Graduation plan

Urban computational fluid dynamics simulations set-up validations

1st supervisor: Clara García-Sánchez
2nd supervisor: Hugo Ledoux
January, 2023

Maren Hengelmolen
4294068

Contents

1 Introduction **3**

2 Related work **4**
- 2.1 Geometric validation 4
- 2.2 Computational Fluid Dynamics (CFD) 5
  - 2.2.1 Urban physics 5
  - 2.2.2 Computational domain 5
  - 2.2.3 Computational grid 6
  - 2.2.4 Region of interest 7
  - 2.2.5 CFD pre-run setups automation 7
  - 2.2.6 CFD simulation software: OpenFoam 7

3 Problem definition **9**
- 3.1 Research question 9
- 3.2 Methodology 9
  - 3.2.1 Approach 9
  - 3.2.2 Research methodology 11
- 3.3 Scope 12
- 3.4 Tools and datasets 13

4 Time planning **14**

Bibliography **15**

1 Introduction

Flows (e.g. wind, heat, pollution) have an impact on the living comfort in urban areas. Extreme weather conditions, such as strong winds and high temperatures, are becoming more common due to climate change, which requires well-constructed buildings and infrastructure that stimulate natural ventilation and can withstand these extreme events (UNEP, 2021). All the while the urban environment keeps on expanding as a cause of population growth. Currently, 56% of the world's population lives in urban area and this is predicted to reach 70% by 2050 (The World Bank, 2022). Issues related to flows in urban areas are thus likely to increase. Flow analysis of 3D urban models is essential to anticipate and solve these upcoming challenges. Computational Fluid Dynamics (CFD) simulation is an important tool for performing such an analysis.

CFD simulations use mathematics, physics and computer science to simulate and visualize fluid flows in many applications, including urban areas. The quality of the geometries within 3D urban models is crucial to perform such simulations (Wagner et al., 2015). Given the increasing use of geometric data for spatial analyses, international standards for geographical information were developed. The standards help to create valid 3D geometric datasets and encourage the interoperability and exchange of geographical data. Validating 3D urban models according to these standards is an important step in acquiring accurate simulation results. In addition, since complex geometries can degrade meshes which are essential for CFD simulations, they should be avoided (Park et al., 2020).

Guidelines were also created for CFD simulations. The guidelines address pre-run setups that highly influence the simulation results (e.g. the computational domain must be large enough to avoid artificial accelerations). Currently, users must manually define in their configuration files their pre-run setups. However, partial automation and automated comparison/control with the CFD guidelines can streamline this process, making it not only faster to acquire a usable model, but also a lot easier, since some expertise is required to interpret these guidelines.

The aim of this thesis is to develop a methodology to compute and verify CFD pre-run setups and validate 3D urban models according to the most recent CFD guidelines and international standards for geographic information, and create a prototype (e.g. in the form of a web application) that executes this method and could be converted to an open source tool.

2 Related work

2.1 Geometric validation

3D urban models, also called 3D city models, contain geographical information in the form of vector data, including geometric primitives (e.g. points, surfaces, solids), and semantics. Buildings are often represented by solids and terrain by surfaces. Many definitions for solids exist, which raises confusion when constructing and using geometric data. To increase the interoperability and exchange of geographical data, the International Organization for Standardization (ISO) and the Open Geospatial Consortium (OGC) developed standards to define basic primitives (ISO19107) (ISO, 2019) and how to digitize them (OGC, 2016, 2011).

The ISO19107 defined geometric primitives from 0D to 3D. It states that a *d*-dimensional primitive is composed of *(d-1)*-dimensional primitives and can be part of another *d*-dimensional primitive to form an aggregate or composite (Arroyo Ohori et al., 2022). As stated before, OGC developed standards to model the geometric primitives according to the ISO19107. However, the standards are often interpreted differently and are not always easy to implement, which leads to invalid geometries (van Oosterom et al., 2005). Most errors seem to be avoidable during the modeling of the 3D urban models (Biljecki et al., 2016).

Since valid datasets provide better computational grids needed for CFD simulations, it is important to validate the geometries. In contrast to 3D datasets, well-defined rules exist for 2D datasets and open source tools are available to perform geometric validations. The most known are JTS Topology Suite and GEOS (Ledoux, 2013). There are also some rules and methods developed to store valid 3D datasets, such as for 3D cadastral data (Shojaei et al., 2017) (Karki et al., 2010) or 3D urban models in general (Bogdahn and Coors, 2010) (Wagner et al., 2015) (Ellul et al., 2013). Several software tools exist that validates 3D datasets (e.g. ArcGIS Pro, Oracle Spatial, and CityDoctor). Yet, according to Ledoux (2018), they do no respect all the definitions within ISO19107 and/or do not support aggregates and composites. Ledoux (2018) developed open source software, named val3dity, that validates 3D primitives according to ISO19107, with the common exception that they need to be linear or planar. These restrictions are also implemented in CityGML, the international standard for 3D modelling of cities developed by OGC (OGC, 2021). The topological relationship is only verified between BuildingParts (primitives forming buildings), meaning that floating buildings and shared faces between buildings are considered as valid for example (Ledoux, 2018). This is an issue while running CFD simulations (e.g. simulated non-existent flows under buildings). One of the objectives of this thesis is to develop tools to perform these missing validations.

