The effect of building footprint uncertainty in CFD simulations

Christos Chontos
Student Number: 5579392

1st supervisor: Clara García-Sánchez
2nd supervisor: Akshay Patil

March, 2025

Contents

List of Figures

List of Tables

1 Introduction

Computational Fluid Dynamics (CFD) is the scientific field that analyzes fluid flows in various applications. In the field of building design and engineering, CFD simulations play a crucial role in predicting and optimizing airflow patterns. As stated by Franke et al. (2004) Computational Fluid Dynamics (CFD) is in general a numerical technique in which equations describing the fluid flow are solved on a computer. However, when we want to reconstruct and simulate wind-flows in urban environments, the accuracy of our simulation and of our model is strongly related to the quality of the input data. These data could be the geometric representation of buildings. They are typically used to raise the buildings, through manual or automated methods, and are a fundamental component in CFD simulations. However, in practice these available data often contain uncertainties that must be addressed.

The building footprints define the shape, dimensions, and orientation of the building, therefore influence the flow patterns and aerodynamic characteristics in the surrounding environment. The quality assessment of the building footprints contains various aspects, based on the primary data, according to Fan et al. (2014). These could be their completeness, position accuracy, shape accuracy and semantic accuracy. Therefore, the process of acquiring building footprints involves various factors of uncertainty and several limitations, and when we perform a Computational Fluid Dynamic (CFD) simulation, could have impact on our final results. These uncertainties can arise from inaccuracies in data collection techniques, errors in measurements, or assumptions made during the modeling process. An example of such bias is rotation (Figs.1a,1b), which can play a significant role according to the directions of the wind. Also, the level of detail (LoD) of the building model can vary – higher LoDs include roof shapes and facade details, while lower LoDs might use only the footprint outline extruded to a certain height. Higher LoDs provide more accuracy, but the requirements of complex data and the modeling effort are higher (Pađen et al., 2022).

(a) Actual (b) Rotated

Figure 1: Building Footprint

The uncertainties in building footprint representation can have a significant impact on CFD simulation results, such as wind speed distributions, turbulence levels, and pollutant dispersion patterns. Prior studies have noted that even minor geometric discrepancies can lead to notable differences in simulation results. For instance, Ricci et al. (2017) compared CFD simulations against wind-tunnel data for a single building block and then for a whole district, using three levels of geometric detail (simplified, medium, and detailed) and found out that the simplifications introduced measurable errors in the predicted wind velocities. Similarly, Hågbo et al. (2021) showed that using an extruded-footprint model (very simplified geometry) in a neighborhood wind simulation led to prominent variations in the wind field compared to using a more accurate building model. In their case, a coarse model based only on 2D footprints produced different wind speed and turbulence patterns, highlighting how sensitive CFD results are to the fidelity of building representations. For our case we will focus on this effect

having as purpose the evaluation of the impact of these uncertainties on the CFD simulation results, determining their spread when the input geometry is uncertain.

The understanding of how the footprint uncertainties translate into different CFD results, is important for several urban wind engineering applications. Therefore, quantifying the sensitivity of CFD outcomes to building footprint uncertainty is an important step toward uncertainty quantification (UQ) in urban simulations. Given the importance of these simulations in real-world decision-making, there is an increasing recognition of the necessity to evaluate and reduce uncertainties. CFD predictions are not deterministic truths but are subject to uncertainty from various sources. This thesis is motivated by this specific need to understand and quantify this uncertainty. Addressing a part of this issue is particularly crucial as high-resolution urban data become more common. By doing so, we aim to provide guidance on data requirements (e.g. when a simple footprint model is sufficient and when a detailed model is warranted). In summary, accurate building footprints are fundamental for reliable urban CFD modeling, and this study seeks to evaluate this effect. This will help wind engineers, urban planners, etc. to take more into account the uncertainty in their analyses.

2 Related work

There are several sources of uncertainty in the results of CFD simulations. To evaluate how well the simulations can predict real-world conditions, it is important to measure the impact of uncertainties in full-scale simulations and validate the results using field experiments. There are two main types of uncertainty: aleatory uncertainty, which comes from natural variations in the system, and epistemic uncertainty, which arises from knowledge gaps or limitations in the model (García-Sánchez et al., 2014). For example, there might be a considerable uncertainty in the prediction of urban flow and dispersion because of variability in the inflow conditions. It is important to make it clear that this is an aleatory uncertainty; it represents physical variability inherent to the system being analyzed (García-Sánchez et al., 2017). Also, according to García-Sánchez et al. (2017), three uncertain parameters were defined to be examined: the velocity magnitude and direction, and the roughness length of the terrain.

Researchers have been working to identify and quantify how each of these uncertainties influences the results. Numerous studies have compared CFD predictions with experimental data to evaluate overall model accuracy and identify uncertainty sources. As it is presented by Robins et al. (2000), multiple teams simulated gas dispersion around buildings and compared their results to wind tunnel experiments. Substantial differences between the models was reported, stemming from user-related settings, differences in source modeling, boundary conditions, and numerical schemes. Also, was noted that none of the simulations matched exactly the measurements. Such inter-comparison studies underscore the importance of standardized best practices and uncertainty analysis in CFD. In response, the CFD community has developed guidelines (e.g. Franke et al. (2007), Tominaga et al. (2008)) to minimize some of these uncertainties by recommending best modeling practices for urban wind simulations. But, even when following the guidelines, uncertainties cannot be eliminated completely– especially those tied to input data like geometric details or inflow turbulence.

2.1 Geometric uncertainties and CFD

In Blocken and Stathopoulos (2013), it is highlighted: 1) the necessity of increased focus on the assessment of pedestrian-level wind comfort and wind safety instead of only on wind speed conditions (mean velocity and turbulence), 2) continuous sensitivity analysis, validation studies and provision of guideline for CFD simulation of pedestrian-level wind conditions. So, within the broader scope of CFD uncertainty research, the impact of geometric uncertainty has been a focal point in recent years. Ricci et al. (2017) provided a detailed analysis of how geometrical simplifications affect urban wind flow simulations. In their study, a part of a city (a district in Livorno, Italy) was simulated (Fig.2) using three different digital models of a representative building block: one with a very simplified footprint and height, one with an intermediate level of detail, and one with a highly detailed shape. By comparing CFD results from these models to wind tunnel measurements, they quantified the deviations caused purely by geometry differences. They found that the simplified model under-predicted wind speed, whereas the detailed model showed much closer agreement with experimental data. This work demonstrated that geometric detail is a significant source of uncertainty: errors in mean wind velocity predictions grew as the building representation became more crude. In a related vein, Hågbo et al. (2021) examined the influence of building geometry input data on CFD simulations for pedestrian wind comfort. Their case study involved a suburban neighborhood in Norway for which four sets of building models were created: varying from a model based on a national GIS building footprint dataset to one derived from high-resolution laser scanning, and even an extreme case where buildings were modeled as simple extruded footprints (Fig.3). Their findings highlighted the importance of geometry reliability – the simulation

using the basic extruded-footprint model showed notable deviations in wind flow patterns compared to simulations with more accurate building shapes. Also, Hågbo et al. (2021) noted that using the moderately detailed GIS data yielded results not far off from the laser-scanned model, but the lowest-detail model (footprint extrusion) had clear differences. This suggests that there may be a threshold of geometric detail beyond which additional fidelity returns less improvements.

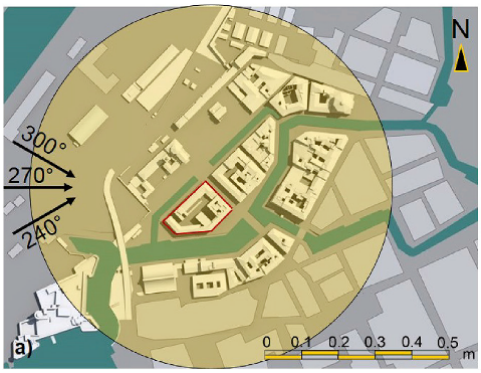


Figure 2: Geometry (Digital model)
Source: Ricci et al. (2017)

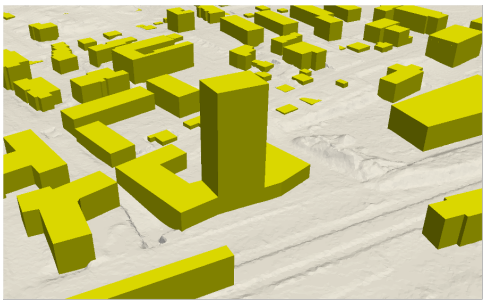


Figure 3: Geometry (Footprint extrusion model)
Source: Hågbo et al. (2021)

Several other works have explored geometric effects on CFD in urban contexts. For example, Franke et al. (2012) conducted a validation of the OpenFOAM CFD code for micro-scale obstacle flows using a standard k–ϵ turbulence model. They followed the German VDI guideline for obstacle-resolving models and compared simulation results with laboratory data for simple building configurations. This validation effort confirmed that, with careful setup, CFD can reasonably predict flow around buildings. However, some discrepancies were observed due to geometrical details that were difficult to resolve on the grid (e.g. sharp-edge separation points). The Franke et al. (2012) study is particularly relevant here because it used the OpenFOAM software and the widely-adopted k–ϵ turbulence model, as do many urban flow studies. It highlighted that even when using the same solver and model, differences in geometry modeling (such as how to represent corners or small structures) can lead to differences in outcomes, since geometric representation is an important contributor to uncertainty.

2.2 Benchmark studies and databases

2.2.1 CEDVAL database

The CEDVAL (Compilation of Experimental Data for Validation of Micro-Scale Dispersion Models) database is an essential resource for validating Computational Fluid Dynamics (CFD) models in urban wind flow and dispersion studies (Uni-hamburg.de, 2024). Developed by the Environmental Wind Tunnel Laboratory at the University of Hamburg, CEDVAL contains a collection of high-quality wind tunnel measurements designed to benchmark numerical simulations. These datasets include velocity profiles, boundary conditions, turbulence parameters, and pollutant dispersion patterns. Researchers have used CEDVAL cases to test the accuracy of their simulations. For instance, one common benchmark is flow and dispersion around a single cubic building (CEDVAL case A1-1), where many studies have compared CFD results (using RANS k–ϵ models) against the measured velocity profiles. Longo et al. (2017) is an example that employed CEDVAL datasets to evaluate the accuracy of advanced turbulence models and boundary conditions in simulating wind flow around different building compositions. Their study demonstrated that while traditional k–ϵ models captured mean velocity

fields reasonably well, they struggled with accurately representing turbulent kinetic energy distributions, particularly in wake regions. CEDVAL database's role is underscored as a fundamental reference for CFD validation, helping to ensure that numerical models can reliably replicate real-world scenarios.

2.2.2 Joint Urban 2003 field study

The Joint Urban 2003 (JU2003) field study was a large-scale urban atmospheric dispersion experiment conducted in Oklahoma City to investigate wind flow and pollutant transport in a realistic urban environment. The study included tracer gas releases and extensive meteorological measurements, providing valuable data for validating Computational Fluid Dynamics (CFD) models (Lee et al., 2004). Scientists simulated these experiments and one finding from those efforts was that accurately modeling of the buildings was crucial. Flaherty et al. (2007) used JU2003 data to evaluate CFD predictions of tracer dispersion. Their findings highlighted that while CFD captured general dispersion patterns, uncertainties were rising from variations in wind direction and source positioning, leading to discrepancies of up to 50% in modeled versus observed concentrations. The study also demonstrated the strong influence of tall buildings (Fig.5) on flow separation and mixing, emphasizing the need for accurate urban geometry representation in CFD. Similarly, García-Sánchez et al. (2014) quantified the uncertainty associated with inflow boundary conditions in CFD simulations of Oklahoma City, using JU2003 field measurements as a benchmark. Their research demonstrated that variations in wind speed, wind direction, and surface roughness significantly affected velocity predictions at multiple measurement stations, reinforcing the importance of accurate inflow parameterization in CFD. These findings reinforce the importance of uncertainty quantification in urban CFD.

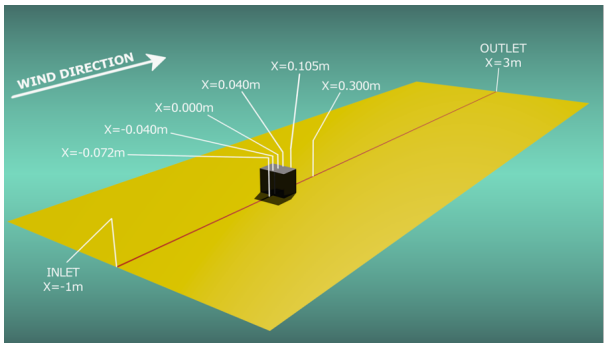


Figure 4: CEDVAL case A1-1
Source: Longo et al. (2017)

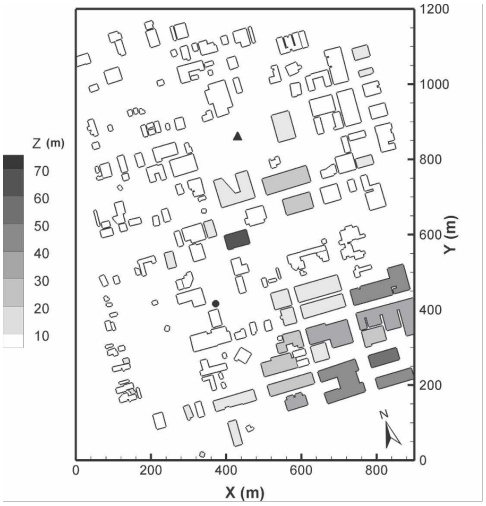


Figure 5: Computational domain
Source: Flaherty et al. (2007)

2.2.3 Architectural Institute of Japan

In Japan, working groups of the Architectural Institute of Japan (AIJ) conducted extensive cross-comparisons between CFD simulation results and high-quality wind-tunnel measurements to support the development of guidelines for practical CFD applications. Part of these efforts were reported by Yoshie et al. (2007) and this group intended to propose the guide-

lines based on the results of their own benchmark tests. In 2008, Tominaga et al. published the "AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings". The feature of these guidelines is that they are based on cross-comparison between CFD predictions, wind tunnel test results and field measurements for seven test cases (Fig.6) used to investigate the influence of different computational conditions for various flow fields (Tominaga et al., 2008).

