Meccanica
https://doi.org/10.1007/s11012-024-01816-z

RESEARCH

An automated geometry generation procedure for CWE applications in OpenFOAM: the case study of a thunderstorm downburst moving over Genoa city

Josip Žužul · Alessio Ricci · Massimiliano Burlando

Received: 29 February 2024 / Accepted: 28 April 2024

Abstract Since a decade, Computational Fluid Dynamics (CFD) simulations for Wind Engineering (WE) on large-scale environments became a norm. Meso-γ scales and microscale can be characterized by a large number of structures as well as complex terrain topographies. CFD simulations over such an area would commonly imply the involvement of a time-consuming geometry preparation in the pre-processing stage. This is also the innovative goal of the present paper for which an automated geometry generation procedure for Computational Wind Engineering (CWE) applications is developed in open-source environments. This paper further expands on the unsteady Reynolds-averaged Navier–Stokes (URANS) simulation of a thunderstorm downburst immersed in a background Atmospheric Boundary Layer (ABL) wind moving over Genoa city, Italy. Recommendations for geometry and grid generation of the present case study are also proposed. The study considers two simultaneous approaching wind conditions, a background ABL wind and a moving downburst (DB), the latter modeled as an impinging jet. The simulated results are compared with available full-scale LiDAR profiler data in terms of the vertical profiles of outflow velocity, showing relatively good agreement in terms of magnitude and profile shape.

Keywords Wind engineering · Thunderstorm downburst · Atmospheric boundary layer wind · Automated procedure · Mesh generation · OpenFOAM

Alessio Ricci and Massimiliano Burlando have contributed equally to this work.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11012-024-01816-z.

J. Žužul (✉) · M. Burlando
Department of Civil, Chemical and Environmental Engineering, University of Genoa, Via Montallegro 1, 16145 Genoa, Italy
e-mail: josip.zuzul@edu.unige.it

M. Burlando
e-mail: massimiliano.burlando@unige.it

A. Ricci
Department of Science, Technology and Society, University School for Advanced Studies IUSS, Piazza della Vittoria 15, 27100 Pavia, Italy
e-mail: alessio.ricci@iusspavia.it

1 Introduction

There is a continuously growing interest in the application of CFD across a range of various disciplines. Although traditionally based on full-scale measurements and wind-tunnel tests, CFD studies became an integral part of studies in WE throughout the past decades as well [1]. As a scientific discipline, the WE deals with the impact of atmospheric flows on the structural integrity, but also considers the influence of wind on the human safety with regards to the

Published online: 23 May 2024

pollutant dispersion [2] and investigation of mitigation measures related to wind discomfort in urban environments [3]. These problems commonly arise as an outcome of mutual interaction of ABL winds with generally quite complex configurations of terrain and urban area [4–7].

However, there is more to wind action than just ABL winds, especially in mid-latitude regions of our planet. The wind action in these regions commonly considers additional contributions of non-synoptic winds like thunderstorm winds (e.g. tornadoes and downbursts) which are caused by unstable atmospheric conditions. Thunderstorm outflows like downbursts (DB) are characterized by a column of vertical cold air descending from the cloud which hits the Earth's surface as an impinging jet, which spreads radially causing strong horizontal near-surface winds due to the propagating vortex ring (i.e. primary vortex) [8, 9]. This feature of radially propagating near-surface winds is particularly interesting in the perspective of wind loading of structures, since the structures are designed to withstand ABL winds but not also nose-shaped vertical velocity profile associated with DB winds. Consequently, structures exposed to DB winds commonly suffer substantial damage or even the collapse in the worst possible cases [10]. In that perspective, the THUNDERR project [11] was initiated to study the characteristics of thunderstorm downbursts and their impact on structures, with the intention of contributing to international codes and standards to aid structural design [12]. This led to a significant increase in available full-scale event recordings through the extensive wind-monitoring network composed of anemometers and several Light Detection and Ranging (LiDAR) scanners and profilers. Unfortunately, the full-scale measurements commonly provide neither the desired spatial nor temporal flow resolution, as the thunderstorm event covers a rather large (meso-γ scale) area of about 20 km [13] and has a limited duration of about 30 min [14]. In that perspective, the present study will consider the recorded full-scale thunderstorm event available in the THUNDERR database through the application of CFD simulations. In particular, the selected thunderstorm event considered in this study was the DB event that hit the city of Genoa (Italy) the 14 August 2018, during the collapse of the Morandi bridge [15]. As claimed by [16], temporal and spatial meso-γ scales commonly simulated by meteorologists using weather forecasting models recently became the subject of CFD analyses for wind engineers. However, to the best knowledge of authors, a similar CFD study covering such a large meso-scale area has not been reported in the literature, and definitively represents an innovative aspect of the present study.

The CFD studies covering WE-related topics commonly consider structures with complex geometries in isolated scenarios, located in complex topographies or amidst urban fabrics. Therefore, it is of paramount importance to have the underlying geometry of the area modeled to start building the case. However, various circumstances may arise in which the geometry/grid generation may be challenging: (i) the geometry dataset is not publicly available for the user, (ii) the dataset is not stored in file formats which could be considered CFD-friendly, (iii) the dataset contains a substantial amount of measurement errors, fault data, or even the missing data. Due to the importance of geometry preparation under such input conditions, which affects the grid quality and CFD simulation accuracy, this topic is gaining increasing interest in the very recent past. There were several studies published in the past years [e.g. 17, 18] which focused on the geometry preparation for CFD simulations in the context of WE applications. Alemayehu and Bitsuamlak [17] adopted the deep learning approach to extract building footprints based on satellite images, which were then used with LiDAR data to create 3D building models. The approach was found promising overall, but it seems to highly depend on the quality of satellite images and LiDAR data. Moreover, the resulting buildings' sharp edges were often not adequately resolved and were rather smoothed, which affects the flow separation. Paden et al. [18] presented a very promising (still under development) open-source tool which takes terrain and buildings data to create a watertight and non-manifold geometrical model of the area of interest. However, it does not allow the filtering of the buildings, so that only "large enough buildings" are considered. To overcome such limitations, this study presents the alternative automated geometry generation procedure tailored for CFD simulations of wind flow over large-scale domains. Although the present procedure will be adopted here to generate a geometry and grid later simulated by `OpenFOAM-v2212` [19], the computational grid could be used with any

other CFD solver. Moreover, the recommendations for grid generation of large-scale urban environments in `snappyHexMesh` will be presented. Finally, this study will showcase the capabilities of `OpenFOAM` in simulating a moving thunderstorm DB immersed in ABL winds over Genoa.

This paper is organized as follows. Section 2 presents the automatized geometry generation procedure developed, and the grid generation procedure for simulating wind flows in large-scale areas. Section 3 presents the breakdown of numerical settings for the case study of the moving thunderstorm immersed in the ABL winds. Section 4 holds the results showcasing the vertical velocity profile comparison with the full-scale measurement data, and the wind flow contours. Section 5 closes the paper with conclusions and future work.

2 Methodology

Conducting CFD simulations over large-scale areas for WE studies commonly involves time-consuming pre-processing steps related to geometry and grid generation. The pre-processing step usually takes more than 50% of the time required to conduct the entire CFD study. Based on the authors' experience of conducting a number of WE studies covering wind flows in complex urban areas [e.g. 7, 16], the geometry simplification and grid generation steps are closely linked and are detrimental to successfully conduct the CFD simulation. These pre-processing steps heavily rely on the experience of the user often leading to the increased time spent for the case preparation. This is also further emphasized by the fact that the underlying geometry required for the grid generation is often also not readily available. This section presents the newly automated geometry generation procedure developed in fully open-source environments. The procedure is based on generating two separate sets of geometries—one for the terrain, and the other one for the buildings. The point cloud representing the terrain elevation data can be retrieved through the Digital Terrain Model (DTM) data (if available) or from the NASA Shuttle Radar Topography Mission (SRTM) repository [20]. The SRTM database, available in the public domain, provides access to terrain elevation data. Interested users can retrieve this data by logging in and specifying the minimum and maximum values for latitude and longitude. This can be done interactively, but also in a programming language of choice (e.g. Python, Matlab) where the retrieved binary file could also be converted into a human-readable (ASCII) elevation data point cloud. These point clouds could rather easily be converted into the surface triangulation. However, the buildings-related datasets are generally not readily available, and in case they are, these are rarely available in surface triangulation formats suitable for CFD studies (e.g. stereolithography, STL). Instead, file formats such as GeoJSON used in geoinformatics, and related disciplines are most common. These file formats are unfortunately not very useful to a CFD user, and based on the current knowledge of the authors, the approach of Paden et al. [18] is the only way available to convert the GeoJSON files into the surface triangulation files. However, the user might be interested in studying the flow over larger buildings only instead of analyzing the flow past every single building, as the latter approach would yield substantial number of computational cells in the mesh. This study will therefore present an approach for the conversion of building dataset data to STL files based on the relevant information from the native GeoJSON file format. Recommendations for the grid generation in `snappyHexMesh` for WE applications closely linked with the geometry generation procedure are also proposed.