For some CFD simulation software, shared vertices, edges and surfaces between buildings should be removed to obtain high quality computational grids. Therias et al. (2022) worked on removing shared faces between buildings in cooperation with Dassault Systèmes. They identified adjacent buildings by adapting the 3D Building Metrics code (3DBM) developed by Labetski et al. (2023), and developed two methods to delete the shared faces. One removes shared faces and creates new faces for the parts that were not intersecting, and the other one uses Nef Polyhedra. During this thesis, the open source CFD software OpenFoam (see Section 2.2.6) will be used to test the developed methodology. In contrast to the software of Dassault Systèmes and many other software, OpenFoam does not seem to encounter problems with shared vertices, edges and surfaces when creating the computational grid (Pađen et al., 2022).

Complex geometries can also affect the quality of the computational grids (Park et al., 2020). To simply these geometries, sharp angles, short edges, small distances between buildings and sliver triangles should be deleted (Pađen et al., 2022). There are some methods developed to automatically simplify these geometries. Piepereit et al. (2018) introduced a sweep-plane algorithm that eliminates edges shorter than a given threshold. Park et al. (2020) proposed a method that simplify geometries using angle and distance thresholds. The detection of these geometries will also be addressed during this thesis.

2.2 Computational Fluid Dynamics (CFD)

2.2.1 Urban physics

Most analyses in urban physics are performed in the lower part of the Atmospheric Boundary Layer (ABL), which is the air layer closest to the surface of the Earth. The ABL heights depends on the surface (e.g. frictional drag or heat transfer from surface) and can vary from several tens metres (stable stratification) to kilometres (unstable stratification or convection) (Blocken, 2015). Within the ABL, weather phenomena occur that can be classified according to their spatial and temporal scale (AMS, nd)(Habby, nd):

- Macroscale: many hundreds of kilometres, days (e.g. cyclones, hurricanes)
- Mesoscale: few to several hundred kilometres, 1 to 24 hours (e.g. thunderstorms, precipitation bands, mountain waves, sea and land breezes)
- Microscale: 2km or less, minutes to hours (e.g. tornadoes, rainbows, convective updrafts and downdrafts)

The flows in urban areas are approximated by the Navier-Stokes (NS) equations, which cover the conservation of mass, momentum, and energy. Depending on the situation, different approaches can be implemented (Blocken, 2015):

- Steady Reynolds-averaged Navier-Stokes equations (RANS): time averaging of the NS equations
- Unsteady RANS (URANS): ensemble-averaging of the NS equations
- Large Eddy Simulation (LES): small turbulent eddies removal by filtering NS equations

While the flow in the ABL is turbulent and an unsteady method should be used, steady RANS remains the most widely used method in urban physics and seems to obtain satisfying results in many fluid analyses, such as natural ventilation, wind energy and air pollution (Blocken, 2015) (Blocken, 2014). LES is the most popular unsteady method, due to the high spatial resolution needed for URANS, and is increasingly used (Franke and Baklanov, 2007). However, this method remains computationally expensive and the scarce availability of information and guidelines to help the users with the implementation remains a drawback (Blocken, 2015).

2.2.2 Computational domain

To avoid strong artificial accelerations, the size of the computational domain representing the ABL must be large enough. Related guidelines are the ones of Franke and Baklanov (2007) and Tominaga et al. (2008). Both make use of the blockage ratio (BR), which is the ratio between the front area of the urban area or building and the one of the domain. Blocken (2015) recommends adding the directional BR, which decomposes the BR in the lateral and vertical direction. The main guidelines for urban areas are summarised in the following bullet points.

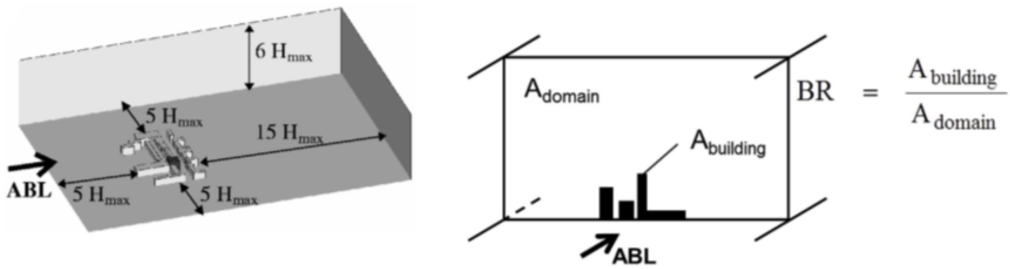

Figure 1: *Note*. Domain boundaries according to Franke and Baklanov (2007). From "Computational Fluid Dynamics for urban physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations" by B. Blocken, 2015, *Science direct*, 91:219-245e

