



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Software development for the optimization of the influence of wind flows within energy applications and sustainable town planning

by
Nicola Germano

ADVERTIMENT La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del repositori institucional UPCommons (<http://upcommons.upc.edu/tesis>) i el repositori cooperatiu TDX (<http://www.tdx.cat/>) ha estat autoritzada pels titulars dels drets de propietat intel·lectual **únicament per a usos privats** emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei UPCommons o TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a UPCommons (*framing*). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del repositorio institucional UPCommons (<http://upcommons.upc.edu/tesis>) y el repositorio cooperativo TDR (<http://www.tdx.cat/?locale-attribute=es>) ha sido autorizada por los titulares de los derechos de propiedad intelectual **únicamente para usos privados enmarcados** en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio UPCommons No se autoriza la presentación de su contenido en una ventana o marco ajeno a UPCommons (*framing*). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the institutional repository UPCommons (<http://upcommons.upc.edu/tesis>) and the cooperative repository TDX (<http://www.tdx.cat/?locale-attribute=en>) has been authorized by the titular of the intellectual property rights **only for private uses** placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading nor availability from a site foreign to the UPCommons service. Introducing its content in a window or frame foreign to the UPCommons service is not authorized (*framing*). These rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.



Earth Systems and Built Environments (UNICH) - Civil Engineering (UPC)

XXXIII Cycle - PON

ING-IND/11

Software development for the optimization of the influence of wind flows within energy applications and sustainable town planning

Thesis by Nicola Germano

Supervisor: Prof. Guido Camata
Co-Supervisors: Prof. Riccardo Rossi
Prof. Sergio Montelpare
PhD Coordinators: Prof. Isabella Raffi (UNICH)
Miguel Cervera Ruiz (UPC)

Academic year 2020/2021

Abstract

This thesis aims to propose and validate an innovative, fully open-source framework capable of performing multiscale analysis for the assessment of local wind flows within the urban fabric. Each part of the framework is fully editable and license-free.

A crucial aspect of the adopted methodology concerns the coupling between the mesoscale-microscale analysis in order to increase the accuracy of the final results. In detail, the procedure is based on the interaction between the mesoscale values and the microscale values obtained considering different wind directions.

The core of the work is the design and development of an open-source application that allows to generate 3D numerical models for microscale analysis in an automatic way and providing only some basic information. The main benefit of such a procedure is the drastic reduction of the time required for the creation of numerical models and the facilitation, in general, of microscale simulations. The process of geodata retrieval and the subsequent 3D modeling phase, in fact, are completely automated. The results obtained with the above application are compared with those obtained with a commercial software, widely used in the sector, in order to test its potential and accuracy.

Finally, the airflows estimated through the application of the whole proposed framework are used as input for dynamic energy simulations to identify the energy consumption, divided into heating and cooling, of a real building located in an urban context.

The framework has been tested assuming a domain located in the city of Pescara (central Italy). However, it is important to emphasize that its application to different urban contexts does not present any constraint related, for example, to the geographical location of the area of interest and that it is, therefore, possible to replicate the analysis in any part of the world.

Keywords:

Multiscale approach, Mesoscale-Microscale coupling procedure, CFD urban analyses, Building Energy Simulations

Resumen

La presente tesis tiene como objetivo proponer y validar un marco de trabajo innovador y totalmente de código abierto (open-source) capaz de realizar análisis multiescala para evaluar los flujos de vientos locales dentro del tejido urbano. Cada etapa del marco es totalmente editable y libre de licencia.

Uno de los aspectos cruciales de la metodología adoptada es el acoplamiento entre las simulaciones mesoescala-microescala, las cuales permiten aumentar la precisión de los resultados finales obtenidos. Concretamente, el procedimiento se basa en la interacción entre los valores mesoescala y microescala obtenidos considerando diferentes direcciones del viento.

El núcleo de la tesis es el diseño y desarrollo de una aplicación open-source capaz de generar automáticamente modelos 3D para el análisis microescala, proporcionando únicamente ciertos datos básicos. El principal beneficio de este procedimiento es la drástica reducción del tiempo requerido para la creación de modelos numéricos y la facilidad, en general, de las simulaciones microescala. El proceso de recuperación de datos de carácter geográficos (geodata) y la posterior fase de modelado 3D están, de hecho, completamente automatizados. Los resultados obtenidos con la aplicación desarrollada son comparados con los generados a través de un software comercial, el cual es usado ampliamente en el sector, con el objetivo de validar y probar el potencial y precisión de la aplicación.

Finalmente, los flujos de aire calculados a través de la aplicación del marco de trabajo propuesto son usados como datos de entrada para la realización de simulaciones energéticas dinámicas, las cuales permiten identificar el consumo de energía, dividido este en los requisitos de calefacción y refrigeración, de un edificio real ubicado dentro de un contexto urbano.

El marco de trabajo y la metodología adoptada han sido testeados asumiendo un dominio local en la ciudad de Pescara, ubicada en el centro de Italia. No obstante, es importante destacar que su aplicación a diferentes contextos urbanos no presenta ninguna restricción, como por ejemplo la ubicación geográfica de interés, siendo por tanto posible replicar el análisis en cualquier parte del mundo independientemente de la ubicación del caso de estudio.

Sommario

La presente tesi intende proporre e validare un framework innovativo, completamente open-source, in grado di eseguire analisi multiscala per la valutazione dei flussi del vento locale all'interno del tessuto urbano. Ogni fase del framework è interamente editabile e a licenza gratuita.

Aspetto cruciale della metodologia adottata riguarda l'accoppiamento tra le analisi mesoscala-microscala al fine di incrementare l'accuratezza dei risultati finali. Nel dettaglio, la procedura si basa sul far dialogare i valori stimati dalla mesoscala con quelli della microscala ottenuti considerando diverse direzioni del vento in ingresso.

Il nucleo principale del lavoro è la progettazione e lo sviluppo di un'applicazione in ambiente open-source che permetta di generare modelli numerici 3D per le analisi microscala in maniera automatica e fornendo solo alcune informazioni basiche. Beneficio principale di una procedura così individuata è la drastica riduzione dei tempi necessari per la realizzazione di modelli numerici e l'agevolazione, in generale, delle simulazioni microscala. Il processo di reperimento dei geodati e la successiva fase di modellazione 3D, infatti, sono completamente automatizzate. I risultati ottenuti con la suddetta applicazione sono confrontati con quelli ottenuti con un software commerciale, largamente utilizzato nel settore, al fine di testarne le potenzialità e l'accuratezza.

Infine, i flussi di aria stimati mediante l'applicazione dell'intero framework proposto, sono impiegati come input per le simulazioni energetiche dinamiche per identificare il consumo energetico, suddiviso in riscaldamento e raffrescamento, di un edificio reale localizzato in un contesto urbano.

Il framework è stato testato assumendo un dominio situato nella città di Pescara (centro Italia). Tuttavia, è importante sottolineare che, la sua applicazione a diversi contesti urbani non presenta alcun vincolo legato, ad esempio, alla posizione geografica dell'area di interesse e che è quindi possibile replicare le analisi in qualsiasi parte del mondo.

Acknowledgements

First and foremost, I would like to thank my supervisors Prof. Guido Camata, Prof. Riccardo Rossi, and Prof. Sergio Montelpare, who made me to reach a Goal that seemed too far away when I started. Indeed, without their knowledge and experience it would have been impossible for me to complete this journey and reach all the other goals I set for myself.

Then, I would like to thank my family for their endless and continuous support in all the choices I made. You helped me become the man I am today.

A special thank you goes to my dear friend Camilla Lops, who has given me way more help than I deserved, often putting aside personal matters in order to support me. I could never thank you enough. You are truly a special human being, and anyone in the world would want to call you friend.

In addition, I would like to thank all my colleagues and friends, in the INGEO department (UNICH) and those at the UPC, who shared this journey with me (including both the ups and downs). Your support meant a lot to me. If I haven't gone crazy (until now), I owe it also to you.

Last but not least, I would like to thank my girlfriend Mariantonietta, who always believed in me (especially when I did not) and always gave me the strength and the enthusiasm to achieve my professional and personal goals. You make me a better person.

Contents

Introduction.....	1
1 Literature review	5
1.1 Applications of the mesoscale- microscale approach.....	11
1.2 The need of geospatial data for microscale simulations	15
2 Theory of Meteorology and Environmental Fluid Dynamics.....	17
2.1 Earth’s Atmosphere	17
2.2 The global wind.....	20
2.3 The local wind	23
2.4 The wind roses	25
2.5 Wind variation laws.....	26
2.6 Atmospheric turbulence.....	28
2.7 Urban Fluid Dynamics	29
2.8 Computational Fluid Dynamics	32
2.8.1 Mass conservation	33
2.8.2 Conservation of momentum.....	33
2.8.3 Energy conservation.....	33
2.9 General aspects of turbulence.....	34
2.10 Direct Numerical Simulation.....	36
2.11 Large Eddy Simulation.....	36
2.12 Reynolds Averaged Navier-Stokes	36
2.13 Turbulent viscosity models.....	38
3 Framework developed	40
3.1 Mesoscale analysis.....	41
3.2 Microscale analysis	42

3.3	Coupling procedure.....	42
3.4	Building Energy Simulations.....	43
4	Mesoscale simulations.....	44
4.1	Numerical Model.....	44
4.2	Numerical set-up.....	50
4.3	Mesoscale results.....	54
5	Microscale simulations.....	66
5.1	Numerical model.....	66
5.2	Numerical set-up.....	72
5.3	Microscale results.....	75
6	Coupling procedure.....	85
6.1	Basic principles.....	85
6.2	Results.....	89
7	Dynamic Energy Simulations.....	91
7.1	Presentation of the case study.....	91
7.2	Numerical model and set-up.....	95
7.3	Results.....	99
8	Open-source application development.....	101
8.1	Geodata processing pipeline.....	102
8.2	Datasets.....	103
8.2.1	Terrain dataset.....	103
8.2.2	Building dataset.....	104
8.3	Numerical model.....	106
8.3.1	Terrain model.....	106
8.3.2	Buildings model.....	109
8.3.3	Final model.....	113
8.4	Boundary conditions.....	124

8.5	Results of CFD analysis	125
9	Comparison between open-source and commercial software.....	128
9.1	Numerical model comparison	129
9.2	Results comparison.....	132
9.3	Considerations.....	134
10	Conclusions.....	136
10.1	Major results.....	136
10.2	Future works.....	137
A	Appendix.....	139
B	Appendix.....	145
C	Appendix.....	153
	Reference.....	162

List of Figures

Figure 1.1 - Temperature from 1854 to 2021 for six data sets (figure from [1])	6
Figure 1.2 - Surface air temperature anomaly for August 2021 (figure from https://climate.copernicus.eu/surface-air-temperature-august-2021).....	6
Figure 1.3 - World population in urban and rural areas (figure from [6])	7
Figure 1.4 - Schematic representation of spatial scales in urban physics (figure based on [7])	9
Figure 1.5 - Coupling procedure performed within the same solver (a) and coupling performed through a software interface that transfers both lateral boundary conditions and other internal domain parameters (b).....	10
Figure 1.6 - Distribution of the locations of the CFD microclimate studies (figure from [10]).....	11
Figure 1.7 - Photogrammetry technique to reconstruct objects in 3D (figure from https://bitfab.io/blog/photogrammetry/)	15
Figure 1.8 - Representation of the five LODs (figure from [36])	16
Figure 2.1 - Atmospheric stratification scheme	18
Figure 2.2 - Atmospheric circulation effects	20
Figure 2.3 - Cyclones and Anticyclones.....	22
Figure 2.4 - The Ekman spiral.....	23
Figure 2.5 - Sea (a) and land (b) breezes.....	24
Figure 2.6 - Example of wind rose	25
Figure 2.7 - Gradient height for various types of surface roughness	26
Figure 2.8 - Roughness length near the ground.....	27
Figure 2.9 - Van der Hoven spectrum	28
Figure 2.10 - Barrier exposed to an orthogonal flow: (a) streamlines; (b) flow zones	30

Figure 2.11 - Simple building exposed to an orthogonal flow: (a) lateral view of streamlines and flow zones (undisturbed flow [A], displacement zone [B], cavity [C] and wake [D]); (b) vertical profiles of velocity and flow zones; plan view of streamlines around a normally stream-oriented building (c) and diagonally to the stream (d).....	31
Figure 2.12 - Flow in different urban scenarios: (a) isolated roughness flow; (b) wake interference flow; (c) skimming flow	32
Figure 3.1 - Characteristic time and horizontal scales of different atmospheric phenomena (modified from Olsen [39])	41
Figure 4.1 - The MM5 modeling system flow chart [41].....	45
Figure 4.2 - Schematic representation of the vertical structure of the model [41].	46
Figure 4.3 - Arakawa B-grid schema [41]	47
Figure 4.4 - Graphical representation of NESTI and NESTJ	52
Figure 4.5 - Domain nesting.....	55
Figure 4.6 - Extraction points of the virtual anemometers	56
Figure 4.7 - Annual wind roses: MM5-MRF@10m.....	57
Figure 4.8 - Annual wind roses: MM5-MRF@92m.....	57
Figure 4.9 - Annual wind roses: MM5-MRF@204m	58
Figure 4.10 - Annual wind roses: MM5-PleimXiu@10m.....	58
Figure 4.11 - Annual wind roses: MM5-PleimXiu@92m.....	59
Figure 4.12 - Annual wind roses: MM5-PleimXiu@204m	59
Figure 4.13 - Annual wind roses: MM5-BK@10m.....	60
Figure 4.14 - Annual wind roses: MM5-BK@92m.....	60
Figure 4.15 - Annual wind roses: MM5-BK@204m	61
Figure 4.16 - Wind roses for the point (14,12): (a) MM5-MRF@10m, (b) MM5-MRF@204m, (c) MM5-PleimXiu@10m, (d) MM5- PleimXiu@204m, (e) MM5-BK@10m, (f) MM5-BK@204m	63

Figure 4.17 - Wind roses for the point (14,19): (a) MM5-MRF@10m, (b) MM5-MRF@204m, (c) MM5-PleimXiu@10m, (d) MM5- PleimXiu@204m, (e) MM5-BK@10m, (f) MM5-BK@204m	64
Figure 4.18 - Wind roses for the point (16,16): (a) MM5-MRF@10m, (b) MM5-MRF@204m, (c) MM5-PleimXiu@10m, (d) MM5- PleimXiu@204m, (e) MM5-BK@10m, (f) MM5-BK@204m	65
Figure 5.1 - (a) division of a sheet into 16 sections, (b) subdivision of a section into four elements	67
Figure 5.2 - Sheet numbers of the Abruzzo region.....	68
Figure 5.3 - CTR result for the area of interest.....	69
Figure 5.4 - Elevation of buildings in the CTR in plan (a) and section (b) view	69
Figure 5.5 - Terrain surface obtained from CTR data.....	70
Figure 5.6 - Circular portion from CTR.....	71
Figure 5.7 - Buildings extracted from CTR: (a) top view and (b) axonometric view	71
Figure 5.8 - Axonometric view with identification of components.....	72
Figure 5.9 - Axonometric view of buildings.....	72
Figure 5.10 - Schematization of the BCs.....	73
Figure 5.11 - Volume mesh of the numerical model.....	74
Figure 5.12 - Detailed mesh on buildings.....	75
Figure 5.13 - Lateral BCs for North (a), East (b), South (c), and West (d)	75
Figure 5.14 – Top view of velocity magnitudes: incoming wind from North (-y) ...	76
Figure 5.15 – Top view of pressure levels and streamlines: incoming wind from North (-y)	77
Figure 5.16 - Section view of velocity magnitudes on the reference building and pressure levels: incoming wind from North (-y).....	77
Figure 5.17 - Axonometric view of streamlines in the urban fabric: incoming wind from North (-y).....	78
Figure 5.18 – Top view of velocity magnitudes: incoming wind from East (-x)	78

Figure 5.19 – Top view of pressure levels and streamlines: incoming wind from East (-x).....79

Figure 5.20 - Section view of velocity magnitudes on the reference building and pressure levels: incoming wind from East (-x).....79

Figure 5.21 - Axonometric view of streamlines in the urban fabric: incoming wind from East (-x).....80

Figure 5.22 – Top view of velocity magnitudes: incoming wind from South (+y)..80

Figure 5.23 – Top view of pressure levels and streamlines: incoming wind from South (+y).....81

Figure 5.24 - Section view of velocity magnitudes on the reference building and pressure levels: incoming wind from South (+y)81

Figure 5.25 - Axonometric view of streamlines in the urban fabric: incoming wind from South (+y).....82

Figure 5.26 – Top view of velocity magnitudes: incoming wind from West (+x) ...82

Figure 5.27 – Top view of pressure levels and streamlines: incoming wind from West (+x)83

Figure 5.28 - Section view of velocity magnitudes on the reference building and pressure levels: incoming wind from West (+x).....83

Figure 5.29 - Axonometric view of streamlines in the urban fabric: incoming wind from West (+x)84

Figure 6.1 - Flowchart microscale side86

Figure 6.2 - Flowchart mesoscale side.....87

Figure 6.3 - Wind roses for the point (16,16): (a) MM5-MRF_pre, (b) MM5-MRF_post, (c) MM5-PleimXiu_pre, (d) MM5- PleimXiu_post, (e) MM5-BK_pre, (f) MM5-BK_post90

Figure 7.1 - Location of the case study92

Figure 7.2 - Reference building for the energy analyses92

Figure 7.3 - Architectural plan of the ground floor93

Figure 7.4 - Architectural plan of the typical floor.....93

Figure 7.5 - South-East elevation.....94

Figure 7.6 - North-West elevation	94
Figure 7.7 - North-East elevation (a) and South-West elevation	94
Figure 7.8 - Thermal zones on the typical floor	95
Figure 7.9 - The numerical model of the case study.....	98
Figure 7.10 - Building heating consumption.....	100
Figure 7.11 - Building cooling consumption.....	100
Figure 8.1 - ASTER-GDEM tile of interest area	104
Figure 8.2 - Multipolygon in OSM. way_1: outer perimeter, way_2: inner perimeter	105
Figure 8.3 - Area analyzed in this work: pre-existing buildings (in yellow) and the areas where buildings are added (in red)	106
Figure 8.4 - ASTER-GDEM tile in which the domain falls (left), and the bounding box of the domain (right)	107
Figure 8.5 - 3D model of the terrain.....	107
Figure 8.6 - Domain positioned at the intersection of 4 tiles.....	108
Figure 8.7 - Portions of the terrain surface.....	109
Figure 8.8 - 3D model of the buildings obtained through the developed procedure (Top view - city of Pescara)	111
Figure 8.9 - 3D model of the buildings obtained through the developed procedure (Axonometric view - city of Pescara)	112
Figure 8.10 - Shifting of buildings on the terrain surface.....	113
Figure 8.11 - Axonometric view of the terrain surface and building geometries.	113
Figure 8.12 - Refinement mesh procedure: (a) distance field step 1, (b) distance field step 2, (c) distance field step 3, and (d) result	114
Figure 8.13 - Large scale test with refinement process: axonometric view	115
Figure 8.14 - Large scale test with refinement process: section on a building....	115
Figure 8.15 - Terrain surface with the building footprints.....	116
Figure 8.16 - Terrain surface with the building geometries.....	116

Figure 8.17 - (a) terrain surface before the "drilling" procedure with the new building nodes highlighted, (b) terrain surface after the "drilling" procedure.....	117
Figure 8.18 - Size of elements for each mesh. In parentheses the default values	117
Figure 8.19 - Tetrahedral mesh on a test case: axonometric view	118
Figure 8.20 - Tetrahedral mesh on a test case: detail view of buildings	118
Figure 8.21 - Numerical model generated with the developed application: terrain and building surfaces (top view).....	119
Figure 8.22 - Numerical model generated with the developed application: boundary surfaces with the top surface in transparency (axonometric view).....	120
Figure 8.23 - Numerical model generated with the developed application: detail on building surfaces (axonometric view).....	120
Figure 8.24 - Tetrahedral mesh of the area under study generated with the developed application: axonometric view	122
Figure 8.25 - Tetrahedral mesh of the area under study generated with the developed application: axonometric view of the buildings	122
Figure 8.26 - Tetrahedral mesh of the area under study generated with the developed application: detailed mesh on the buildings	123
Figure 8.27 - Tetrahedral mesh of the area under study generated with the developed application: section on the buildings.....	123
Figure 8.28 - Top view of velocity magnitudes: incoming wind from North (-y)..	126
Figure 8.29 - Top view of pressure levels and streamlines: incoming wind from North (-y)	126
Figure 8.30 - Section view of velocity magnitudes on the reference building: incoming wind from North (-y).....	127
Figure 8.31 - Axonometric view of streamlines in the urban fabric: incoming wind from North (-y).....	127
Figure 9.1 - Numerical model of terrain and buildings obtained with open-source software (a) and commercial software (b): top view.....	129