The present research study is embedded inside the framework of the ADAPTNOW Project where the Municipality of Genoa is set as a target area for the evaluation of the wind hazard map. The domain size was therefore based on the previous studies [21–23] which used larger areas, but also on the characteristics of the thunderstorm event. In that perspective, the present study does not imply all similar WE applications should necessarily have such a large scale domain size. Instead, domain size should be adjusted based on specifics of the analyzed case study. The thunderstorm characteristics were observed through the output data of the LiDAR station installed in the port of Genoa and radio sounding, satellite and Radar images acquired on the area of interest. The output data also clarified some of the basic characteristics of the DB, as: the touchdown location, the downdraft diameter (D) of 4 km, the storm translation speed (V_{storm}) of 5 m/s, and the direction of the storm propagation (i.e. 30° with respect to the North)

from the touchdown location. The storm duration was about 30 min, which coincides with the storm arrival to the coastline [15]. It was found in previous studies that the DB outflow still exhibits severe wind speeds due to the propagating vortex rings up to the radial distance of approximately $2D$ away from the touchdown [24]. Accordingly, the target domain area was selected as 32 km (width W) × 32 km (length L) centered around the DB touchdown location, to account for these vortex ring structures. The schematics of the thunderstorm event is depicted in Fig. 1a, which shows the initial touchdown location (denoted with the X mark), the initial storm location (black dot), and the storm center location at the moment of reaching the shoreline (denoted with red star). Blue circle denotes the downdraft, i.e. the area characterized by the vertically downwards descending air from the thunderstorm cloud. Red dashed line indicates the trajectory of the traveling storm from the touchdown towards the shore. Figure 1b shows the LiDAR station (i.e. LiDAR scanner working in LiDAR profiler operational mode) which recorded the vertical profiles of DB outflow velocity throughout the event, also indicated in Fig. 1a with red rectangle.

2.1 Geometry generation procedure

2.1.1 Geometry of the topography

Geometry generation step is very often a critical pre-processing step required in order to produce a high-resolution computational grid of complex built environments. The typical geometry relevant for such cases commonly include the topography (i.e. terrain) varying in altitude, sea surface, and various types of structures like buildings and bridges. About the terrain data, the underlying point cloud of the elevation data could be freely retrieved from the open database of NASA SRTM mission [20]. Eventually, the local authorities might provide DTM dataset usually of higher resolution with respect to the NASA SRTM dataset. The point cloud can then be converted to the triangulated STL file by means of a variety of different software as Python and Matlab. It is also often the case there might be several different types of roughness classes [25] that could be associated with the terrain surface of the area of interest, of primary importance to properly develop the approach ABL and DB wind. In this study, two main roughness

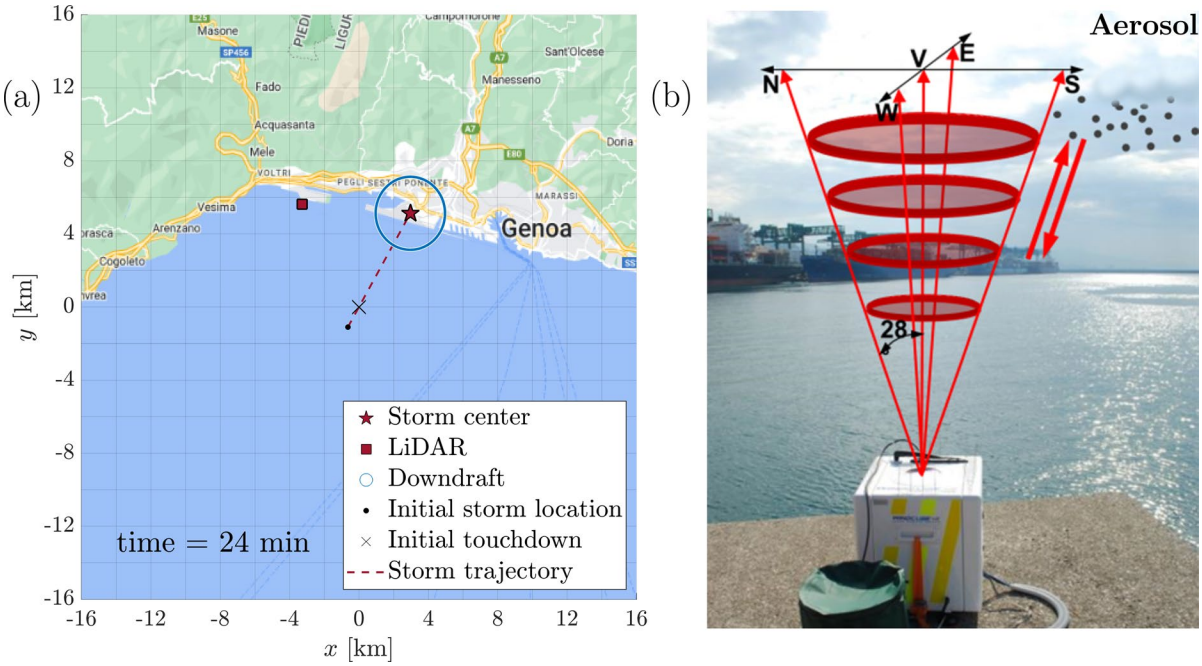

Fig. 1 Downburst event occurred in Genoa, on 14 August 2018: **a** schematics of the storm trajectory with the indication of the initial DB touchdown, location of the LiDAR station, and storm center location with the downdraft moving along the trajectory (hereby presented for $t = 24$ min); **b** LiDAR station located in the port of Genoa

classes were preliminarily considered in terms of aerodynamic roughness length (z_0) to be imposed on the corresponding patches, namely "sea" and "land". Since each roughness patch is associated with a different z_0, it is a good practice to define the patches (and the corresponding z_0) prior to the grid-generation stage. Moreover, when simulating ABL flows, multiple wind directions are usually analyzed over a certain area, with a specific interval (e.g. 0°, 30°,..., 330°) to cover the entire wind rose. In that perspective, it is also a good practice to consider this aspect at the early stage of case preparation. Therefore, a certain degree of terrain topography modification close to the boundaries might be required in order to avoid numerical instabilities for cases which high irregularities like mountains. These numerical instabilities may lead to convergence issues, as there is not enough space available in the domain to allow for the proper development of the ABL wind. In such circumstances the modifications to the ground patch are highly recommended for the sole purpose of making sure the proper inflow conditions are indeed imposed. These terrain modifications are hereby referred to as the "buffer region". The buffer region is an "extra region" artificially added to the external boundaries of the target domain (Fig. 1a), which does not necessarily represent the reality but is therefore required to facilitate the simulation stability when releasing an ABL flow. Hereby, the buffer region is divided in two segments: (i) the outer segment, characterized by a flat surface at the outer 25% of the total buffer length; (ii) the inner segment, characterized by a linear increase of the terrain height from the flat surface to the actual terrain height. The altitude of the outer segment is determined by the reference altitude at the boundaries of the considered target domain area. As the sea surface level was the reference altitude in the present case (see Fig. 1a), the 0 m altitude was chosen for the outer segment throughout the whole buffer region. A buffer length of 2 km was adopted in this study. The created terrain topography with the additional buffer region, and a separation of terrain into individual patches based on the roughness classification is shown in Fig. 2.

2.1.2 Generation of the buildings geometry

Dealing with buildings' data may be more challenging compared to topography. First, the buildings' data are generally not always publicly available. Secondly, when the dataset is available, it is commonly stored in file formats typically used in geoinformatics and related fields. The typical example is the GeoJSON file format, which is not easily transferable into STL format. However, it is an ASCII file with coordinates (x, y) of each building polygon and building height above the sea surface level (z). Such data can be extracted into separate .txt files (e.g. for every building) allowing for the building reconstruction. It is important to stress out that the georeferenced data in GeoJSON files is not exempt from LiDAR measurement errors. These measurement errors typically represent themselves in a cluster of points very close one to the other. Firstly, such detailed set of points is redundant as it is not required for the realistic description of the building shape (i.e. alpha shape).