Franke and Baklanov (2007):

- Domain boundaries: top H_{max}, lateral $5H_{max}$, outlet $15H_{max}$, inflow $5H_{max}$ (H_{max}: tallest building height). For the the outlet and inflow boundaries, smaller distances can be used.
- BR $\leq$ 3%

Tominaga et al. (2008):

- Domain boundaries: lateral $5H_{max}$ from target building, top terrain category of the surrounding, upwind area covered by a smooth floor in the wind tunnel from building, outflow $10H_{max}$ (H_{max}: tallest building height). For the inlet and outflow boundaries, these guidelines are not specifically mentioned for urban areas, but are valid for single buildings.
- BR $<$ 3%

Blocken (2015):

- BRL = $L_{building}/L_{domain} \leq 17\%$
- BRH = $H_{building}/H_{domain} \leq 17\%$

Rectangular, round and oval computational domains can be computed with these guidelines (Pađen et al., 2022).

2.2.3 Computational grid

To obtain accurate results (no discretization errors and convergence problems), it is essential to generate high-quality computational grids with sufficient resolution and well-defined cells (Blocken, 2015). Franke and Baklanov (2007) and Tominaga et al. (2008) developed guidelines addressing the computational grid for buildings and urban areas. In the following bullet points, their main guidelines are listed.

- Franke and Baklanov (2007): 10 cells per cube root of the building volume, 10 cells between buildings, evaluation height (1.5-2m) located at the 3rd or 4th grid cells from the ground, number of systematically refined grids $\geq$ 3, linear refinement factor (combination of all three direction) $\geq$ 3.4, stretching ratio $\leq$ 1.3 (especially in regions with a high velocity gradient), hexahedra cells are preferred over tetrahedral cells, and grid lines perpendicular to wall (no tetrahedral cells at walls)

- Tominaga et al. (2008): 10 grid cells at each side of the buildings, evaluation height (1.5-5m) located at the 3rd or higher grid cell from the ground, linear refinement factor (combination of all three direction) $>$ approx. 3.375 (nr. fine meshes $>$ 1.5 nr. coarse meshes), stretching ratio ≤ 1.3 (especially in regions with a high velocity gradient), and minimum grid resolution approx. 1/10 of the building scale (about 0.5-5m)

2.2.4 Region of interest

Guidelines were developed to define the region of interest (RoI), which is very useful given that modelling all buildings in the computational domain is expensive (Pađen et al., 2022) and the level of details of buildings can lead to diverse wind flows (García-Sánchez et al., 2021). Tominaga et al. (2008) proposed to clearly model buildings within a radius of 1 or 2 times the height of the target building and an additional street block in each direction. Tong et al. (2016) recommended to model explicitly 2 or 3 layers of surroundings depended on the street canyons and wind directions. Liu et al. (2018) suggests using detailed building structures around the target building within a radius of at least three times the largest building dimension. According to Pađen et al. (2022), the third one is the most suitable for automation.

2.2.5 CFD pre-run setups automation

Pađen et al. (2022) developed City4CFD (https://github.com/tudelft3d/City4CFD), a tool that automatically reconstruct 3D city geometries for microscale urban flow simulations from 2D geographical datasets (e.g. cadastral data, topographic datasets) and aerial pointcloud-based elevation data. This tool is able to generate terrains, buildings, surface layers, domain buffers, influence regions and domain boundaries. The two last features are needed to construct the mesh for the CFD simulations and are in line with guidelines stated above. They tested the tool by performing CFD simulations with OpenFoam and obtained satisfying results. The thesis will also focus on features required for the meshing process.

2.2.6 CFD simulation software: OpenFoam

This thesis focuses on a methodology that prepares 3D urban models for CFD simulations until the meshing process in OpenFoam, an open source CFD software. The following commands are used (OpenFoam, nd):

- *surfaceFeatures*: identifies points and edges in a surface geometry and writes them to a file (.eMesh). Running this command helps to recognize sharp edges, which results to a better mesh (snappyHexMesh).
- *blockMesh*: creates the background mesh, which means that the domain is decomposed in one or more three dimensional hexahedral blocks.
- *snappyHexMesh*: creates the final mesh (takes into account the geometry), containing hexahedral and split-hexahedral blocks.

Figure 2 illustrates the meshing process in OpenFoam. After running these commands, the CFD simulation can start with one of the available solvers. This is out of scope.