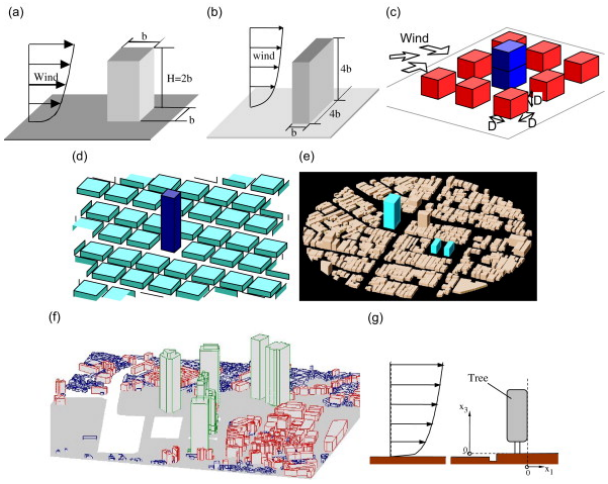


Figure 6: AIJ cases
Source: Tominaga et al. (2008)

According to Tominaga et al. (2005), a comparison between CFD results and wind-tunnel measurements had already been conducted some years earlier, for two different building cases. The first was located at Niigata (Japan), which consisted of low-rise houses and the second was the Shinjuku sub-central area (Japan), which consisted of high-rise buildings. For the first case overall, CFD matched the wind tunnel results well, when assessing the wind environment. For the second case, the high-rise buildings presented a good match, while for some low-rise houses the accuracy of CFD results depended a lot on grid resolution. Differences appeared when the grid was not fine enough to capture details properly. These kind of differences we will try to capture in the current thesis.

Another similar comparative study was conducted by Tominaga and Blocken (2015), where velocity was measured in a cross-ventilated flow. The geometry that was used imposes significant similarities with the one that will be used in the current thesis. However, Tominaga et al. (2008) presents a set of guidelines that summarizes important points in using the CFD technique for appropriate prediction of pedestrian wind environment. The reason was that the influence of the computational conditions (grid discretization, domain sizes, boundary conditions, etc.), on the prediction accuracy had not been systematically investigated.

In summary, the related work in this domain establishes that:

- Uncertainty in CFD simulations of urban flow is inevitable and multifaceted.
- Geometric uncertainty is a particularly influential factor for urban wind and dispersion predictions.
- Numerous validation efforts (wind-tunnel and field studies) have helped quantify these uncertainties, showing where and by how much CFD results might deviate.

The insights from previous studies form the foundation for this thesis. The aim is to contribute

to the ongoing effort of uncertainty quantification in CFD, ultimately guiding better modeling practices and more reliable predictions for urban wind engineering applications.

3 Research questions

3.1 Objectives

The objective of this thesis is to assess the impact of building footprint uncertainty on CFD simulations. By running multiple simulations with different geometries, or wind angles, the research aims to identify how variations in building geometry, dimensions, and orientation influence the flow patterns in our domain. As a result, our aim is to really understand the magnitude of geometry uncertainty impact and identify the parameters that have the most significant influence on the CFD simulation results.

Some more detailed goals and research questions are:

- Validation of our CFD cases, in order to achieve an acceptable difference, in comparison with the wind tunnel data (approximately 10% or lower).
- What is the level of uncertainty in our CFD cases and how is this affected into our CFD workflow, in order to enhance the accuracy and reliability of simulations?
    1. How does this uncertainty deviate if we perturb the buildings' geometry?
    2. How does this uncertainty deviate if we rotate the buildings' footprint?
    3. What is the impact on the uncertainty if we change the wind direction?

3.2 Scope of research

By accomplishing this thesis, we seek to improve the understanding of uncertainty in CFD simulations. To achieve these objectives, during the research we will run multiple CFD simulations. By systematically perturbing building geometries, adjusting footprint orientations, and varying wind directions, this research aims to quantify how such uncertainties propagate through CFD simulations and influence the results. The primary goal is to assess the magnitude of uncertainty effects on predicted wind flow patterns, ultimately identifying the most influential geometric parameters. A critical aspect of this study involves validating CFD results against wind tunnel data from AIJ Case C, ensuring an acceptable error margin (approximately 10% or lower). Additionally, the research will explore how uncertainty behaves across different stages of the CFD workflow—ranging from preprocessing and meshing to solution convergence and post-processing—providing insights into how modeling choices can enhance simulation accuracy and reliability. The study's findings will contribute to a better understanding of uncertainty quantification in urban CFD and update the best practices for geometric representation in wind flow modeling.

4 Tools and dataset used

4.1 Tools

The necessary tools that will be used in order to elaborate the Thesis topic, are:

Tool	Purpose
Excel	Pre-processing of numerical data
MeshLab	Visualisation of geometry
OpenFOAM	CFD Simulations
Visual Studio Code	Handling OpenFOAM and connection to remote server
Paraview	Post processing and visualisation of geometry
Python	Coding, processing and plotting

4.2 Dataset

The case that we will work on, is the Case C - Simple building blocks, from the Architectural Institute of Japan and is presented below (Fig.7). It is an example of a canonical case for simulations and is a hypothetical case study that has been created to represent a typical realistic urban-case scenario. It consists of nine building blocks in cubic shape. Each cube has a dimension of 0.2 meters and the distance between every cube is 0.2 meters respectively. There are 3 different cases that should be studied and they are presented below (Figs.8a,8b,8c). For all the three cases the only building that its height is changing is the middle one. The report related to this case can be found in Nonomura et al. (2003).

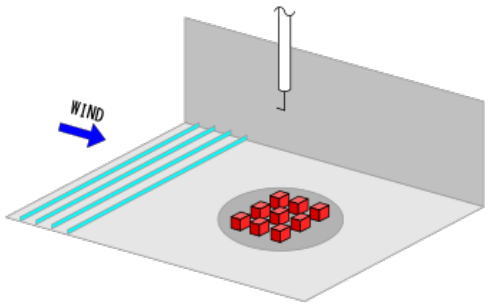
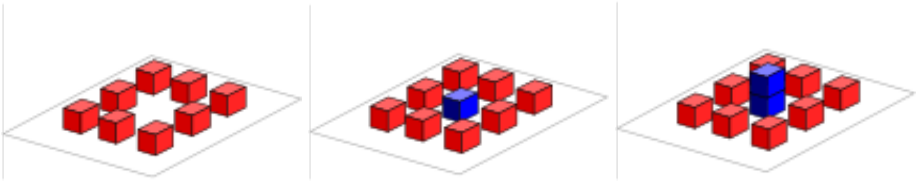


Figure 7: Case C
Source: https://www.aij.or.jp/jpn/publish/cfdguide/index_e.htm.



(a) Case 0H (b) Case 1H (c) Case 2H

Figure 8: Geometry
Source: https://www.aij.or.jp/jpn/publish/cfdguide/index_e.htm.

The data that are provided to us, stem from a wind tunnel experiment and is the inflow velocity U(m/s) values that correspond to different heights Z(m), alongside with the standard deviation of the velocity σ(m/s) (Fig.9). We used Python's matplotlib to create the following plot, where we plot the points for every U and Z value, on the xy plane and then we drew a

curved line that approximates the points. It becomes clear from the following plot, that we are having a logarithmic wind velocity profile (Fig.10).

Z(mm)	Z(m)	U (m/s)	σ_u (m/s)
10	0,01	2,372	0,56
20	0,02	2,434	0,564
30	0,03	2,548	0,563
50	0,05	2,912	0,695
70	0,07	3,042	0,71
100	0,1	3,392	0,743
150	0,15	3,483	0,78
200	0,2	3,654	0,801
250	0,25	3,82	0,812
300	0,3	4,019	0,786
350	0,35	4,205	0,83
400	0,4	4,343	0,809
600	0,6	4,985	0,796
800	0,8	5,713	0,587
1000	1	6,12	0,283
1200	1,2	6,201	0,148

Figure 9: Inflow values
Source: https://www.aij.or.jp/jpn/publish/cfdguide/index_e.htm.

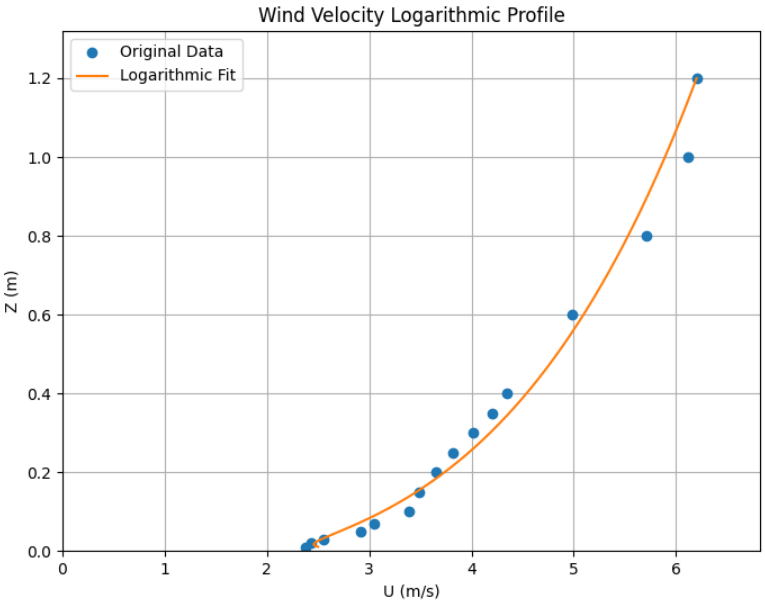


Figure 10: Wind velocity profile

Furthermore, besides given the dimensions of the building domain and the dimension of each cube, we also have the inflow angle, which takes three different values: 0°, 22.5° and 45° degrees. Below we see a top view of our domain with its dimensions (Fig.11). The red points are locations where the wind was measured experimentally. They are 120 points and the velocity value of the wind there is known to us. Their height is 0.02m from the ground. Specifically, for them we have the velocity ratio of every height case (0H, 1H, 2H) and for every angle case (0°, 22.5°, 45°), (9 different cases in total). The data were aquired from www.aij.or.jp, where the Case C is stored.

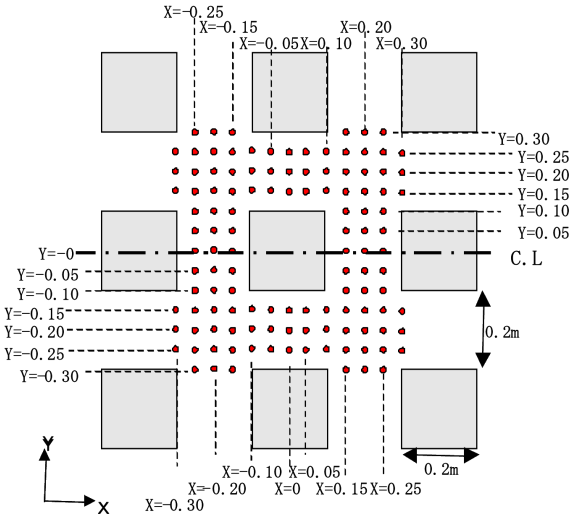


Figure 11: Top view and probes
Source: https://www.aij.or.jp/jpn/publish/cfdguide/index_e.htm.

5 Methodology

To setup a case and run a CFD simulation in OpenFOAM, we need the following steps. A general overview of the whole procedure is:

- Preparation of our geometry
- Initial and boundary conditions
- Mesh generation
- Simulation execution
- Analysis and post-processing of the results

5.1 Preparation of geometry

To begin with, we had to obtain the geometry of our buildings. In our case, the geometry was not given in OBJ format, so we created an OBJ file named *cubes.obj*. For the creation of the OBJ file, we used a simple text editor. According to the given dimensions and location of the cubes, we calculated the coordinates of every vertex. Then, we formed the edges and faces. Our geometry is triangulated and the total amount is of vertices: 72 (8 for every cube), edges: 162 (18 for every cube), faces: 108 (12 for every cube). It is a simple geometry so we could do this process manually. We created 3 OBJ files, for the 0H, 1H and 2H case, respectively. Finally we imported the vertex coordinates, edges and faces to the TXT file and with MeshLab we were able to visualize our final triangulated geometry and it is represented below (Fig.12).

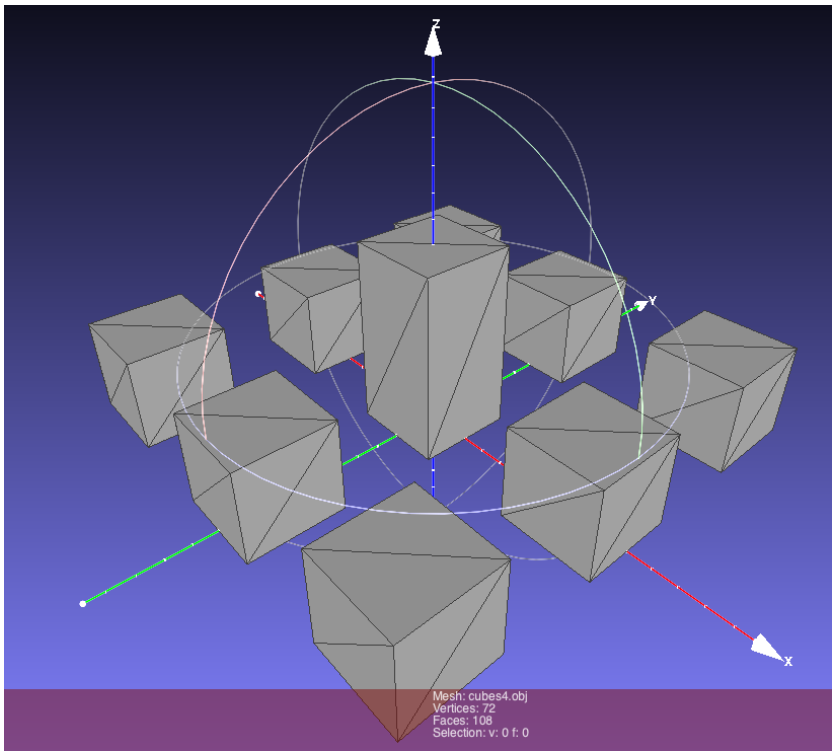


Figure 12: CaseC.obj (2H, MeshLab view)

5.2 Initial and boundary conditions

5.2.1 Initial conditions

The next step was to specify our boundary and initial conditions of our simulation. In RANS family of turbulence models, it is very common to use the k-ϵ turbulence model with a logarithmic velocity profile. In the *case/0* directory, all the physical properties of the wind are located. Specifically, the directory *0/ABLConditions* of our case, is the folder responsible for this logarithmic profile of the wind (OpenFOAM.com). The values of the initial conditions are stored here and for their calculation and initialization of the wind flow, a reference height and inflow velocity was chosen. This selection was based on the logarithmic wind profile and the characteristics of our building domain.