Figure 9.2 - Numerical model obtained with the open-source software (a) and with the commercial software (b): axonometric view 130

Figure 9.3 - Buildings obtained through the open-source (a) and commercial (b) software: top view 131

Figure 9.4 - Detail on buildings located in the center of the domain: open-source (a) and commercial (b) software 131

Figure 9.5 - Velocity magnitude for microscale analysis with open-source software: incoming wind from North (-y) 132

Figure 9.6 - Velocity magnitude for microscale analysis with commercial software: incoming wind from North (-y) 133

Figure 9.7 - Streamlines in the urban area obtained with the open-source software: incoming wind from North (-y) 133

Figure 9.8 - Streamlines in the urban area obtained with the commercial software: incoming wind from North (-y) 134

Figure C.1 - OSM of Rome (Italy) 154

Figure C.2 - Satellite image of Rome (Italy) 154

Figure C.3 - 3D terrain of Rome (Italy) 155

Figure C.4 - 3D building in an area located in Rome (Italy) 155

Figure C.5 - Photo of an area in Rome (Italy) 156

Figure C.6 - Detail of terrain surface and 3D buildings: Rome (Italy) 156

Figure C.7 - Terrain surface and 3D buildings: Rome (Italy) 157

Figure C.8 - OSM of San Giovanni Rotondo (Italy) 158

Figure C.9 - Satellite image of San Giovanni Rotondo (Italy) 158

Figure C.10 - 3D terrain of San Giovanni Rotondo (Italy) 159

Figure C.11 - 3D building in an area located in San Giovanni Rotondo (Italy) 159

Figure C.12 - Photo of an area in San Giovanni Rotondo (Italy) 160

Figure C.13 - Detail of terrain surface and 3D buildings: San Giovanni Rotondo (Italy) 160

Figure C.14 - Terrain surface and 3D buildings: San Giovanni Rotondo (Italy) 161

List of Tables

Table 1 - Mesoscale domains	55
Table 5.1 - BCs type	73
Table 5.2 - Mesh parameters	74
Table 7.1 - Building envelope components.....	96
Table 7.2 - Input parameters for the simulations	97
Table 8.1 - Size of regions and meshes	121
Table 8.2 - BCs assigned.....	124
Table 9.1 - Number of mesh elements	130
Table 9.2 - Final considerations.....	135

Glossary

AEP	Annual Energy Production
AMS	American Meteorological Society
ASL	Atmospheric Surface Layer
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
BC	Boundary conditions
BES	Buiding Energy Simulation
CFD	Computational Fluid Dynamics
COP	Coefficient of Performance
CTR	Carta Tecnica Regionale (regional cartography in Italian word)
d	Distance of the anemometer from the center of the domain
DEM	Digital Elevation Model
d_i	Distance of the i -th node from the center of the domain
dir	Direction of the velocity vector of the i -th anemometer
DNS	Direct Numerical Simulation
e	Specific internal energy
E	Total energy
f	Coriolis parameter
F_{Coriolis}	Coriolis force
F_{Gradient}	Gradient force
f_i	Component along the i -direction of the volume force
FNL	Final Global Data Assimilation System
GDEM	Global Digital Elevation Model
k	Turbulent kinetic energy

k_{tc}	Thermal conductivity of the fluid
k_{vk}	Von Karman constant
L	Characteristic length of the considered phenomenon
LES	Large Eddy Simulations
LiDAR	Light Detection and Ranging
LOD	Levels of Details
LSM	Land-Surface Model
m	Particle's mass
MERRA	Modern-Era Retrospective Analysis for Research and Applications
MM5	Mesoscale Model Fifth Generation
MRF	Medium-Range Forecast
NCAR	National Center for Atmospheric Research
NCEP	National Centers for Environmental Prediction
NWP	Numerical Weather Prediction
OSM	OpenStreetMap
p	Pressure
p_0	Reference-state pressure
PBL	Planetary Boundary Layer
ps_0	Reference-state surface pressure
PSU	Pennsylvania State University
p_t	Constant top pressure
QGIS	Quantum Geographic Information System
RANS	Reynolds Averaged Navier-Stokes
r_{boundary}	Radius of the domain
Re	Reynolds number
r_{ground}	Inner radius of the annular portion with simplified orography

RNG	Renormalization Group
SGS	SubGrid Scale
SRTM	Shuttle Radar Topography Mission
STAR-CCM+	Simulation of Turbulent flow in Arbitrary Regions - Computational Continuum Mechanics
T	Temperature
t	Time
U	Velocity characteristic of larger motion scales
u	Local flow velocity
U(z)	Wind velocity
u*	Friction velocity
UFD	Urban Fluid Dynamics
UHI	Urban heat island
u _i	Velocity component along the direction i
u' _i	Time-dependent fluctuating component
UN	United Nations
U-RANS	Unsteady Reynolds-Averaged Navier-Stokes
u _η	Velocity scales
V	Particle's velocity
v	Velocity of the i-th anemometer at a given altitude
WMO	World Meteorological Organization
WRF	Weather Research and Forecasting
z	Height at which the wind speed is calculated
z ₀	Altitude at which the average wind speed is zero
z _i	Coordinate of the i-th node
z _{min}	Minimum elevation of the domain
z _{ref}	Reference height at which the velocity is known

α	Coefficient that depends on the terrain type
β	Reductive coefficient
C_μ	Characteristic coefficient of the model
δ	Gradient height
δ_{ij}	Kronecker delta
ε	Turbulent kinetic energy dissipation rate
η	Kolmogorov length
μ	Molecular viscosity
ν	Kinematic diffusivity
ρ	Fluid density
τ	Time
τ_{ij}	Viscous stress tensor
ϕ	Latitude of the air particle
Ω	Earth's angular velocity
ω	Specific dissipation rate

Introduction

In recent years, the analysis of complex phenomena that occur inside urban areas has become a wide investigated topic thanks to the consciousness of their adverse effects on human beings from one side and available increment in computational power from the other.

Considering, in fact, the general and constant temperature increase due to climate change and the parallel increment of inhabitants in urban centers, the correct evaluation of these problems is crucial for identifying the optimum mitigation strategies. Furthermore, extreme climate conditions or people's deaths, directly or indirectly connected to weather scenarios, are continually on the agenda, and, for this reason, they can not be neglected anymore.

In the last decades, various works started to investigate these phenomena to find standardized and validated procedures, thus defining guidelines that could be followed by researchers worldwide. Most of these investigations adopt in situ experimental campaigns, wind tunnel tests or, finally, perform numerical simulations. Among the possible approaches, computational fluid dynamics analyses are the most widespread thanks to their versatility in predicting different scenarios and variables within a defined computational domain. Moreover, the literature review underlines that the estimations obtained by adopting this method are characterized by high accuracy in comparison to measured data. In addition, they also could be referred to future conditions, not predictable by means of in situ experimental campaigns.

Based on the aforementioned considerations, this dissertation aims to propose and validate an innovative framework, completely open-source, able to conduct multiscale analyses for estimating local wind fluxes inside urban areas. The suggested approach allows to perform coupled mesoscale-microscale simulations by setting only a few simple parameters. Furthermore, the adopted coupling procedure permits interconnecting the results obtained from different scales, from the synoptic to the local one, thus increasing the predictions' accuracy. Moreover, various wind incoming directions are taken into account for ensuring higher precision levels during the coupling stage. The central core of the framework is the design of an open-source application, developed for carrying on microscale analyses and able to generate 3D numerical models in a total automatic way. The capability and reliability of the application is largely tested by comparing the generated models with those obtained from commercial software. Finally, the air fluxes estimated by adopting the whole developed framework are assumed as input parameters for performing dynamic energy simulations, thus evaluating the impact that better boundary conditions have on the energy consumption predictions.

The proposed procedure here presented is tested, only for illustrative purposes, on the city of Pescara (Central Italy), but it can be applied without any limitations or variations to every location all over the world.

Chapter 1 describes the causes that underlie the increase of temperatures inside urban areas and their negative impacts on the health of the inhabitants. Moreover, various works that adopt the multiscale approach are reported, and some mitigation strategies are analyzed. Finally, the information regarding different geospatial data, essential for generating the numerical model for microscale analyses, is presented.

Chapter 2 introduces some main concepts of meteorology and analyzes the main physical quantities involved in the applications of the present work. Furthermore, detailed descriptions about the wind phenomenon, the laws governing its behaviour, the mechanisms of the formation and development of turbulence in the atmosphere are explained.

Chapter 3 describes the developed framework, clarifying the interconnection among the mesoscale, microscale, and energy analyses.

Chapter 4 is based on the presentation of the mesoscale simulations performed on the selected case study area. First, information about the numerical model is given by analyzing the main components of the adopted software. Then, the numerical set-up assumed for the conducted analyses is reported. In addition, the obtained results are presented in terms of annual wind roses at three points located inside the domain of interest.

Chapter 5 shows the numerical model generated for the microscale simulations carried on using commercial software. In detail, the first part of the chapter is centred on the definition of the dataset and the main procedures, both essential for creating the modelling phase. Then, the set-up for performing the microscale analysis is presented, and, finally, the obtained results are shown.

Chapter 6 depicts the coupling procedure between the mesoscale-microscale results. In particular, the method adopted for the intercommunication between the two analyses is schematized, and the estimated outputs are presented and compared.

Chapter 7 describes the dynamic energy simulations carried on by assuming as a case study a real building located in a suburb area of Pescara. Comparisons, expressed in terms of annual building energy consumptions, are reported for underlining their variability related to the use of climatic files obtained before and after the coupled procedure.

Chapter 8 presents the developed open-source application. Detailed information about the adopted pipeline is described and the same is done for the selected datasets referred to the terrain and building geometry. Moreover, the two used strategies for transforming the geospatial data into body-fitted numerical mesh are explained. Finally, the results obtained from the performed computational fluid dynamic simulations are depicted.

Chapter 9 is focused on the comparisons between the numerical model and the results generated, respectively, through the open-source and commercial procedure, highlighting similarities and main differences.

Chapter 10 draws the main conclusions of the current work and discusses possible future developments.

Finally, the main features of the developed codes are reported on Appendix A and B, whereas Appendix C shows a few examples of the potentiality of the application designed for the microscale.

This dissertation was carried out under a Co-tutoring Agreement among the “G. d’Annunzio” University of Chieti-Pescara (Italy) and the Universitat Politècnica de Catalunya (Spain) within a Double Degree Industrial PhD Programme. The proposed framework is meant to allow the combination and contamination of research skills from Academia with the user-oriented and business skills of Industry.

1

Literature review

In recent years, the rising of temperatures within cities has become a well-known phenomenon and is closely linked to climate change. The analyses conducted by the World Meteorological Organization (WMO) [1] highlight how the average temperature of the period 2017-2021 is among the hottest ever recorded in history. In particular, temperatures have increased between 1.06°C and 1.27°C in comparison to the pre-industrial era (1850-1900). Figure 1.1 shows the temperature trends for the mentioned periods (data on the year 2021 were derived based on averages from January to June).

The map of the temperature anomalies in August 2021 is, instead, depicted in Figure 1.2. In this period, precisely on August 11, a temperature value of 48.8°C was recorded in the region of Sicily in Italy. This temperature is the highest ever measured in the territory of continental Europe.

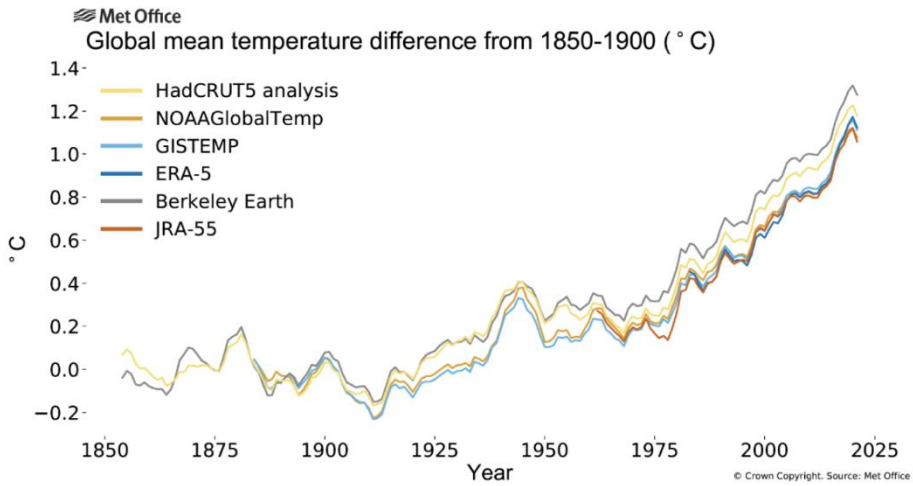


Figure 1.1 - Temperature from 1854 to 2021 for six data sets (figure from [1])

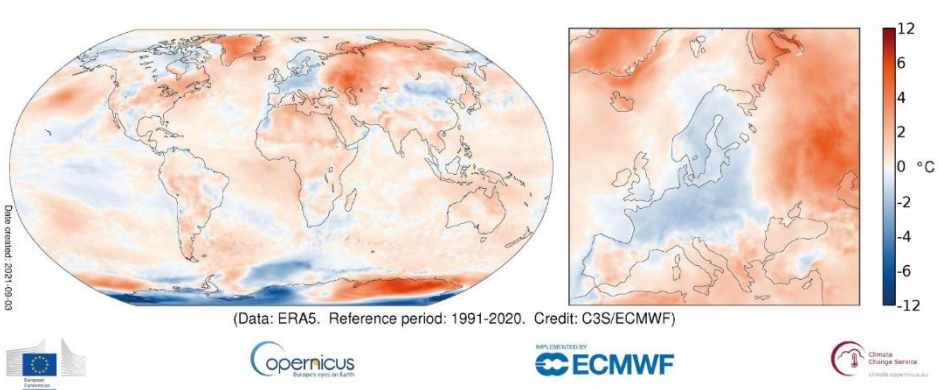


Figure 1.2 - Surface air temperature anomaly for August 2021 (figure from <https://climate.copernicus.eu/surface-air-temperature-august-2021>)

A second negative aspect is an increment in carbon dioxide concentration, which has reached almost 150% more than pre-industrial levels.

The scenario just described gives an idea of the adverse effects on people's health. In fact, the researchers conducted by Vicedo-Cabrera et al. [2] show that 37% of heatwave-related deaths can be attributed to climate change (reference period 1991-2018). In addition, it is necessary to consider the "indirect" effects that these heatwaves have on people's health. One example is the increment in the number of

Chapter 1

days when people are exposed to a very high risk of wildfires. These lead to higher air pollutants, a significant risk factor for respiratory and cardiovascular diseases [3].

Another problem that negatively interests the cities is the urban heat island (UHI) that causes more elevated local temperatures than those expected inside the surrounding less urbanized areas [4]. This phenomenon was defined for the first time in 1818 when it was observed in London by Luke Howard, a meteorologist [5].

The study of UHIs and, more generally, the urban microclimate is essential for identifying the possible impacts that such complex phenomena could have on the health of inhabitants. The research conducted by the United Nations (UN) in 2014 [6] shows that the number of people choosing to live in an urban center rather than a rural area is steadily increasing. Figure 1.3 indicates that in 1950 about 30% of the population lived in urban centers, whereas in 2014 it reached up to 54%. At this rate in 2050, a value of 66% is expected. Therefore, considering the ever-increasing number of citizens involved, the necessity of tackling all these problems that daily affect our cities represents an urgent aspect to be taken into account.

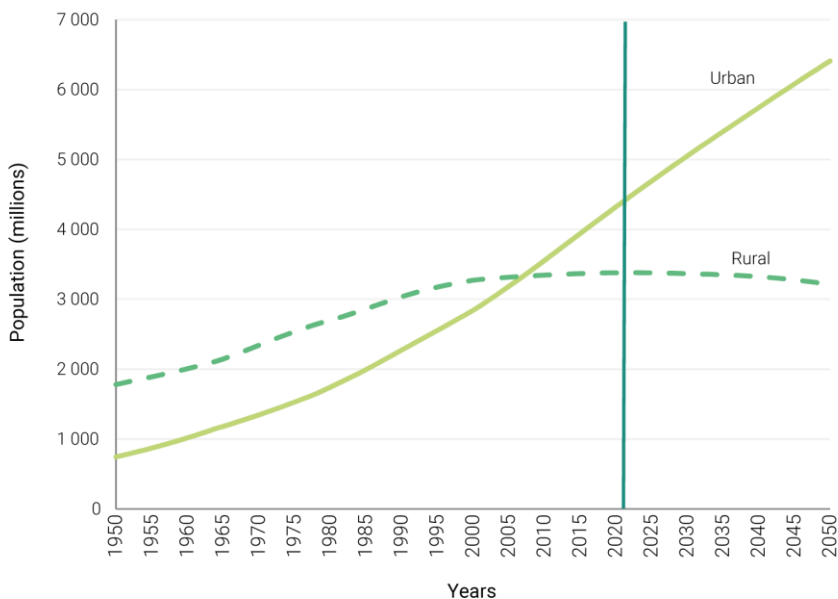


Figure 1.3 - World population in urban and rural areas (figure from [6])

For this purpose, “*urban physics*” aims to study the urban environment and its interaction with people [7] to find more efficient mitigation strategies.

Urban physics is a discipline that relates different scientific areas, even apparently very distant from each other, obtaining a more detailed picture of the phenomena that come in contact. In this way, it is possible to design the urban fabric more consciously than in the past years.

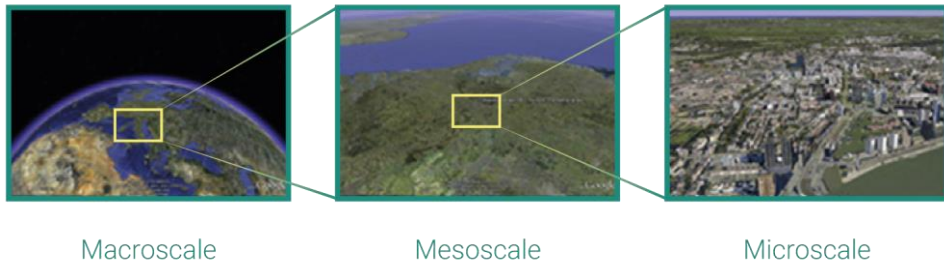
Many studies have been conducted on UHIs; Toparlar Y. et al. [8] provided a detailed list of possible causes of the UHI effects: increased shortwave and longwave radiation, decreased dispersion of longwave radiation, heat generated by anthropogenic sources, higher heat accumulation, lower evapotranspiration and wind speed. The list underlines that mitigation strategies for this phenomenon are not straightforward, and many factors must be considered.

There are several strategies to mitigate this problem. The most explored option is undoubtedly knowing how to exploit local winds to increase the convective heat and urban ventilation to reduce expected temperatures.

The study of the urban context is analyzed taking into account different spatial scales. According to the definitions provided by the American Meteorological Society (AMS), three different spatial scales can be defined [9]:

- *The Macroscale* (or synoptic scale): the scale of atmospheric motions with a typical scale of hundreds of kilometers, including phenomena such as cyclones and tropical cyclones.
- *The Mesoscale*: the scale of atmospheric phenomena with horizontal scales ranging from a few kilometers to several hundred kilometers, including thunderstorms, gale lines, fronts, precipitation bands in tropical and extratropical cyclones, and topographically generated weather systems such as mountain waves and sea and land breezes.
- *The Microscale*: the scale of atmospheric motions with Rossby numbers greater than 200.0 or spatial scales of 2.0 kilometers or less.