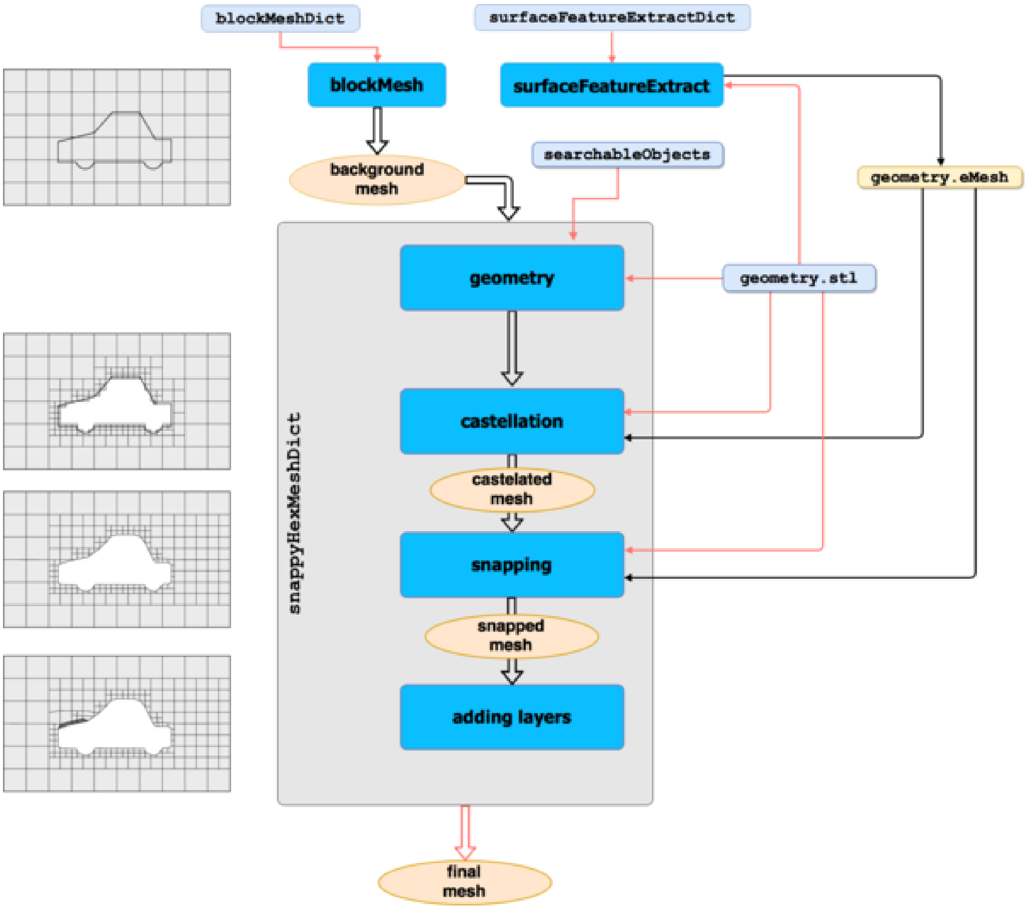

Figure 2: *Note.* Mesh generation in OpenFoam. From *snappyHexMesh*, by OpenFoam, n.d. (https://www.openfoam.com/documentation/guides/latest/doc/guide-meshing-snappyhexmesh.html)

3 Problem definition

3.1 Research question

The main question of this thesis is: *which methodology can validate geometries and identify CFD pre-run setups of 3D urban models to simplify the use of CFD simulations for urban areas?*

The following sub-questions are addressed to provide an exploration to the main research question:

- Which method(s) can validate 3D city models according to international standards for geographical information, verify the topological relationships between building objects and terrain surfaces, and detect complex geometries causing problems in creating computational grids?
- Which methods can compute and verify CFD pre-run setups to generate high quality meshes according to the most recent CFD guidelines?
- Can CFD guidelines be formalized and translated to axioms and code?
- How to report errors, warnings and results to the users?
- How to validate the quality of the methodology?

3.2 Methodology

3.2.1 Approach

User perspective
Figure 3 illustrates the concept of the prototype (e.g. web application). The graphical user interface enables users to upload their 3D model (CityJSON, OBJ, PLY, or STL) and complete their CFD pre-run setups. After filling in the input data, the validation and computation processes start and return the errors, warnings and calculated CFD setups.