So, we chose a representative height that captures the wind conditions above our objects and we used a height slightly above the highest structure in our domain. In our case, this was $z_{\text{ref}} = 0,6m$. Our tallest object is 0,4 meters and this was the next available measurement. We also used the corresponding velocity at this selected height (Fig.9), which is $U(z_{\text{ref}}) = 4,985m/s$, running along the x axis, with positive direction. The reason behind this selection is because we want to capture free-stream conditions. Above the tallest structure, the wind flow is less affected by roughness elements and at the same time, we avoid the generation of turbulence and flow separation, which may take place at lower heights.

So, based on the following steady-state, incompressible, Reynolds-averaged Navier-Stokes equations that were presented by Richards and Hoxey in their pioneering paper (Richards and Hoxey, 1993), we calculated the initial values for the turbulence parameters:

$$(1)\ U(z) = \frac{u_*}{\kappa} \ln\left(\frac{z+z_0}{z_0}\right), \quad (2)\ k(z) = \frac{u_*^2}{\sqrt{C_\mu}}, \quad (3)\ \epsilon(z) = \frac{u_*^3}{\kappa(z+z_0)}$$

where, the von Karman constant κ = 0.41 and constant C_μ = 0.09. Also the height in our geometry $z_{\text{ground}} = 0m$, since we don't have any altitude. Finally, the direction of the flow (*flowDir*) was set only along the x-axis (1 0 0), so the wind angle is 0°. It is important to mention that in later stages of this thesis, the cases where the wind angle is 22.5° and 45° will be examined.

Another important parameter that we had to calculate, is the aerodynamic roughness length (z_0). According to Yoshie et al. (2007), the experimental the value that was used was z_0 = 0.00018. However, we decided to stay coherent and calculate z_0 through our inflow data (Fig.9), since this is a better justified and independent method. Equation (1) could be rewritten as:

$$U(z) = \frac{u_*}{\kappa} ln(z) - \frac{u_*}{\kappa} ln(z_0)$$

By performing linear regression, we fit our line, such that $U(z) = mz + \alpha$, where $m = \frac{u_*}{\kappa}$ is the slope of the line and α is the intercept. From α we can calculate the aerodynamic roughness length. So z_0 would be derived by fitting a line through the velocity profile in the log-law region. Using Python we created the following graph (Fig.13) , that represents the line-fit and the calculated values are $z_0 = 0.0073m$ and friction velocity $u_* = 0,478m/s$. According to the Wieringa (1992) roughness classification, the z_0 value indicates that we have featureless land surface without any noticeable obstacles (smooth terrain).

Then, the calculations that followed were the turbulent kinetic energy $k = 0.762$, according to equations (2) and the dissipation rate, turbulent epsilon $\epsilon = 0.428$, according to equation (3).

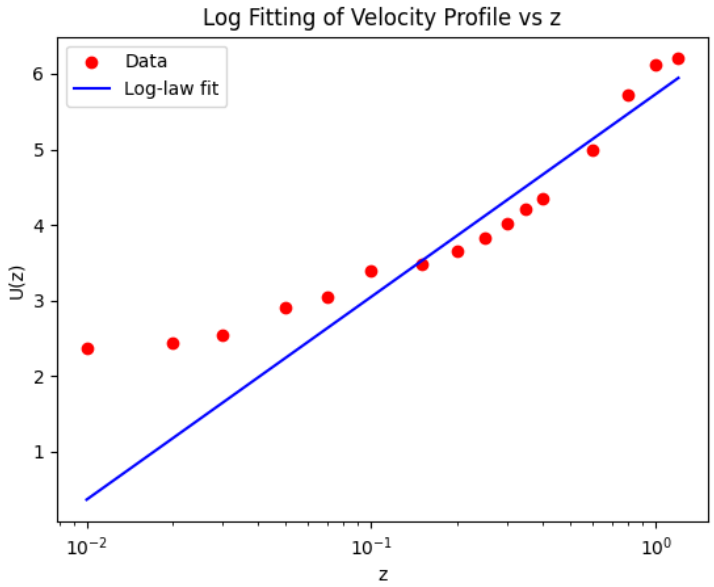


Figure 13: Log-fitting line

5.2.2 Boundary conditions

The boundary conditions were set in order to describe how the flow behaves at the boundaries of our mesh. Our computational domain consists of boundaries and has a specific volume. From some of these boundaries the wind flows by entering or leaving it. In general, only the bottom of the domain corresponds to physical a boundary in reality, and the top and side boundaries are artificial (Blocken, 2015). For that reason, the atmBoundaryLayerInlet conditions that were used had as purpose to ensure a realistic wind profile for our urban case at the inlet. For the buildings and ground the *kqRWallFunction*, *epsilonWallFunction* and *nutkWallFunction* were used, as they considered as rough surfaces. For the sides and top surfaces, *symmetry* condition was used. Finally, the *inletOutlet* condition was used at the outflow, where the wind exits our domain freely. Below (Table 1) are presented the boundary conditions that were used.

Patch name	$U(m/s)$	$p(m^2/s^2)$	$k(m^2/s^2)$	$\epsilon((m^2/s^3)$	$nut(m^2/s)$
Inlet	atmBoundaryLayerInletVelocity	zeroGradient	atmBoundaryLayerInletK	atmBoundaryLayerInletEpsilon	calculated
Outlet	inletOutlet	uniformFixedValue	inletOutlet	inletOutlet	calculated
Ground	uniformFixedValue	zeroGradient	kqRWallFunction	epsilonWallFunction	nutkAtmRoughWallFunction
Buildings	uniformFixedValue	zeroGradient	kqRWallFunction	epsilonWallFunction	nutkWallFunction
Sides and Top	symmetry	symmetry	symmetry	symmetry	symmetry

Table 1: Boundary conditions

5.2.3 Scheme selection

The scheme selection impacts the accuracy and stability of our CFD simulation and it refers to the numerical method that is used to approximate the differential equations governing the fluid flow (in our case, Navier-Stokes equations).

The gradient scheme that was used is *CellLimited Gauss Linear*. It is a second-order scheme, ideal for preventing excessive oscillations while the accuracy is maintained. For the divergence schemes it is used the *bounded Gauss linearUpwind limited* option and *bounded Gauss limitedLinear*, which are also second order schemes.

The reason behind these selections is that with first-order discretization schemes, numerical diffusion is caused. According to the best practice guidelines Blocken (2015), it is important to use high-quality grids and higher-order discretization schemes, since they allow us to have more accurate results, without having to compromise significantly with the convergence behavior.

5.3 Mesh generation

The computational domain plays an important role in our simulation since it discretizes the geometry into smaller computational cells and also encloses the open flow inside a box.

5.3.1 Creation of the computational domain

In order to define the dimensions of this background mesh, we took into account the building domain. At first, we found the height of the tallest building, which in our case is H_{max}=0.4m for the 2H case and 0.2m for the 1H and 0H cases. As mentioned in Blocken (2015), according to Type-1 guidelines by Franke et al. (2004), the ideal domain size depends on H_{max}. For the x-axis we have $5H_{\text{max}}$ to the -x direction and $15H_{\text{max}}$ to the +x direction. On the y-axis we have on both sides $5H_{\text{max}}$ and for the z-axis we have $6H_{\text{max}}$. These distances for every direction are very specific (Fig.14), since the boundaries should be located far enough from the building domain to allow the wind flow to develop freely. The outflow boundary is set at $15H_{\text{max}}$ in order to allow the wake flow behind the buildings to fully develop (Blocken, 2015).

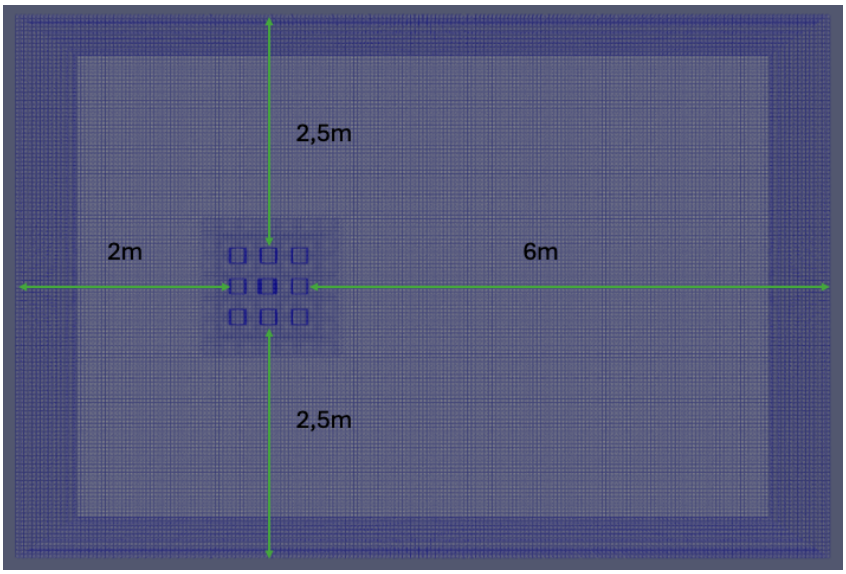


Figure 14: Size of Computational Domain

Another factor that had to be taken into consideration, before we finalize our mesh dimensions, is the Blockage Ratio (BR). According to type-2 guidelines by Blocken (2015), looking at our domain from a side view, where is the inflow of the wind, the ratio of the area covered by the building domain, to the total area of the computational domain, should be less than 3%: $BR = \frac{A_{\text{building}}}{A_{\text{domain}}}$. Also at Blocken (2015), it is mentioned that the type-3 guidelines is a combination of type 1 and 2. Specifically, what should be considered is the Directional Blockage Ratio, where separately the length and height of the buildings should be less than 17% of the length and height of our entire domain: $BR_{\text{L}} = \frac{L_{\text{building}}}{L_{\text{domain}}} = \frac{1m}{6m} < 17\%$ and $BR_{\text{H}} = \frac{H_{\text{building}}}{H_{\text{domain}}} = \frac{0.4m}{2.4m} = \frac{0.2m}{1.2m} < 17\%$. Our dimensions also cover this criteria.

The final dimensions of our computational domain are 9 x 6 x 2.4 meters (Table 2). The BlockMesh that we created is presented below (Fig.15).

Variables	Building Domain (m)	BlockMesh(m)	Mesh Dimensions(m)
x_{min}	-0,5	-2,5	
x_{max}	0,5	6,5	Dx = 9
y_{min}	-0,5	-3	
y_{max}	0,5	3	Dy = 6
z_{min}	0	0	
z_{max}	0,4	2,4	Dz = 2,4

Table 2: BlockMesh dimensions

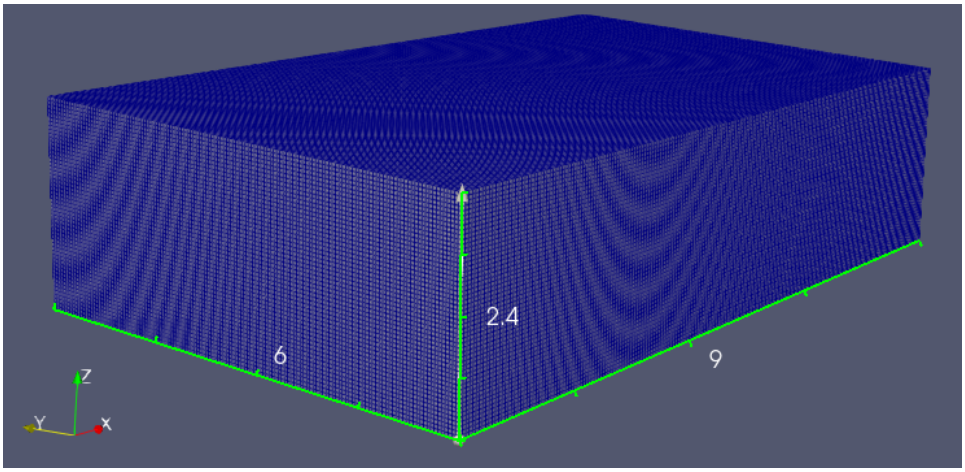


Figure 15: Blockmesh

5.3.2 Grid resolution

One more functionality of *blockMesh* is to define the resolution of the mesh, by dividing the space inside the domain to hexaxedral cells. The criteria that were followed for the selection of the appropriate cell size are analyzed below:

- According to Blocken (2015), the grid should use 10 cells per cube root of the building volume and also between every two buildings. In our dataset we have 9 cubes, as buildings, with 0.2 meters dimension, and equal distance from each other 0.2 meters. So, $0.2/10 = 0.02m$ cell size can be a first estimation.
- Another guideline that should be followed is that the height of 1st cell from the ground should not be less than our z_0 (Blocken, 2015), which is equal to 0.0073m. In our case, 0.02 is higher than 0.0073.
- This AIJ case simulates urban environment since these cubes represent simple building blocks. So, according to Blocken (2015), for pedestrian level wind studies, the 3rd or 4th cell should reach the height of the probes ($0.02m$). It is important not to extract data from the first cell closer to the ground because our wall function can heavily influence our results. So by refining the areas around them, the desired height is achieved.
- Another important aspect to be considered is the quality of the computational cells in terms of shape . The most preferable is to divide our domain into cubic cells. According to Blocken (2015) guidelines, it is advised that the stretching ratio should be kept below 1.3 in regions of sharp gradients (e.g., near buildings), to limit the truncation error. Another reason is to achieve grid independence and convergence. By using cubic cells, we can uniformly refine the grid in all three dimensions without introducing ad-

ditional complexity. According to Blocken (2015), the refinement of grids systematically is the key to achieving grid-independent results, and cubic grids are ideal for uniform refinement.