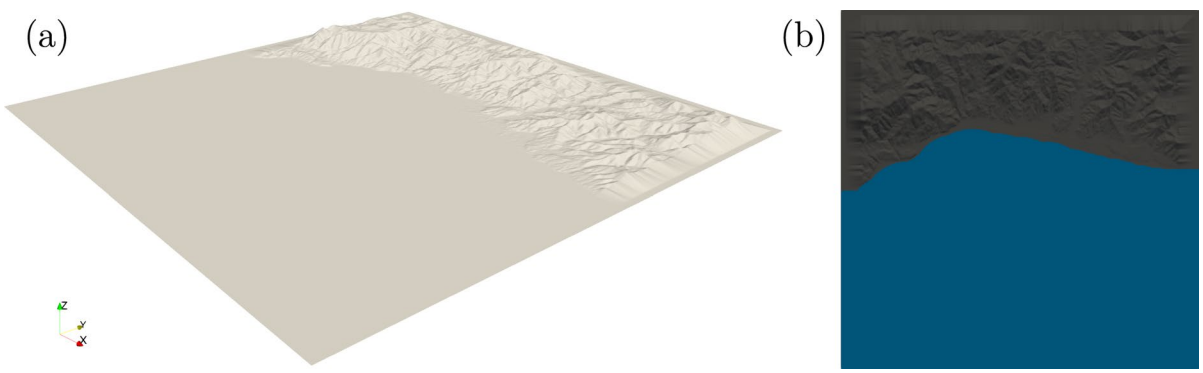

Fig. 2 Terrain geometry of the target area: **a** isometric view showing the buffer region at the boundaries; **b** separation of the terrain geometry in two patches: sea (blue) and land (dark gray)

Moreover, generation of 3D objects in such scenarios is likely to experience rendering errors and failure in creating STL files. In these scenarios, the application of Ramer–Douglas–Peucker (RDP) algorithm for the point reduction is recommended [26, 27]. It is able to effectively reduce the cluster of points into a single point such that it does not substantially distort the overall shape (i.e. alpha shape) of a building polygon. In addition, the RDP algorithm would also remove outliers in the input building polygons which are also occasionally present. These outlier points should indeed be removed, as their consideration would in turn represent themselves in the buildings having severely acute angles, and finally likely producing highly skewed cells (i.e. source of numerical errors). Finally, the input GeoJSON file can sometimes hold overly detailed dataset. In such cases, the user might consider filtering the buildings based on certain criteria, for instance, the influence of smallest buildings can be accounted for in terms of implicit roughness (e.g. z_0). In the present study, the following building removal criteria were used: (i) the threshold building roof cross-section area was set to 25 m^2, and (ii) the threshold building height was set to 5 m. Buildings which satisfied these conditions were removed and were not exported for STL generation. It is crucial to emphasize that the threshold values adopted are closely linked to the domain and target area size. While they might suit the present study, it is imperative to consider a further reduction of these threshold values for scenarios involving less extensive domains. In this context, users need to carefully balance the trade-off between the geometry/grid resolution on one side and the extent of the area of interest on the other.

As the number of buildings can be rather huge, generating these manually with a Computer-Aided Design (CAD) may be very time consuming. In that perspective, an open-source CAD software OpenSCAD [28] is proposed in this study. OpenSCAD is an open-source software which makes it possible to generate CAD through coding, using its own programming language. It essentially works by operating on 2D outlines (such as polygon point clouds or alpha shape) by applying the linear extrusion operations of these point cloud sets to obtain 3D objects. However, OpenSCAD lacks basic features every modern programming language has. Therefore, the procedure to use OpenSCAD was written in Python and it makes use of `solidPython2` library that essentially returns the compiled OpenSCAD code. The latter is then used for rendering in OpenSCAD software which exports the target STL file composed of all buildings (without self-intersections or similar issues). The proposed procedure could also be applied to other scenarios, different than large built environment cases, which require the generation of the CAD primitive shapes like cubes, cylinders, and so on. As an example, Appendix A provides the code to read the polygon coordinates of every single building's roof and performs its linear extrusion in the negative z direction. The provided code forces the building extrusion below the reference altitude (here the sea surface level) up to an arbitrary height of z = -5 m. This will ensure the terrain-building intersection in every region of the domain by avoiding possible zero thickness issues. The output of the code is the SCAD file which by straightforward rendering in OpenSCAD returns the STL file of all the buildings. The STL representation of all buildings covering the entire area in Genoa is presented in Fig. 3, while Fig. 4 shows the topography and the buildings on a large portion of Genoa city.

2.2 Grid generation procedure

As mentioned earlier, the present case study considers two simultaneous approaching wind conditions: (i) a background ABL wind, and (ii) a moving vertical DB wind modeled as an impinging jet. As the DB inflow jet location is changing over time (see Fig. 1a), this study utilizes the dynamic meshes (sliding meshes) approach. In this approach two independent meshes (i.e. mesh region) are created: the bottom static mesh close to the terrain and buildings, and the top moving mesh in the form of a narrow disk. The latter translates by sliding at the top of the bottom mesh during the simulation to mimic the storm propagation.

The bottom mesh was built in `snappyHexMesh`, and the procedure to create high-quality hex-dominant mesh for WE applications is briefly explained herein. To create the mesh, one first needs to have the relevant geometry (i.e. STL files for the sea, the terrain, and the buildings), discussed in the Sect. 2.1. The buildings are bluff bodies with sharp edges which highly determine the flow separation. Therefore, the buildings' sharp edges need to be adequately resolved and they should also have a sufficient mesh resolution. This can be effectively achieved through