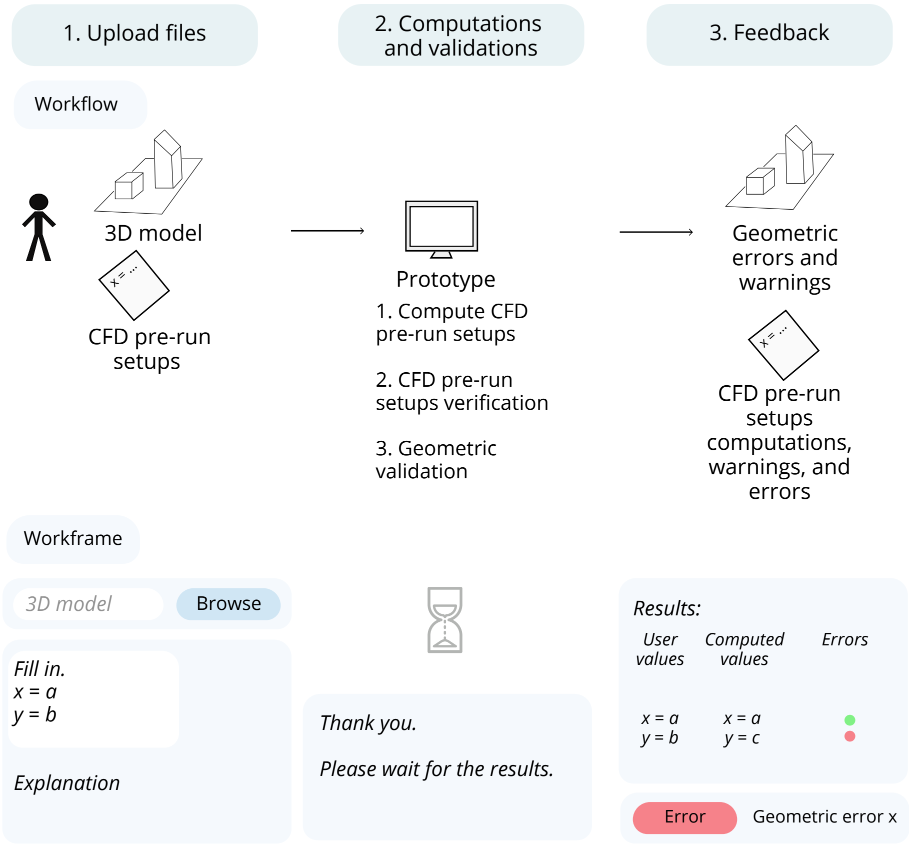

Figure 3: User perspective

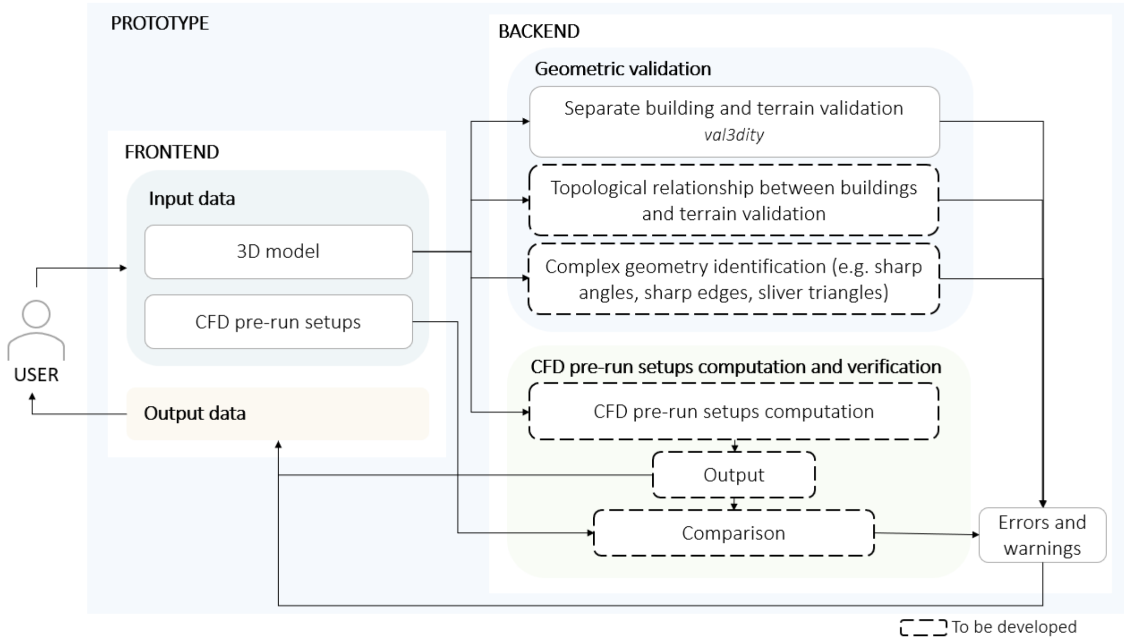

Figure 4: Prototype architecture. A part of the prototype is already built. The parts indicated in dashed lines are yet to be designed and implemented.

