates that the inclusion of more accurate and detailed geometry in wind simulation models could lead to more accurate predictions of wind behavior around buildings and other structures in urban areas. However, while a higher LoD geometry can improve the accuracy of wind simulation results in urban environments, there is a trade-off between the level of detail and the computational time required to reconstruct the 3D model and perform simulations. Specifically, in my case, the automatically reconstructed LoD 1.2 model took approximately one day to complete, while the manually reconstructed LoD 2.1 model that was given to me took up to two months. Overall, the choice of LoD geometry in wind simulation models in urban environments should consider the desired level of accuracy, the available computational resources, the characteristics of the simulation model and the available time. A balance between these factors can help to optimize the simulation results while minimizing computational costs.

5.2 LIMITATIONS AND RECOMMENDATIONS

5.2.1 Limitations introduced from the input models

The CFD simulation models used in the current study did not include trees. However, excluding trees from the wind analysis can have several impacts on the results.

Trees play a vital role in shaping wind patterns, and their exclusion can alter wind speed and direction. Trees can act as barriers or deflectors, causing turbulence and creating vortices downstream. By excluding trees from the simulation, the drag force and turbulence effects may be underestimated, leading to unreliable results. Therefore, it is recommended to include trees in the wind analysis model to ensure a more accurate and comprehensive analysis.

Additionally, heat sources and atmospheric conditions are excluded from our models. This exclusion can limit the accuracy of the simulation results. Buildings, vehicles, and other sources of heat can create temperature variations in different parts of the city, and including information about these sources can help to create a more accurate and comprehensive model. Similarly, including information about atmospheric conditions such as temperature and humidity can improve the simulation results.

However, it is crucial to maintain a balance between the completeness of the model and the simulation execution time. The inclusion of more factors in the model can significantly increase the execution time of the simulation. Therefore, it is necessary to carefully consider which factors should be included in the model to strike a balance between model completeness and simulation time.

5.2.2 Limitations introduced from CFD simulation

The study has some limitations due to the assumptions and simplifications made to simplify the problem. The simulation involved solving steady-state incompressible flow, which assumes constant flow conditions over time, and may not accurately represent real-world condition. Unsteady flow models are capable of capturing time-dependent variations in the flow field and can provide more accurate results in city-level simulations compared to steady-state models. In a city-level simulation, the flow field is complex and highly variable, with flow patterns quickly changing due to several factors such us buildings, and other obstacles like trees that impact the flow. However, unstable models are more computationally expensive, demand more computing power, and have longer simulation time.

Additionally, meshing is a crucial factor that can impact the accuracy of simulation results. The quality of the mesh can introduce errors and inaccuracies in the solution. In our study, automatic mesh creation was employed, but manually creating the mesh can provide greater control over the mesh quality, especially in areas where the flow is expected to be more complex or where accuracy is crucial. Manual meshing may also be more suitable for complex geometries with non-standard shapes that cannot be meshed automatically precisely. However, manual meshing, on the other hand, can be time-consuming and difficult to generate, particularly for complex geometry. The decision to use manual or automatic meshing depends on the specific needs of the simulation and the available resources. Manual meshing might be a better choice for simulations with complicated geometries that need a high degree of accuracy.

The wind speed and direction used in our simulation were obtained from the nearest weather station, which may not accurately represent the actual conditions in a complex urban environment. To enhance the accuracy of the simulation, a better approach would be to estimate the wind speed and direction at the inlet boundary based on the simulation results. To accomplish this, the simulation could be run with various wind speeds and directions as inputs, and the results could be compared to measurements from sensors within the area of interest to identify the input values that best match the observed values.

In the presented thesis, the decision was made to rotate the geometry to align the wind direction with the y-axis in order to eliminate the numerical diffusion effect. When the wind direction is not aligned with a coordinate axis, the numerical approximations required to discretize the flow equations in multiple directions can increase numerical diffusion. This is because the approximations of derivatives in each direction can be influenced by the gradients in the other directions. By rotating the domain to align with the wind direction, the equations can be discretized in the direction of the flow, which can reduce numerical diffusion. However, it should be noted that aligning the wind direction with the coordinate axes can also introduce numerical errors due to coordinate transformation. Therefore,it is essential to balance the advantages and disadvantages based on the simulation setup and objectives.