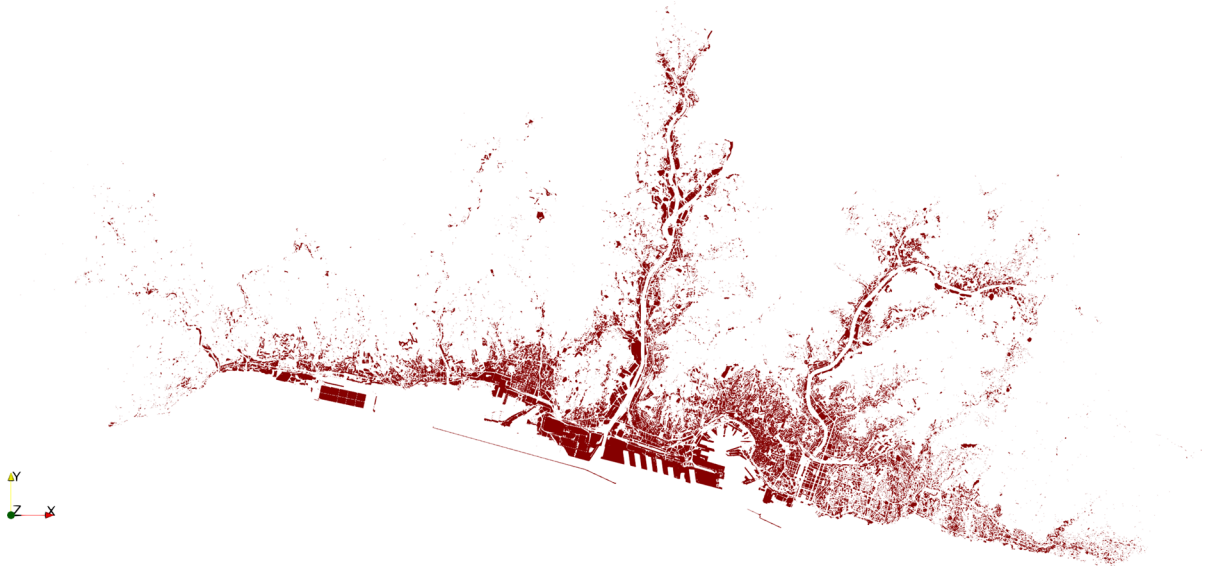

Fig. 3 Top view of the buildings of Genoa city generated with OpenSCAD tool

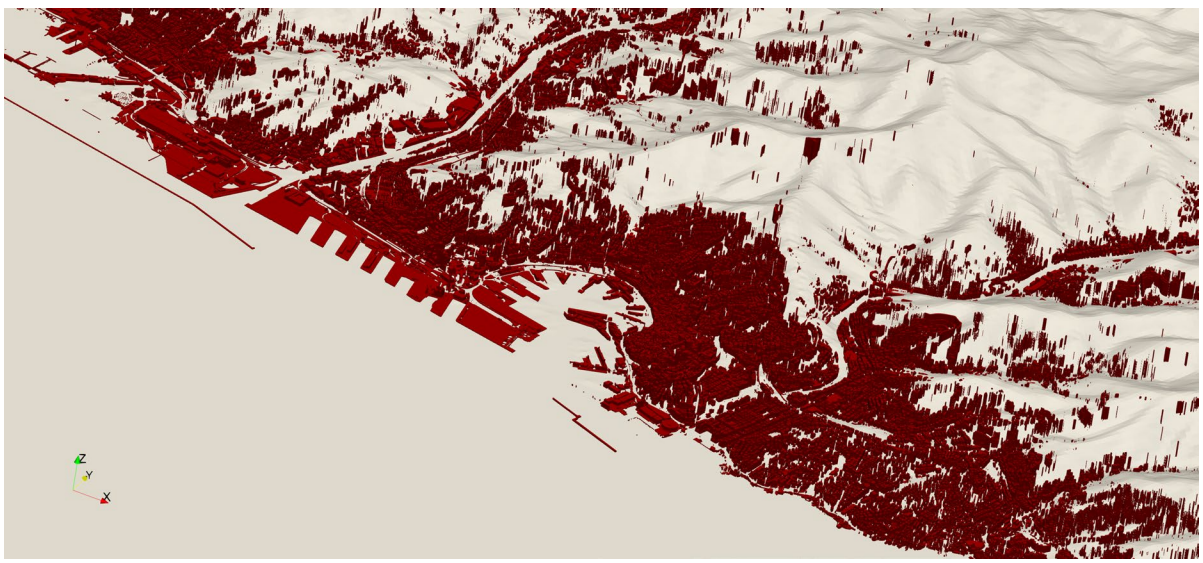

Fig. 4 Topography and buildings of a large portion of Genoa city generated with OpenSCAD tool

explicit feature edge extraction using the utility `surfaceFeatureExtract`. As the geometry generation procedure (Sect. 2.1) does not create the conformal match among vertices of surface triangulation files (i.e. buildings and terrain), it is necessary to also specify the feature edges which occur at the intersection between terrain and buildings with `surfaceFeatureExtract`. The meshing was based on the `blockMesh` background mesh. It was created based on the topography dimensions extended by the buffer region (i.e. 36 km × 36 km), and considering in height the storm cloud up to 4.5 km.

The `snappyHexMesh` was then used for mesh castellation, snapping, and layering accordingly.

Figure 5 shows the computational domain with the indication of domain boundaries: `ablInlet` (ABL inflow patch), `outlet`, `terrain` and `sea` (indicated as ground in Fig. 5a), `buildings` (not reported in Fig. 5a), and `top`. The Fig. 5a also shows the sliding top mesh, which was created with `blockMesh` with the assistance of the `m4` macro. Figures 5b–d show three views of the computational grid: (b) top view of the domain, (c) perspective view to visualize the different altitude between sea and

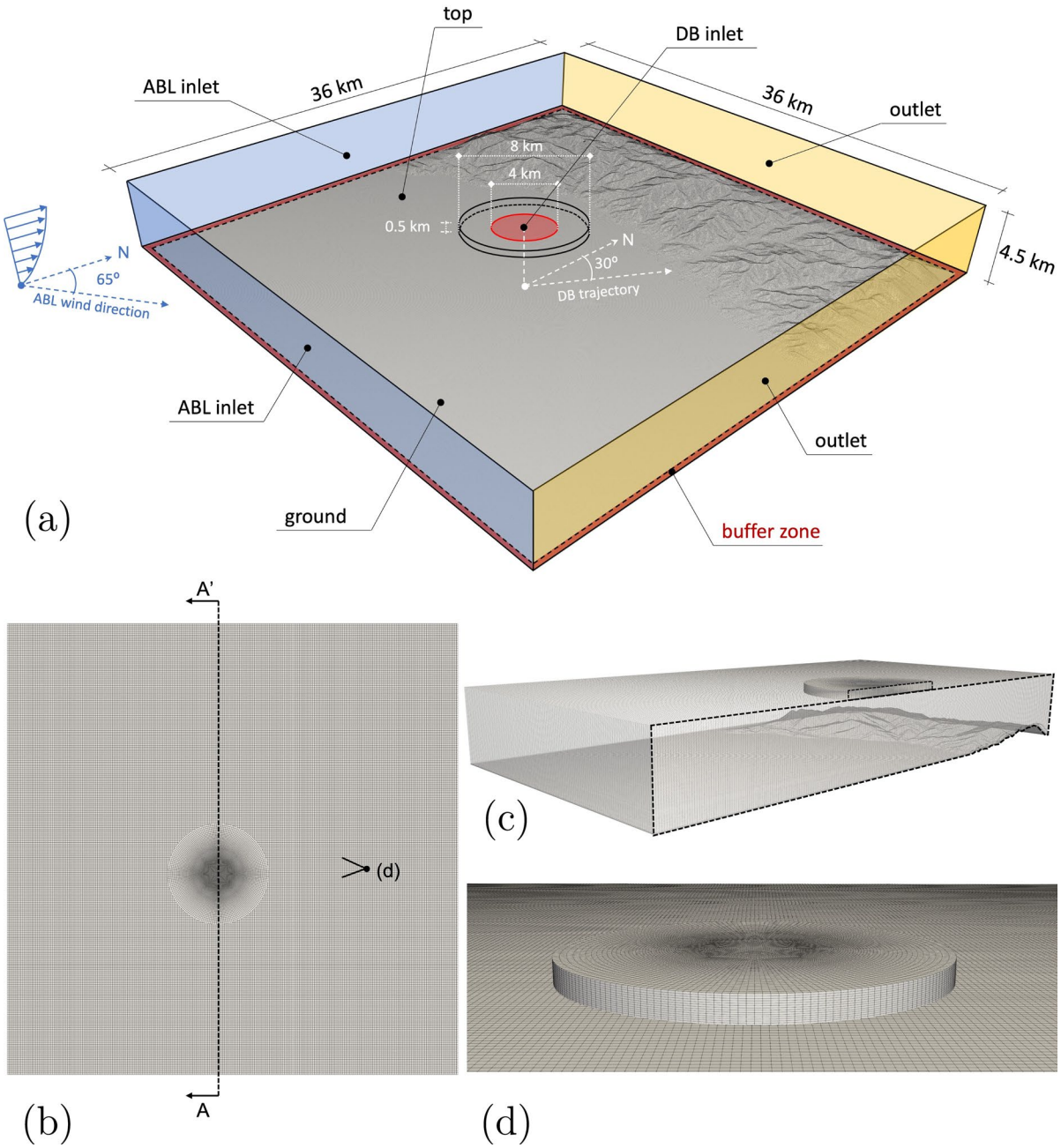

Fig. 5 Schematics of the computational domain with indications of domain boundaries for the near-surface mesh, and the depiction of the sliding mesh approach (**a**); top view of the grid (**b**); grid around the terrain (**c**); and details of computational grid in the proximity of the moving top mesh (**d**)

mountainous terrain, (d) detailed view of the interface between the static bottom mesh and the sliding mesh at the top. The top sliding mesh with an indication of the relevant domain boundaries is also presented in detail in Fig. 6. More specifically, the top mesh is composed of four patches: `stormInlet` (the actual inflow patch of the vertical DB jet having diameter D), `slidingPatch` (the patch sliding over the `top` patch of the static bottom mesh), `stormOutlet`, and `stormSides`. The top mesh has a diameter equal to $2D$ and a height of 0.5 km. The computational grid counts 14 million cells, approximately. Ultimately, Fig. 7 provides a condensed overview illustrating the entire process of geometry and grid generation essential for carrying out CFD simulations of wind flow over large and complex areas.