Architecture

The main components of the prototype are defined as follows:

- **Input data**: as stated in the user perspective, the users can import their 3D urban model and CFD pre-run setups and start the validation process.
- **Geometric validation**: the geometries within the 3D urban model are verified according to the ISO19107, with the common exception that they need to be linear or planar. The building objects and terrain surfaces are separately validated by using the open source software val3dity. As stated in Section 2.1, val3dity does not verify the topological relation between building solids and terrain surfaces, which means that floating buildings and buildings that are intersected by terrain surfaces are considered to be valid. Also, complex geometries should be simplified. Both issues can lead to inappropriate meshing and inaccurate and/or incorrect CFD results, and should be identified.
- **CFD pre-run setups computation and verification**: the 3D urban model is used to compute the CFD pre-run setups required for the meshing process. The relevant setups are domain size, computational grid and influence region. The resulting setups are compared with the ones of the users.
- **Output data**: the errors from the geometric and CFD pre-run setups validations and the results of the CFD pre-run setups computations are retrieved and reported to the user. They are classified into errors and warnings, depending on their severity. Based on the feedback, the users can adapt their input values.

3.2.2 Research methodology

To reach the objectives set in Section 2.1, a methodology involving literature research, backend and frontend development, prototype validation, and termination is defined. This methodology is summarised in Figure 5.

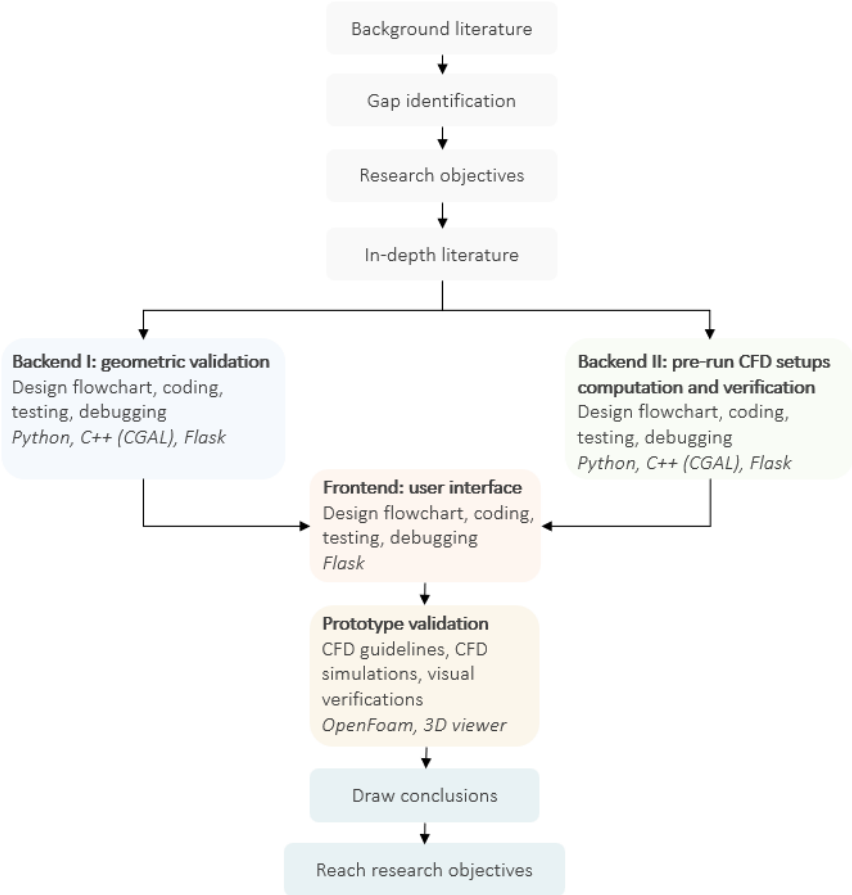

Figure 5: Research methodology

Now that the research objectives have been identified, a more in-depth literature study will follow which aims to design the two backend development blocks. Once a method is chosen based on found guidelines and articles, it will be designed and implemented in Python and C++. The next step will be to execute the frontend development block, to enable users to interact with the user interface in an intuitive way. Backend and frontend development are addressed in the previous Section 3.2.1. One of the last steps is to validate the prototype to ensure that it operates correctly. This will be done by running test cases and validate the results against guidelines and simulations.

3.3 Scope

The focus lies on identifying and reporting geometric errors and computing CFD pre-run setups for generating the mesh in OpenFoam for 3D urban models. The prototype focuses on urban 3D models intended for microscale (<2km) RANS simulations and should work with different formats such as CityJSON, OBJ, STL, and PLY. Semantic validation functions will not be integrated into the prototype. Different types of domains can be generated (rectangular, round, and oval) following the guidelines discussed in Section 2.2.2. This thesis focuses on rectangular domains, since they are common in CFD simulations in urban areas (Mirzaei and Carmeliet, 2013).