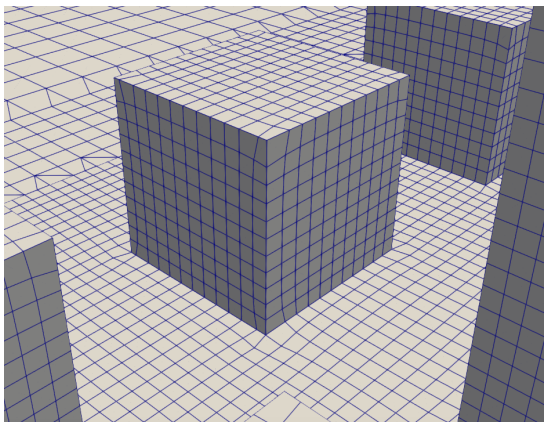
To make sure that we will run the simulation having designed the most appropriate mesh, we had to design 3 meshes with progressive refinement (a coarse, a medium and a fine mesh), run the simulation with each one of them and finally decide which one is the most suitable by comparing the simulation time, level of convergence and velocity values at the probes. As it is mentioned by Blocken (2015), we can find the resolution of the medium and coarse mesh by multiplying our cell size with a factor of 1.5 consecutively. As a result, we had 0.02*1.5 = 0.03m resolution for the medium mesh and 0.03*1.5 = 0.045 resolution for the coarse mesh.

However, it is important to mention that due to the simplicity of the geometry, the excessive computational time and the lack of convergence of the residuals led us to the conclusion that we couldn't proceed with the 0.02m resolution mesh as our fine mesh, so we considered the 0.03m cell size for our fine mesh and we simplified our coarse mesh even more, 0.045*1.5$\simeq$0.07. So, the final cell sizes for each mesh, alongside with the number of cells for every dimension, are 0.03m, 0.045m, 0.07m and are presented below (Table 3). Also by using ParaView software, we present an overview of our mesh with the three different resolutions (Figs.16a,16b,16c).

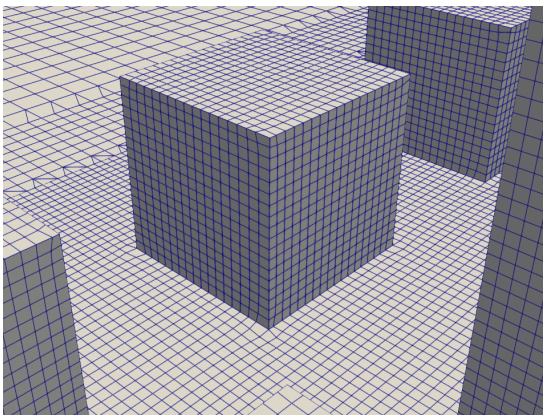
Mesh type	**Cell size (m)**	**nCells x**	**nCells y**	**nCells z**	**Total cells**
Coarse Mesh	0,07 x 0,07 x 0,07	128	86	34	374.272
Medium Mesh	0,045 x 0,045 x 0,045	200	133	53	1.409.800
Fine Mesh	0,03 x 0,03 x 0,03	300	200	80	4.800.000

Table 3: Cells per mesh

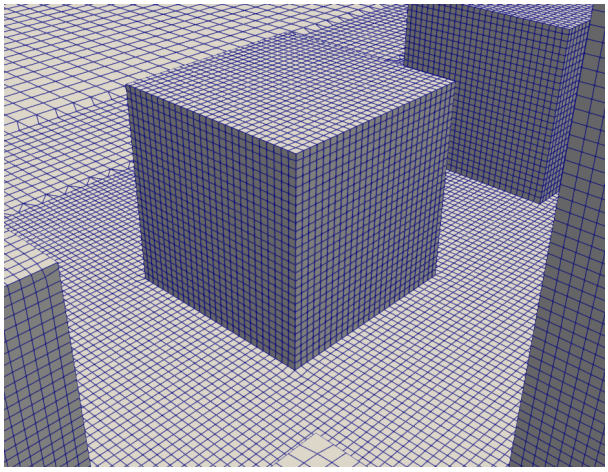
In "blockMeshDict" file we defined the meshing parameters that we mentioned so far, including the dimensions of our domain, the number of cells and the cell size. So, by applying *blockMesh* command, the background mesh was generated (Fig.10). Then, we used *surfaceFeatures* command to improve the generation of the Mesh as it allows to extract the outlines of the buildings which are going to be used in the final Mesh generation. Also this command helps to detect the location of the buildings and their shape.



(a) Coarse Mesh: 0,07m Resolution



(b) Medium Mesh: 0,045m Resolution



(c) Fine Mesh: 0,03m Resolution

Figure 16: The three meshes

5.3.3 Mesh refinement

Another important parameter that we should consider for our mesh is to achieve higher resolution in our region of interest. *SnappyHexMesh* was used to perform this refinement and adapt the mesh as close as possible to the real geometry and to our building domain. For the mesh refinement, we used the buildings model boundaries and we defined 2 levels of refinement. Specifically, we created 2 refinement boxes and 2 refinement levels in total. It was applied to all three meshes, to refine at its maximum the area of interest, which in our case is the domain of the cubes. A refinement level of 2, which is the most detailed one, was applied on the surfaces of the cubes. Then, there is a refinement box that spreads across our whole computational domain at a relatively small height, enclosing all our probes (testing points) and has dimensions (-2.5, -3, 0)x(6.5, 3, 0.04)m. This also has a refinement level of 2. It is applied throughout the entire domain and matches the lowest level of refinement along the buildings, so as to ensure that the wind profile is not heavily interpolated. Our second box includes our building site and with dimensions (-0.9, -0.9, 0)x(0.9, 0.9, 0.8)m, it has a refinement level of 1. The height of it was chosen accordingly so as not to surpass the height of $2H_{max}$. The rest of the computational domain is considered level 0. The criteria of choosing these refinement boxes was to be equally distributed in space, to include the whole building domain, our probes and to have a smooth transition from the sparser grid of the free wind flow to the denser grid that is around the cubes and reaches their walls. Since our geometry consists of plain surfaces and simple

shapes like cubes without any complicated composition or abrupt angles, it was deemed appropriate not to carry out further refinement. It is worth mentioning that the same refinement regions were maintained for the medium and coarse mesh, while the cell size was increasing.

Below, we see the refinement boxes (Fig.17) and our domain (Fig.18), in Paraview.

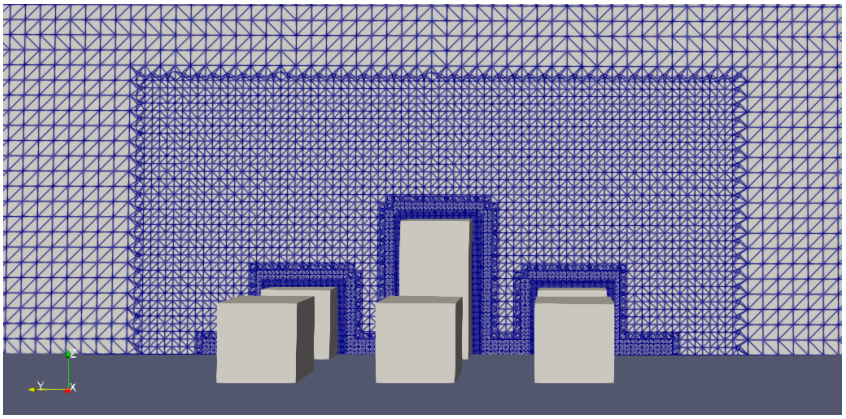


Figure 17: Refinement Boxes (Case 2H)

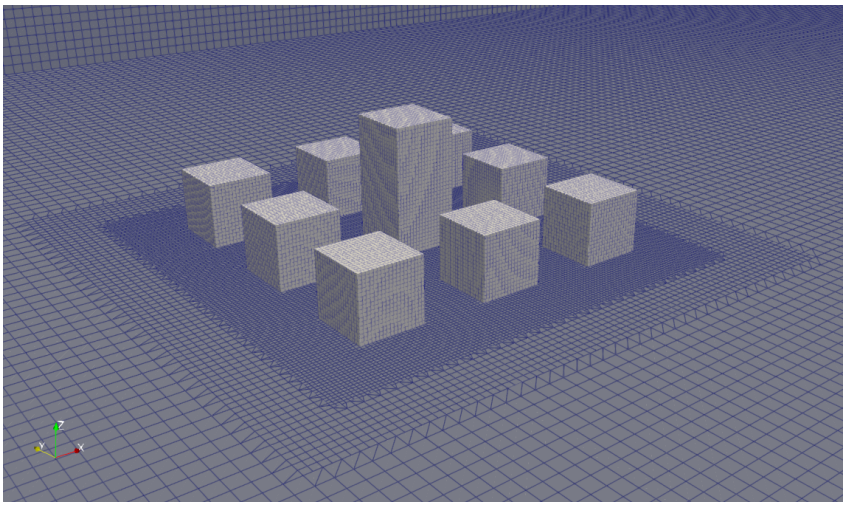


Figure 18: Refined Mesh (Case 2H)

5.4 Run the simulation

5.4.1 Preparation and solver type

The simulation is performed by an iterative method, through which the variables U, P, k, ϵ and nut will be calculated by OpenFOAM, for every individual cell of our mesh. The solver is responsible for resolving numerically our set of equations by representing particular physical phenomena. Our cases are set up for a steady-state, incompressible, turbulent flow simulation, so the solver that is used, is the *simpleFoam*. The GAMG solver is used for pressure, and a smoothSolver for velocity and turbulence, with GaussSeidel smoother in order to smooth out errors. Convergence limit has been set to 10^{-9}, for the residuals, ensuring high accuracy for our model.

5.4.2 Simulation time

We run our case with the 3 different meshes (coarse, medium and fine) and for three different building heights (0H, 1H, 2H). Then, we compared the results with the experimental values provided by the dataset. It is important to mention here that the number of processors used both for the creation of the meshes and for the simulations was 16. This number didn't change throughout the whole process, so as to compare properly the running times.

The finer the mesh, the more time it needs to converge. For our case the number of iterations was set to 3000. This is amount of iterations needed so as our fine mesh to fully converge. For the comparison, we used the same amount of iterations also for the coarse and the medium mesh. However, after several attempts, we found out that the coarse and medium mesh could converge with even less iterations. Below, at tables 4, 5 and 6, is presented in detail the time of each simulation, the time required to create the refined mesh and the amount of cells that were created. Each table represents a different geometry case. As expected, every category increases from coarse to fine and also from smaller to bigger *H*. The following numbers were taken from the *.log* files that were used to monitor the simulation.

Mesh type	**Cells after refinement**	**Meshing time (sec)**	**Running time (sec)**
Coarse mesh	449.116	10,56	254
Medium mesh	1.683.546	41,88	915
Fine mesh	5.693.109	210,86	3483

Table 4: Mesh: Case 0H

Mesh type	**Cells after refinement**	**Meshing time (sec)**	**Running time (sec)**
Coarse mesh	450.863	11,41	253
Medium mesh	1.686.534	44,4	906
Fine mesh	5.696.681	211,99	3479

Table 5: Mesh: Case 1H

Mesh type	**Cells after refinement**	**Meshing time (sec)**	**Running time (sec)**
Coarse mesh	452.688	10,85	250
Medium mesh	1.689.398	42,32	972
Fine mesh	5.703.041	214,42	3506

Table 6: Mesh: Case 2H

5.5 Analysis and post-processing of the results

5.5.1 Residuals

The first step related to the post-processing of the results is the convergence of the residuals. In practice, through the folder *Post-processing/Residuals* that was created, we plotted the residuals of the field values through Python and observed if they converge according to our standards. A representative resulting graph of the case 0H is presented below (Fig.19). The rest of the resulting graphs are presented in the appendix (Figs.25a, 25b, 25c, 26a, 26b, 26c, 27a, 27b, 27c). Also, we plotted the velocity magnitude of five probes and see that they have stabilized (Fig.20).