3 Other computational settings

In addition to the storm characteristics such as the downdraft diameter D, the full-scale measurements and radar images assisted in identifying several other properties important for setting up the CFD case. For the thunderstorm, the vertical DB inflow jet velocity component (w_{jet}) was set to 11 m/s. For the ABL wind, some key parameters were defined as follows: the friction velocity (u^*) was set equal to 0.121 m/s; the aerodynamic roughness length of the sea ($z_{0,\text{sea}}$) and land ($z_{0,\text{land}}$) were set to 0.0002 m and 0.03 m based on Davenport classification modified by [25], respectively; the approach ABL wind direction was set to 65° with respect to the North. `OpenFOAM-v2212` was used to set the ABL, due to the availability of `atmosphericModels` library. For the two inflow conditions, the `ablInlet` was used for the ABL wind imposed on the two sides of the domain, while the `stormInlet` was used to reproduce the vertical DB wind imposed on the moving disk (see Sect. 2), by using the dynamic mesh (i.e. sliding mesh). It is a common practice in WE to consider several ABL wind directions with a specific interval over certain area (e.g. every 30°) to cover the entire wind rose. Although this particular study focused on a very specific ABL direction, the possibility was still left for future studies to facilitate straightforward configuration of cases with other ABL wind directions. As the ABL direction was 65° with respect to the North in the present study, the eastern and southern domain boundaries were set as the `ablInlet` boundary. Although rotating the entire computational domain might be acceptable on some occasions, it is commonly elected to keep it in the original North-South orientation to keep the clear correlation between the locations in the model and in reality.

For the dynamic mesh, two independently constructed meshes were firstly combined together by means of `mergeMeshes` utility, which creates two distinct mesh regions within the combined mesh. Such an approach allows one to the impose the dynamic mesh motion (i.e. the storm propagation with the speed V_{storm} along the storm trajectory) to

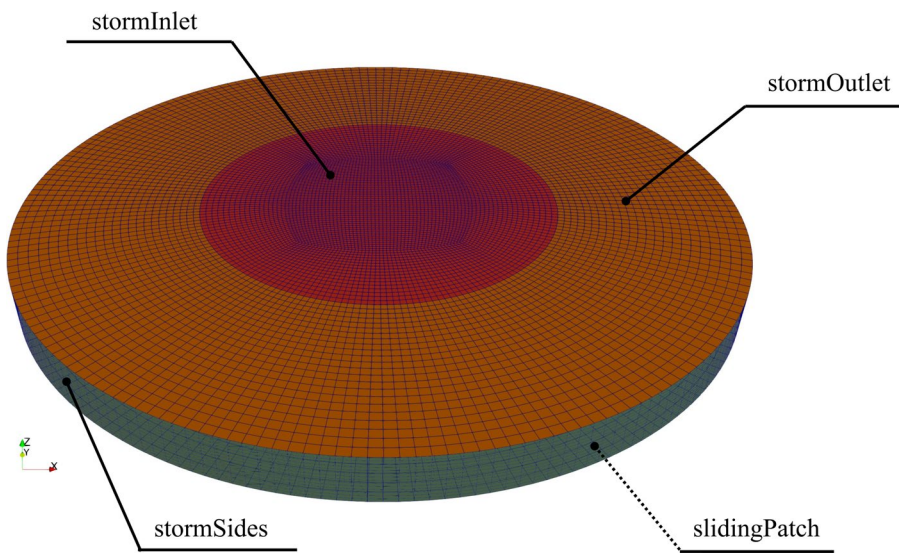

Fig. 6 The mesh of the moving disk which slides on the top of the near-surface mesh. Boundaries of the top mesh are also indicated in the figure

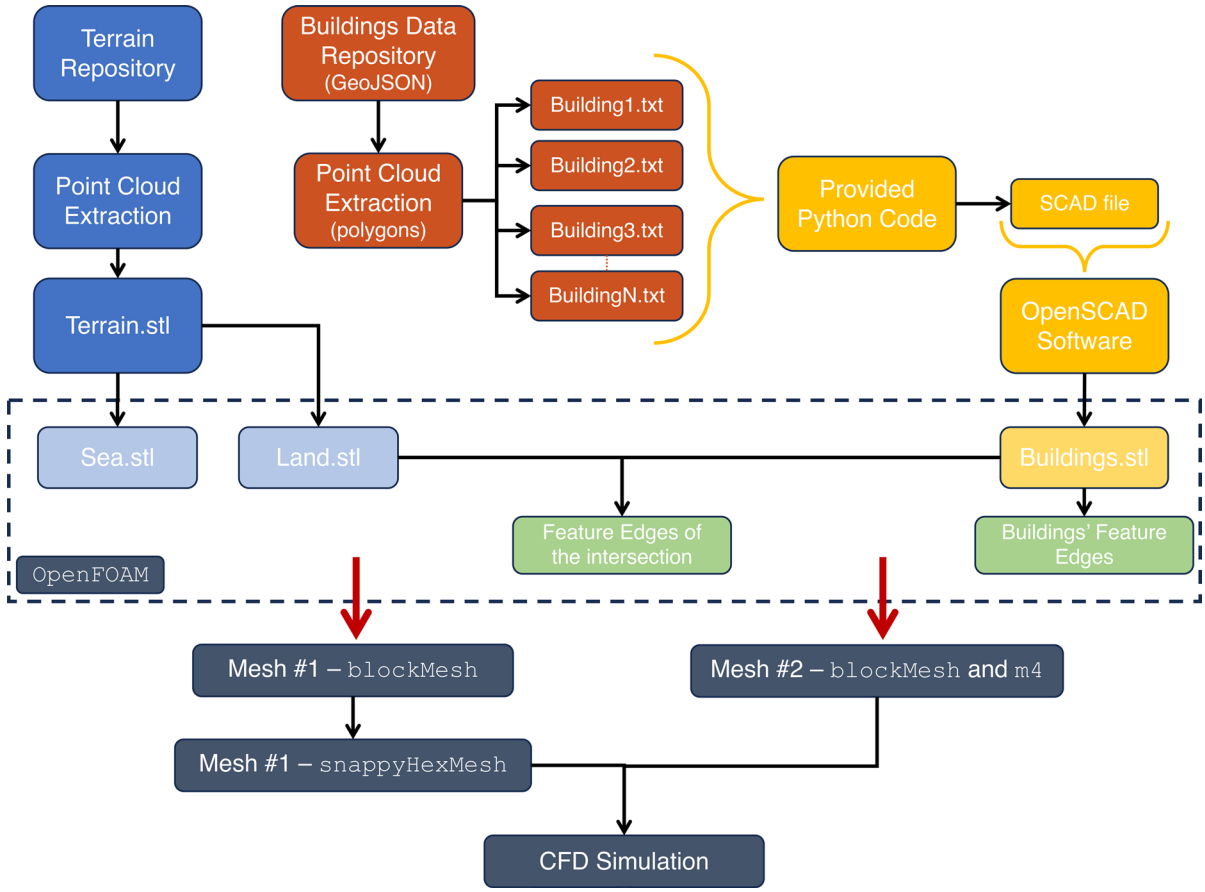

Fig. 7 Flowchart of the procedure: from geometry and grid generation towards the CFD simulation

the selected `cellZone` composed of the top mesh cells. This is achieved by restricting the dynamic mesh motion in `dynamicMeshDict` to the `linearMotion` with 2.5 m/s, and 4.33 m/s translational velocity in x and y direction, respectively. Although the two meshes are combined, the conformal match between the mesh nodes at both sides (`slidingPatch` of `top` mesh, and `top` patch of the bottom mesh) could not be guaranteed. Therefore, an interface was introduced to allow the vertical DB inflow into the bottom mesh. This was done by introducing the partially overlapping sliding mesh interface, namely the Arbitrary Coupled Mesh Interface `cyclicACMI`. In that perspective, the two sets of patch pairs (4 in total) were created: 2 from the `slidingPatch` patch (`sliding_couple` and `sliding_blockage`), and 2 from the `top` patch (`top_couple` and `top_blockage`). Hereby, `sliding_couple` and `top_couple` are indeed defined as `cylicACMI` patch types defining the interface itself, while the `sliding_blockage` and `top_blockage` are of a generic `patch` type. Therefore, the `sliding_couple` defines the `top_couple` as its `neighbourPatch`, while it requires its pair `sliding_blockage` to be specified as the `nonOverlapPatch`. The same approach is done vice-versa, so the `top_couple` defines `sliding_couple` as its `neighbourPatch`, and `top_blockage` is specified as the `nonOverlapPatch`. Such an approach allows to use the interface at the locations of DB inflow from the top mesh to the bottom mesh, while another boundary condition could be used for the non-overlapping part of the `top` patch.