3.4 Tools and datasets

The tools and datasets needed for this thesis are listed below:

- **Source codes and web interface**: for the backend and frontend development (geometric validation, pre-run CFD setups, and user interface) code will be programmed in Python and C++. Tools from the C++ library CGAL (https://www.cgal.org/) will be used to perform the validations and verifications. Flask will be used to build the framework of the user interface.
- **val3dity**: the validity of the building objects and terrain surfaces will be tested with this tool (https://github.com/tudelft3d/val3dity).
- **Dataset**: an OBJ file, representing a part of the TU Delft campus with the EWI building, will be used as test file.
- **Prototype validation**: the computed CFD pre-run setups will be verified with the CFD guidelines, meshes will be generated and evaluated in OpenFOAM, and an 3D viewer will be used to verify some of the geometric errors.

4 Time planning

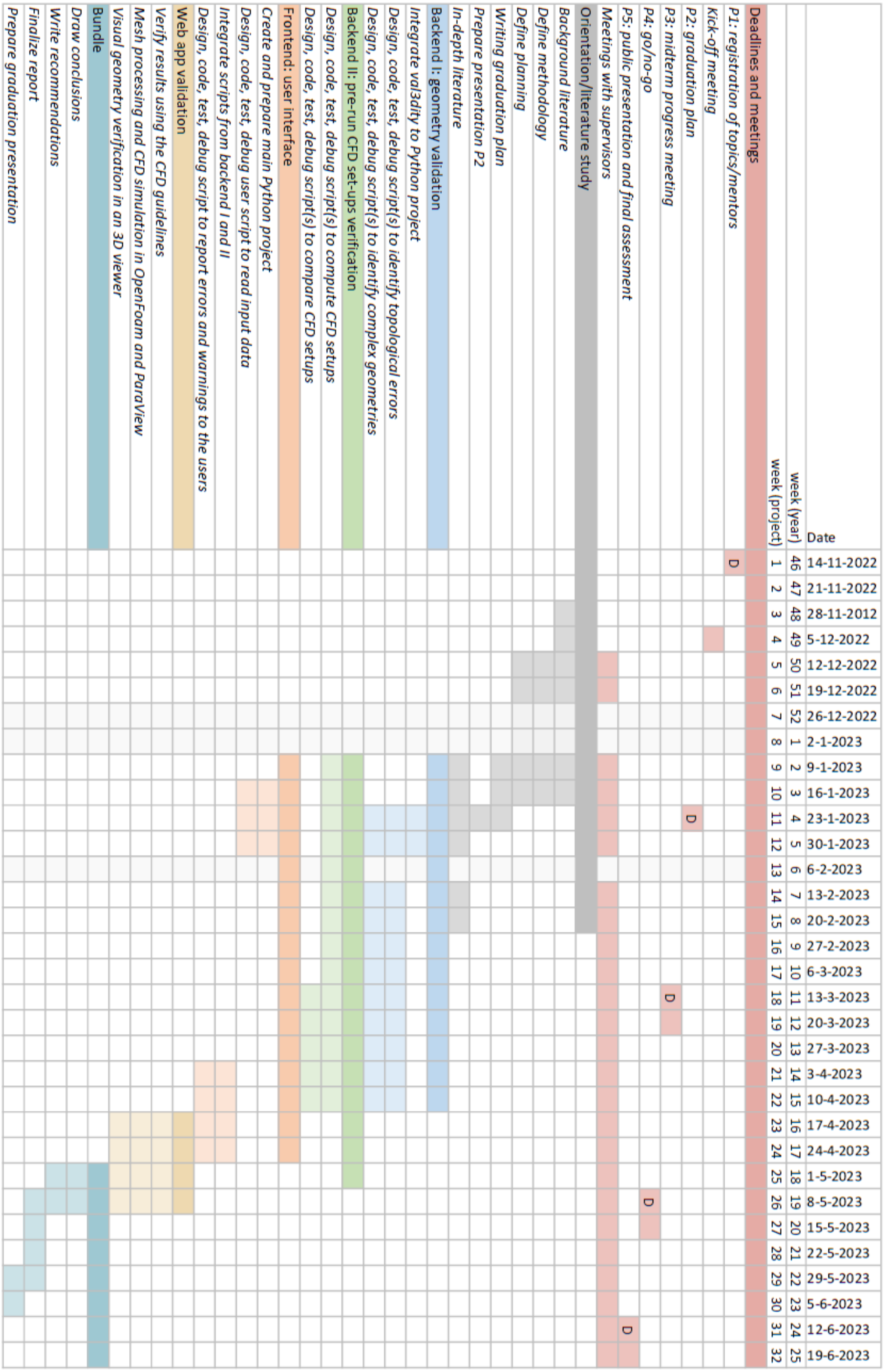

Figure 6: Workflow

Bibliography

AMS (n.d.). Glossary of meteorology. `https://glossary.ametsoc.org/wiki/Welcome`.

Arroyo Ohori, K., Ledoux, H., and Peters, R. (2022). *3D modelling of the built environment*, volume v0.8.