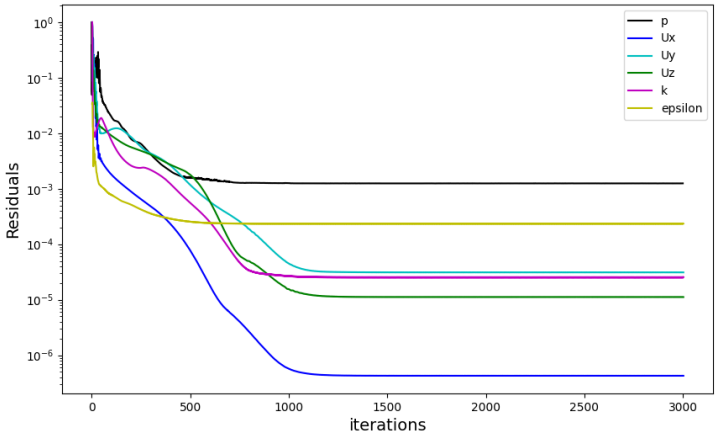


Figure 19: Residuals: Case 0H (Medium Mesh)

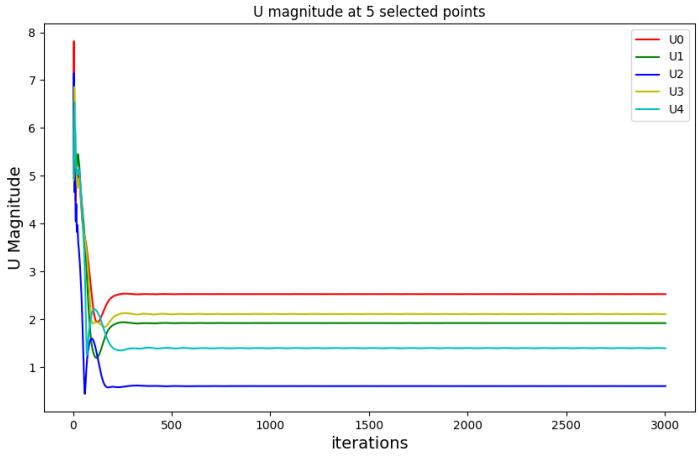


Figure 20: Velocity magnitude over time for five monitoring points

As we can see, the behavior of our mesh is pretty similar in the three cases, since the geometry of them doesn't change drastically. As it is expected the U_x variable is the one that converges faster, since the wind flows parallel to the x axis and our case simulates urban environment. In different urban environments though, that the geometry would be characterized by high complexity it would be harder to achieve small values of residuals which explains the results that we obtain in our cases. All the cases converge to the 10^{-6} or lower. We see that the coarse

and medium meshes reached the converged solution faster that that fine mesh. For the coarse mesh approximately 500 iterations needed to reach convergence and for the medium, about 1000. For the fine mesh, convergence is achieved within 2500 iterations, much longer than the other two. As it is expected, the finer the mesh, the more iterations it takes to get closer to zero, since the amount of cells is also significantly higher. Nevertheless, in terms of time and amount of iterations, the coarse and the medium mesh have a clear advantage in all three cases.

5.5.2 Field plots

Also in order to compare effectively the three meshes, we created field plots in Paraview, that are displayed below. A representative field plot of Ux for the three meshes,of case 0H is presented here (Fig.21). The rest of the plots are presented in the appendix (Figs.28a, 28b, 28c, 28d, 28e, 28f, 29a, 29b, 29c, 29d, 29e, 29f, 30a, 30b, 30c, 30d, 30e, 30f). Specifically, the tool that was used was the *'plot over line'* from the data analysis tools in *Paraview*. The line that was chosen, was the (-2.5, 0.15, 0.1), (6, 0.15, 0.1). The criteria of choosing this line was 1) to follow the development of the wind parallel to the x-axis at a height of interest, 2) to include entirely our mesh, 3) to be located at a proper height were our variables present some variance and 4) not to be obstructed by our geometry at any point.

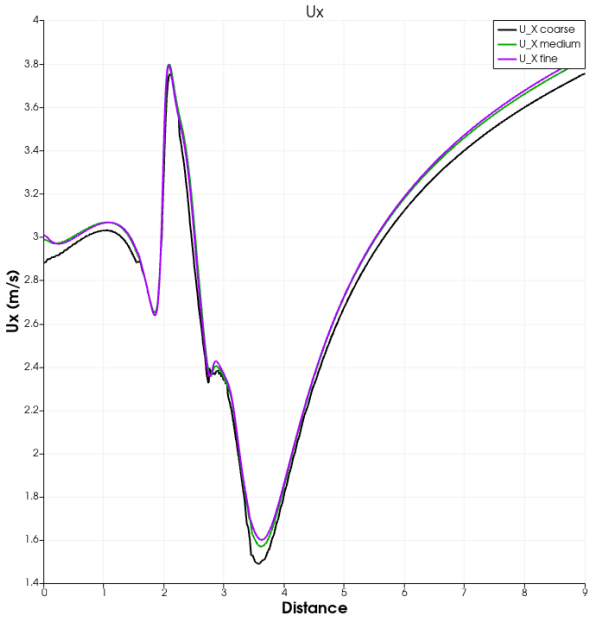
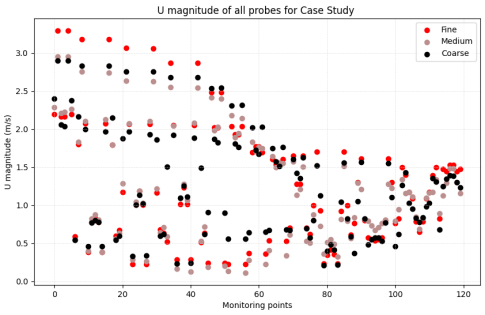


Figure 21: Ux field plot for the three meshes: Case 0H

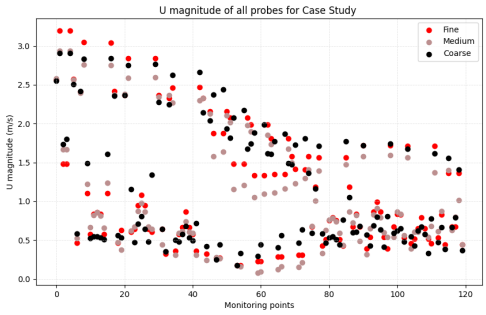
Through this tool we plotted the values of our fields of interest Ux, Uy, Uz, p, k and ϵ through out this line . As we could see from these plots, the values were closer in the cases of the fine and medium meshes, while the gap between the coarse and medium meshes seems larger.

5.5.3 Probes

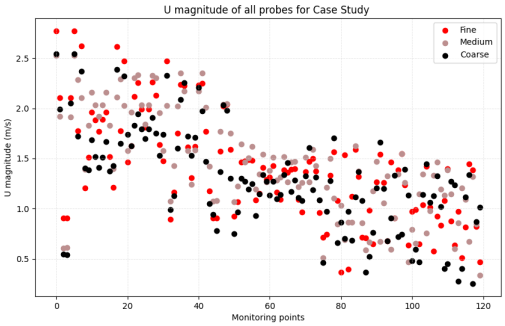
Another criteria for the selection of the most optimal mesh, is the comparison of the wind velocity at the points of interest (probes). Another folder that was created is the *postProcessing/Probes*, from where again using a Python script we plotted the velocity values of the probes. In the following plots (Figs.22a,22b,22c) we can see the velocity values of the probes from all three meshes. Each plot corresponds to a different case (0H, 1H, 2H)



(a) Case 0H



(b) Case 1H



(c) Case 2H

Figure 22: Probes

As we can see, the values of the fine and medium meshes are very similar in most locations. On the contrary, the coarse values do not seem to be so close. So the values provided by medium and fine mesh seem to be more realistic, which is beneficial for them in all three cases. However, the medium mesh reached these values much sooner (almost 75% faster) than the fine mesh.

5.5.4 Grid Convergence Index (GCI)

Since the abovementioned comparison is pretty visual, and therefore unclear to tell how much the velocity magnitude differs between our three meshes, we had to use a more meaningful metric to quantify the differences between them. Grid convergence is often analyzed using the Grid Convergence Index (GCI), a formal method proposed by Roache (1994) for quantifying grid convergence and the level of uncertainty related to grid discretization. It is considered as a more accurate and understandable way to present our results and it involves computing an error estimate based on our solutions at different grid levels. The lower GCI value, the more grid-independent our solution is. Also, lower GCI value indicates that the solution is less sensitive to further mesh refinement, suggesting grid convergence.

The implementation and validation algorithm was based on the procedure outlined by Celik et al. (2008), and was implemented through Python. The key steps involved in the GCI calculation and the results are presented below.

The necessary inputs for the calculation of GCI are the representative cell sizes (h), and the quantity of interest (phi) of the three meshes. The key requirement for selection of phi is that it should be a representative scalar quantity that is sensitive to grid refinement (Celik et al., 2008). The average velocity at our probes fits this criterion. So, we calculated the arithmetic mean of the velocity magnitudes across the 120 points:

		Case 0H	**Case 1H**	**Case 2H**
Fine Grid:	$h_1 = 0.03m,$	$phi_1 = 1.270m/s,$	$phi_1 = 1.147m/s,$	$phi_1 = 1.399m/s$
Medium Grid:	$h_2 = 0.045m,$	$phi_2 = 1.226m/s,$	$phi_2 = 1.080m/s,$	$phi_2 = 1.445m/s$
Coarse Grid:	$h_3 = 0.07m,$	$phi_3 = 1.281m/s,$	$phi_3 = 1.164m/s,$	$phi_3 = 1.297m/s$

Some key steps were the calculation of:

- Grid Refinement Ratio: $r_{21} = h_2/h_1 = 1.5$ and $r_{32} = h_3/h_2 = 1.556$. These ratios represent how much finer one grid is compared to the next coarser one.
- Error Estimation: $e_{21} = phi_2 - phi_1$ and $e_{32} = phi_3 - phi_2$. These are the differences in the computed quantities between successive grids, for our three cases.
- Apparent order of convergence (p): A measure of how fast the solution of the numerical simulation improves as the grid is refined. It is calculated using Richardson extrapolation.

So, for our three meshes, we calculated two Grid Convergence Indexes (GCIs): 1) GCI_{21}: between the medium and coarse grid and 2) GCI_{32}: between the fine and medium grid. We computed the GCI using Roache's formula where: $GCI_{21} = \frac{1.25\left|\frac{phi_2 - phi_1}{phi_1}\right|}{r_{21}^p - 1} \times 100\%$ and the results are presented below:

	Case 0H	**Case 1H**	**Case 2H**
GCI_{21}:	18.235%,	30.273%,	1.997%
GCI_{32}:	23.611%,	40.309%,	6.222%

Based on our results, we come to the observations, that the coarse mesh is not recommended because of the higher difference in all three cases, meaning that the results may not be the most reliable. Moreover, the fine mesh is the most accurate but it is computationally more expensive. Since the $GCI_{21} \approx 2\%$, further refinement might improve accuracy but with a high computational cost.

By observing the results, we see that for the case 2H the values of GCI are much lower meaning that the refinement is performed in a proper way, compared to the cases 1H and 0H that we see much higher values meaning that the grid refinement is not performed properly and we have large difference both between medium-coarse and medium-fine grids. The reason behind this inconsistency could be the slight non-uniformity, since r_{21} and r_{32} are not exactly the same. Another reason could be the necessity of further refinement at the area around our probes that the velocity value is calculated.

Finally, according to the given results, we conclude that the medium mesh is the best choice, according to GCI. Since, $GCI_{21} < GCI_{32}$, in all three cases, it means that the medium mesh is significantly more accurate than the coarse mesh and that using the fine mesh only improves

accuracy slightly. It is considered a balanced choice, since it combines reasonable accuracy, without the high computational cost of the fine mesh.

5.5.5 Mesh convergence

Achieving grid convergence in CFD indicates that the solution of the discretized equations and of our simulation becomes grid independent as the mesh is refined, and tends to the exact solution of the differential equations. It is ensures that the results are not significantly influenced by the size or distribution of the computational cells in the mesh. Also, grid convergence is important for the 1) validation of our results, since it ensures our numerical results are accurate and not dependent on grid resolution and 2) reproducibility, since it confirms that others can reproduce your results with different grids.

Finally, according to these four criteria: 1) convergence time of the residuals, 2) number of iterations needed, 3) results of the probes and 4) GCI, the mesh chosen to perform the analysis of the results is the **medium mesh**. It offers a good balance between accuracy and computational cost. In case we need higher accuracy, we could verify our results by running a few key simulations on the fine mesh but continuing the full analysis with the medium one.

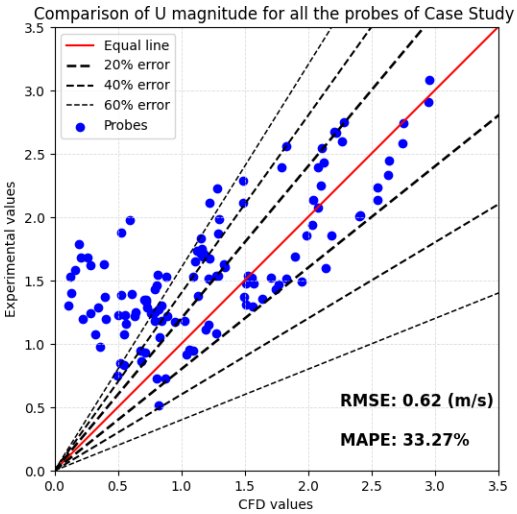
5.5.6 Comparison with experimental data

The dataset provides use with the velocities of the abovementioned probes for the three cases. We observe that for these 120 probes, the given velocity values, that are experimentally measured, are normalized by the inflow velocity at the same height (0,02 meters). This means that the velocity values that we have for the probes are expressed as a ratio relative to the inflow velocity at the same height. We can assume that the formula used for this transformation is:

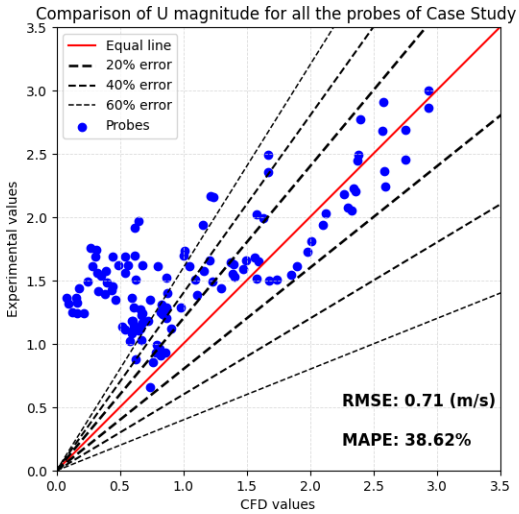
$$U_{\text{normalized}} = \frac{U_{\text{probe}}}{U_{\text{inflow}}(z)}$$

After running the simulations, the values of velocity that we have for the probes is the velocity magnitude at $0,02m$ height. For that reason we need to denormalize the velocity ratio and compare the results. On the other hand the probes' file *U.dat* that was created after running our simulations, contains a set of 3 numbers for every probe. These numbers are the components of velocity in the Cartesian coordinate system (U_x, U_y, U_z). So, in order to compute the magnitude of velocity, we used the following formula: $U = \sqrt{U_x^2 + U_y^2 + U_z^2}$.