The CFD simulations were performed in two stages: (i) the ABL initialization through the domain (without considering the moving disk), and (ii) the moving storm immersed in the initialized background ABL wind. The ABL wind was initialized by means of a steady RANS

simulation using the atmosphericModels library and its corresponding set of boundary conditions, with the velocity field U at the top patch determined based on the fixed shear stress (without considering the top mesh and the translating disk). In that perspective, the ABL inflow profiles in terms of velocity U, turbulent kinetic energy (k) and specific dissipation rate (omega) were specified through the ablInlet patch positioned at the eastern and southern boundary of the domain. The velocity profile and directionality of the wind are determined through the ablInlet based on the $z_{0,sea}$, friction velocity (u^*), and flowDir vector (for the directionality). The wall-type patches (sea, land, and buildings) were set to noSlip conditions. Standard wall functions were used for k and omega. The turbulent viscosity field nut of the buildings patch adopted the standard wall function, while the topography-related wall-type patches (land and sea) adopted wall function for atmospheric flows required to maintain the imposed U, k, and omega profiles and prevent the occurrence of unintended streamwise gradients [29]. As stressed out by [29], a special care has to be taken when simulating atmospheric flows in conjunction with the wall functions to model the near-wall fluid behavior. It is the case as the standard wall functions were developed to facilitate fluid flow modeling by using the imposed roughness of a significantly smaller scale (compared to atmospheric flow applications) based on the experimental data for sand-grain roughened pipes. As a consequence of not treating the surface roughness appropriately, nonphysical changes in the imposed ABL profiles might occur near the ground level as the flow is being propagated in the domain from the inlet boundary. Zero-static gauge pressure p was specified at the outlet patch, while outflow conditions for U, k, and omega, respectively. The adopted boundary conditions are summarized in Table 1 (last 6 table entries). Note that the top patch during the ABL initialization stage adopted the same set of boundary conditions used for the top_blockage in the second stage.

Following the successful ABL initialization throughout the bottom domain, the two meshes are merged (with mergeMeshes), the cyclicACMI interface is created, and the dynamic motion of the moving cellZone is configured as discussed earlier in this section. The simulation is conducted by setting the DB vertical inflow jet at the stormInlet patch boundary through fixedValue, using u_{jet} and v_{jet} components (based on storm translation speed and direction) and w_{jet}. The stormOutflow and stormSides patches were assigned the outflow conditions, while the slidingPatch was converted into the cyclicACMI patch-pair interface (i.e. sliding_couple and sliding_blockage patches) with cyclicACMI patch-pair based on the top patch (i.e. top_couple and top_blockage patches). Hereby the non-overlapping cyclicACMI patches (i.e. top_blockage and sliding_blockage) were assigned the velocity boundary condition based on the fixed shear stress to maintain the ABL profile. The complete set of boundary conditions used for the simulation is presented in Table 1. This research study focused on presenting the automatized procedure to geometry and grid generation for complex urban areas typical of WE. In that perspective, the study in its present form did not consider the sensitivity analyses of grid resolution and turbulence model on the numerical results.

Unsteady RANS approach coupled with SST k–ω turbulence model was used to simulate the storm event of 30 min of real time (t) with the time step of 0.2 s using the second-order discretization schemes, and the case was solved with pimpleFoam solver. The simulation was carried out by with a High-Performance Computing (HPC) system by using nodes composed of Intel Xeon Platinum 8276–8276 L (2.4Ghz) processors. In total 480 processor cores were used for the simulation (10 nodes of 48 processor cores/node) with 300 GB RAM per node. In total, 9600 CPU hours were used, approximately.

4 Results

This section showcases the preliminary results to demonstrate the applicability of the workflow presented in earlier sections. Figure 8 shows the comparison of CFD results with the available full-scale measurement data gathered by the LiDAR station located in the port of Genoa (see Fig. 1). The comparison was performed in terms of the vertical profiles of radial outflow velocity for three different time steps. The profiles were presented in the normalized form, where the outflow velocity (U) was normalized with the inflow jet velocity magnitude (V_{jet}), while the heights (z) above the sea surface level was normalized by the height of LiDAR measurements associated

with the outflow velocity maximum (Z_{max}). The comparison shows a relatively good agreement in time, magnitude, and vertical profile shapes.

Figure 9 shows the wind velocity contours with superposition of velocity vectors, normalized by V_{jet}. Specifically, Fig. 9a shows the wind velocity contours on the horizontal plane made at $z = 175$ m above the sea surface level for $t = 30$ min (end of the simulation). The wind velocity field shows the characteristic features of the propagating vortex ring affecting the area around the downdraft, with amplified wind speeds at the propagating front [24, 30–32]. The translating DB by interacting with the underlying terrain and buildings has the potential to intensify the local wind speeds. More specifically, the wind speed seems to be further amplified by the topological effects of the area, the mountains, and the valley. Indeed, it was the valley location where the highest wind speeds were observed due to possible canyoning effects. It is also observed that the area affected by the highest wind speeds during the DB event coincides with the location of the collapsed Morandi bridge (denoted with the black dot in Fig. 9a). Figure 9b–d show the wind velocity contours on a vertical plane made along the storm propagation trajectory (i.e. 30° with respect to the North) for time instances $t = 20$, 25 and 30 min, respectively. The wind velocity contours in vertical planes show the development of the ring vortex typical for the downburst outflows. This ring vortex structure forms due to the wind shear aloft, which then propagates radially outwards above the ground surface after the impingement. This is the case for stationary (non-translating storms) as well [24], and not only for the translating storms. However, compared to the stationary storms, which create the axisymmetric flow field around the touchdown [24], the translating storms induce the flow asymmetry with the strongest winds downstream of the propagating front. Although the vertical cross-sections in Fig. 9b–d were presented along the storm trajectory (which coincides with the lowest terrain elevation gradients as it passes through the valley), the terrain roughness still seems to play the role in lifting the vortex core to higher altitudes. This effect has also been observed by [33]. This dynamic interaction has a consequential impact on the vortex core, which leads to expansion of high-velocity regions across a broader range of altitudes. The flow field at the very opposite side is characterized by the collision of ABL wind with the DB outflow. In such circumstances, the ring vortex assumes a standing-vortex-like behavior instead of propagating.

5 Conclusions and future work

The present study aimed to address the common challenges in conducting CFD analyses for WE applications which typically consider complex and large areas. Such cases often involve the entire preprocessing pipeline including the tedious and time-consuming tasks of geometry generation. Indeed, it is the geometry generation procedure which is typically the critical component of the entire study as it affects the grid generation procedure, which in turn has an impact on the quality of CFD results. In this paper, a comprehensive break-down of an entire framework to conduct a CFD study for WE applications in open-source environments was presented on the case study of a moving thunderstorm outflow over a complex area of Genoa city, Italy.

WE studies typically consider areas composed of a large number of buildings, scattered around a terrain topography of various complexity level. Therefore, a generic CFD study on built environment eventually ends up with complex geometries of terrain and buildings. This is also the reason why the geometry preparation procedures for environmental applications are an actively investigated topic nowadays. In this study, an automated procedure for the geometry and grid generation was presented. Essentially, the presented Python-based code reads the coordinates of the input buildings' roof vertices and compiles the OpenSCAD file, that is used for straightforward rendering of the 3D surface triangulation set of buildings in STL format. Although developed particularly for dealing with a large number of buildings in the perspective of WE, the procedure could also be adopted for similar purposes in other disciplines where handling a large number of primitive shapes might be required. These could include assessing the effects of newly designed buildings within well-settled environments, in terms of pedestrian-level wind and thermal comfort, ventilation as well as pollutant concentration. Recommendations for the terrain topology preparation were also discussed, in an attempt to facilitate the simulation setup. In particular, the importance of introducing the buffer region at the boundaries of a complex

Table 1 The summary of adopted boundary conditions

	Fields				
patchName	U	p	k	omega	nut
stormInlet	fixedValue	zeroGradient	[1]	[2]	calculated
stormOutlet	inletOutlet	fixedValue	inletOutlet	inletOutlet	calculated
stormSides	inletOutlet	fixedValue	inletOutlet	inletOutlet	calculated
sliding_couple	cyclicACMI	cyclicACMI	cyclicACMI	cyclicACMI	cyclicACMI
sliding_blockage	[3]	zeroGradient	zeroGradient	zeroGradient	calculated
top_couple	cyclicACMI	cyclicACMI	cyclicACMI	cyclicACMI	cyclicACMI
top_blockage	[3]	zeroGradient	zeroGradient	zeroGradient	calculated
ablInlet	[4]	zeroGradient	[5]	[6]	calculated
outlet	inletOutlet	fixedValue	inletOutlet	inletOutlet	calculated
land	noSlip	zeroGradient	[7]	[8]	[9]
sea	noSlip	zeroGradient	[7]	[8]	[9]
buildings	noSlip	zeroGradient	[7]	[8]	[10]