The above-mentioned spatial scales are schematized in Figure 1.4.



*Figure 1.4 - Schematic representation of spatial scales in urban physics
(figure based on [7])*

At the microscale, the phenomena affecting urban centers can be studied with different approaches as field measurements, full-scale or small-scale wind tunnel measurements, and numerical simulations.

Field measurements allow to obtain exact values of an area under study but are not very suitable in the design phase (both on an urban scale and for a single building). For this reason, wind tunnel measurements are more appropriate for the purpose. In addition, wind tunnel allows conducting tests on scale models of entire urban districts (as well as on the individual building) to study different scenarios. However, these analyses are costly and particularly complicated in the realization of the scale model. Thus, generally, the study on whole urban quarters is performed through numerical simulations, and the main approach is Computational Fluid Dynamics (CFD).

As underlined by Toparlar et al. [10], most of the studies (96%) in the field of CFD models referred to urban areas are conducted by adopting the RANS formulation. Only 2.8% of these works uses LES, and 1.1% adopts both LES and RANS as an approximate form of the governing Navier-Stokes equation. Even if the LES analyses are known to predict with high accuracy the estimations, they have an elevate computational effort, and, for this reason, they have received less attention compared to the RANS models [11].

A crucial aspect to be taken into account in the CFD simulations is the turbulence model. Among the possible options, the standard $k-\epsilon$ is the most adopted [12]. However, it underestimates the turbulence kinetic energy around the buildings and, instead, tends to overestimate this value in the proximity of the frontal corners of

buildings [13]. Besides this model, the realizable $k-\epsilon$ [14] and the renormalization group (RNG) $k-\epsilon$ [15] are gaining occurrence in recent works. Numerical simulations have become increasingly popular due to the increment in computational power [16], the possibility of being able to perform comparative analyses based on different scenarios [7], and the ability to derive information about each variable investigated within the computational domain [17], [18].

Considering the complexity of the scenarios under evaluation, coupling procedures between mesoscale and microscale simulations can be a successful choice. However, in the coupling approach, one of the main difficulties to be taken into account is overcoming the inability to transfer relevant atmospheric flow parameters among the mesoscale and microscale domains or codes [19]. Therefore, two different methods can be adopted to tackle these barriers. The first considers only one solver for the mesh refinement and the physics for both the mesoscale and microscale (Figure 1.5a). Instead, the other adopts two different solvers, linked via an interface through relevant flow parameters (Figure 1.5b).

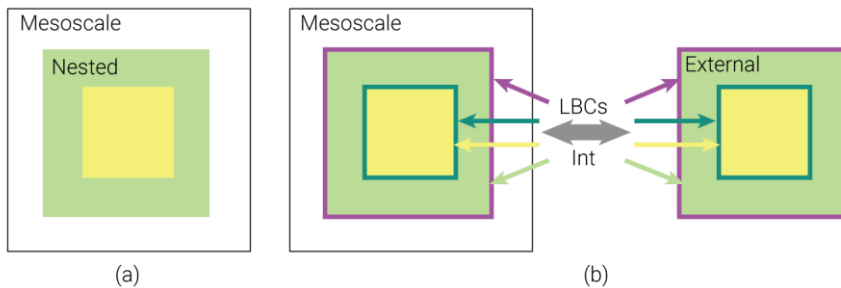


Figure 1.5 - Coupling procedure performed within the same solver (a) and coupling performed through a software interface that transfers both lateral boundary conditions and other internal domain parameters (b)

The reviews conducted by Toparlar et al. [10] and Kadaverugu et al. [20] show how many studies on the coupling procedure and CFD in urban settings are carried on to analyse different issues. Some of these topics, described in the following subparagraph, can be summarized as follows:

- pedestrian wind/thermal comfort,
- UHI simulations,

Chapter 1

- pollutant dispersion,
- snow drift,
- wind flow around buildings,
- thermal comfort and building energy consumption.

Moreover, the state-of-the-art highlights that most of the researchers centred on these topics are located in mid-latitude climates and developed regions of the world. Topalar et al. [10] counted 122 studies on real urban areas, identifying only 74 cities in 30 countries. In detail, the top five urban locations are Phoenix (USA) (7 studies), Hong Kong (China) and Cairo (Egypt) (both 6 studies), Tokyo (Japan) (5 studies) and Wuhan (China) (4 studies). Similarly, the top five countries are China (23 studies), Japan (12 studies), USA (11 studies), Germany (9 studies) and Greece (8 studies). Figure 1.6 reports a graphical schematization of the data mentioned above.

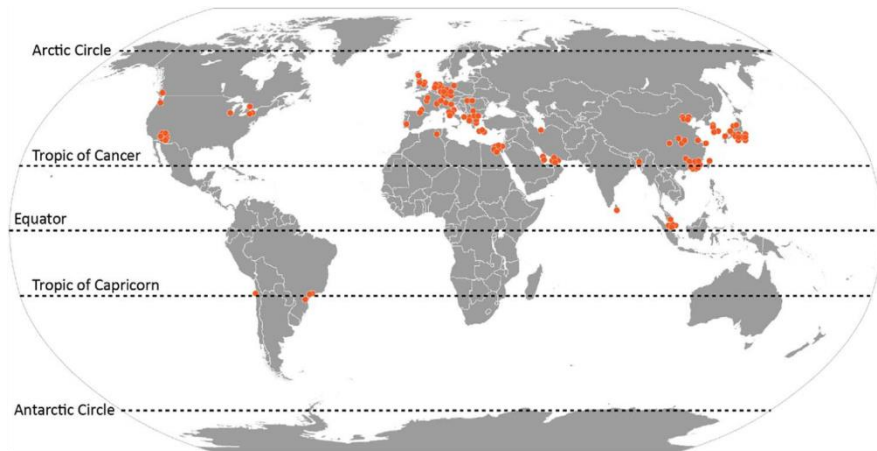


Figure 1.6 - Distribution of the locations of the CFD microclimate studies (figure from [10])

1.1 Applications of the mesoscale- microscale approach

Varies studies conducted by Miao et al. [21], [22] and Zheng et al. [23] aim to estimate the adverse effects of toxic chemical release inside dense areas by adopting a multi-scale atmospheric dispersion model. In detail, the Weather Research and Forecasting (WRF) model is combined with the OpenFOAM solver and a Lagrangian dispersion model. Based on the comparisons with wind tunnel experiments carried on by the authors, the atmospheric dispersion model confirms

1.1 Applications of the mesoscale- microscale approach

to be capable of predicting the flow pattern and concentration distribution on the investigated scales from the urban to the suburban area. Similar analyses are performed by Coirier et al. [25] to estimate the velocity and dispersion fields inside realistic cityscapes generated from Geographic Information System (GIS) data. Moreover, further investigations in this sector are done by Tewari et al. [24] that adopt two different approaches for tracing the transport and dispersion of contaminants on an urban scale. In the first case, the evaluations are based on observation obtained from a single sounding site during the experimental campaign and, in the second one, by using WRF outputs in quasi-steady mode. The study is conducted in Salt Lake City (USA). The results underline reduced velocities, turbulence, and thermodynamic fields, comparing the estimations before and after the coupling procedure, which is essential for obtaining better quality outputs.

Other works based on the use of the multiscale focus are centered on the comfort of pedestrians. For example, Blocken et al. [25] perform CFD analyses for pedestrian wind comfort and wind safety inside urban areas. In detail, the research develops a framework for obtaining information when measurements are not available. The suggested framework is applied to a real case study, the Eindhoven University of Technology campus. The results, expressed in terms of wind speeds, are compared with recorded data obtained from on-site experimental campaigns. The comparisons underline that the overall average deviation between simulated and measured wind speed is 14%.

CFD analyses are also performed by Toparlar et al. [8] to predict the urban temperature in the built-up area, assuming the Bergpolder Zuid region in Rotterdam as a case study. The research aims to provide insights for designing climate-friendly urban regions based on deterministic analyses. 3D unsteady Reynolds-averaged Navier-Stokes (U-RANS) simulations with the realizable $k-\epsilon$ turbulence model are performed on a high-resolution computational grid. In addition to the wind flow, the evaluations also include the heat transfer by conduction, convection, and radiation. Furthermore, high-resolution infrared satellite imagery is used to validate surface temperatures. The study showed that it is possible to estimate urban surface temperature with an average deviation of 7.9% from experimental data with CFD analysis. Therefore, CFD simulations are an excellent tool to identify problematic urban areas and evaluate the effects of climate adaptation measures.

Chapter 1

Besides evaluating local temperatures inside urban areas, various works are carried on to investigate local air fluxes. Li et al. [26] simulated the microscale wind in the Mount Chongli region in China to ensure success in hosting the 2022 Winter Olympic Games. In fact, understanding the characteristics of the wind field in the region of interest is essential for the safety of athletes involved in the disciplines of ski jumping, cross-country skiing, and alpine skiing. Considering the selected purpose of the work, the numerical simulation is the only possible solution in defining the wind field since the data obtained from observations is limited to a small area. Thus, the WRF and CFD models are combined as a one-way off-line nested model system and are applied to a CFD simulation. The starting point of this combination is the consideration that a mesoscale model can provide more realistic boundary conditions, while the CFD is able to produce the wind-field evaluation with a higher spatial resolution. The spatial difference between the two models is overcome by adopting the Cressman interpolation method, the best fit for coupling the WRF and the CFD model over complex terrains. These results improve the performance of the WRF-CFD model and signify a more accurate microscale wind simulation in the considered region.

Carvalho et al. [27] test the performance of three different coupling procedures to estimate the wind speed and direction and annual energy production (AEP). The WRF software is used for mesoscale analyses, whereas the microscale evaluations are carried on with WAsP software. Numerical simulations are compared with the data obtained from the measurement campaigns. With the first approach, the mesoscale outputs are fed directly into the microscale model. In this way, the wind speed is underestimated because the mesoscale model fails to correctly represent the complex terrain characteristics. In the second approach, the wind data from the mesoscale are introduced into the microscale model obtaining results like the data measured in situ. Finally, with the third procedure the mesoscale terrain data are completely ignored in the wind simulation using geostrophic winds in the wind atlas calculation. The results show a significant improvement over the first methodology but exhibit much worse performance than the second approach.

Another interesting investigation in the sector is the study of Bilal et al. [28] over a complex terrain near the Norwegian-Swedish border (Nygårdsfjell), centered on wind data from the mesoscale as input in the microscale model. Two different mesoscale analyses are conducted. The first set of mesoscale winds is modeled

1.1 Applications of the mesoscale- microscale approach

with the WRF numerical tool, whereas the second set of mesoscale winds is taken from the Modern-Era Retrospective Analysis for Research and Applications (MERRA) data system. For the microscale simulation, the adopted tool is WindSim. According to the obtained results, the performance of the microscale model is strongly dependent on the quality of the inputs provided by the mesoscale. Moreover, improvements can be obtained during the cold weather, and the coupling between WRF and WindSim provide better outputs than the MERRA-WindSim one. Over the three cases analyzed, the root means square error decreased (RMSE) by 70.9% for the February case, 61.5% for October, and 14.4% for June.

Finally, the literature review underlines that microscale-mesoscale analyses can also be adopted for performing highly accurate dynamic energy simulations. In this case, the outputs obtained from the coupling procedure, as the estimated wind velocity and direction, become part of the climatic parameters, essential for the evaluations. Various works highlight the huge impact weather data have on building energy simulations [29] [30]. The selection of reliable climatic data, in fact, is an important issue both for the correct estimation of the building energy performance and the design of retrofit solutions. According to Bhandari et al. [31], the predicted annual and monthly energy consumption can vary, respectively, by $\pm 7\%$ and $\pm 40\%$ as a function of the provided location's weather data. Thus, the correct selection of the climatic data represents a crucial phase, especially in comparing different technological options by adopting dynamic simulation programs to optimise energy performance and cost-effectiveness in various locations and climate zones.

Moreover, better accuracy in the predictions referred to real building contexts helps architects and engineers in designing energy-efficient buildings and contributing to global warming mitigation [32]. Unfortunately, recorded data are not always available for every site, and the use of estimation models or software for their generation represents the only choice. Furthermore, it is essential to bear in mind that predictions obtained from climatological models are subjected to uncertainty due to limitations of knowledge on the climate system, pollution, and available computing sources [33].

An interesting example of using the mesoscale-microscale coupling procedures for generating better quality climatic data to be adopted for dynamic simulations is the one carried on by Montepare et al. [34]. The research is based on the city of Ancona (Italy) and is centered on evaluating the feasibility of the suggested holistic

approach for improving the energy analysis of buildings inside complex urban environments.

1.2 The need of geospatial data for microscale simulations

The central core of microscale analyses is the necessity of high-resolution geospatial data to be used for evaluations. In fact, one of the main challenges in this type of simulations is the necessity of obtaining 3D models of cities, essential for the numerical domain.

In general, three different methods can be adopted to obtain reliable information for modeling locations, as identified by Kadaverugu et al. [20]. The first is Photogrammetry which is the science of making measurements from photographs, important for recovering the exact positions of surface features. This method is widely used for generating 3D surfaces on the base of the surveys recorded by micro-aerial vehicles or flying drones. However, the use of this approach is limited and inaccurate in the case of the reconstruction of large-scale objects, as underlined by Daftry et al. [35]. Figure 1.7 reports a schematization of a photogrammetric survey and its results.



Figure 1.7 - Photogrammetry technique to reconstruct objects in 3D
(figure from <https://bitfab.io/blog/photogrammetry/>)

A second approach is the Light Detection and Ranging (LiDAR) that generates 3D-based point clouds. This emerging technique uses laser light to sample the surface of the earth. The LiDAR allows to obtain highly accurate data, but it also requires elevated computational facilities. Moreover, it covers limited parts of the world,

1.2 The need of geospatial data for microscale simulations

considering their generation through the specific survey, and assuming different levels of details (LOD) [36]. For example, Kolbe [37] defines five LODs: LOD0 represents the ground footprint of the building, LOD1 is a coarse prismatic model obtained by extrusion of LOD0, LOD2 is a model with a simplified roof shape, LOD3 is an architecturally detailed model, and LOD4 includes indoor features. The representation of the five LODs is depicted in Figure 1.8.



Figure 1.8 - Representation of the five LODs (figure from [36])

Finally, the last method uses machine learning algorithms combined with 2D city data to create 3D structures by adopting multiple attributes and their combinations. Over et al. [38] selected this approach for the generation of 3D city models based on the free geodata available from the OpenStreetMap (OSM) project and public domain height information provided by the Shuttle Radar Topography Mission (SRTM).

2

Theory of Meteorology and Environmental Fluid Dynamics

The chapter consists of a brief compendium necessary to understand what will be described in the following chapters. It briefly describes the Earth's atmosphere and some basic concepts of meteorological phenomena and fluid dynamics.

The first part reports some meteorology principles and analyzes the main physical quantities involved in the research. Then, it explains what the wind is, the laws that describe the behavior, and the mechanisms of formation and development of turbulence in the atmosphere.

2.1 Earth's Atmosphere

The atmosphere is the gaseous envelope surrounding the Earth up to 700-1000 km. Commonly, it is divided into several layers with similar characteristics of the vertical temperature profile (vertical thermal gradient) or particular physical phenomena or specific chemical reactions occurring at certain altitudes. The layers with homologous properties are named "spheres", while the boundary bands are denominated "pauses" (Figure 2.1).

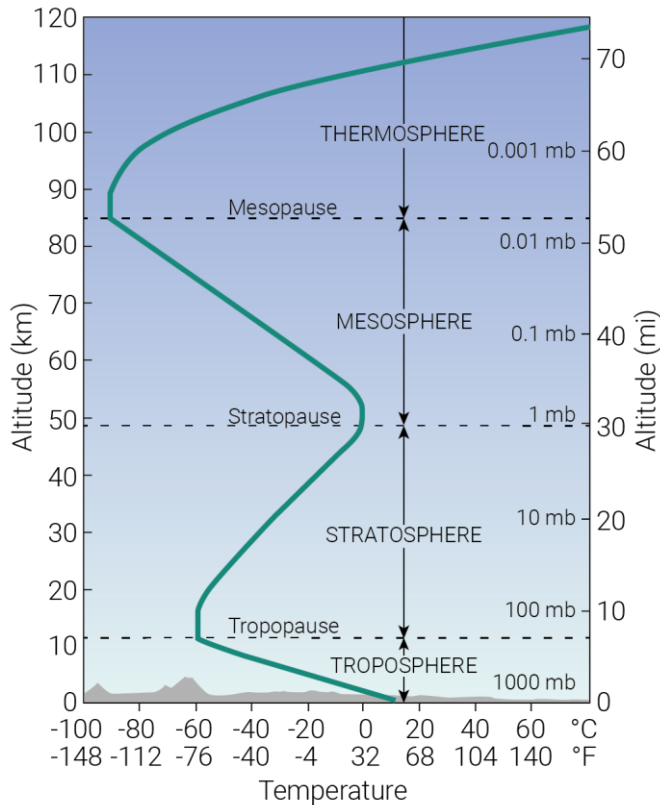


Figure 2.1 - Atmospheric stratification scheme

The closest layer to the Earth's surface (about 10 km) is called the *troposphere* and presents an average decrease in air temperature of 6.5°C every 1000 m. The temperature decrement at a specific altitude varies over time. Therefore, some days are characterized by a more intense reduction with increasing quota and others where this phenomenon is less pronounced or situations where there is a thermal inversion (the temperature increases with altitude). The temperature rise near the Earth's surface is due to the action of the sun that heats the ground, consequently increasing the temperature above it.

A fundamental characteristic of this layer is its high capacity of vertical mixing between the layers near the ground and those above. Under certain special conditions (such as those associated with thunderstorms), air particles can be transported from the terrain to troposphere limits in a few tens of minutes. This

Chapter 2

rapid mixing allows aerosols a relatively short average residence time in the troposphere (between 5 and 20 days). Aerosols are small solid or liquid particles suspended in the atmosphere as droplets or pollutant substrates of natural or anthropogenic origin.

Almost all atmospheric water vapor is present in the troposphere. For this reason, clouds and precipitation only affect this layer. The tropospheric water vapor is also the leading cause of the *greenhouse effect*.

Above the troposphere, the *tropopause* coincides with the zone where the temperature no longer decreases with height but remains constant. The next layer is called the *stratosphere* and consists of a strip of about 10 km. In this layer, first, the temperature remains constant and, subsequently, between 25 and 30 km, it increases with the altitude.

The vertical thermal gradient in this stratum is positive and generates a vertical stratification of the atmosphere (this is why it is called the "*stratosphere*"). This stratification prevents the diffusion of vertical currents from the troposphere within the stratosphere and, at the same time, limits the vertical currents within the stratosphere.

The temperature increase in the area between 20 and 50 km is due to the high absorption of solar ultraviolet rays by ozone, which reaches its most intense concentration in this part of the atmosphere. At higher altitudes, where the ozone density is lower, the highest concentrations of ultraviolet rays arrive with low wavelengths ($\sim 0.175 \mu\text{m}$), allowing an increment of the kinetic energy of the particles with a consequent temperature rise.

At an altitude of about 50 km from the ground, there is the *stratopause* and, later, the *mesosphere*. At this altitude, the temperature decreases again as the altitude increases. The low presence of ozone means that the particles' thermal balance is negative, and there is a cooling phenomenon.

At the end of the mesosphere, after the *mesopause*, the *thermosphere* begins. In this last layer, the temperature rises again with the altitude as the oxygen and nitrogen molecules absorb solar radiation at very low wavelengths. Here, the Northern Lights are due to the interaction of air molecules and charged particles from the sun.

2.2 The global wind

At a global level, the wind comes from combining the different Earth warming by the sun and the rotation around its axis. The situation is more extreme near the equator because the drag speed is higher where the heating is more intense for the solar irradiation.

The hot air at the equator (low pressure) tends to rise and close the convective cell around 30° latitude (high pressure). As it descends towards the ground, this mass of air divides into two recirculation cells: "Tropical Hadley's cell" and "Ferrel's cell" (Figure 2.2). This process generates the trade winds (from NE to SW) and the westerly winds (from SW in the northern and NW in the southern hemisphere).