5.3 FUTURE WORK

A potential direction for future work in the current thesis could be to enhance the accuracy of the model by integrating the dispersion of pollution and the influence of trees. Specifically, the dispersion of pollution is particularly important because it can affect the air quality and human health in the area, and the behavior of pollutants can be complex and vary depending on environmental factors. By including this in the model, we gain a better understanding of how pollutants move and behave in the environment and it would be possible to evaluate the risks associated with different pollutant sources and their impact on the environment.

Furthermore, the presence of trees can have a notable impact on wind flow patterns, which can influence pollutant dispersion and the micro-climate of the area. Incorporating the impact of trees into the existing model would enable to simulate the real-world scenario more effectively, thereby providing more precise results. Finally, incorporating the effect of trees in the model would enable a better understanding of their impact on the environment and help to evaluate their potential benefits as a means of mitigating pollution and climate change.

Overall, by integrating the dispersion of pollution and the influence of trees into the existing model, it would be possible to improve the accuracy of the simulation and gain a better understanding of the real-world conditions being simulated. This could have important implications for policy decisions related to air quality and environmental management.

A REPRODUCIBILITY SELF-ASSESSMENT

A.1 MARKS FOR EACH OF THE CRITERIA

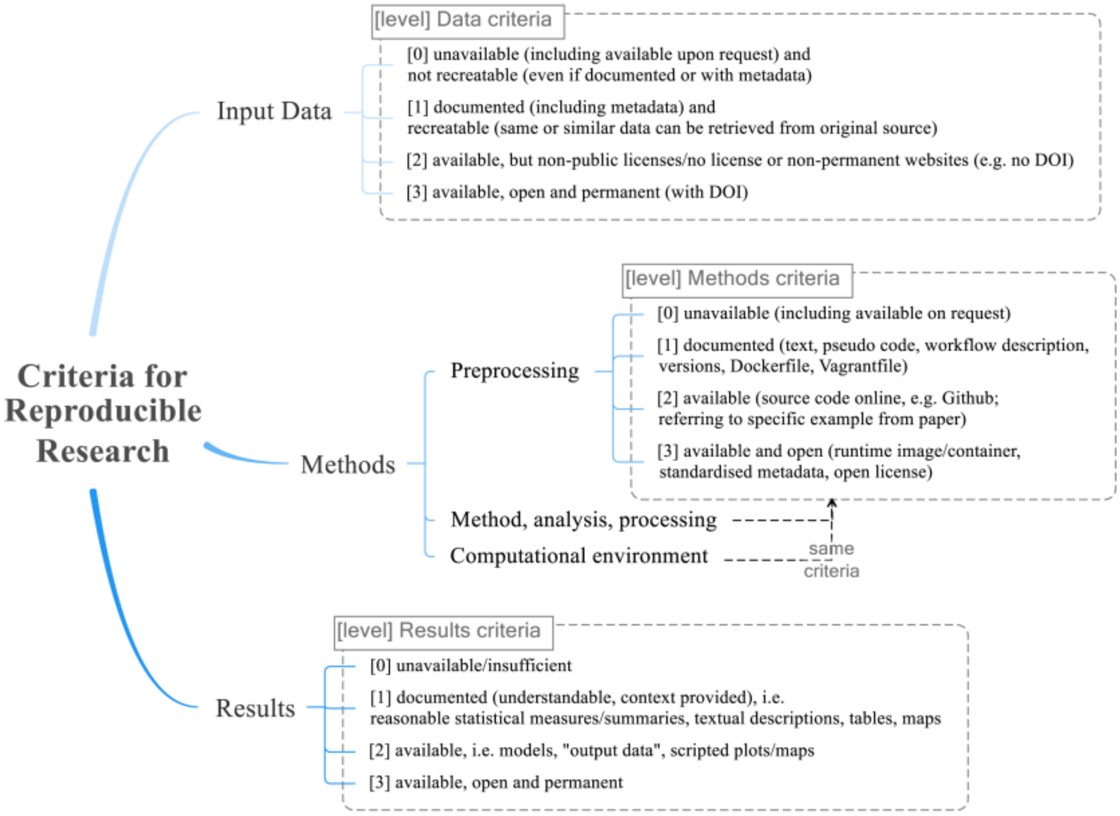

Criteria for Reproducible Research
Input Data
[level] Data criteria
[0] unavailable (including available upon request) and not recreatable (even if documented or with metadata)
[1] documented (including metadata) and recreatable (same or similar data can be retrieved from original source)