[1] turbulentIntensityKineticEnergyInlet; [2] turbulentMixingLengthFrequencyInlet; [3] fixedShearStress; [4] atmosphericBoundaryLayerInletVelocity; [5] atmosphericBoundaryLayerInletK; [6] atmosphericBoundaryLayerInletOmega; [7] kqRWallFunction; [8] omegaWallFunction; [9] atmNutkWallFunction; [10] nutkWallFunction

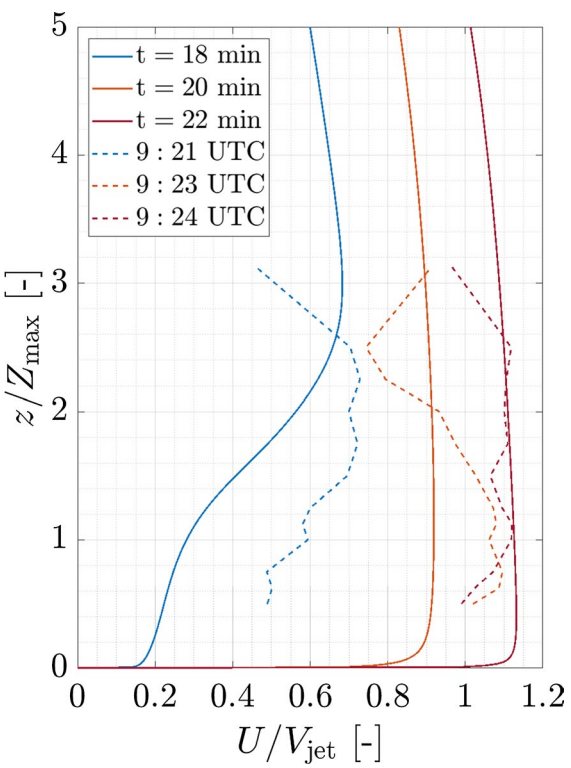

Fig. 8 Comparison between the CFD and LiDAR vertical profiles of radial velocity

(e.g. mountainous) terrain was emphasized to avoid numerical instabilities. Moreover, the grid generation in snappyHexMesh was elaborated and suggestions on effective handling of the grid generation procedure was presented. The multi-inflow case study of a moving thunderstorm immersed in a background ABL wind was covered, and an approach to handling such cases in OpenFOAM was provided through the application of sliding meshes with ACMI interface. The bottom mesh was used to initialize the ABL wind throughout the domain by using the atmosphericModels library. The secondary inlet related to the translating storm and vertically descending impinging jet was modeled by adopting the dynamic mesh motion applied to the top mesh region. The configuration of boundary conditions tailored for common WE applications was discussed and summarized.

The CFD results were compared to the available full-scale LiDAR measurements and the relatively good agreement in terms of velocity magnitude and profile shape was observed. The wind velocity contours in horizontal plane showed wind velocity amplifications in the valley of the Morandi bridge location. These amplifications are due to the mutual confluence of various factors like the channeling effects due to the terrain topography and thunderstorm trajectory. The terrain was found to have an additional

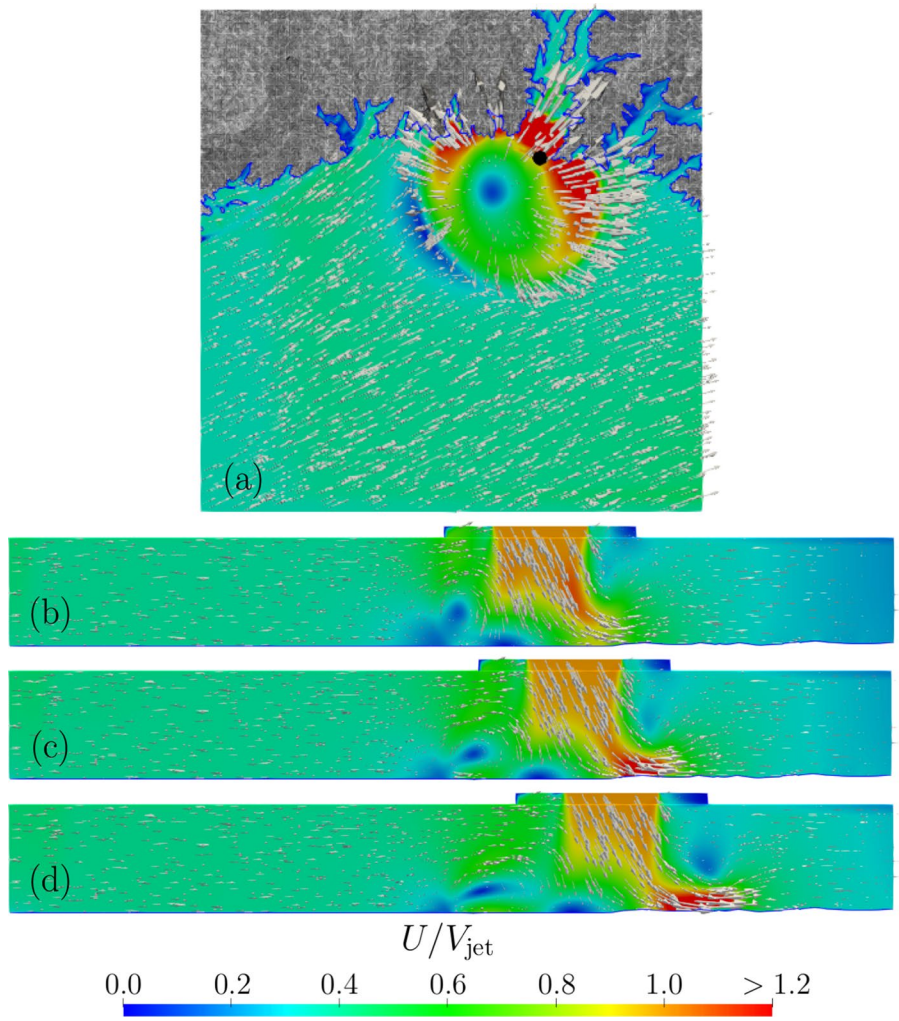

Fig. 9 Wind velocity contours on the horizontal plane made at $z = 175$ m above sea surface level for $t = 30$ min (**a**); wind velocity contours on the vertical plane taken along the storm trajectory for $t = 20$ min (**b**), $t = 25$ min (**c**), and $t = 30$ min (**d**). Black dot indicates the location of the collapsed Morandi bridge

effect of keeping the vortex core at higher levels from the ground at the downstream locations, causing the high-velocity regions to span across larger range of heights. In conclusion, this study demonstrated the capabilities of `OpenFOAM`, and its reliability for meshing and simulating large-scale complex areas for WE applications.

The current study represents an initial phase in a more extensive research endeavor aimed at establishing a comprehensive framework for conducting CWE studies within intricate urban settings. This framework envisions the integration of a fully automated process for geometry and mesh generation, specifically tailored to large-scale simulations. Moreover, the framework will incorporate additional optional wind conditions, such as thunderstorm winds like downbursts and tornadoes. A critical facet of the forthcoming research involves a meticulous sensitivity analysis which aims to address the level of detail in input geometry, and the impact of various numerical settings such as the grid resolution and turbulence model selection. This sensitivity analysis would seek to identify an optimal equilibrium among three key elements: (i) the input geometry resolution, (ii) the obtained grid resolution, and (iii) the accuracy of the numerical solution. The insights derived from such an analysis are expected to yield best practice guidelines for grid generation in the intricate context of large and complex urban areas, in a longer term perspective of advancing the robustness and reliability of future CWE studies.