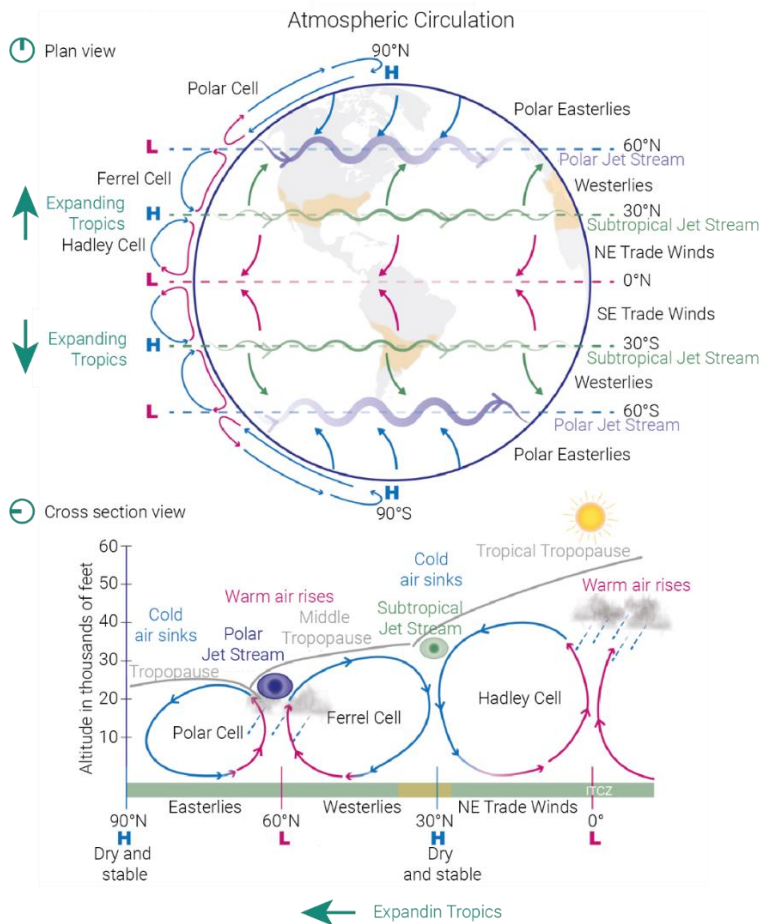


Figure 2.2 - Atmospheric circulation effects

Chapter 2

At the latitude of 60°, Western winds meet the cold air coming from the pole, incrementing the so-called polar front. The hottest tropical air tends to rise and move towards the poles on this front, creating a new cell called the “Polar Hadley’s Cell”.

Winds, therefore, are formed by the pressure difference between two points. The displacement of air masses from one point to another on the planet is due to the horizontal distribution of atmospheric pressure. The driving force that causes the movement of air masses is called the gradient force:

$$F_{\text{Gradient}} \sim \Delta p \quad (\text{Eq. 2.1})$$

Theoretically, air masses move from high-pressure areas to low-pressure areas, following a direction parallel to the pressure gradient; however, an apparent force influences the wind direction. This force is known as the *Coriolis force*:

$$\mathbf{F}_{\text{Coriolis}} = m \mathbf{f} \mathbf{V} \quad (\text{Eq. 2.2})$$

where m is the particle’s mass and V is the particle’s velocity relative to a coordinate system integral to the Earth’s surface. The term f , called the Coriolis parameter, is equal to:

$$f = 2 \Omega \sin \phi \quad (\text{Eq. 2.3})$$

where ϕ representing the latitude of the air particle and Ω the earth’s angular velocity.

Initially, there is a movement orthogonally to the isobar lines. Then, the gradient force balances with the Coriolis force, making the wind speed constant (defined as the gradient velocity).

If air movement is established around an area of high or low pressure, the balance of forces is modified, and cyclonic and anticyclonic regions are formed. In the northern hemisphere, a low-pressure zone generates a counterclockwise motion called *cyclone*. On the contrary, an area of high pressure has a clockwise motion called *anticyclone*. In the southern hemisphere, it happens exactly the opposite.

A particle starts to move from a high-pressure area to a low-pressure one and is diverted to the right (north hemisphere) by the Coriolis force. In the case of low pressure, the particle starts to move from outside to inside. Once the gradient force, added to the centrifugal one, equals the Coriolis force, the system reaches an equilibrium condition, and the air mass rotates among the isobars without intercepting them. In the case of high pressure, the particle begins to move from the center outwards. When the Coriolis force equals the sum of gradient and centrifugal forces, the motion is parallel to isobars and clockwise (Figure 2.3).

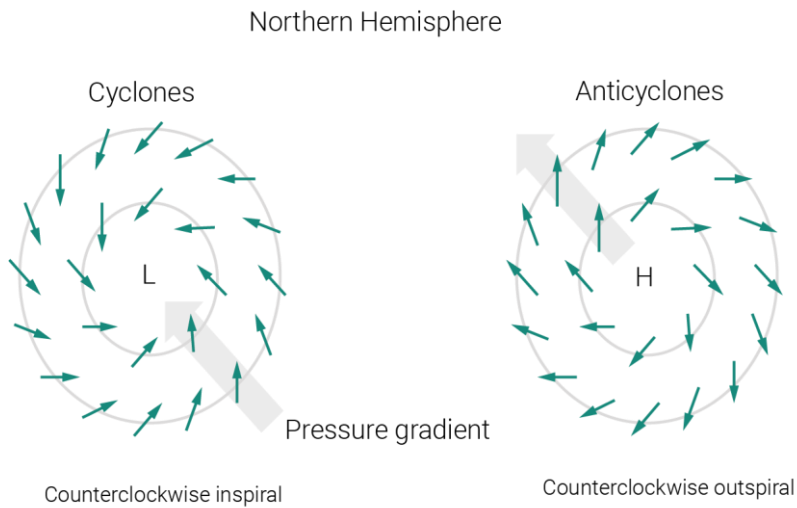


Figure 2.3 - Cyclones and Anticyclones

The effect of the friction force on particles is more significant near the Earth's surface, becoming lower at the upper layer. The main consequence of this phenomenon is the variation of the velocity modulus and direction. Specifically, the friction force causes the velocity vector to rotate counterclockwise in the northern hemisphere. The velocity vector, represented at various altitudes, describes an *Ekman spiral curve* (Figure 2.4).

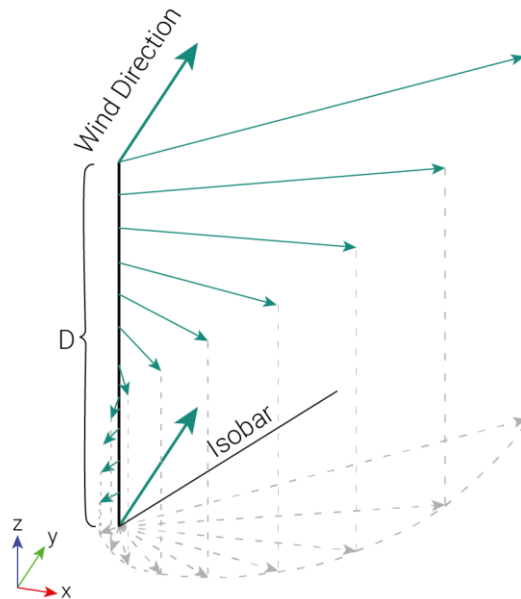


Figure 2.4 - The Ekman spiral

2.3 The local wind

At the local level, winds always arise because of the differential heating of the Earth's surface and, consequently, air masses. Typical cases are the sea or land breezes that occur with a daily frequency.

The thermal inertia during the day heats the surface with more intensity than the sea. Therefore, the air over the land tends to rise, creating a depression that draws air from the sea. Going up, the air above the ground cools down, forms cloud formations, and goes down over the sea, replacing the air masses recalled to the land. The demarcation zone between the cold, moist air from the sea and the warm, dry air over land creates the so-called sea breeze front (

Figure 2.5a). This breeze peaks typically in the afternoon hours (between 2:00 pm and 4:00 pm) with speeds of 4-8 m/s.

During the night, the opposite phenomenon happens: the sea becomes warmer than the ground, and the air recalls towards the sea. Thus, the formation of low-pressure areas over it can cause sea clouds. The *landward breeze* (

Figure 2.5b) peaks between 5:00 am and 6:00 am, reaching speeds of 2-3 m/s.

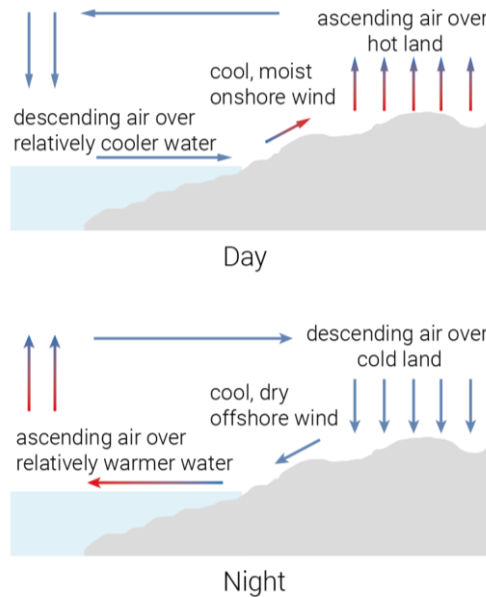


Figure 2.5 - Sea (a) and land (b) breezes

As described for the sea and land breezes, *slope breezes* blow in a cross-slope direction. During the day, the slope is warmed by the sun and draws air from the valley (*anabatic winds*). At night, instead, the valley cools more slowly draws air from the peaks (*katabatic winds*). The anabatic winds occur mainly in the warm semester and have their maximum intensity during the afternoon with a speed of 1.5-3.5 m/s. In contrast, the katabatic ones are most intense just before sunrise and are more frequent during the cold semester.

The katabatic winds coming from the ridges, at night, channel into the valley and reach the plain below. This flow is encouraged by the baric gradient that decreases towards the warmer plain (*mountain breeze*). In the presence of significant plateaus, the katabatic winds can reach high intensities, whereas gradients not very pronounced can trigger cool breezes not particularly intense. An example is the

Chapter 2

Bora, which starts from the plateaus above Serbia and Croatia where cold winds from Russia are channeled. A second case is the Mistral that descends on the Rhone valleys from the French mountains to plunge into the Mediterranean.

During the day, however, the warmer mountains draw air from the plains (valley breeze), feeding the anabatic winds. These breezes are generally much more intense than the slope breezes.

2.4 The wind roses

The wind rose shows the direction and speed of the wind for a given sampling period. Each petal of the rose shows the direction from which the wind is blowing, whereas its length indicates how often the wind is blowing from that specific direction. In the example reported in Figure 2.6, the wind rose exhibits that, during a given sampling period, the wind blows from the east 12% of the time, from the north-northwest 9% of the time, etc. The different color of the petal gives information about the wind speed and direction (in m/s). In this case, the wind blows from the east at a speed between 5.7-8.8 m/s (blue zone) 1% of the time, between 3.6-5.7 m/s (red zone) 3% of the time, between 2.1-3.6 m/s (yellow zone) 4% of the time, and between 0.5-2.1 m/s (light green zone) 4% of the time.

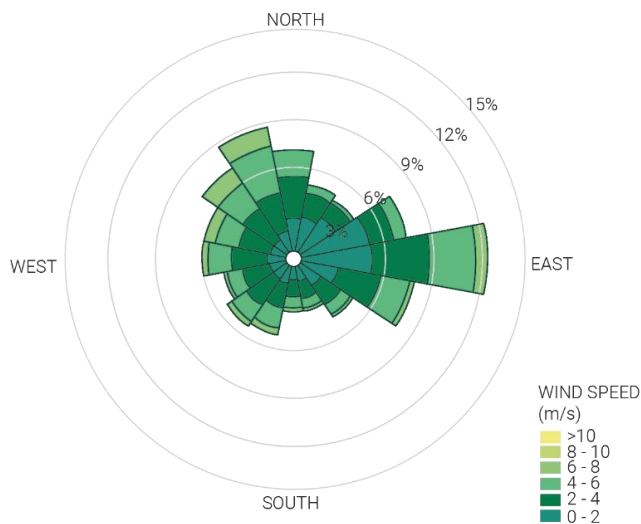


Figure 2.6 - Example of wind rose

2.5 Wind variation laws

As mentioned earlier, as you get closer to the ground, the wind speed decreases due to the friction of the air with the Earth's surface. This effect occurs within a certain thickness in the atmosphere known as the *Planetary Boundary Layer* (PBL). In this layer, the average wind speed increases with altitude up to a height beyond which its value remains constant. This height is called the *gradient height* (δ) (Figure 2.7) and depends on the air velocity and the type of Earth surface over which it flows. For this reason, the height is not constant over the entire Earth's surface: for places with slight roughness, such as desertic areas or sea surfaces, the PBL reaches about 300 m; whereas for regions with high roughness, like heavily urbanized areas, it can be up to 500 m. This phenomenon appears in the case of atmospheric thermal stability conditions when there is the absence of vertical thermal gradients in the air or, at least, less than -1°C per 100 m of ascent.

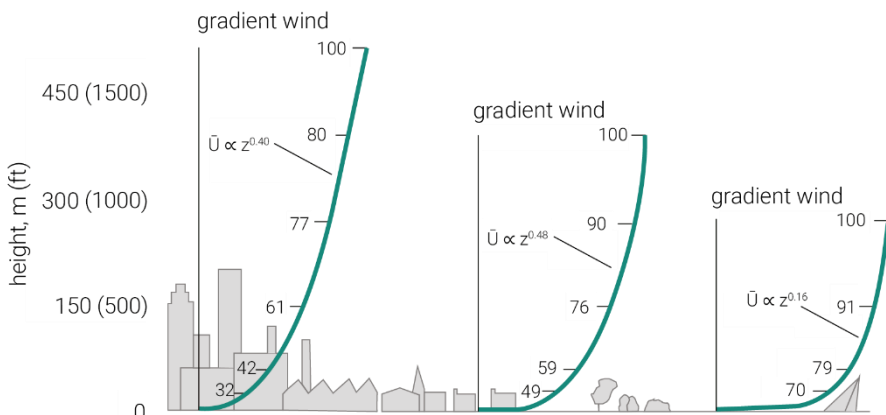


Figure 2.7 - Gradient height for various types of surface roughness

Two formulas describe the mean velocity trend as a function of the height: the *exponential law* and the *logarithmic law*. These formulations accurately describe the vertical velocity profile up to elevations of 100-200 m above the ground called the Atmospheric Surface Layer (ASL), equal to about 10% of the PBL.

Chapter 2

The *exponential law* was the first to describe the vertical profile of wind speed. It is an empirical law based on experimental data:

$$U(z) = U(z_{rif}) \left(\frac{z}{z_{rif}} \right)^\alpha \quad (\text{Eq. 2.4})$$

where α is a coefficient that depends on the terrain type, z_{ref} is the reference height at which the velocity is known, and z is the height at which the wind speed is calculated.

The logarithmic law can well describe the speed profile up to an altitude of 100 m from the ground and, in the presence of winds greater than 20 m/s can reach an altitude of 300 m. The average wind speed is given by:

$$U(z) = \frac{u^*}{k_{vk}} \ln \left(\frac{z}{z_0} \right) \quad (\text{Eq. 2.5})$$

where z_0 (roughness length) is a constant representing the surface roughness considering a horizontal, flat portion of the Earth's surface, k_{vk} is the von Karman constant (~ 0.4), and u^* is the friction velocity.

Mathematically, z_0 represents the altitude at which the average wind speed is zero, while geometrically, it is the average size of the vortices that form on the roughness of the ground (Figure 2.8).

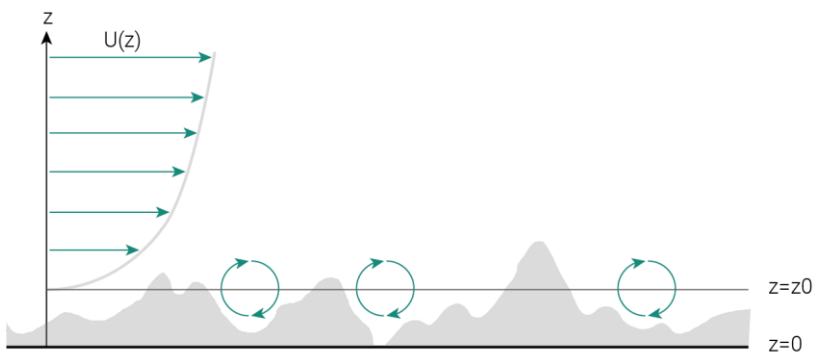


Figure 2.8 - Roughness length near the ground

2.6 Atmospheric turbulence

The phenomena involved in the atmospheric circulation are summarized by the *Van der Hoven spectrum* (Figure 2.9), a plot of the energy associated with winds as their frequency varies. Expressly, the ordinate represents the spectral density of the wind speed, i.e., the portion of energy associated with vortices of a given magnitude and time period (frequency); the abscissa reports the time steps and frequencies of variations of the wind speed.

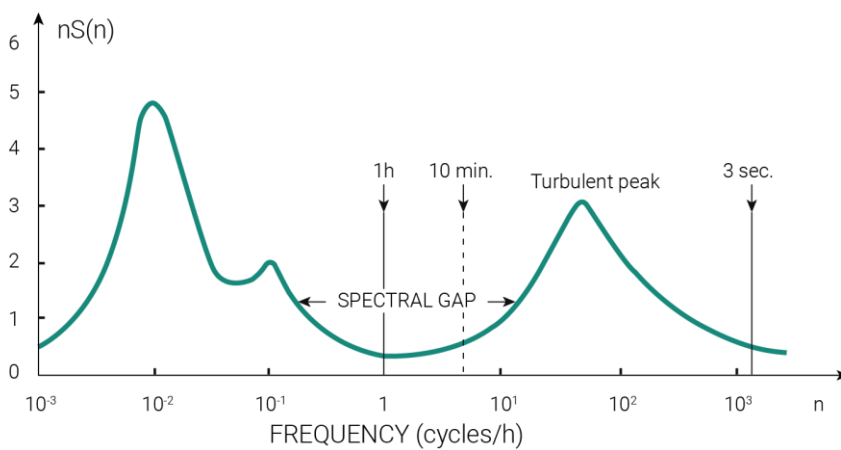


Figure 2.9 - Van der Hoven spectrum

The Van der Hoven spectrum is different for each site analyzed because the atmospheric turbulence varies with the specific terrain orography and climatic conditions. However, some general considerations can be drawn. For all frequencies, the wind speed spectral density always takes on values different from zero. This means that for all frequencies, there are energy contributions and therefore vortices of all sizes.

A zone of low energy content separating two prominent peaks is present. This zone is called a *spectral gap* and is associated with periods between 10 minutes and 1 hour. The presence of this zone indicates that no particular events are occurring in this time frame and, therefore, no velocity variations.

Chapter 2

The most significant peak, present at low frequencies, corresponds to a time scale in the order of a few days; another peak, around 24 hours, explains the fluctuations between day and night.

The peak after the spectral gap area covers periods on the order of 1 minute. This is characteristic of the microscale because it considers vortices generated by surface roughness (with dimensions on the order of cm or mm).

From a physical point of view, atmospheric turbulence is created by the mechanical stretching of moving air masses. The different air layers, positioned at different heights above the ground, move with different average horizontal velocities between them. This causes a stretching of the air particles that are contained between the layers because they are dragged with a greater speed in their higher part and, at the same time, they are slowed down in their lower part by lower velocities. As a result, a rotation of the air cell and the formation of a vortex is created. Vortices move with a speed that vectorially sums the velocity of the mean flow.

The mechanical stretching of air masses produces anisotropic vortices of considerable size, which move with speeds comparable to those of the average flow and interact with it; inside them, other vortices, considered isotropic, are characterized by smaller sizes. As a result, there is a continuous transfer of energy from the largest vortices, which take energy from the average flow to the smallest ones that, in turn, dissipate it throughout the viscous effect. This energy transfer occurs through the medium-sized vortices, also considered isotropic in which there is a balance between the forces of inertia and the viscous ones. The whole process is called the *energy cascade*.