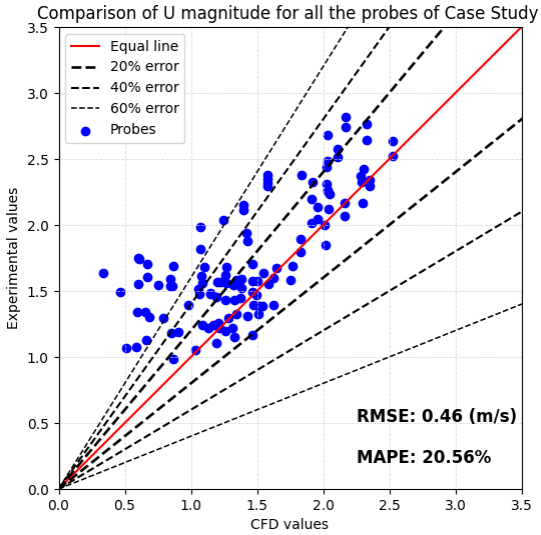
In order to proceed to the evaluation of our simulations against the experimental values, we elaborated a point by point comparison, by placing the experimental values at the y-axis and the CFD-values at the x-axis. We compared our velocity results with the measured values from the experiment. For the proper comparison, descriptive statics methods were used, where the Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), where calculated. RMSE quantifies how far are the simulation values from the actual values on average, while MAPE expresses in percentage the difference between CFD simulation and wind-tunnel experiment. Using these metric we can compare the results and determine, which geometry has better results and how our estimations are affected. The red line symbolizes the equal line and the points located on this line means that the experimental with the simulation values are equal. We used a Python script that reads the denormalized values of the probes' velocity from a CSV file and displays them alongside with the simulation values from our medium meshes, at the plots that are presented below, for every case (Figs.23a,23b,23c).



(a) Case 0H



(b) Case 1H



(c) Case 2H

Figure 23: Point by Point Comparison AIJ-CFD values

Observing the plots we can see that the best results appear in the 2H case with a $MAPE = 20\%$, while in the other two case we see that the middle cube's height, affects the error increasing it to 33% and 39% respectively. Alongside with this difference, is observed that the experimental values are higher against the values of the simulation. We can also see that the points of interest with lower velocity magnitude present higher variance compared to points with higher velocity value. The reasons behind these differences could vary and a certain set of solutions can be considered to resolve it. The problem could stem from various factors:

- The lack of proper refinement near the probes area, which is close to the ground
- Different boundary conditions. Wall effects may alter the velocity field, since the experiment was done in a wind tunnel

In future analysis, different approaches will be considered to further understand the reasons behind this difference and where the issue could arise from. By fixing our setup, our results can be improved and reach closer to the experimental results. Moreover, different wind angles will be considered.

6 Time planning

For the functionality and effectiveness of this thesis project, a time planning has been set, to picture the graduation calendar. In order to visualize it, we created a Gantt chart in Excel, for an analytical and clear representation (Fig.24).

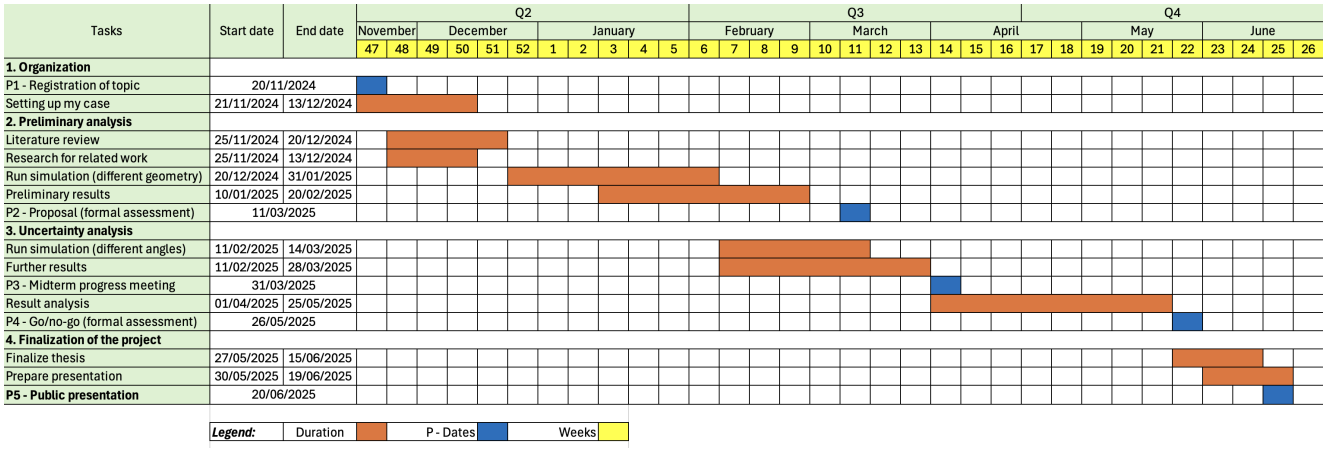


Figure 24: Time Planning

Moreover, the GitHub link is provided in order to gain access to all the simulation results, datasets, case setup, Python scripts and files related to this case study:

https://github.com/tudelft3d-theses/2023-Chontos

References

B. Blocken. 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, 2015. doi: 10.1016/j.buildenv.2015.02.015. URL https://doi.org/10.1016/j.buildenv.2015.02.015.

B. Blocken and T. Stathopoulos. Cfd simulation of pedestrian-level wind conditions around buildings: Past achievements and prospects. *Journal of Wind Engineering and Industrial Aerodynamics*, 121:138–145, 2013. doi: 10.1016/j.jweia.2013.08.008. URL https://doi.org/10.1016/j.jweia.2013.08.008.

I. Celik, U. Ghia, P. Roache, C. Freitas, H. Coleman, and R. P.E. Procedure for estimation and reporting of uncertainty due to discretization in cfd applications. *Journal of Fluids Engineering*, 130(7):078001, 07 2008. ISSN 0098-2202. doi: 10.1115/1.2960953. URL https://doi.org/10.1115/1.2960953.

H. Fan, A. Zipf, Q. Fu, and P. Neis. Quality assessment for building footprints data on openstreetmap. *International Journal of Geographical Information Science*, 28(4):700–719, 2014. doi: 10.1080/13658816.2013.867495. URL https://doi.org/10.1080/13658816.2013.867495.

J. Flaherty, D. Stock, and B. Lamb. Computational fluid dynamic simulations of plume dispersion in urban oklahoma city. *Journal of Applied Meteorology and Climatology*, 46(12):2110–2126, 2007. doi: 10.1175/2006jamc1306.1. URL https://doi.org/10.1504/ijep.2000.000526.

J. Franke, C. Hirsch, G. Jensen, H. Krüs, S. Miles, M. Schatzmann, P. Westbury, J. Wisse, and N. Wright. Recommendations on the use of cfd in wind engineering, 2004. URL https://www.researchgate.net/publication/251814717_Recommendations_on_the_use_of_CFD_in_wind_engineering. Published as part of COST Action C14, Impact of Wind and Storms on City Life and Built Environment,Von Karman Institute, Sint-Genesius-Rode, Belgium.

J. Franke, A. Hellsten, H. Schlünzen, and B. Carissimo. BEST PRACTICE GUIDELINE FOR THE CFD SIMULATION OF FLOWS IN THE URBAN ENVIRONMENT. Technical report, COST European Cooperation in Science and Technology, May 2007. URL https://hal.science/hal-04181390.

J. Franke, M. Sturm, and C. Kalmbach. Validation of openfoam 1.6.x with the german vdi guideline for obstacle resolving micro-scale models. *Journal of Wind Engineering and Industrial Aerodynamics*, 104-106:350–359, 2012. ISSN 0167-6105. doi: https://doi.org/10.1016/j.jweia.2012.02.021. URL https://www.sciencedirect.com/science/article/pii/S0167610512000451. 13th International Conference on Wind Engineering.

C. García-Sánchez, D. Philips, and C. Gorlé. Quantifying inflow uncertainties for cfd simulations of the flow in downtown oklahoma city. *Building and Environment*, 78:118–129, 2014. doi: 10.1016/j.buildenv.2014.04.013. URL https://doi.org/10.1016/j.buildenv.2014.04.013.

C. García-Sánchez, G. Van Tendeloo, and C. Gorlé. Quantifying inflow uncertainties in rans simulations of urban pollutant dispersion. *Atmospheric Environment*, 161:263–273, 2017. doi: 10.1016/j.atmosenv.2017.04.019. URL https://doi.org/10.1016/j.atmosenv.2017.04.019.

T.-O. Hågbo, K. Giljarhus, and B. Hjertager. Influence of geometry acquisition method on pedestrian wind simulations. *Journal of Wind Engineering and Industrial Aerodynamics*, 215: 104665, 2021. doi: 10.1016/j.jweia.2021.104665. URL https://doi.org/10.1016/j.jweia.2021.104665.

L. Lee, T. Humphreys, and S. Chan. Cfd simulations of joint urban atmosphere dispersion field study 2003. Technical report, U.S. Department of Energy, OSTI (Office of Scientific and Technical Information), 2004. URL https://www.osti.gov/biblio/15014068. [Accessed 5 Mar. 2025].

R. Longo, M. Ferrarotti, C. Sánchez, M. Derudi, and A. Parente. Advanced turbulence models and boundary conditions for flows around different configurations of ground-mounted buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, 167:160–182, 2017. doi: 10.1016/j.jweia.2017.04.015. URL https://doi.org/10.1016/j.jweia.2017.04.015.

Y. Nonomura, N. Kobayashi, Y. Tominaga, and A. Mochida. The cross comparison of cfd prediction for flow field around building blocks(part 3). *Summaries to Technical Papers of Annual Meeting, Japan Association for Wind Engineering*, 2003:41–41, 2003. doi: 10.14887/jaweam.2003.0.41.0. URL https://doi.org/10.14887/jaweam.2003.0.41.0.

OpenFOAM.com. Openfoam: User guide: atmboundarylayer. Online. URL https://www.openfoam.com/documentation/guides/latest/doc/guide-bcs-inlet-atm-atmBoundaryLayer.html. [Last accessed 31 Jan. 2025].

I. Pađen, C. García-Sánchez, and H. Ledoux. Towards automatic reconstruction of 3d city models tailored for urban flow simulations. *Frontiers in Built Environment*, 8, 2022. doi: 10.3389/fbuil.2022.899332. URL https://doi.org/10.3389/fbuil.2022.899332.

A. Ricci, I. Kalkman, B. Blocken, M. Burlando, A. Freda, and M. Repetto. Local-scale forcing effects on wind flows in an urban environment: Impact of geometrical simplifications. *Journal of Wind Engineering and Industrial Aerodynamics*, 170:238–255, 2017. doi: 10.1016/j.jweia.2017.08.001. URL https://doi.org/10.1016/j.jweia.2017.08.001.

P. Richards and R. Hoxey. Appropriate boundary conditions for computational wind engineering models using the k- turbulence model. *Journal of Wind Engineering and Industrial Aerodynamics*, 46-47:145–153, 1993. doi: 10.1016/0167-6105(93)90124-7. URL https://doi.org/10.1016/0167-6105(93)90124-7.

P. Roache. Perspective: A method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering*, 116(3):405–413, 1994. doi: 10.1115/1.2910291. URL https://doi.org/10.1115/1.2910291.

A. Robins, R. Hall, I. Cowan, J. Bartzis, and A. Albergel. Evaluating modelling uncertainty in cfd predictions of building affected dispersion. *International Journal of Environment and Pollution*, 14(1/2/3/4/5/6):52, 2000. doi: 10.1504/ijep.2000.000526. URL https://doi.org/10.1504/ijep.2000.000526.

Y. Tominaga and B. Blocken. Wind tunnel experiments on cross-ventilation flow of a generic building with contaminant dispersion in unsheltered and sheltered conditions. *Building and Environment*, 92:452–461, 2015. doi: 10.1016/j.buildenv.2015.05.026. URL https://doi.org/10.1016/j.buildenv.2015.05.026.

Y. Tominaga, R. Yoshie, A. Mochida, H. Kataoka, and T. Nozu. Cross comparisons of cfd prediction for wind environment at pedestrian level around buildings part 2: Comparison

of results for flowfield around building complex in actual urban area. Online, 2005. URL https://www.researchgate.net/publication/240624782. [Last accessed 31 Jan. 2025].

Y. Tominaga, A. Mochida, R. Yoshie, H. Kataoka, T. Nozu, M. Yoshikawa, and T. Shirasawa. Aij guidelines for practical applications of cfd to pedestrian wind environment around buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(10-11):1749–1761, 2008. doi: 10.1016/j.jweia.2008.02.058. URL https://doi.org/10.1016/j.jweia.2008.02.058.

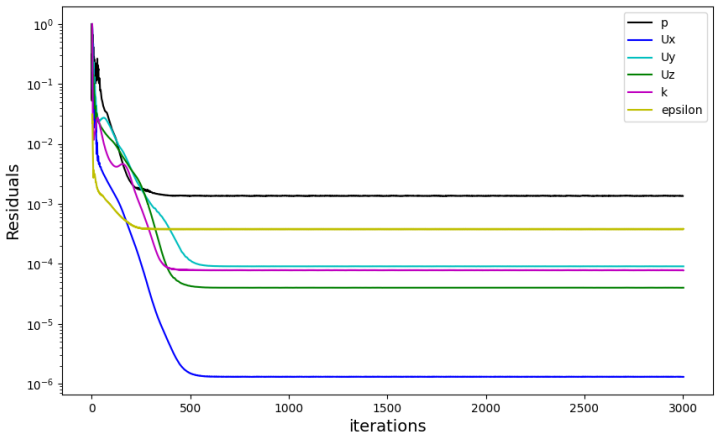
Uni-hamburg.de. data sets. Online, 2024. URL https://www.mi.uni-hamburg.de/en/arbeitsgruppen/windkanallabor/data-sets.html. [Accessed 4 Mar. 2025].

J. Wieringa. Updating the davenport roughness classification. *Journal of Wind Engineering and Industrial Aerodynamics*, 41(1):357–368, 1992. doi: 10.1016/0167-6105(92)90434-C. URL https://doi.org/10.1016/0167-6105(92)90434-C.