[2] available, but non-public licenses/no license or non-permanent websites (e.g. no DOI)
[3] available, open and permanent (with DOI)
Methods
Preprocessing
[level] Methods criteria
[0] unavailable (including available on request)
[1] documented (text, pseudo code, workflow description, versions, Dockerfile, Vagrantfile)
[2] available (source code online, e.g. Github; referring to specific example from paper)
[3] available and open (runtime image/container, standardised metadata, open license)
Method, analysis, processing
Computational environment
same criteria
Results
[level] Results criteria
[0] unavailable/insufficient
[1] documented (understandable, context provided), i.e. reasonable statistical measures/summaries, textual descriptions, tables, maps
[2] available, i.e. models, "output data", scripted plots/maps
[3] available, open and permanent

Figure A.1: Reproducibility criteria to be assessed.

Criteria	Grade	Justification
Input data	2	The vector data of footprints and vegetation as well as the point cloud are available on Github of the project but not DOI. Also, the data for measurements is not open data
Preprocessing	3	Available and open on the Github page of the project
Methods	3	Available and open on the Github page
Computational environment	3	Open source software
Results	1	The results take up 1TB of space and they are not able to be hosted in Github; although they can easily be reproducible based on the report

A.2 SELF-REFLECTION

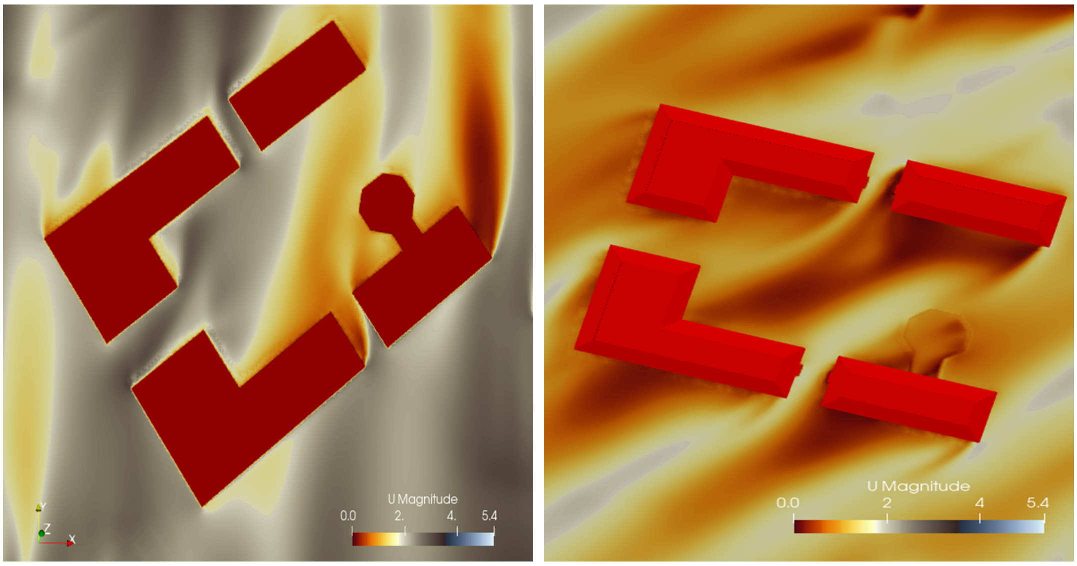
This thesis's self-assessment is based on the criteria explained in Section A.1. Concerning input data, vector data of footprints and vegetation and point clouds was acquired by the author and are available as open data in https://github.com/pkountouri/Impacts-of-LoDs-in-wind-engineering-/tree/main/City4CFD. However, additional data, such as measurements from sensors and the weather station, were obtained from Stanford University and are not available to the public.