2.7 Urban Fluid Dynamics

Urban areas are characterized by a high degree of roughness, and consequently, the motion field is very unstable. This is because the flow field is influenced by the local effects of topography: shape and size of buildings, roads, vegetation, etc.

Analyzing the simplest case of a barrier exposed to an orthogonal flow (Figure 2.10) or an isolated building with a flat roof (Figure 2.11), it is possible to distinguish three

main zones. In the first zone, the *displacement zone*, the flow first moves upwards and then downwards to avoid the barrier. After the barrier, a very turbulent *wake zone* is created, and the flow is separated. Just behind the border, however, stationary downwind eddies (*cavity zone*) are made.

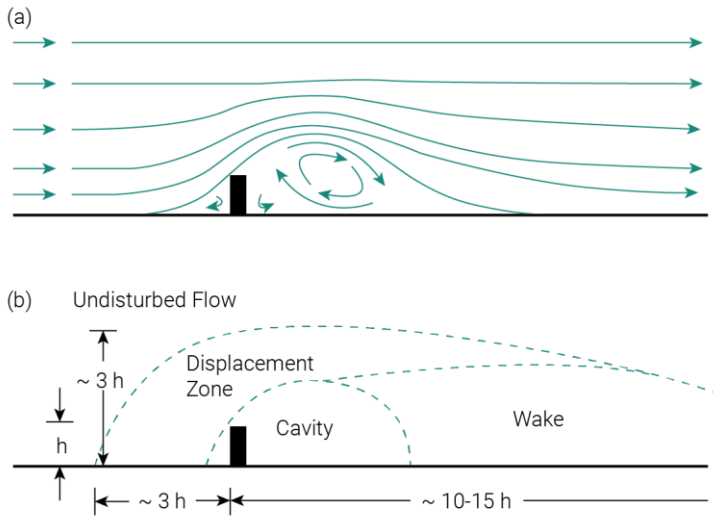


Figure 2.10 - Barrier exposed to an orthogonal flow: (a) streamlines; (b) flow zones

In the case of a more complex urban context, the geometric relationship between the width W and the height H plays a significant role. W is the dimension along the wind direction, and H is the height of the building. Figure 2.12 shows the cases when the buildings are spaced enough apart and when, instead, they are closer together. In the first case, the W/H ratio is greater than 3.33, and the flow conditions remain similar to the case with the single obstacle. While in the other two cases, the W/H ratio is less than 3.33, and the flow zones interfere with each other. This is a typical phenomenon that occurs when the wind direction is orthogonal to the road axis. With wind parallel to the road axis, the wind may be faster than the undisturbed flow (*urban canyon*).

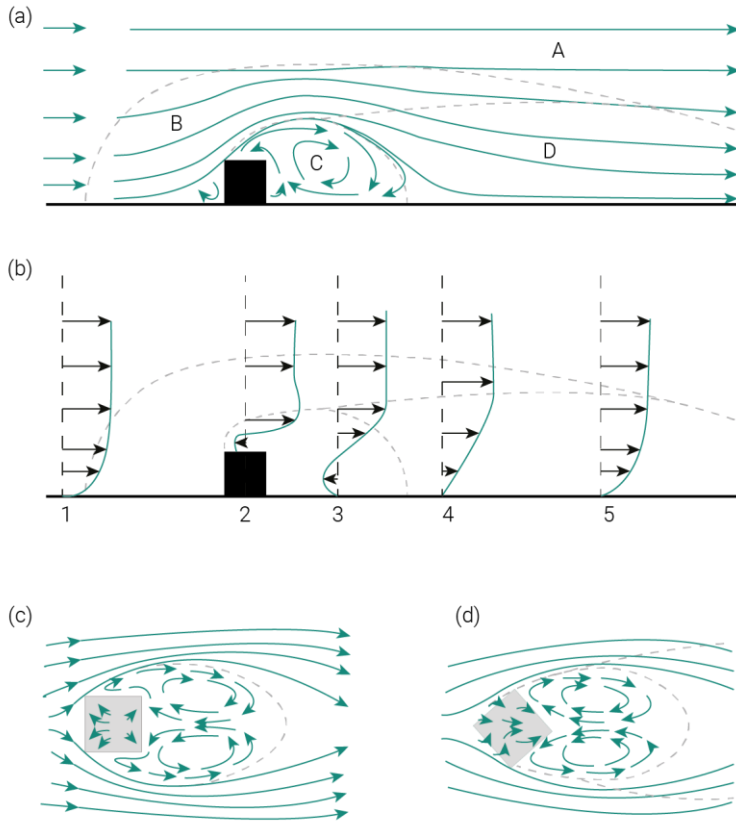


Figure 2.11 - Simple building exposed to an orthogonal flow: (a) lateral view of streamlines and flow zones (undisturbed flow [A], displacement zone [B], cavity [C] and wake [D]); (b) vertical profiles of velocity and flow zones; plan view of streamlines around a normally stream-oriented building (c) and diagonally to the stream (d)

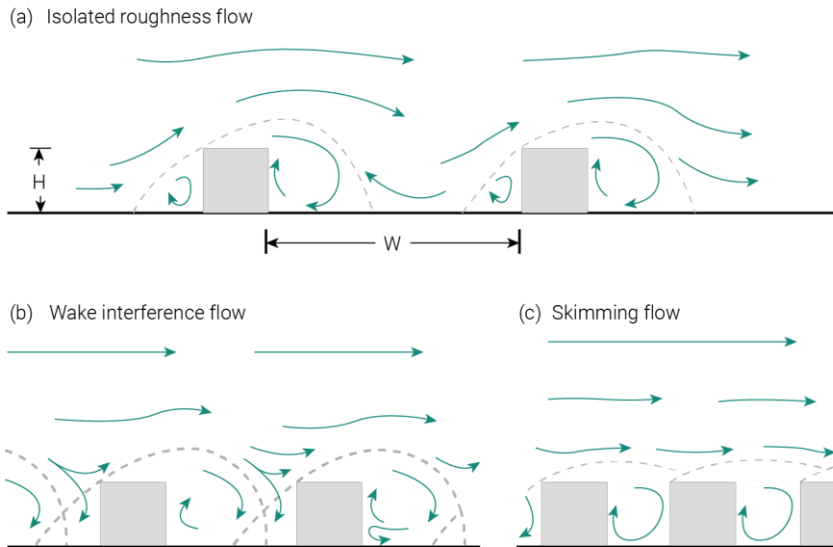


Figure 2.12 - Flow in different urban scenarios: (a) isolated roughness flow; (b) wake interference flow; (c) skimming flow

2.8 Computational Fluid Dynamics

The ever-increasing computational power made available by technology has meant that computational fluid dynamics (CFD) has become increasingly popular. This approach involves numerically solving the equations governing fluid flow at a discrete set of points in space and time.

CFD is based on the continuous medium hypothesis, and three fundamental conservation equations can describe the physics of fluid dynamics:

- conservation of mass (equation of continuity),
- conservation of momentum,
- conservation of energy (first principle of thermodynamics).

These three equations are complex because they are nonlinear, coupled and challenging to solve. It is hard to prove that there is a straightforward solution with particular boundary conditions with existing mathematical models. On the other hand, the Navier-Stokes equations accurately describe the flow of a Newtonian fluid, and only in a restricted and simplified group of cases, it is possible to obtain

Chapter 2

an analytical solution. For these reasons, numerical methods are commonly adopted.

2.8.1 Mass conservation

The mass continuity equation states that mass can neither be created nor destroyed within the control volume without a generating term. In the generic Cartesian coordinate x_i , the equation can be written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (\text{Eq. 2.6})$$

where ρ is the fluid density, t the time and u_i the velocity component along the direction i .

2.8.2 Conservation of momentum

The momentum equation is derived from Newton's second law, in which the change in momentum is equal to the summation of the forces acting on the fluid in each direction. In Cartesian coordinates, the equation can be written as:

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = \rho f_i - \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \quad (\text{Eq. 2.7})$$

where f_i is the component along the i -direction of the volume force and p is the pressure. The viscous stress tensor τ_{ij} is defined as:

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \quad (\text{Eq. 2.8})$$

with μ being the molecular viscosity and δ_{ij} being the Kronecker delta.

2.8.3 Energy conservation

The conservation of energy equation can be expressed as:

$$\frac{\partial E}{\partial t} + \frac{\partial [u_j (E + p)]}{\partial x_j} - \frac{\partial (u_i \tau_{ij} - q_j)}{\partial x_j} = 0 \quad (\text{Eq. 2.9})$$

E represents the total energy calculated as:

$$E = \rho \left(e + \frac{1}{2} u_i u_i \right) \quad (\text{Eq. 2.10})$$

with e equal to the specific internal energy. Assuming Fourier's law for heat conduction, the heat flux in the i -direction becomes:

$$q_i = -k_{tc} \frac{\partial T}{\partial x_i} \quad (\text{Eq. 2.11})$$

where k_{tc} is the thermal conductivity of the fluid and T is the temperature.

2.9 General aspects of turbulence

From the resolution of the Navier-Stokes equations, it emerges that turbulence is generated because of the laminar flow instability.

A parameter commonly adopted in fluid dynamics to determine if a fluid flow is laminar or turbulent is the Reynolds number. It is a dimensionless number that relates the forces of inertia to the viscous forces and can be defined as:

$$Re = \frac{uL}{\nu} = \frac{\rho u L}{\mu} \quad (\text{Eq. 2.12})$$

where u is the local flow velocity (in m/s), L is the characteristic length of the considered phenomenon (in m), ν is the kinematic diffusivity (in m²/s). The latter is given by the ratio between the dynamic viscosity (μ expressed in Pa) and the density (ρ in kg/m³).

For low Reynolds numbers ($Re < 2000$), the fluid layers flow neatly over each other, and the molecular viscosity dampens the random disturbances that occur on a small scale at high frequencies. This type of flow is referred to as *laminar flow*.

For $2000 \leq Re \leq 4000$, the flow loses its stationarity and propagates along a wavy trajectory (*transient flow*).

Instead, when the Reynolds number increases ($Re > 4000$), the effects of viscosity are too weak to prevent the random disturbances from amplifying. On the contrary,

Chapter 2

they gradually intensify until they become nonlinear and interact with the adjacent perturbations. This type of regime is called *turbulent flow* and is characterized by:

- a disorderly and seemingly random motion,
- three-dimensionality (dependence on time and rotation),
- non-stationarity,
- non-deterministic velocity fluctuations.

The simultaneous presence of multiplicity scales and phenomena in the same region of the domain makes the solution of the fluid dynamic problem difficult. The instability, characterized by high Reynolds numbers, creates vortical structures that evolve into progressively smaller ones until they are dissipated through viscosity.

The first to introduce the concept of "*energy cascade*" was Lewis Richardson (1922), while Andrej Kolmogorov developed the first statistical theory of turbulence in (1941).

According to Kolmogorov's law it is possible to calculate the characteristic scales of vortices in the universal equilibrium zone:

$$\eta = \frac{L}{Re^{0,75}} \quad (Eq. 2.13)$$

$$\tau = \frac{t}{Re^{0,5}} \quad (Eq. 2.14)$$

$$u_\eta = \frac{U}{Re^{0,25}} \quad (Eq. 2.15)$$

where η , τ , and u_η denote the Kolmogorov length, time, and velocity scales, while U and L denote the velocity and length characteristic of larger motion scales. The t , on the other hand, represents the time scale of large-scale motions.

In order to numerically solve the non-stationary Navier-Stokes equations, it is necessary to compute a wide range of spatial and temporal scales. This results in very dense computational grids and temporal discretization that require significant computational effort.

2.10 Direct Numerical Simulation

Direct Numerical Simulation (DNS) is the approach that simulates turbulence by solving the Navier-Stokes equations without approximations or averaging. This method is equivalent to performing a laboratory test of short duration.

The computational domain must be at least as large as the biggest turbulent vortex. In order to capture the dissipative phenomenon in its entirety, the grid cannot be larger than the Kolmogorov length scale η . For the homogeneous turbulence (the simplest), the number of grid points in each direction must be at least L/η . As already mentioned, this ratio is proportional to the quantity $Re^{3/4}$. Considering that this number of points must be used in each of the three dimensions and that the time discretization is proportional to the grid size, the computational cost of the simulation increases with Re^3 . Thus, this approach can only be implemented with relatively low Reynolds numbers.

2.11 Large Eddy Simulation

Another mathematical model used to study turbulent phenomena is the *Large Eddy Simulations* (LES), which consists of simulating the largest scales of the turbulence motion and approximating the smallest scales by means of an appropriate model (SubGrid Scale - SGS).

Larger turbulent structures have anisotropic characteristics dependent on the specific problem and constitute the most considerable energy contribution. On the other hand, smaller structures (dissipative) are essentially isotropic and universal and, therefore, do not depend on the specific problem.

The LES approach is intermediate between DNS and RANS (which will be discussed in the next paragraph). It is more computationally expensive than RANS models but less expensive than DNS models.

2.12 Reynolds Averaged Navier-Stokes

From an engineering point of view, the most widely used approach is undoubtedly the one called *Reynolds Averaged Navier-Stokes* (RANS). The RANS equations are derived by applying a statistical procedure (Reynolds' decomposition) that expresses all the flow variables as the sum of an average contribution and a fluctuating one. The velocity, for example, is given by:

Chapter 2

$$u_i(t) = \bar{u}_i + u'_i \quad (\text{Eq. 2.16})$$

where u'_i is the time-dependent fluctuating component and \bar{u}_i is the time-independent average term because it is obtained from a time-averaging operation:

$$\bar{u}_i = \lim_{T \rightarrow +\infty} \frac{1}{T} \int_t^{t+T} u_i(t) dt \quad (\text{Eq. 2.17})$$

The RANS equations are defined from the continuity and momentum conservation equations by averaging over time all the present terms. In the continuity equation, there are no fluctuations because the velocity is linear. In the equation of conservation of momentum, however, the fluctuating components do not vanish, and a new term $\overline{u'_i u'_j}$, called *Reynolds' stress*, appears. Thus, the result is given by:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = \rho f_i - \frac{\partial p}{\partial x_i} + \frac{\partial (\tau_{ij} - \overline{\rho u'_i u'_j})}{\partial x_j} \quad (\text{Eq. 2.18})$$

where all parameters must be considered as average values.

The stress tensor consists of three parts:

- isotropic strain induced by the mean pressure field,
- viscous strain caused by mean velocity gradients,
- Reynolds' stress, which, as mentioned, arises from the average of the products of the fluctuating components.

The fluid dynamics problem presents four equations (continuity equations plus the three Navier-Stokes equations) and ten unknowns: three components of the mean velocity, the mean pressure and the six components of the Reynolds stress tensor. For this reason, the problem of closing this set of equations arises, and it is solved by adopting a turbulence model.

2.13 Turbulent viscosity models

Three categories of turbulent viscosity models can be defined: zero, one or two equations. All three are based on the *Boussinesq hypothesis*, which can be written as:

$$-\overline{\rho u_i' u_j'} = \mu_T \left[\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] - \frac{2}{3} \delta_{ij} \rho k \quad (\text{Eq. 2.19})$$

In zero-equation models, the last term is eliminated.

Two-equation models, unlike the others, can be used to predict the property of a given turbulent flow without knowing a priori the turbulent structure or geometry of the flow. The starting point is the Boussinesq hypothesis and a pair of partial derivative differential equations describing the transport of two variables. The first of these variables is the turbulent kinetic energy k :

$$\frac{Dk}{Dt} = \tau_{ij} \frac{\partial u_i}{\partial x_j} - \varepsilon + \frac{\partial}{\partial x_j} \left[\nu \frac{\partial k}{\partial x_j} - \frac{1}{2} \overline{u_i' u_i' u_j'} - \frac{1}{\rho} \overline{p' u_j'} \right] \quad (\text{Eq. 2.20})$$

where:

$\tau_{ij} \partial u_i / \partial x_j$	production of turbulence;
$\nu \partial k / \partial x_j$	molecular diffusion;
$\overline{u_i' u_i' u_j'} / 2$	turbulent flow of k ;
$\overline{p' u_j'} / \rho$	pressure diffusion.

On the other hand, the second variable is arbitrary, and two of the most popular are the turbulent kinetic energy dissipation rate ε and the specific dissipation rate ω .

For the k - ε model, the turbulent viscosity used in the Boussinesq hypothesis is calculated as:

$$\nu_T = C_\mu \frac{k^2}{\varepsilon} \quad (\text{Eq. 2.21})$$

Chapter 2

with C_μ the characteristic coefficient of the model. This model is the most common and is probably the best compromise between the generality of application, accuracy, simplicity of implementation and numerical stability.

While for the k - ω model the turbulent viscosity is given by:

$$\nu_T = \frac{k}{\omega} \quad (\text{Eq. 2.22})$$

This model is more robust and accurate than the previous one in calculating heat transfer coefficients and wall shear stresses. It requires a less dense grid and has strong sensitivity of ω to boundary conditions outside the boundary layer.

A further hybrid model, given by the combination of the k - ϵ and k - ω model, considers the k - ω model near the walls and the k - ϵ model in the rest of the domain.

3

Framework developed

The state-of-the-art analysis highlights that numerical weather prediction (NWP) models commonly forecast atmospheric fluxes. However, they are unable to estimate small-scale fluctuations caused by urban roughness. Therefore, computational fluid dynamics (CFD) simulations are widely adopted for assessing airflows around buildings inside urban areas.

These two approaches have been developed to operate at different spatial scales. However, the increase in computational efficiency has allowed NWP models to solve numerical problems at the smallest scales, whereas CFD models at the largest ones. As a result, the models, converging at intermediate scales, gradually overlap in problem-solving.

As mentioned above, Figure 3.1 diagrams atmospheric phenomena according to the horizontal (abscissas) and the time (ordinates) scales.

The coupling procedure between NWP and CFD codes, adopted in the present work, allows generating a reliable model that can accurately predict the influence of the orography or the architectural configuration on wind directions and velocities.

Chapter 3

These quantities profoundly affect the accuracy of the Building Energy Simulations (BES).

The chapter describes the methodology adopted for the mesoscale and microscale approach, the coupling procedure, and the energy simulation of the case study.

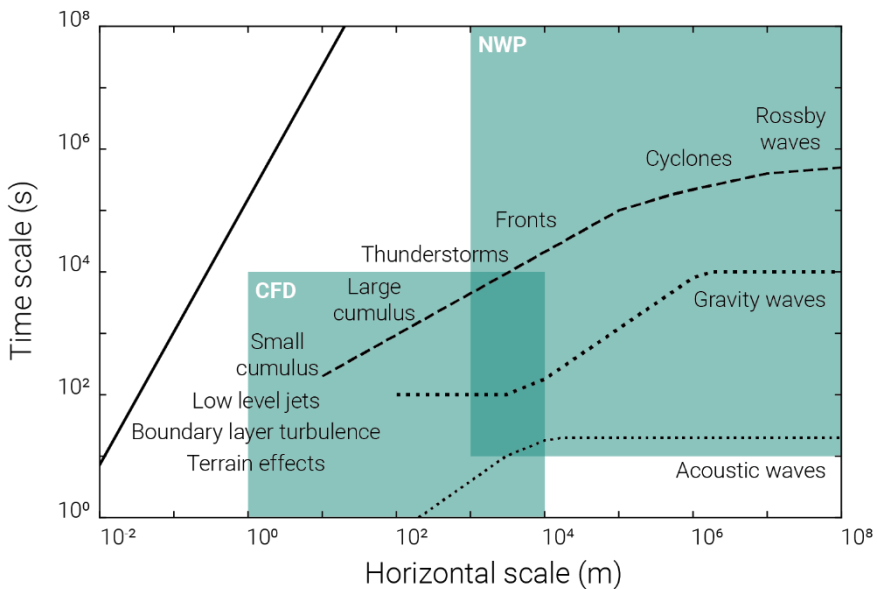


Figure 3.1 - Characteristic time and horizontal scales of different atmospheric phenomena (modified from Olsen [39])

3.1 Mesoscale analysis

The Mesoscale analysis describes the climatology and evaluates the wind inside an area under study, and it is carried out for one year.