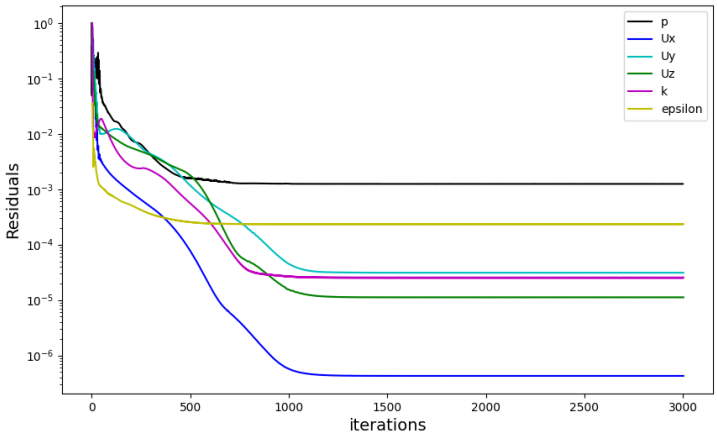
www.aij.or.jp. Guidebook for cfd predictions of urban wind environment. Online. URL https://www.aij.or.jp/jpn/publish/cfdguide/index_e.htm. [Last accessed 31 Jan. 2025].

R. Yoshie, A. Mochida, Y. Tominaga, H. Kataoka, K. Harimoto, T. Nozu, and T. Shirasawa. Cooperative project for cfd prediction of pedestrian wind environment in the architectural institute of japan. *Journal of Wind Engineering and Industrial Aerodynamics*, 95(9-11):1551–1578, 2007. doi: 10.1016/j.jweia.2007.02.023. URL https://doi.org/10.1016/j.jweia.2007.02.023.