Biljecki, F., Ledoux, H., Du, X., Stoter, J., Soon, K. H., and Khoo, V. H. S. (2016). The most common geometric and semantic errors in CityGML datasets. IV-2/W1:13–22.

Blocken, B. (2014). 50 years of computational wind engineering: Past, present and future. 129:69–102.

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

Bogdahn, J. and Coors, V. (2010). Towards an automated healing of 3D urban models.

Ellul, C., Zlatanova, S., Rumor, M., and Laurini, R. (2013). Geometric validation of 3D city models based on standardized quality criteria. In *Urban and Regional Data Management*, pages 203–216. CRC Press, 0 edition.

Franke, J. and Baklanov, A. (2007). *Best Practice Guideline for the CFD Simulation of Flows in the Urban Environment: COST Action 732 Quality Assurance and Improvement of Microscale Meteorological Models*.

García-Sánchez, C., Vitalis, S., Pađen, 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.

Habby, J. (n.d.). Scales of motion. `https://www.theweatherprediction.com/habyhints3/733/`.

ISO (2019). ISO 19107:2019(en) geographic information — spatial schema.

Karki, S., Thompson, R., and McDougall, K. (2010). Data validation in 3D cadastre. In Neutens, T. and Maeyer, P., editors, *Developments in 3D Geo-Information Sciences*, pages 92–122. Springer Berlin Heidelberg, Berlin, Heidelberg.

Labetski, A., Vitalis, S., Biljecki, F., Arroyo Ohori, K., and Stoter, J. (2023). 3d building metrics for urban morphology. 37(1):36–67.

Ledoux, H. (2013). On the validation of solids represented with the international standards for geographic information: On the validation of solids represented with the international standards. 28(9):693–706.

Ledoux, H. (2018). val3dity: validation of 3d GIS primitives according to the international standards. 3(1):1.

Liu, S., Pan, W., Zhao, X., Zhang, H., Cheng, X., Long, Z., and Chen, Q. (2018). Influence of surrounding buildings on wind flow around a building predicted by CFD simulations. 140:1–10.

Mirzaei, P. A. and Carmeliet, J. (2013). Dynamical computational fluid dynamics modeling of the stochastic wind for application of urban studies. *Building and Environment*, 70:161–170.

OGC (2011). OpenGIS® implementation standard for geographic information - simple feature access - part 1: Common architecture. OGC 06-103r4, version 1.2.1.

OGC (2016). OpenGIS® geography markup language (GML) encoding standard. OGC 07-036r1, version 3.2.2.

OGC (2021). OGC city geography markup language (CityGML) part 1: Conceptual model standard. OGC 20-010, version 3.0.

OpenFoam. OpenFOAM: User Guide: snappyHexMesh.

OpenFoam (n.d.). OpenFOAM: User Guide: OpenFOAM®: Open source CFD : Documentation.

Pađen, I., García-Sánchez, C., and Ledoux, H. (2022). Towards automatic reconstruction of 3d city models tailored for urban flow simulations. 8:899332.

Park, G., Kim, C., Lee, M., and Choi, C. (2020). Building Geometry Simplification for Improving Mesh Quality of Numerical Analysis Model. *Applied Sciences*, 10(16):5425.

Piepereit, R., Deininger, M., Kada, M., Pries, M., and Voß, U. (2018). A sweep-plane algorithm for the simplification of 3D building models in the application scenario of wind simulations. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W10:151–156.

Shojaei, D., Olfat, H., Quinones Faundez, S. I., Kalantari, M., Rajabifard, A., and Briffa, M. (2017). Geometrical data validation in 3D digital cadastre - A case study for Victoria, Australia. *Land Use Policy*, 68:638–648.

The World Bank (2022). Overview. `https://www.worldbank.org/en/topic/urbandevelopment/overview`.

Therias, A., Theodoridou, E., Papadimitriou, C., Visser, F., Zhang, F., and Panagiotidou, I. (2022). Removing shared faces in 3d datasets for numerical simulations.

Tominaga, Y., Mochida, A., Yoshie, R., Kataoka, H., Nozu, T., Yoshikawa, M., and Shirasawa, T. (2008). AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings. 96(10):1749–1761.

Tong, Z., Chen, Y., and Malkawi, A. (2016). Defining the influence region in neighborhood-scale CFD simulations for natural ventilation design. 182:625–633.

UNEP (2021). 5 ways to make buildings climate change resilient. `http://www.unep.org/news-and-stories/story/5-ways-make-buildings-climate-change-resilient`.

van Oosterom, P., Quak, W., and Tijssen, T. (2005). About Invalid, Valid and Clean Polygons. In *Developments in Spatial Data Handling*, pages 1–16. Springer Berlin Heidelberg, Berlin, Heidelberg.

Wagner, D., Alam, N., Wewetzer, M., Pries, M., and Coors, V. (2015). Methods for geometric data validation of 3D city models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W5:729–735.