The software selected for this type of analysis is the one developed by Pennsylvania State University (PSU) and the National Center for Atmospheric Research (NCAR), called Mesoscale Model Fifth Generation (MM5). It is a limited-area, non-hydrostatic, terrain-following sigma-coordinate model designed to simulate or predict mesoscale atmospheric circulation.

Weather phenomena from the synoptic to the local scale are considered using the two-way nesting procedure over five nested domains. These domains are characterized by the same center and number of grid points with an increasing spatial dimension ratio of 3:1.

Through the domain at the synoptic scale, it is possible to study the main perturbation phenomena that occur weekly or monthly. In contrast, the domain at the local scale allows describing, with higher accuracy, local convective phenomena such as sea breezes or hill breezes occurring with a frequency of 12 or 24 hours.

3.2 Microscale analysis

On the microscale side, the proposed methodology is first tested with the commercial software (Chapter 5) to check its effectiveness and, only later, performed by adopting an open-source software (Chapter 8).

The commercial software chosen for the CFD analysis is *STAR-CCM+*.

The numerical model consists of a cylindrical domain where the terrain orography and building shapes are obtained from the regional cartography. The center of the domain is represented by the case study selected for carrying on energy simulations. The same center is also chosen for the Mesoscale model.

The lowest profile of the cylindrical domain follows the shapes of the terrain and buildings, while the upper surface is flat. On the other hand, the lateral surface is divided into 12 sectors to account for the different directions of the incoming wind.

In contrast, *Kratos Multiphysics* is the open-source software selected for the simulations.

Unlike the commercial one, the procedure to create the numerical model is fully automated. Furthermore, data about the terrain orography and building shapes come from open-source databases. Specifically, from ASTER-GDEM for the former and OpenStreetMap for the latter.

3.3 Coupling procedure

The idea at the base of the coupling procedure is to calculate the wind speed and direction in the central point of the domain starting from a representative number

Chapter 3

of virtual anemometers located inside the selected computational domain. As mentioned above, the building for the energy simulations is situated in the center of the domain.

The information extracted from the virtual anemometers is the wind speed and direction. The position of these anemometers is the same for both Mesoscale and Microscale results.

Moreover, on the microscale side, information is extracted at the domain's central point to calculate the speed-up matrix and the directional inflow matrix. In detail, the speed-up matrix is obtained from the ratio among the velocity expected at the center of the domain and the one extracted from the virtual anemometers. Instead, the directional inflow matrix is the angle between the velocity vector in the central points and the velocity vector in extracted points.

On the mesoscale side, the extracted points' velocities are multiplied with the speed-up coefficients, and the wind directions are summed with the inflow coefficients previously calculated at the same point.

The final wind speed and direction are defined considering each anemometer's distance from the center of the domain. Thus, anemometers positioned closer to the center have more influence than anemometers located further away.

3.4 Building Energy Simulations

Finally, energy analyses are carried on to evaluate the impact of local winds on thermal comfort conditions and energy consumption of buildings inside an urban area. Thus, the outputs obtained from the coupling procedure become the inputs to energy analyses used to conduct better quality simulations. The present step is crucial, especially in the case of advanced technological systems, as Double Skin Façade buildings, in which the local environmental conditions strongly influence the energy performance.

4

Mesoscale simulations

The annual average wind intensity assessment and the distribution of its occurrence inside the area of interest are performed either by installing meteorological stations or using the information recorded close to the location of interest.

Unfortunately, the amount of meteorological stations is not high, and, most of the time, they are placed in specific areas (e.g., airports) that rarely coincide with the site under analysis. In addition, the measurements are point-source, and local flux variations are only valid over a limited area (a few meters to a dozen kilometers) that depends on the complexity of the terrain.

For these reasons, to describe the climatology of an area, Numerical Weather Prediction (NWP) is increasingly used in lieu of in situ measurements.

4.1 Numerical Model

The software selected for mesoscale analysis is the Mesoscale Model Fifth Generation (MM5) [40] developed by Pennsylvania State University (PSU) and the National Center for Atmospheric Research (NCAR).

MM5 is a limited-area, nonhydrostatic, terrain-following sigma-coordinate model designed to simulate or predict mesoscale atmospheric circulation.

Figure 4.1 depicts the main components of the flow-chart of the complete modeling system and the order of the modules. In detail, *Terrain* and *Regrid* modules horizontally interpolate meteorological data from a latitude-longitude mesh to a high-resolution variable domain on either a Mercator, Lambert Conformal, or Polar Stereographic projection. The *Interp* module, on the other hand, performs vertical interpolation from the pressure levels to the *MM5* sigma coordinate system.

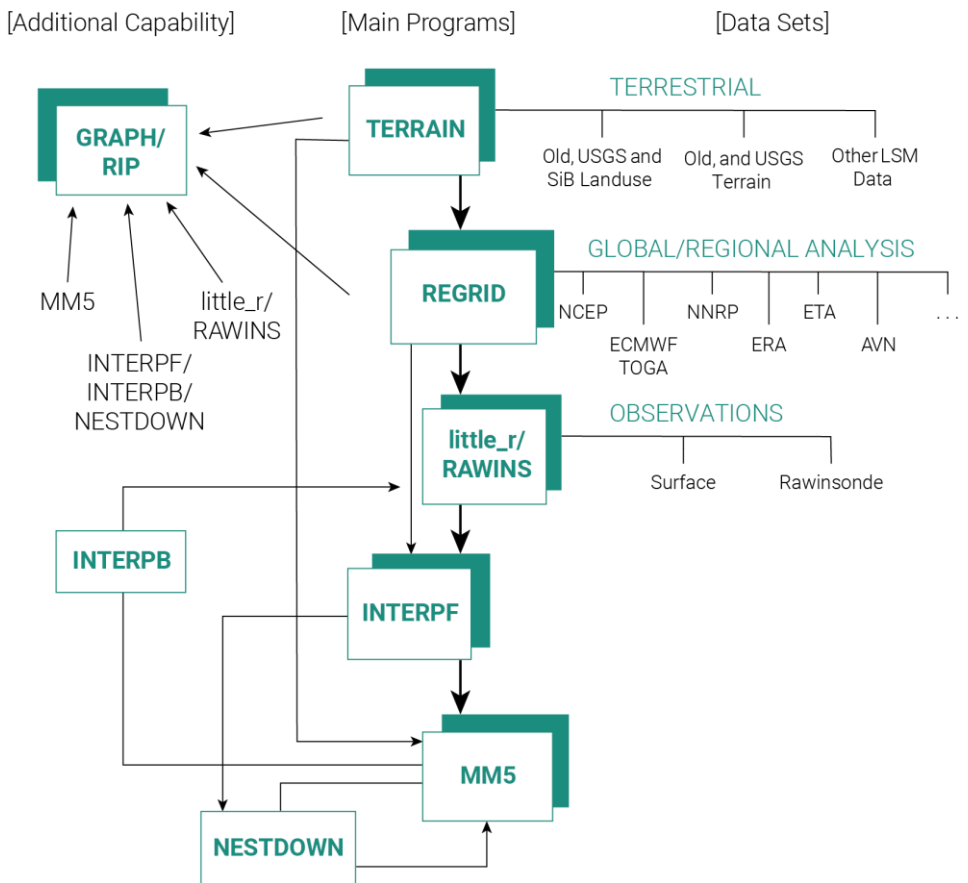


Figure 4.1 - The MM5 modeling system flow chart [41]

The vertical coordinate is terrain-following (Figure 4.2), meaning that the lower grid levels follow the terrain while the upper surface is flat. At intermediate levels, the pressure progressively decreases toward the top pressure chosen.

The levels of the model are defined by:

$$\sigma = (p_0 - p_t)/(p_{s0} - p_t) \tag{Eq. 4.1}$$

where p_0 is the reference-state pressure, p_t is a specified constant top pressure, and p_{s0} is the reference-state surface pressure.

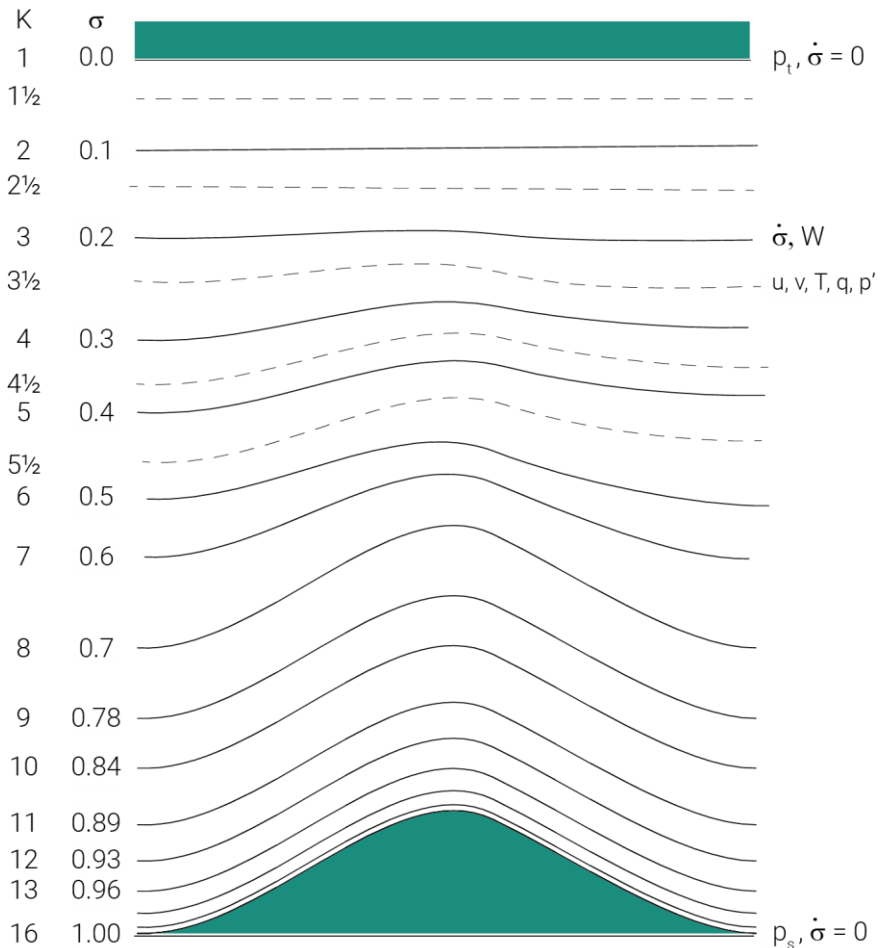


Figure 4.2 - Schematic representation of the vertical structure of the model [41]

Chapter 4

The horizontal grid has an Arakawa-Lamb B-staggering of the velocity variables concerning the scalars (Figure 4.3). The scalars (i.e., T , q) are defined at the center of the grid square, while the eastward (u) and northward (v) velocity components are collocated at the corners. The center nodes of the grid squares are referred to as cross points, and the corner nodes are dots.

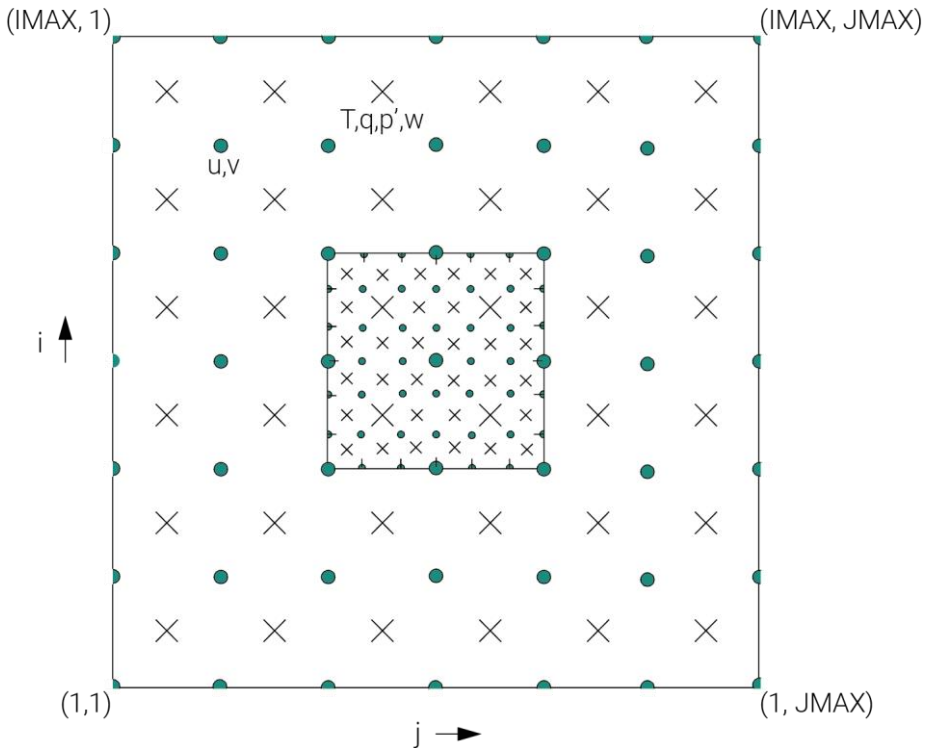


Figure 4.3 - Arakawa B-grid schema [41]

The first program run in a complete forecast simulation, in the MM5 procedure, is *Terrain*. It horizontally interpolates the regular latitude-longitude terrain elevation and vegetation on selected domains. When the Land-Surface Model (LSM) is used in the MM5 model, it generates soil types, vegetation fraction, and annual deep soil temperature fields.

The main tasks performed by this module are essentially two: set up the domain grids (from coarse to fine) and produce terrestrial data fields for all domains.

Before the analysis, the *TERRAIN/terrain.deck* file is updated with the necessary information. The modified variables are listed below:

<i>PHIC</i>	central latitude of the coarse domain in degrees North (in South is negative),
<i>XLONX</i>	central longitude of the coarse domain in degrees East,
<i>MAXNES</i>	maximum number of domains on which the analysis is performed,
<i>NESTIX</i>	the i-direction dimensions for each of the domains,
<i>NESTJX</i>	the j-direction dimensions for each of the domains,
<i>DIS</i>	the grid distance for each of the domains (in km),
<i>NESTI</i>	the i-location in its parent domain of the nest domain's low-left corner,
<i>NESTJ</i>	the j-location in its parent domain of the nest domain's low-left corner,
<i>NTYPE</i>	the source terrain height and land-use data type for each of the domains.
	1: 1 deg (~111 km)
	2: 30 min (~56 km)
	3: 10 min (~19 km)
	4: 5 min (~9 km)
	5: 2 min (~4 km)
	6: 30 sec (~0.9 km)
	7: 3 sec (~0.09 km)

The second module, named *Regrid*, is a suite of programs to manage various tasks. The main functions are data input and interpolation to the MM5 grid. The module is divided into two sub-modules: *pregrid* and *regridder*.

The input meteorological data (handle from *pregrid*) are the NCEP (National Centers for Environmental Prediction) ds083.2 datasets which belong to FNL (Final Global Data Assimilation System) category. These data are available from 1999 to

Chapter 4

current days, and they have a grid resolution of 1x1 degree with a time interval of 6 hours (00:00, 06:00, 12:00, 18:00 UTC).

The *Interpf* module manages the transformation process required by the analysis programs to the mesoscale model. After the horizontal interpolations of the orography and weather data, this module performs the vertical interpolation of the weather data on the levels required in the domain discretization phase. Finally, the data provided by the previous modules are used to define the boundary conditions for the simulation.

The last module that is run is *MM5*, where the actual analysis is performed.

Three different combinations of the Planetary Boundary Layer (PBL) and Land Surface Model (LSM) parameterization schemes are evaluated. The first scheme is the PBL of Medium-Range Forecast model (MRF) developed by Hong and Pan [42] in combination with the NOAA LSM (hereafter called *MM5-MRF*). The second one, implemented by Pleim and Chang [43], is coupled with the LSM created by Pleim and Xiu [44] (hereafter called *MM5-PleimXiu*). The last model analyzed is the one developed by Blackadar [45] (hereafter called *MM5-BK*).

According to Dudhia J. et al. [41], the modified variables in the *MM5* module are briefly described below:

IBLTYP an integer value defines PBL schemes.

- 2: Blackadar PBL (suitable for high-resolution PBL, e.g., five layers in lowest km, surface layer < 100 m thick).
- 5: MRF PBL (suitable for high-resolution in PBL and it must be used with ISOIL=1 or 2).
- 7: Pleim-Chang PBL (this scheme only works with ISOIL=3, and it is a derivative of the Blackadar PBL scheme).

ISOIL surface schemes.

- 1: Five-Layer Soil model (temperature predicted in approximately 1,2,4,8,16 cm layers).
- 2: Noah Land-Surface Model (the land-surface model can predict soil moisture and temperature in four layers: 10, 30, 60, and 100 cm thick).

3: Pleim-Xiu Land-Surface Model (this model represents soil moisture and temperature in two layers: surface layer at 1 cm and root zone at 1 m).

TIMAX forecast length in minutes.

TISTEP time step (in seconds) for the coarsest domain (recommend $3 \cdot dx$).

XENNES ending time in minutes for each domain.

A *one-way* or *two-way nesting* procedure can also be performed with this module. The nesting procedure is a technique that produces finer grid resolutions limited to the area of interest. The finer grid is embedded in a parent domain that has a coarser resolution.

When the execution of single or multiple domains is completed, the outputs are used as input into the domain with higher resolution. In one-way nesting, the information goes from the parent domain to its subdomain. In contrast, in the two-way nesting, each domain takes information from its parent domain every timestep and executes three timesteps for each step before returning the information to the parent domain.

4.2 Numerical set-up

Based on what is described in the previous paragraph, some snippets of the code used to launch the analysis are reported below.

First, the *TERRAIN/terrain.deck* file is updated by inserting coordinates and information about the domains:

```

1 | &MAPBG
2 | PHIC = 42.442679,
3 | XLONC = 14.207077,
4 | IEXP = .F.,
5 | AEXP = 360., ; approx expansion (km)
6 | IPROJ = 'LAMCON', ; Lambert-conformal map
   | projection

```

```

7 | ;IPROJ = 'POLSTR', ; Polar stereographic map
   | projection
8 | ;IPROJ = 'MERCAT', ; Mercator map projection
9 | &END
10 | &DOMAINS
11 | MAXNES = 5,
12 | NESTIX = 31, 31, 31, 31, 31,
13 | NESTJX = 31, 31, 31, 31, 31,
14 | DIS = 32.4, 10.8, 3.6, 1.2, 0.4,
15 | NUMNC = 1, 1, 2, 3, 4, ; parent domain ID
16 | NESTI = 1, 11, 11, 11, 11,
17 | NESTJ = 1, 11, 11, 11, 11,
18 | RID = 1.5, 1.5, 1.5, 3.1, 2.3, 2.3, ; radius
   | of influence in grid units (IFANAL=T)
19 | NTYPE = 5, 6, 6, 6, 7,
20 | NSTTYP = 1, 2, 2, 2, 2, 2, ; 1 -- one-way nest,
   | 2 -- two-way nest
21 | &END

```

(The values after the semicolon are comments)

The datum used to identify the center of the domain is *WGS84* (also known as *EPSG:4326*).

In the *NESTIX* and *NESTJX*, the number of nodes is the same in all computational domains. With the ratio of 3:1 (necessary for the two-way nesting), the square domains ensure the complete overlapping between the nodes of the parent domain and its sub-domain.

In the *DIS* variable, the 3:1 ratio is clearly visible. Domain 1, in fact, has a distance of 32.4 km; the second domain has a distance of 10.8 km (which is one-third of the previous one) and so on until the last domain (0.4 km).

Figure 4.4 illustrates a graphical representation of *NESTI* and *NESTJ*.