Chapter 3 explains the pre-processing steps required to automatically reconstruct the 3D model in Lod1.2. The dataset used for the reconstruction is openly available, and different users can reproduce the steps to reconstruct the 3D model of their interest. The CFD simulation of both LoD1.2 and LoD2.1 models in OpenFOAM can be found on the Github page, including the necessary folders to run the simulations https://github.com/pkountouri/Impacts-of-LoDs-in-wind-engineering-/tree/main/OpenFOAM.

For the implementation of the thesis several open-source software, such as QGIS, Cloud Compare, Meshlab, OpenFOAM, and Paraview, as well as different programming languages, such as C++, for CFD geometry preparation and OpenFOAM implementations, and Python for creating result plots.

Overall, the thesis's methodology can be applied in any urban area where computational wind engineering is used to analyze wind flow patterns and design strategies for improving ventilation and comfort. By comparing the outcomes of different LoD models with on-site wind measurements, the most appropriate model can be chosen for CFD simulations, providing more accurate and reliable results. This approach can help urban planners and designers to make informed decisions regarding the design and layout of buildings and outdoor spaces to improve pedestrian comfort, reduce energy consumption, and enhance urban livability.

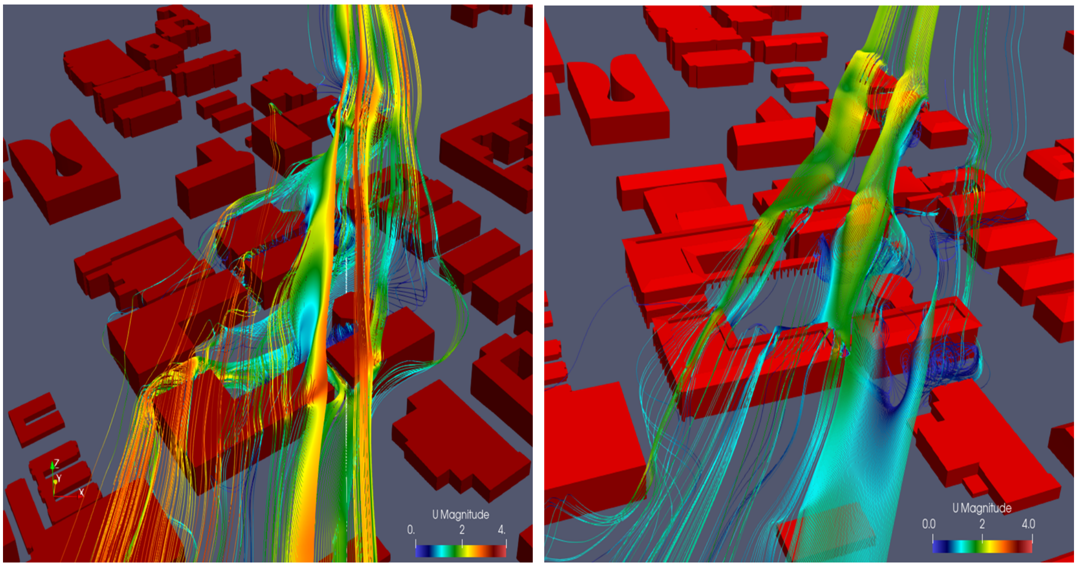
B ADDITIONAL RESULTS



(a) Slice of LoD1.2 at 25m

(b) Slice of LoD2.1 model at 25m

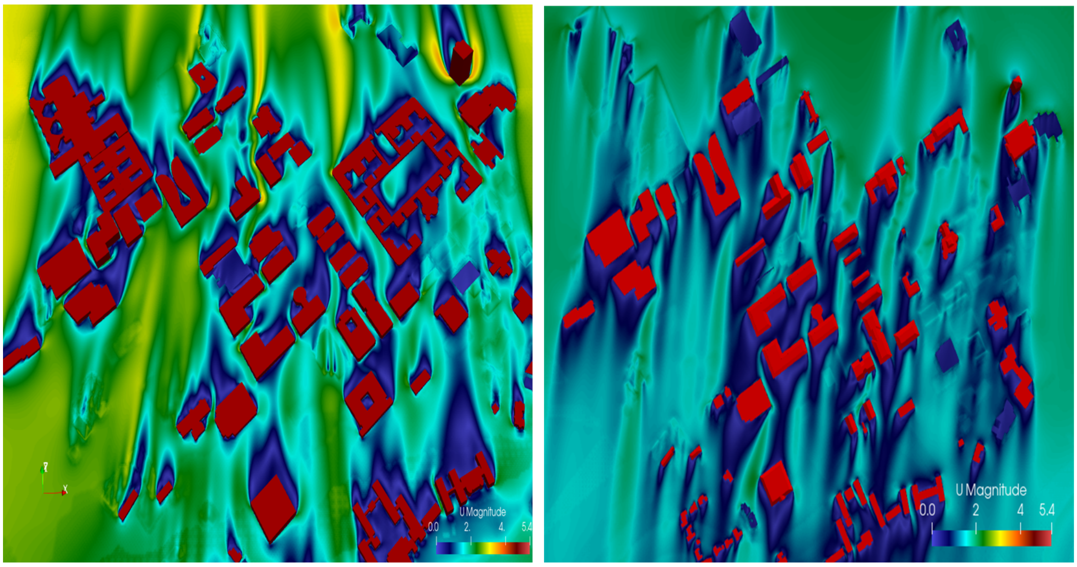
Figure B.1: U magnitude at the roof level



(a) Stream tracer of LoD1.2 model at 25m

(b) Stream tracer of LoD2.1 model at 25m

Figure B.2: Stream tracer of U magnitude at the roof level



(a) Slice of LoD1.2 at 10m

(b) Slice of LoD2.1 model at 10m

Figure B.3: U magnitude at 10m

BIBLIOGRAPHY

Akhatova, A., Kassymov, A. B., Kazmaganbetova, M., and Rojas-Solórzano, L. R. (2016). Cfd simulation of the dispersion of exhaust gases in a traffic-loaded street of astana, kazakhstan. *Journal of Urban and Environmental Engineering*, 9:158–166.

Biljecki, F., Ledoux, H., Stoter, J., and Zhao, J. (2014). Formalisation of the level of detail in 3d city modelling. *Computers, Environment and Urban Systems*, 48:1–15.