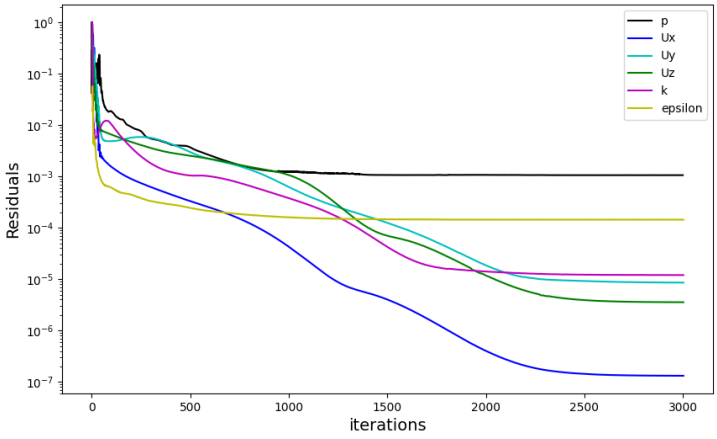
7 Appendix



(a) Coarse Mesh

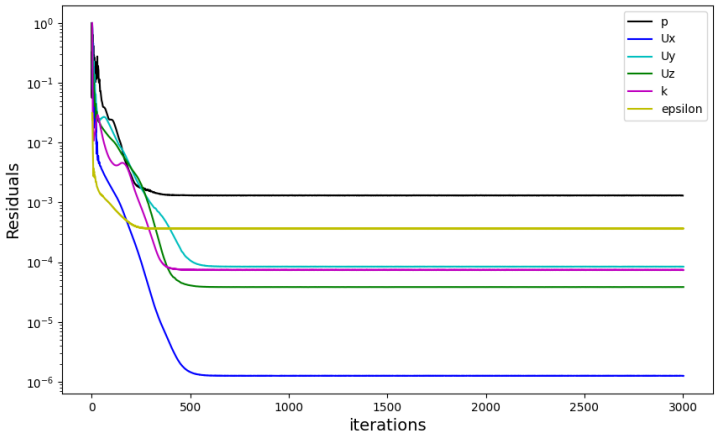


(b) Medium Mesh

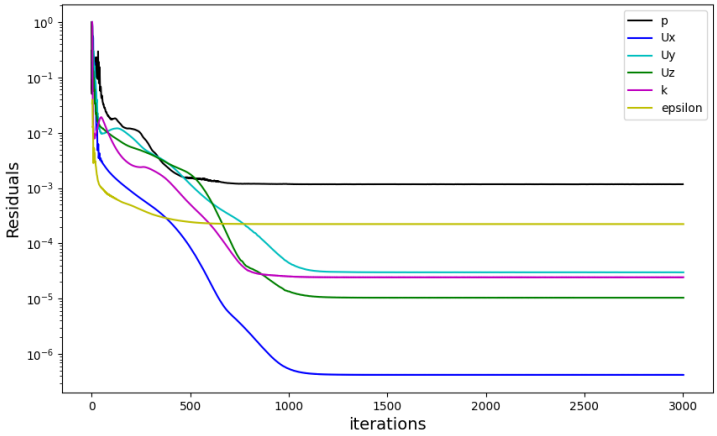


(c) Fine Mesh

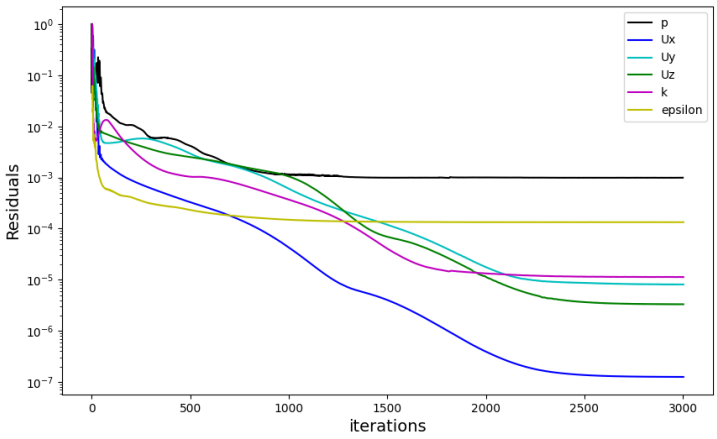
Figure 25: Residuals: Case 0H



(a) Coarse Mesh

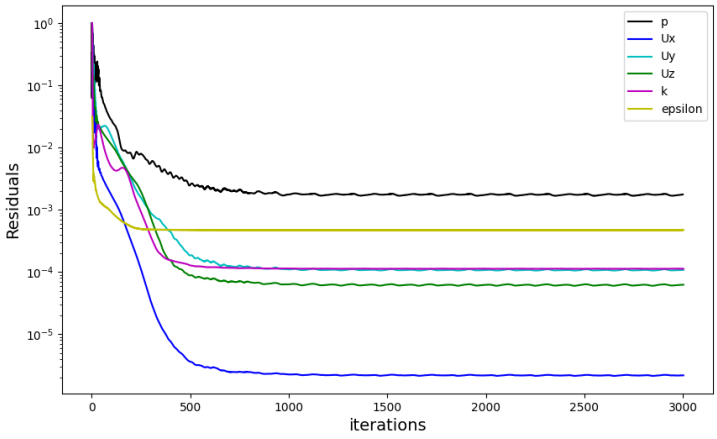


(b) Medium Mesh

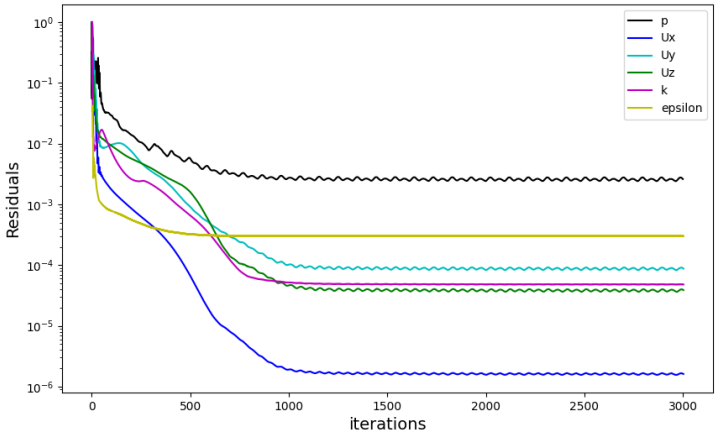


(c) Fine Mesh

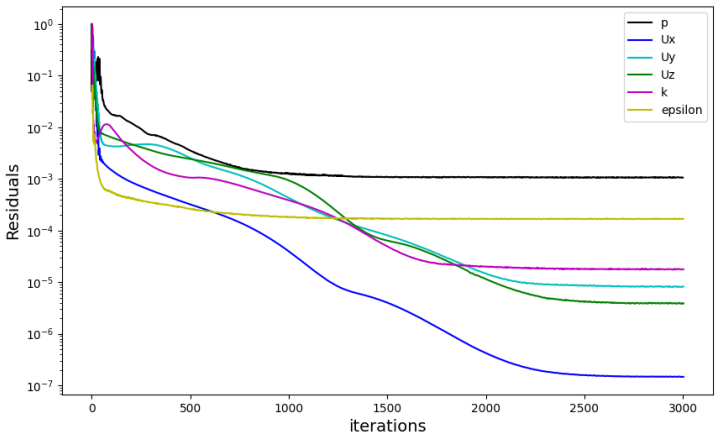
Figure 26: Residuals: Case 1H



(a) Coarse Mesh

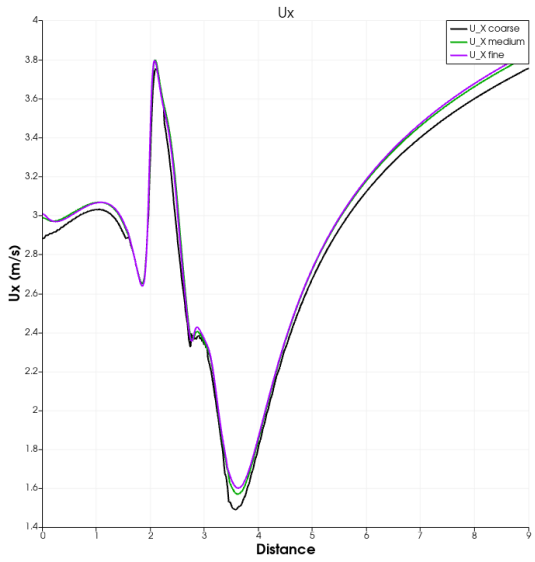


(b) Medium Mesh

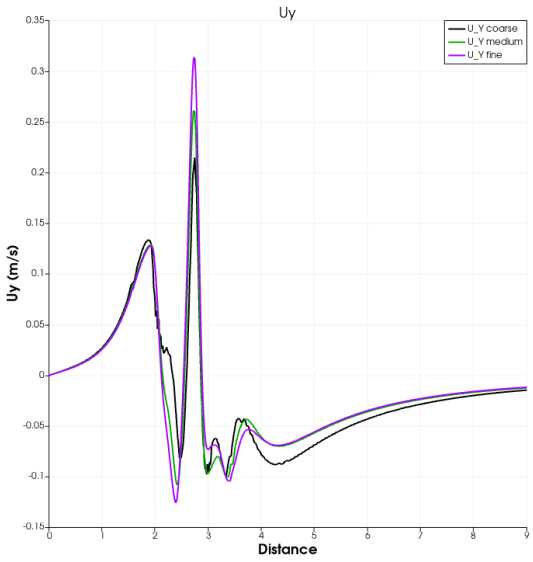


(c) Fine Mesh

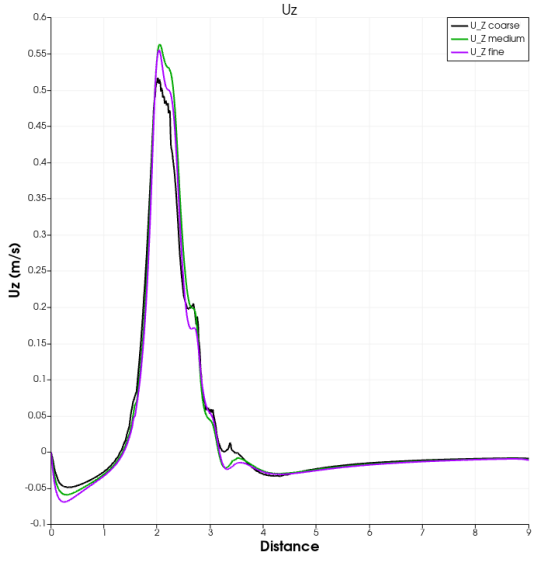
Figure 27: Residuals: Case 2H



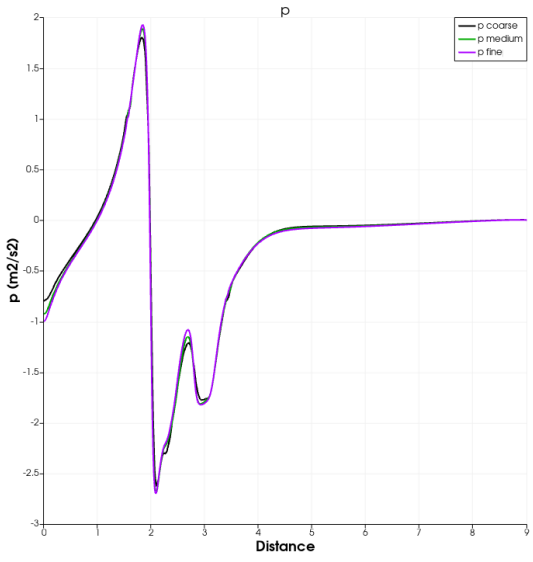
(a) Ux plot for the three meshes



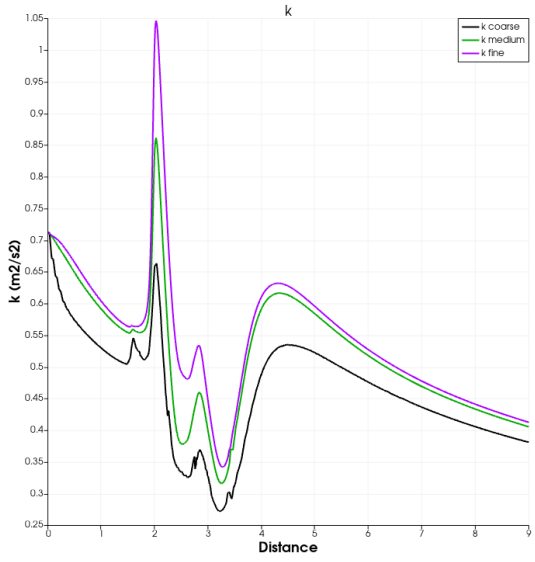
(b) Uy plot for the three meshes



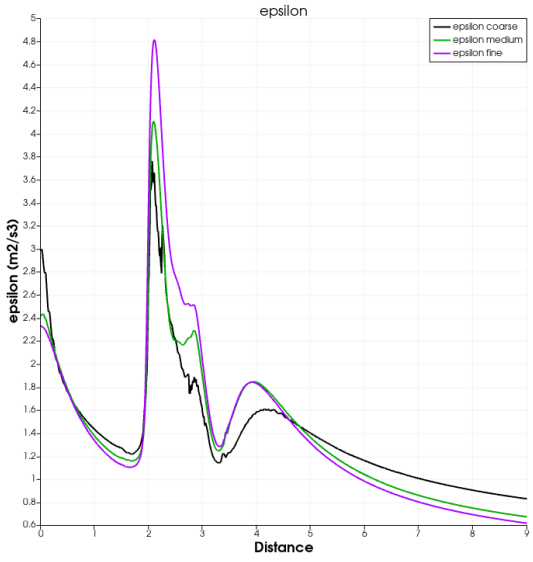
(c) Uz plot for the three meshes



(d) P plot for the three meshes

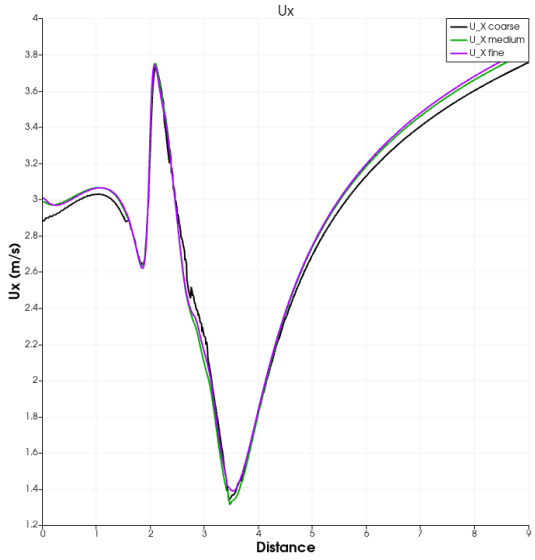


(e) K plot for the three meshes

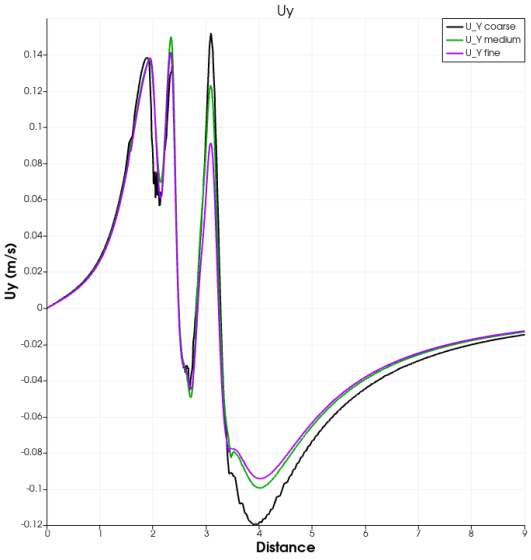


(f) ϵ plot for the three meshes

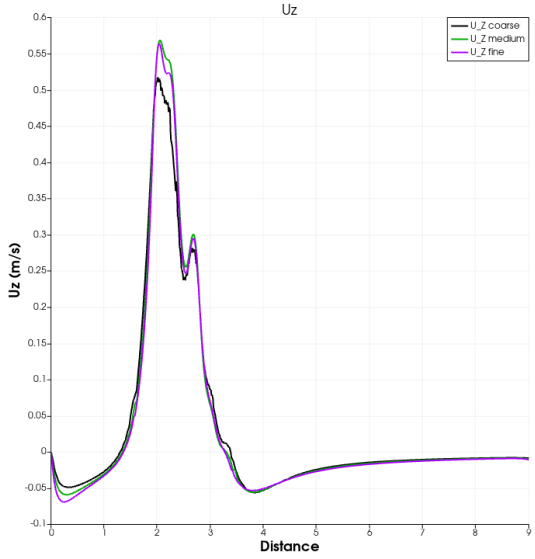
Figure 28: Field Plots: Case 0H



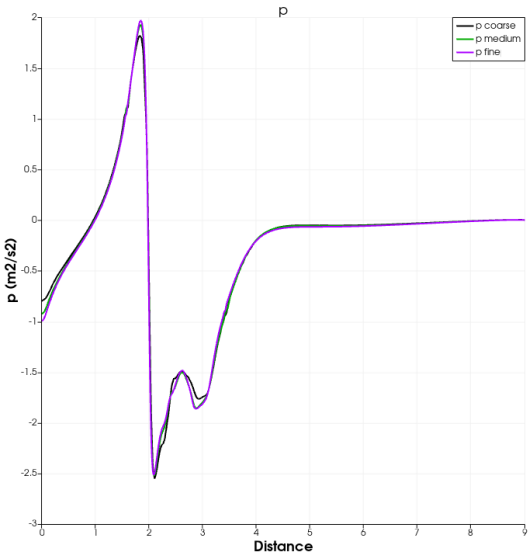
(a) Ux plot for the three meshes



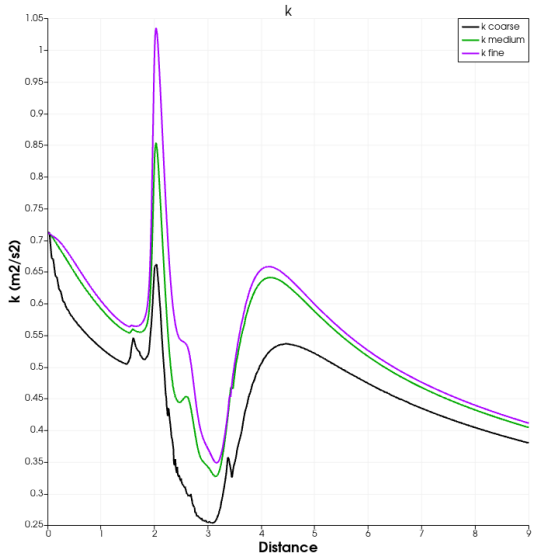
(b) Uy plot for the three meshes



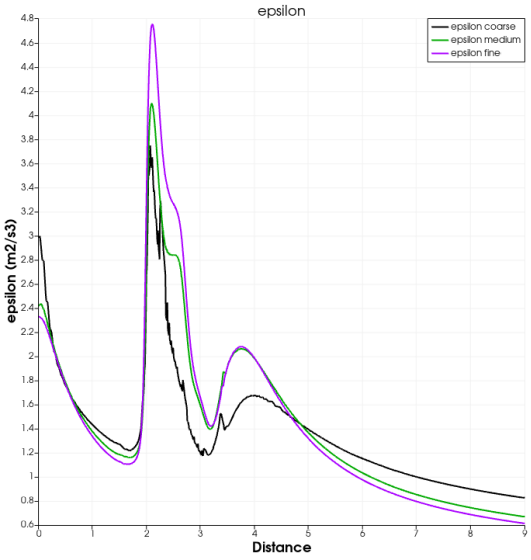
(c) Uz plot for the three meshes



(d) P plot for the three meshes

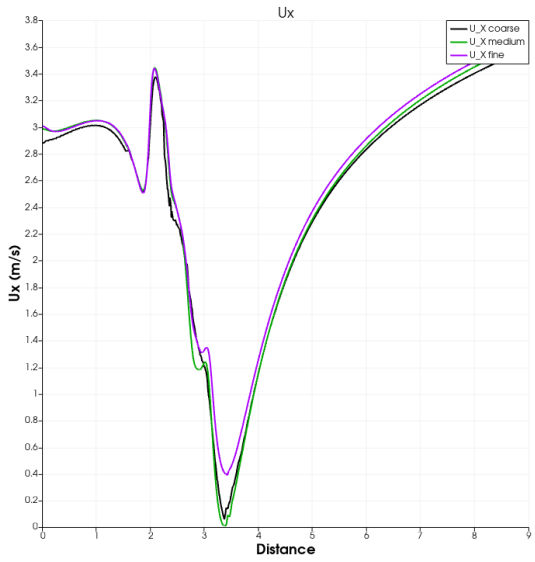


(e) K plot for the three meshes

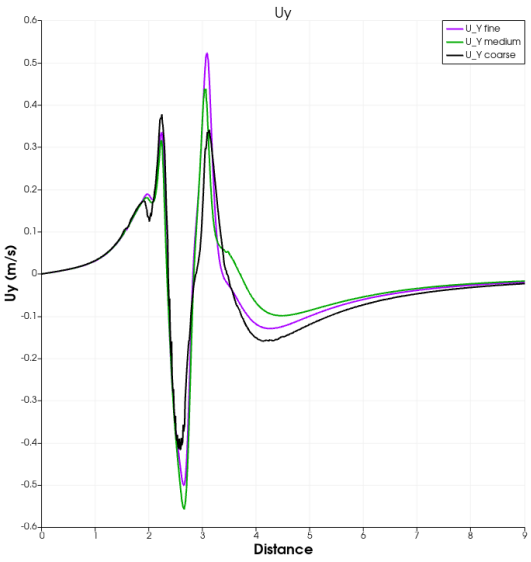


(f) ϵ plot for the three meshes

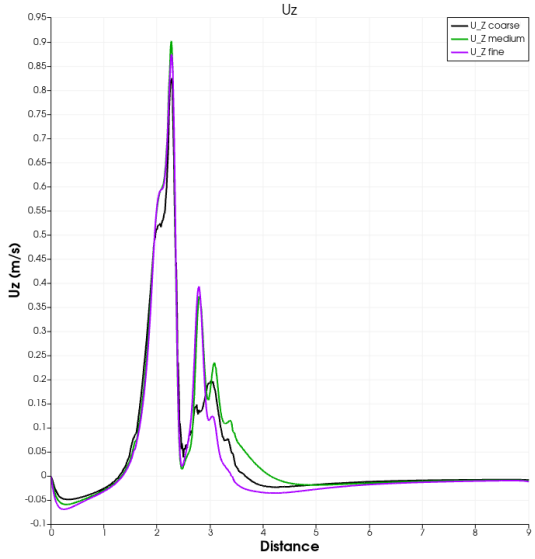
Figure 29: Field Plots: Case 1H



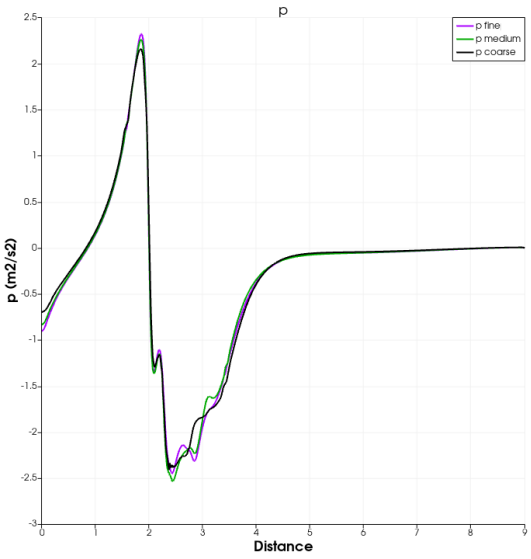
(a) Ux plot for the three meshes



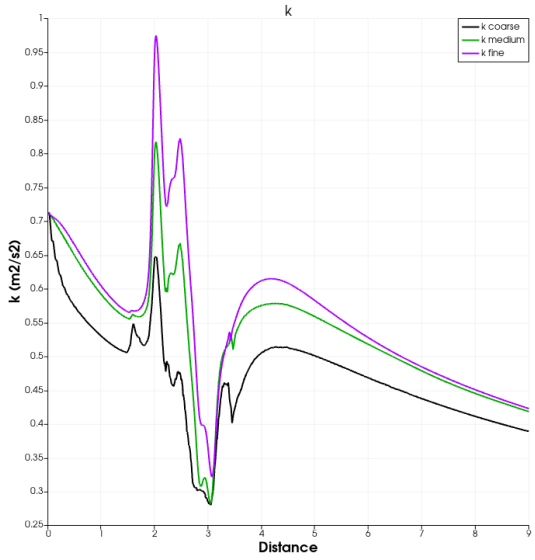
(b) Uy plot for the three meshes



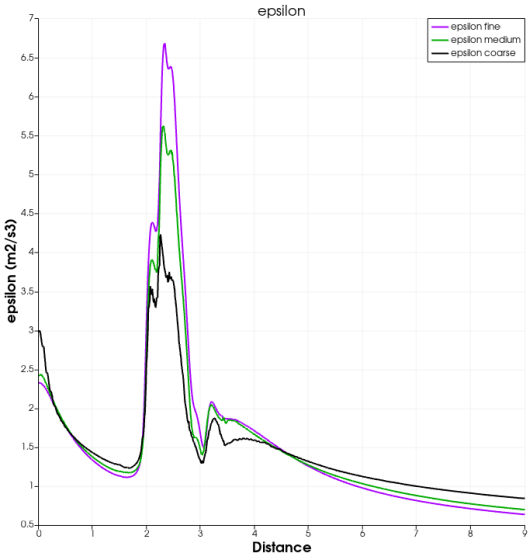
(c) Uz plot for the three meshes



(d) P plot for the three meshes



(e) K plot for the three meshes



(f) ϵ plot for the three meshes

Figure 30: Field Plots: Case 2H