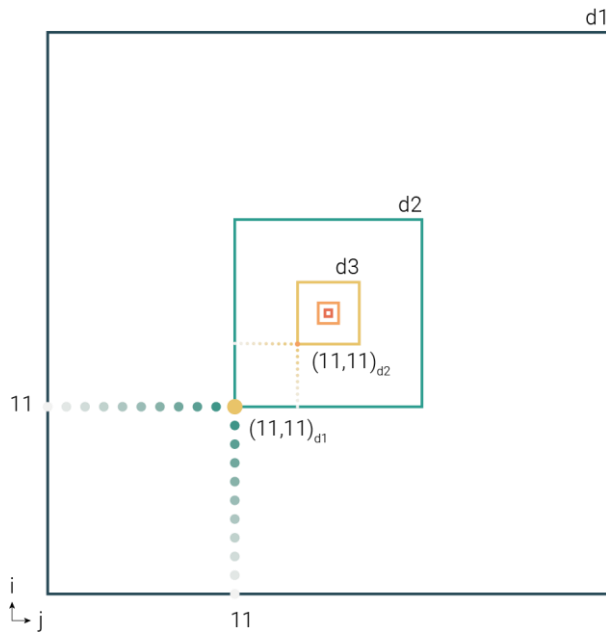


Figure 4.4 - Graphical representation of NESTI and NESTJ

Then the starting and ending periods are also set in the *Regrid/pregrid/pregrid.csh*, *Regrid/regridder/namelist.input* and *Interpf/namelist.input* files:

```

1 | START_YEAR = 2017 # Year (Four digits)
2 | START_MONTH = 01 # Month (01 - 12)
3 | START_DAY = 01 # Day (01 - 31)
4 | START_HOUR = 00 # Hour (00 - 23)
5 |
6 | END_YEAR = 2017 # Year (Four digits)
7 | END_MONTH = 12 # Month (01 - 12)
8 | END_DAY = 31 # Day (01 - 31)
9 | END_HOUR = 18 # Hour (00 - 23)

```

(The values after the tag [#] are comments)

In the last module, the file *MM5/configure.user* and *MM5/mm5.deck* are modified with the domain information and PBL parameters.

Chapter 4

The following code snippet shows the *MM5/configure.user* file. The highlighted lines indicate the different PBL parameters identified by various colors.

MM5-MRF parameters

MM5-PleimXiu parameters

MM5-BK parameters

```
45 | # MAXNES (int): max num of Domains in simulation
46 | MAXNES = 5
47 | #
48 | # MIX,MJX (int): max Dimensions of any Domain
49 | MIX = 70
50 | MJX = 70
51 | # MKX (int): num of half sigma levels in model
52 | MKX = 26
   | ...
81 | # IBLTYP: for planetary boundary layer (array,int)
82 | #     0=no PBL fluxes,1=bulk,2=Blackadar,
83 | #     3=Burk-Thompson,4=Eta M-Y,5=MRF,
84 | #     6=Gayno-Seaman,7=Pleim-Xiu
85 | IBLTYP = "5,5,5,5,5,0,0,0,0,0"
   | ...
97 | # ISOIL: for multi-layer soil temp model (int)
98 | #     0=no,1=yes (only works with IBLTYP=2,4,5,6)
99 | #     2=Noah land-surface scheme (IBLTYP=4,5 only)
100 | #     3=Pleim-Xiu LSM (IBLTYP=7 only)
101 | ISOIL = 2
```

(The values after the tag [#] are comments)

```
85 | IBLTYP = "7,7,7,7,7,0,0,0,0,0"
   | ...
101 | ISOIL = 3
```

```
85 | IBLTYP = "2,2,2,2,2,0,0,0,0,0"
   | ...
101 | ISOIL = 1
```

MM5/mm5.deck code snippet with modified parameters:

```

50 | ; *** FORECAST TIME AND TIME STEP ***
51 | ;
52 | TIMAX = 260640, ; forecast length in minutes
53 | TISTEP = 120, ; coarse domain DT in model
   | ...
161 | ; *** NEST AND MOVING NEST OPTIONS ***
162 | ;
163 | LEVIDN = 0,1,2,3,4,4,4,4, ; level of nest for
   | each domain
164 | NUMNC = 1,1,2,3,4,4,4,4, ; ID of parent domain
   | for each nest
165 | NESTIX = 31, 31, 31, 31, 31, ; domain size i
166 | NESTJX = 31, 31, 31, 31, 31, ; domain size j
167 | NESTI = 1, 11, 11, 11, 11, ; start location i
168 | NESTJ = 1, 11, 11, 11, 11, ; start location j
169 | XSTNES = 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., ;
   | domain initiation
170 | XENNES = 260640, 260640, 260640, 260640, 260640,
   | 720., 720., 720., 720., 720.; domain termination
171 | IOVERW = 1, 2, 2, 2, 2, 2, 2, 2, 0, 0, ; overwrite
   | nest input
172 | ; 0=interp. from coarse mesh (for nest domains)
173 | ; 1=read in initial domain conditions
174 | ; 2=read in nest terrain file

```

(The values after the semicolon are comments)

4.3 Mesoscale results

The mesoscale analyses are performed in the Adriatic city of Pescara (Italy). The center of the domain is 42.442679 latitude and 14.207077 longitude.

As mentioned above, there are five nested domains (Figure 4.5) with the same grid center and the same number of compute nodes.

Domain 1 covers almost the entire Italian surface and the Croatian coasts; at this scale are studied the main perturbation phenomena that occur weekly or monthly (Figure 2.9). Domain 2 evaluates the regional climate influenced by mountain

Chapter 4

ranges and seawater. Domains 3 and 4 allow a better description of the principal valleys and the Apennine chain. Domain 5, instead, describes local convective phenomena such as sea breezes or hill breezes that occur with a frequency of 12h or 24h.

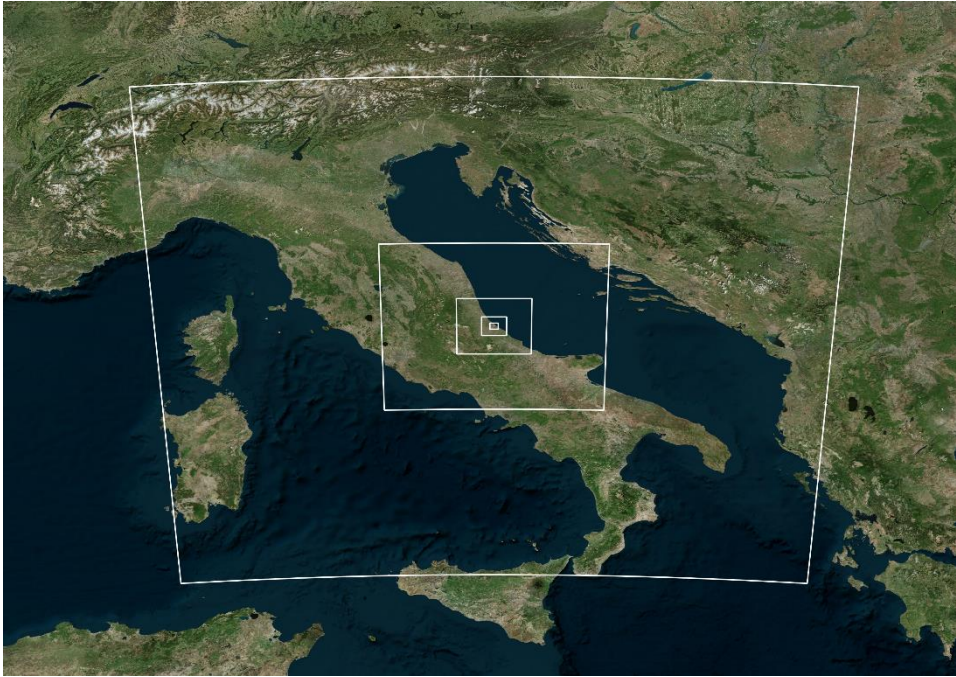


Figure 4.5 - Domain nesting

The geometric difference between the domains is the grid spacing as summarizes in Table 1.

Domain	Grid spacing [km]	Grid points	Domain extension [km ²]
D5	0.4	31x31	12x12
D4	1.2	31x31	36x36
D3	3.6	31x31	108x108
D2	10.8	31x31	324x324
D1	32.4	31x31	972x972

Table 1 - Mesoscale domains

The simulation results are weather files with a 4-minute time step. These files are post-processed to recreate virtual anemometers at each grid point inside the computational domain and obtain the chosen period's temporal history.

Figure 4.6 reports the extracted virtual anemometers with the created wind roses. The point (16, 16) corresponds to the center of the computational domain.



Figure 4.6 - Extraction points of the virtual anemometers

The wind roses show information about the direction of wind occurrence, the intensity of the wind, and the number of occurrences in a specific direction. This information is visualized through the direction, color, and size of the petals. In particular, the direction of the petal indicates the direction of the wind (with a variation of $\pm 15^\circ$); the color of the petal indicates a specific intensity, and the radial magnitude the number of occurrences.

The corresponding wind roses have been created for each analysis (*MM5-MRF*, *MM5-PleimXiu*, *MM5-BK*) and each virtual anemometer.

Chapter 4

From Figure 4.7 to Figure 4.9 are reported the wind roses at altitudes 10 m, 92 m, and 204 m of the *MM5-MRF* analysis.



Figure 4.7 - Annual wind roses: MM5-MRF@10m



Figure 4.8 - Annual wind roses: MM5-MRF@92m

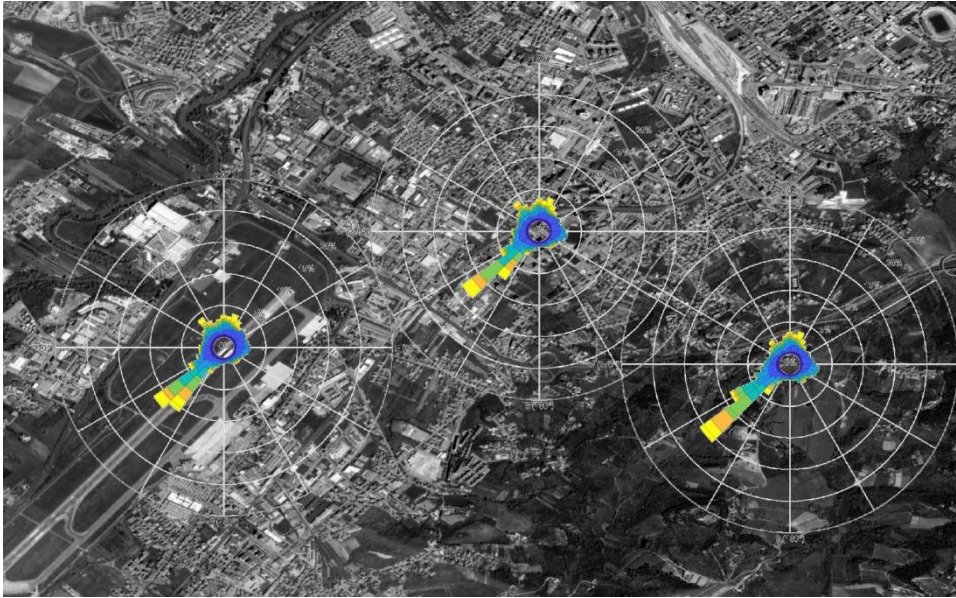


Figure 4.9 - Annual wind roses: MM5-MRF@204m

The wind roses corresponding to the *MM5-PleimXiu* analysis, on the other hand, are depicted from Figure 4.10 to Figure 4.12.



Figure 4.10 - Annual wind roses: MM5-PleimXiu@10m

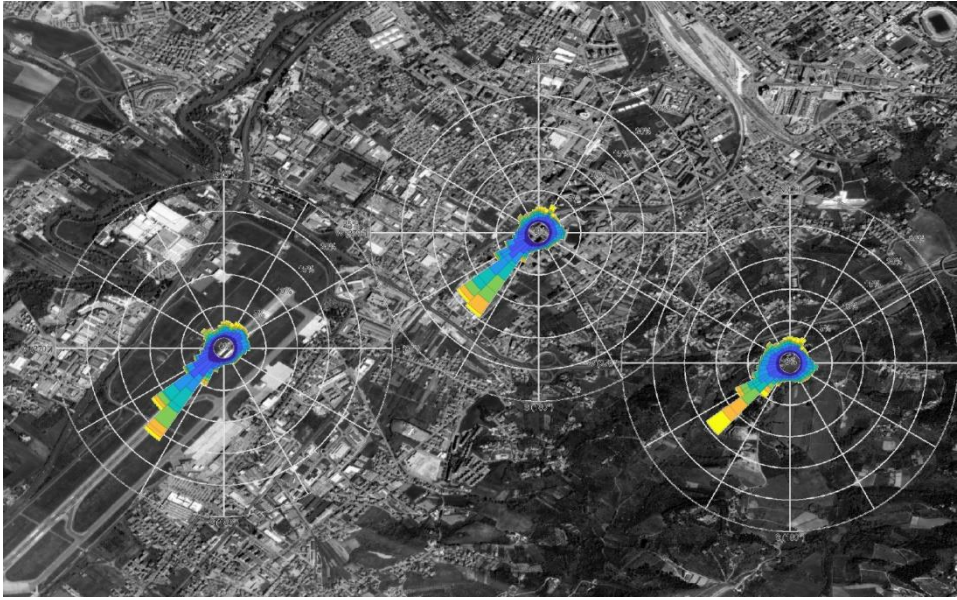


Figure 4.11 - Annual wind roses: MM5-PleimXiu@92m



Figure 4.12 - Annual wind roses: MM5-PleimXiu@204m

Lastly, from Figure 4.13 to Figure 4.15, the wind roses for the MM5-BK analysis are shown.

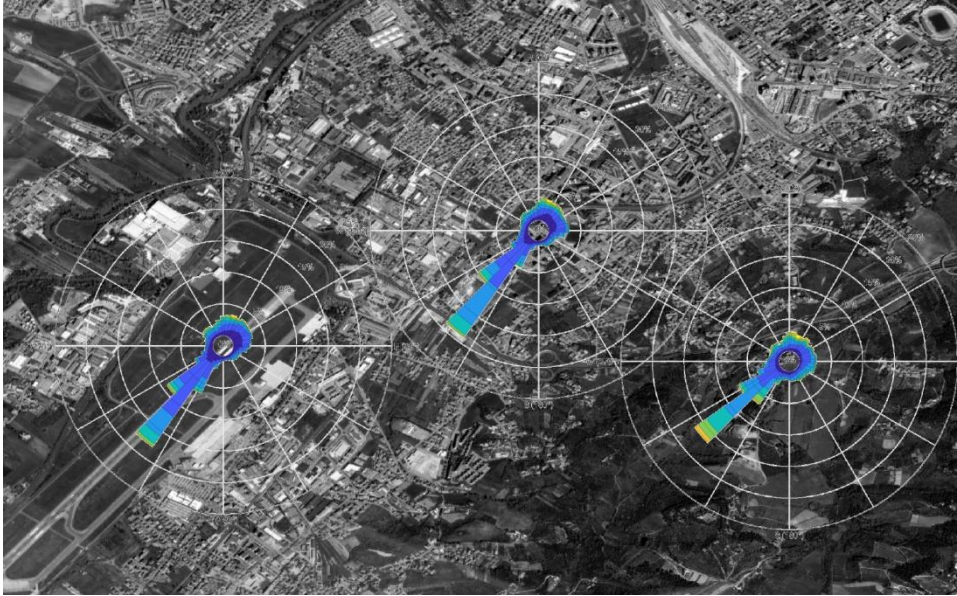


Figure 4.13 - Annual wind roses: MM5-BK@10m



Figure 4.14 - Annual wind roses: MM5-BK@92m



Figure 4.15 - Annual wind roses: MM5-BK@204m

For the sake of brevity, only three significant points of the domain are analyzed, comparing the results obtained with the performed analyses. Point (14,12) is positioned near the airport of Pescara ("Aeroporto Internazionale d'Abruzzo") and, therefore, in a predominantly flat area. Point (14,19) is localized in a slightly hilly area in the S-E of the city of Pescara. Finally, point (16,16) is located within the city's urban fabric and corresponds to the building on which the energy analysis will be carried out.

Figure 4.16 shows the wind roses of the point (14,12) at quotas 10.0 m and 204.0 m for each analysis. The three simulations produce similar results to each other. With the *MM5-MRF* at elevation 10.0 m (Figure 4.16a), the predominant direction is S-W for a total time of about 30%. The maximum velocity is 6.0-8.0 m/s, with 2.0-4.0 m/s being the most frequent. At 204.0 m (Figure 4.16b), the dominant direction is still S-W for 25% of the time, and the maximum velocity is greater than 12.0 m/s. With the *MM5-PleimXiu* analysis at 10.0 m (Figure 4.16c) above ground level, the direction is S-W 30% of the time and S-SW 11% of the time. The maximum velocity is 4.0-6.0 m/s, although the most frequent velocity is 2.0-4.0 m/s. At elevation 204.0 m (Figure 4.16d), the main direction is S-W 20% of the time, and the maximum velocity is greater than 12.0 m/s. The wind rose obtained with the *MM5-BK* analysis

at elevation 10.0 m (Figure 4.16e) still shows a dominant direction from S-W 35% of the time and 7% from S-SW with a maximum speed of 6.0-8.0 m/s and the most frequent 2.0-4.0 m/s. At 204.0 m (Figure 4.16f), the dominant direction is S-W 19% of the time, and the maximum velocity is greater than 12.0 m/s.

Figure 4.17 shows the wind roses for the point (14,19). As for point (14,12), the dominant direction is S-W even if there is a slight deviation to the W due to the hill positioned a little farther E. With the *MM5-MRF* analysis, at elevation 10.0 m (Figure 4.17a), the dominant direction is S-W (about 35% of the time) with speeds 4.0-6.0 m/s most of the time and a maximum of 6.0-8.0 m/s. At 204.0 m above the ground level (Figure 4.17b), the dominant direction (S-W) is present 20% of the time, with peaks more outstanding than 12.0 m/s. With the *MM5-PleimXiu* analysis at 10.0 m (Figure 4.17c), the dominant direction is S-W for a time greater than 40%. The maximum velocity is 8.0-10.0 m/s, although the most frequent velocity is 4.0-6.0 m/s (8% of the time). At 204.0 m (Figure 4.17d), the prevailing direction is still S-W (15%), but there is a peak (5.5%) in the N-W direction. The maximum velocity recorded is greater than 12.0 m/s. From the *MM5-BK* analysis at 10.0 m (Figure 4.17e), the dominant direction (S-W) has a maximum velocity of 10.0-12.0 m/s and an overall time of about 30%. At 204 m (Figure 4.17f), the distribution is even with a peak in the S-W direction.

At point (16,16) (Figure 4.18), the wind roses obtained from the three analyses still show a main direction to the S-W. From the *MM5-MRF* analysis at 10.0 m (Figure 4.18a), the main direction occurs 30% of the total time with a maximum speed of 4.0-6.0 m/s and the most frequent of 2.0-4.0 m/s. At 204.0 m (Figure 4.18b), there is a significant peak to the S-W (13.5%) with maximum velocities greater than 12.0 m/s. From the *MM5-PleimXiu* analysis at 10.0 m (Figure 4.18c), the dominant direction is S-W, although more shifted toward S-SW. The total time is 45%, of which about 10% is at a velocity of 4.0-6.0 m/s and 2.0-4.0 m/s for 28% of the total time. At 204.0 m (Figure 4.18d), the behavior is very similar to the previous analysis, with a peak in the S-W direction. Finally, the wind roses obtained from the *MM5-BK* at an elevation of 10.0 m (Figure 4.18e) present an S-W main direction and a maximum speed of 6.0-8.0 m/s. At 204.0 m (Figure 4.18f), the results are homogeneous with a peak to the S-W with a maximum velocity higher than 12.0 m/s.