Biljecki, F., Ledoux, H., and Stoter, J. E. (2016). An improved lod specification for 3d building models. *Comput. Environ. Urban Syst.*, 59:25–37.

Blocken, B. (2015). Computational fluid dynamics for urban physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations. *Building and Environment*, 91:219–245. Fifty Year Anniversary for Building and Environment.

Blocken, B. and Persoon, J. (2009). Pedestrian wind comfort around a large football stadium in an urban environment: Cfd simulation, validation and application of the new dutch wind nuisance standard. *Journal of Wind Engineering and Industrial Aerodynamics*, 97(5):255–270.

Blocken, B., Roels, S., and Carmeliet, J. (2004). Modification of pedestrian wind comfort in the silvertop tower passages by an automatic control system. *Journal of Wind Engineering and Industrial Aerodynamics*, 92(10):849–873.

Blocken, B., Stathopoulos, T., and Carmeliet, J. (2007). Cfd simulation of the atmospheric boundary layer: wall function problems. *Atmospheric Environment*, 41(2):238–252.

Blocken, B., Stathopoulos, T., and van Beeck, J. (2016). Pedestrian-level wind conditions around buildings: Review of wind-tunnel and cfd techniques and their accuracy for wind comfort assessment. *Building and Environment*, 100:50–81.

Deininger, M. E., von der Grün, M., Piepereit, R., Schneider, S., Santhanavanich, T., Coors, V., and Voß, U. (2020). 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(11).

Franke, J., HELLSTEN, A., Schlünzen, H., and Carissimo, B. (2010). The best practice guideline for the cfd simulation of flows in the urban environment: an outcome of cost 732. *Proceedings of the 5th International Conference on Computational Wind Engineering, Chapel Hill, USA*.

Franke, J., Hirsch, C., Jensen, G., Krüs, H., Miles, S., Schatzmann, M., Westbury, P., Wisse, J., and Wright, N. (2004). Recommendations on the use of cfd in wind engineering. In *Proceedings of the International Conference on Urban Wind Engineering and Building Aerodynamics*, pages C.1.1–C1.11. conference; COST Action C14, Impact of Wind and Storm on City Life and Built Environment; 2004-05-05; 2004-05-07 ; Conference date: 05-05-2004 Through 07-05-2004.