Acknowledgements This study was funded by ADAPTNOW Project—"ADAPTation Capacity Strengthening for Highly

Affected and Exposed Territories in the Alps NOW" (Grant No. ASP0100048). ADAPTNOW is co-financed by the European Regional Development Fund through the Interreg Alpine Space programme. The authors also acknowledge the support of the Municipality of Genoa (Italy) in providing the relevant datasets used in the framework of this research study. The authors acknowledge the support of the Italian supercomputing center CINECA for providing access to their High-Performance Computing (HPC) facilities needed to carry out the research presented in this paper. The authors are very grateful to the Organizing committee of the 18th OpenFOAM workshop for the invitation to the technical session "Civil and Wind Engineering" and to the present special issue.

Author contributions Josip Žužul: Code development, CFD simulations, Formal analysis, Data curation, Writing—original draft. Alessio Ricci: Supervision, Data curation, Writing—review & editing. Massimiliano Burlando: Conceptualization, Supervision, Writing—review & editing, Funding acquisition, Project administration.

Funding Open access funding provided by Università degli Studi di Genova within the CRUI-CARE Agreement. This study was funded by ADAPTNOW Project—"ADAPTation Capacity Strengthening for Highly Affected and Exposed Territories in the Alps NOW".

Data availability Data will be made available on request.

Code availability Code is provided within the manuscript or supplementary information files.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

Appendix 1: Python code for geometry generation

```
# ------------------------------------------------------------ #
# import SolidPython2 library
from solid2 import *

def myBuildings():

    # NOTE: union is an OpenSCAD operator for Boolean operations
    allBuildings = union()

    # Loop across every building class
    for ii, region in enumerate(regionID):

        # Loop through all files
        for jj, file in enumerate (files):
            .
            .
            .
            # Extract (x,y,z) coords of every point
            for point in lines:

                # Variable to hold (x,y) coords
                coords = []

                # Group building coords
                line = point.split()
                coords.append(float(line[0]))
                coords.append(float(line[1]))
                zCoord = float(line[2])

                # Collect (x,y) coords to define polygon
                buildingShape.append(coords)

            # 1) Create building from polygon data and
            #    increase its height for 5 m
            myBuilding = \
                linear_extrude(height=zCoord+5)\
                    (polygon(buildingShape))

            # 2) Translate downwards for 5 m to
            #    guarantee the terrain intersection
            myBuilding = translate(0, 0, -5) \
                (myBuilding)

            allBuildings.add(myBuilding)
        .
        .

            .
    return (allBuildings)
# ------------------------------------------------------------ #

# ------------------------------------------------------------ #
if __name__ == '__main__':
    out_dir = sys.argv[1] if len(sys.argv) > 1 else os.curdir
    file_out = os.path.join(out_dir, 'output.scad')
    a = myBuildings()
    scad_render_to_file( \
        a, file_out, file_header='$fn = %s;' % SEGMENTS)

print('End.')
# ------------------------------------------------------------ #
```

References

1. Solari G (2019) Wind science and engineering: origins, developments, fundamentals and advancements (Springer tracts in civil engineering). Springer, Cham
2. Leng S, Li SW, Hu ZZ, Wu HY, Li BB (2022) Development of a micro-in-meso-scale framework for simulating pollutant dispersion and wind environment in building groups. J Clean Prod 364:132661
3. Janssen W, Blocken B, van Hooff T (2013) Pedestrian wind comfort around buildings: comparison of wind comfort criteria based on whole-flow field data for a complex case study. Build Environ 59:547–562
4. Toparlar Y, Blocken B, Vos P, van Heijst G, Janssen W, van Hooff T, Montazeri H, Timmermans H (2015) CFD simulation and validation of urban microclimate: a case study for Bergpolder Zuid, Rotterdam. Build Environ 83:79–90
5. Blocken B, van der Hout A, Dekker J, Weiler O (2015) CFD simulation of wind flow over natural complex terrain: case study with validation by field measurements for Ria de Ferrol, Galicia, Spain. J Wind Eng Ind Aerodyn 147:43–57
6. Antoniou N, Montazeri H, Neophytou M, Blocken B (2019) CFD simulation of urban microclimate: validation using high-resolution field measurements. Sci Total Environ 695:133743
7. Ricci A, Vasaturo R, Blocken B (2023) An integrated tool to improve the safety of seaports and waterways under strong wind conditions. J Wind Eng Ind Aerodyn 234:105327
8. Fujita TT (1981) Tornadoes and downbursts in the context of generalized planetary scales. J Atmos Sci 38(8):1511–1534
9. Hjelmfelt M (1988) Structure and life cycle of microburst outflows observed in Colorado. J Appl Meteorol 27:900–927
10. Repetto MP, Burlando M, Solari G, De Gaetano P, Pizzo M, Tizzi M (2018) A web-based GIS platform for the safe management and risk assessment of complex structural and infrastructural systems exposed to wind. Adv Eng Softw 117:29–45
11. Solari G, Burlando M, Repetto MP (2020) Detection, simulation, modelling and loading of thunderstorm outflows to design wind-safer and cost-efficient structures. J Wind Eng Ind Aerodyn 200:104142
12. Stathopoulos T, Alrawashdeh H (2020) Wind loads on buildings: a code of practice perspective. J Wind Eng Ind Aerodyn 206:104338
13. Orlanski I (1975) A rational subdivision of scales for atmospheric processes. BAMS 56(5):527–530
14. Byers HR, Braham R (1949) The thunderstorms. U.S. Govt. Printing Office, Washington
15. Burlando M, Romanic D, Boni G, Lagasio M, Parodi A (2020) Investigation of the weather conditions during the collapse of the Morandi bridge in Genoa on 14 August 2018 using field observations and WRF model. Atmosphere 11(7):724
16. Ricci A, Burlando M, Repetto M, Blocken B (2022) Static downscaling of mesoscale wind conditions into an urban canopy layer by a CFD microscale model. Build Environ 225:109626
17. Alemayehu TF, Bitsuamlak GT (2022) Autonomous urban topology generation for urban flow modelling. Sustain Cities Soc 87:104181
18. Paden I, García-Sánchez C, Ledoux H (2022) Towards automatic reconstruction of 3D city models tailored for urban flow simulations. Front. Built Environ 8
19. Weller H, Tabor G, Jasak H, Fureby C (1998) A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 12:620–631
20. NASA JPL (2013) NASA Shuttle Radar Topography Mission Global 1 arc second. distributed by NASA EOSDIS Land Processes Distributed Active Archive Center. https://doi.org/10.5067/MEaSUREs/SRTM/SRTMGL1.003
21. Blocken B, Janssen W, van Hooff T (2012) CFD simulation for pedestrian wind comfort and wind safety in urban areas: general decision framework and case study for the Eindhoven University campus. Environ Model Softw 30:15–34
22. Ricci A, Blocken B (2020) On the reliability of the 3D steady RANS approach in predicting microscale wind conditions in seaport areas: the case of the IJmuiden sea lock. J Wind Eng Ind Aerodyn 207:104437
23. Ricci A, Janssen W, van Wijhe H, Blocken B (2020) CFD simulation of wind forces on ships in ports: case study for the Rotterdam cruise terminal. J Wind Eng Ind Aerodyn 205:104315
24. Žužul J, Ricci A, Burlando M, Blocken B, Solari G (2023) CFD analysis of the WindEEE dome produced downburst-like winds. J Wind Eng Ind Aerodyn 232:105268
25. Wieringa J (1992) Updating the davenport roughness classification. J Wind Eng Ind Aerodyn 41(1):357–368
26. Ramer U (1972) An iterative procedure for the polygonal approximation of plane curves. Comput Graph Image Process 1(3):244–256
27. Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Can Cartogr 10(2):112–122
28. OpenSCAD (2024) The programmers solid 3D CAD Modeller. https://openscad.org/
29. Blocken B, Stathopoulos T, Carmeliet J (2007) CFD simulation of the atmospheric boundary layer: wall function problems. Atmos Environ 41(2):238–252
30. Canepa F, Massimiliano B, Solari G (2020) Vertical profile characteristics of thunderstorm outflows. J Wind Eng Ind Aerodyn 206:104332
31. Xhelaj A, Burlando M, Solari G (2020) A general-purpose analytical model for reconstructing the thunderstorm outflows of travelling downbursts immersed in ABL flows. J Wind Eng Ind Aerodyn 207:104373

32. Xhelaj A, Burlando M (2022) Application of metaheuristic optimization algorithms to evaluate the geometric and kinematic parameters of downbursts. Adv Eng Softw 173:103203
33. Canepa F, Burlando M, Romanic D, Hangan H (2024) Effect of surface roughness on large-scale downburst-like impinging jets. Phys Fluids 36(3):036610