García-Sánchez, C., Vitalis, S., Paden, I., and Stoter, J. (2021). The impact of level of detail in 3d city models for cfd-based wind flow simulations. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVI-4/W4-2021:67–72.

García-Sánchez, C., Van Tendeloo, G., and Gorlé, C. (2017). Quantifying inflow uncertainties in rans simulations of urban pollutant dispersion. *Atmospheric Environment*, 161:263–273.

Jung, M. C., Park, J., and Kim, S. (2019). Spatial relationships between urban structures and air pollution in korea. *Sustainability*, 11(2):1–17.

Kim, B., Lee, D.-E., Preethaa, K. S., Hu, G., Natarajan, Y., and Kwok, K. (2021). Predicting wind flow around buildings using deep learning. *Journal of Wind Engineering and Industrial Aerodynamics*, 219:104820.

Ledoux, H., Biljecki, F., Dukai, B., Kumar, K., Peters, R., Stoter, J., and Commandeur, T. (2021). 3dfier: automatic reconstruction of 3d city models. *Journal of Open Source Software*, 6(57):2866.

Liu, S., Pan, W., Zhang, H., Cheng, X., Long, Z., and Chen, Q. (2017). Cfd simulations of wind distribution in an urban community with a full-scale geometrical model. *Building and Environment*, 117:11–23.

Mandal, B. and Mazumdar, H. (2015). The importance of the law of the wall. *International Journal of Applied Mechanics and Engineering*, 20(4):857–869.

Mirzaei, P. (2021). Cfd modeling of micro and urban climates: Problems to be solved in the new decade. *Sustainable Cities and Society*, 69:102839.

Nielsen, P. V., Allard, F., Awbi, H. B., Davidson, L., and Schälin, A. (2007). Computational fluid dynamics in ventilation design rehva guidebook no 10. *International Journal of Ventilation*, 6(3):291–294.

Nieuwstadt, F. and Duynkerke, P. (1996). Turbulence in the atmospheric boundary layer. *Atmospheric Research*, 40(2):111–142. Trophospheric turbulence.

Paden, I., García-Sánchez, C., and Ledoux, H. (2022). Towards automatic reconstruction of 3d city models tailored for urban flow simulations. *Frontiers in Built Environment*, 8.

Pađen, I. (2021). Automatic reconstruction of 3d city modelstailore d to urban flow simulations. `chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://3d.bk.tudelft.nl/ipaden/files/phd_proposal.pdf`.

Peters, R., Dukai, B., Vitalis, S., Liempt, J., and Stoter, J. (2021). Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands.

Santiago, J., Dejoan, A., Martilli, A., Martin, F., and Pinelli, A. (2010). Comparison between large-eddy simulation and reynolds-averaged navier–stokes computations for the must field experiment. part i: Study of the flow for an incident wind directed perpendicularly to the front array of containers. *Boundary-Layer Meteorology*, 135:109–132.

Sousa, J., García-Sánchez, C., and Gorlé, C. (2018). Improving urban flow predictions through data assimilation. *Building and Environment*, 132:282–290.

Sousa, J. and Gorlé, C. (2019). Computational urban flow predictions with bayesian inference: Validation with field data. *Building and Environment*, 154:13–22.

Tu, J., Yeoh, G.-H., and Liu, C. (2018). Chapter 7 - practical guidelines for cfd simulation and analysis. In Tu, J., Yeoh, G.-H., and Liu, C., editors, *Computational Fluid Dynamics (Third Edition)*, pages 255–290. Butterworth-Heinemann, third edition edition.

Vitalis, S., Arroyo Ohori, K., and Stoter, J. (2019). Incorporating topological representation in 3d city models. *ISPRS International Journal of Geo-Information*, 8(8).

Yoshie, R., Mochida, A., Tominaga, Y., Kataoka, H., Harimoto, K., Nozu, T., and Shirasawa, T. (2007). Cooperative project for cfd prediction of pedestrian wind environment in the architectural institute of japan. *Journal of Wind Engineering and Industrial Aerodynamics*, 95(9):1551–1578.

COLOPHON

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.