Chapter 4

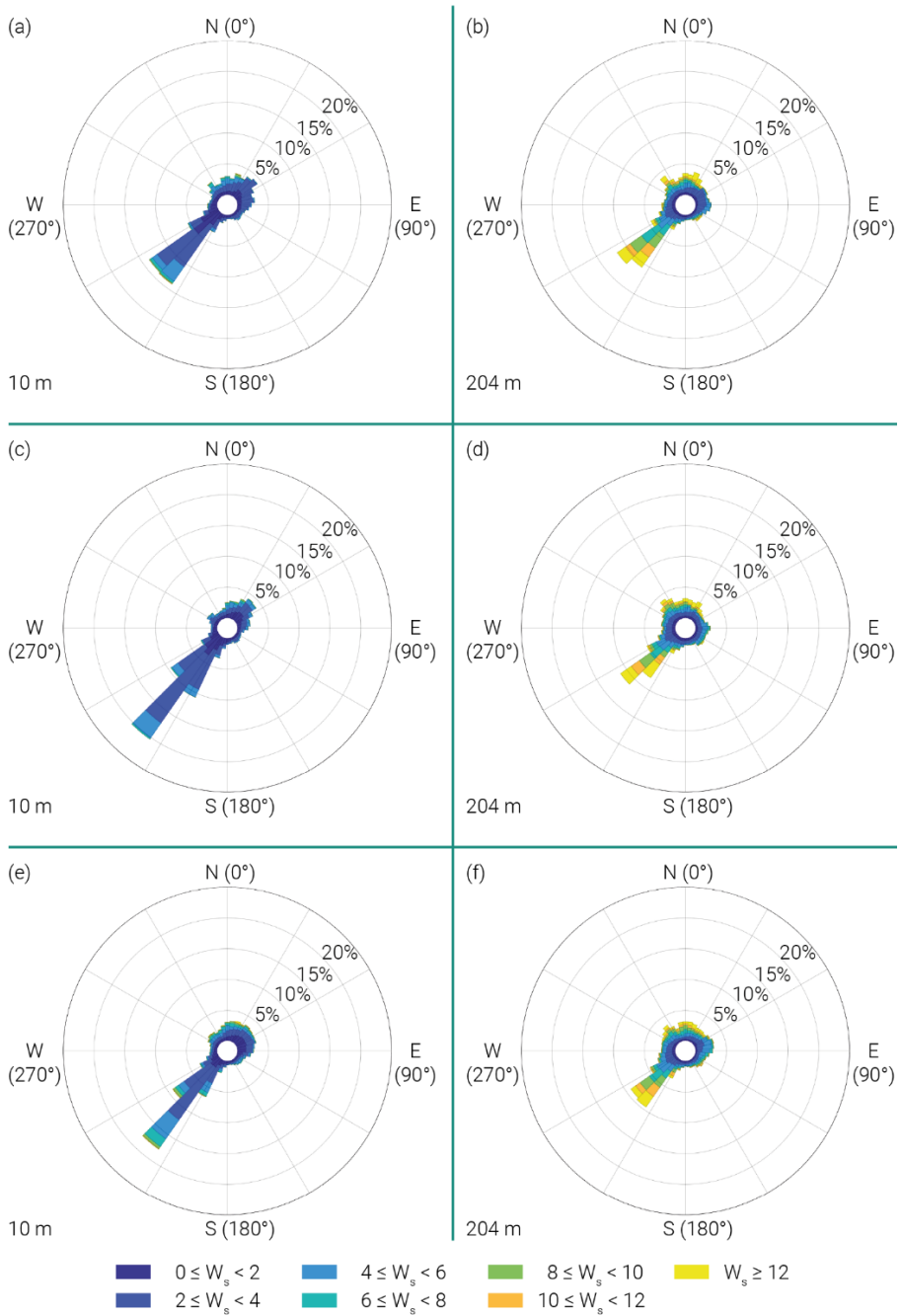


Figure 4.16 - Wind roses for the point (14,12):

(a) MM5-MRF@10m, (b) MM5-MRF@204m, (c) MM5-PleimXiu@10m,
 (d) MM5- PleimXiu@204m, (e) MM5-BK@10m, (f) MM5-BK@204m

4.3 Mesoscale results

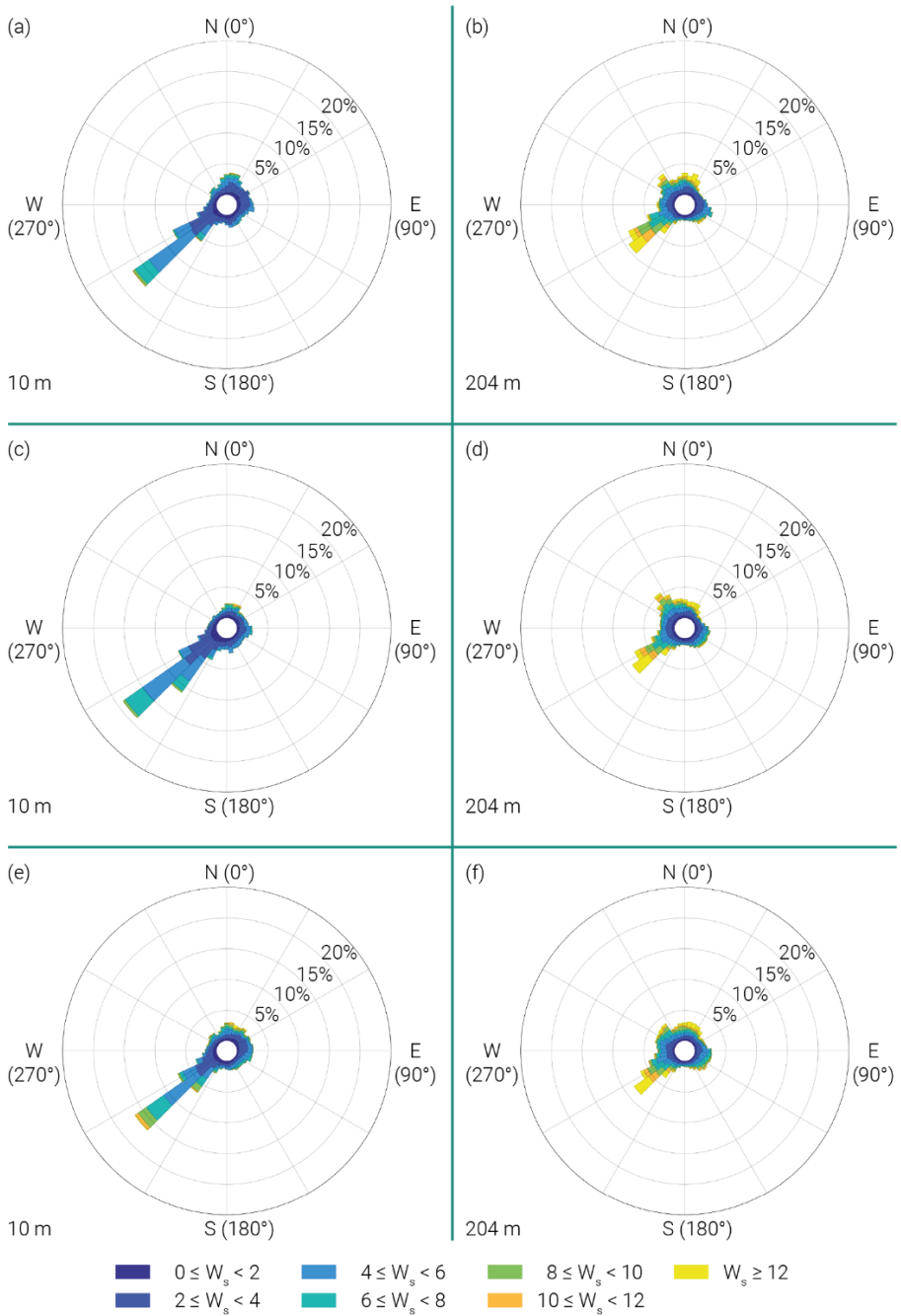


Figure 4.17 - Wind roses for the point (14,19):
 (a) MM5-MRF@10m, (b) MM5-MRF@204m, (c) MM5-PleimXiu@10m,
 (d) MM5- PleimXiu@204m, (e) MM5-BK@10m, (f) MM5-BK@204m

Chapter 4

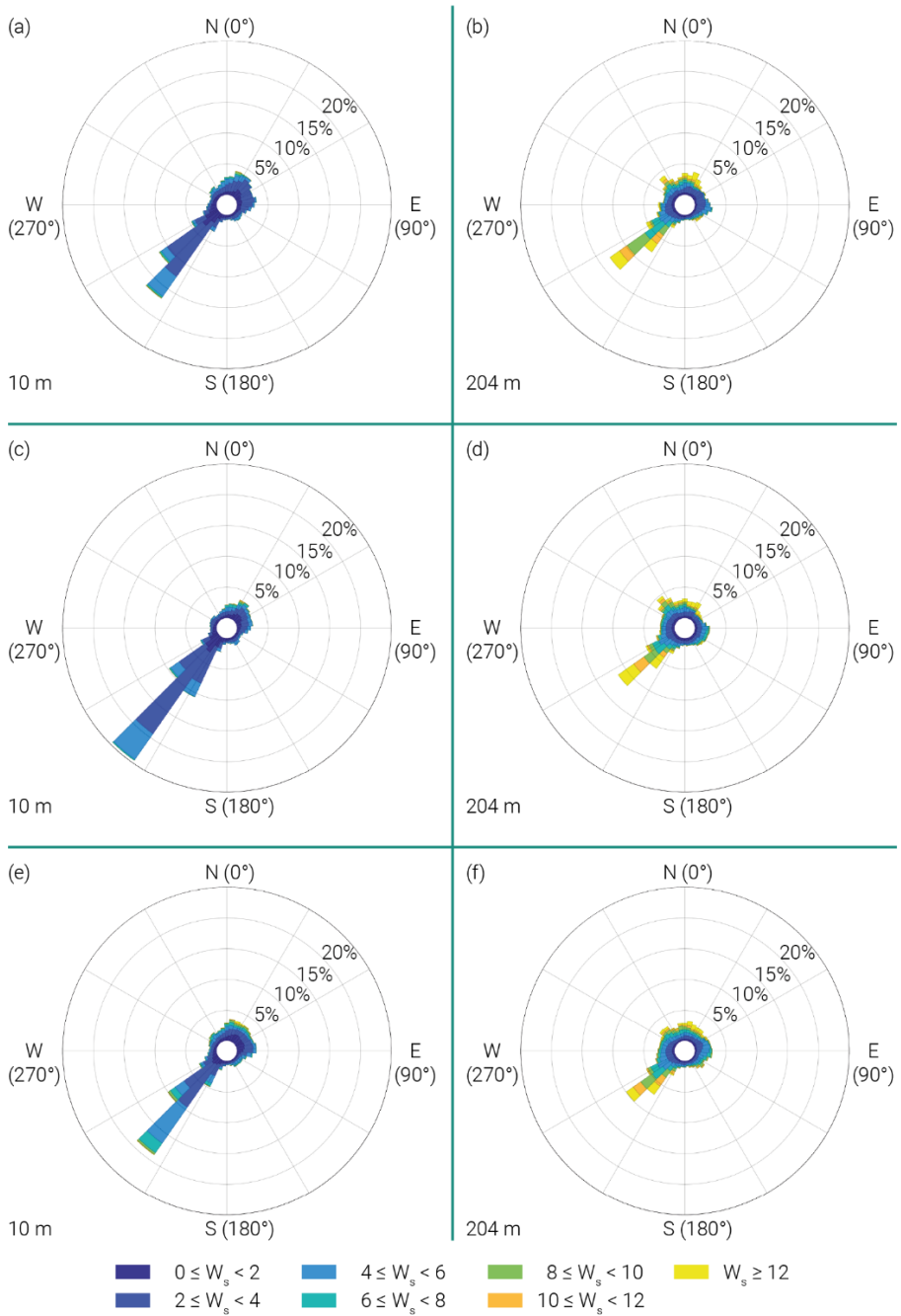


Figure 4.18 - Wind roses for the point (16,16):
 (a) MM5-MRF@10m, (b) MM5-MRF@204m, (c) MM5-PleimXiu@10m,
 (d) MM5- PleimXiu@204m, (e) MM5-BK@10m, (f) MM5-BK@204m

5

Microscale simulations

The previous chapter underlines as the mesoscale analysis cannot correctly estimate flows within a built environment. Therefore, this chapter aims to quantify the effect of buildings on wind direction and velocity inside an urban area.

In the first part, the dataset essential for creating the numerical model is presented, then the procedure to transform this initial data into the numerical model is shown. The next part is centred on the set-up necessary for performing the four microscale analyses. In detail, the direction of each simulation coincides with the cardinal points. Finally, the outputs for every incoming wind direction are reported. These results will be coupled in the following chapters with those obtained from the mesoscale analysis.

5.1 Numerical model

The information, valid for the realization of the numerical model, is taken from the Italian regional cartography (Carta Tecnica Regionale - *CTR* in Italian acronym).

Chapter 5

This cartography is a product of each Italian region and does not present distortions; therefore, elements are represented with their real projection. The standard scale is 1:5000 or 1:10000, although there may be higher scales.

The CTR data are organized in numbered tiles defined as follows:

- the first three digits are the identification number of the sheet of the Italian map (from 001 to 652),
- the following two digits identify the section in the 4x4 grid by numbering it progressively from West and North (from 01 to 16),
- the last digit is 0 if it refers to the whole section or 1-4 if it is related to one of the elements (numbered clockwise starting from the northeast).

Figure 5.1 summarizes the points mentioned above. The elements in Figure 5.1b are collocated within each section of Figure 5.1a.

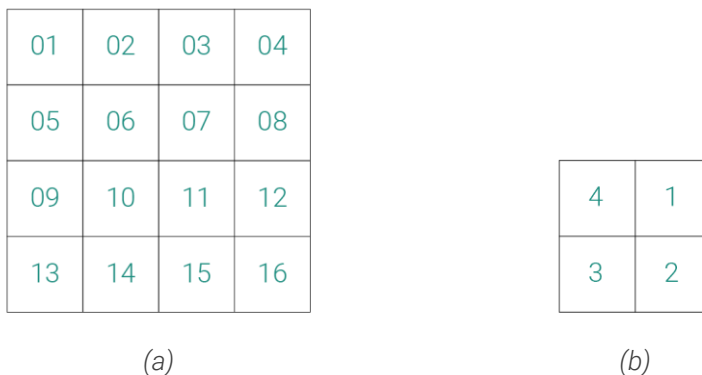


Figure 5.1 - (a) division of a sheet into 16 sections,
(b) subdivision of a section into four elements

The data are open to download from the site of the region of interest. In this case, the 1:5000 scale files for the city of Pescara (Abruzzo, Italy) are downloaded from the Abruzzo region portal¹.

Figure 5.2 shows the sheet numbers for the Abruzzo region, with a focus on the sheet selected for being referred to the investigated area.

¹ http://opendata.regione.abruzzo.it/opendata/CTRN_Regione_Abruzzo_1_5000

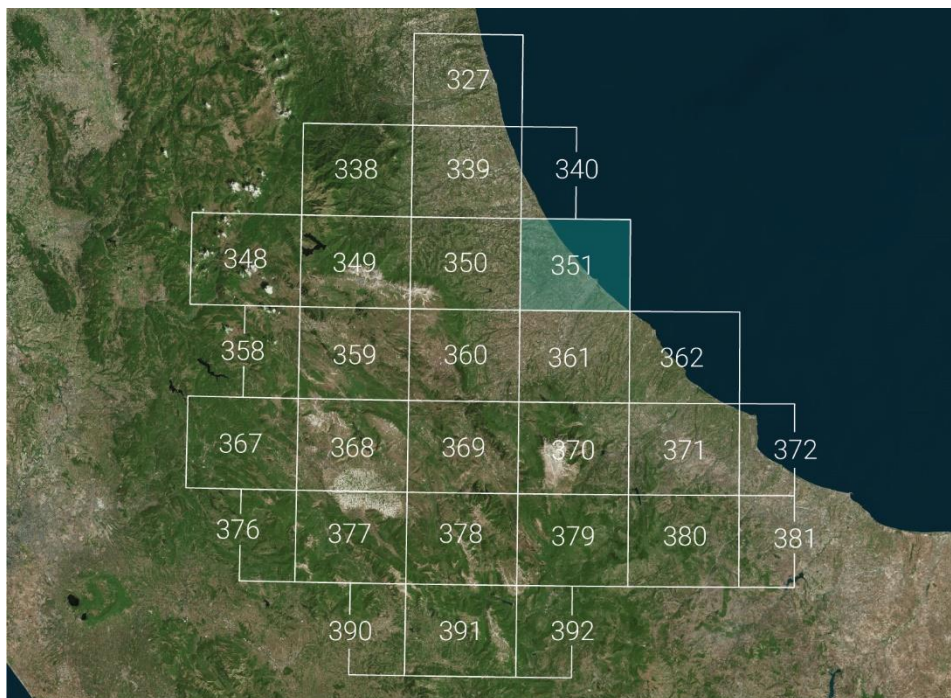


Figure 5.2 - Sheet numbers of the Abruzzo region

In detail, the downloaded files, in *dxf* format, are:

- 351102 (sheet number: 351, section number: 10, element number: 2),
- 351112 (sheet number: 351, section number: 11, element number: 2),
- 351113 (sheet number: 351, section number: 11, element number: 3),
- 351141 (sheet number: 351, section number: 14, element number: 1),
- 351142 (sheet number: 351, section number: 14, element number: 2),
- 351151 (sheet number: 351, section number: 15, element number: 1),
- 351152 (sheet number: 351, section number: 15, element number: 2),
- 351153 (sheet number: 351, section number: 15, element number: 3),
- 351154 (sheet number: 351, section number: 15, element number: 4).

The result is show in Figure 5.3.

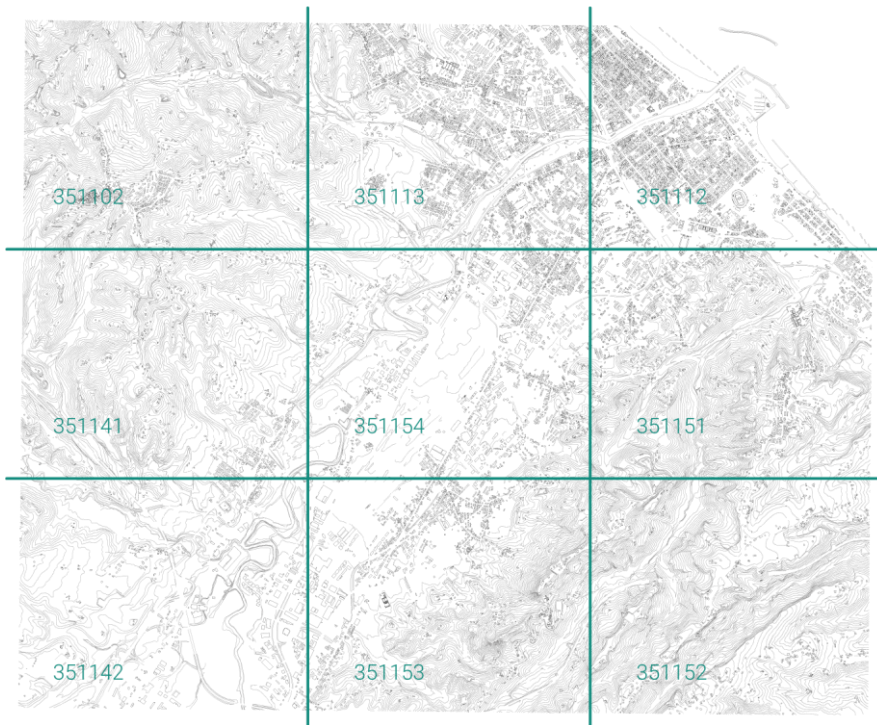


Figure 5.3 - CTR result for the area of interest

The peculiarity of the CTR data is the presence of information regarding the elevation of the elements. The buildings, for example, are represented by closed polygons positioned at the ridge height of the roof. The scheme presented in Figure 5.4 clarifies what is here explained.



Figure 5.4 - Elevation of buildings in the CTR in plan (a) and section (b) view

Afterwards, the files are processed with the commercial software *Rhinceros*² version 7 developed by Robert McNeel & Associates. Figure 5.5 shows the terrain surface obtained from interpolation of the CTR contour lines.

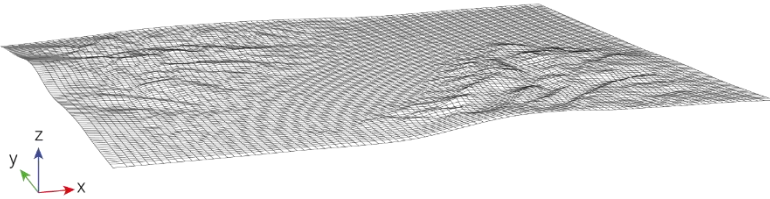
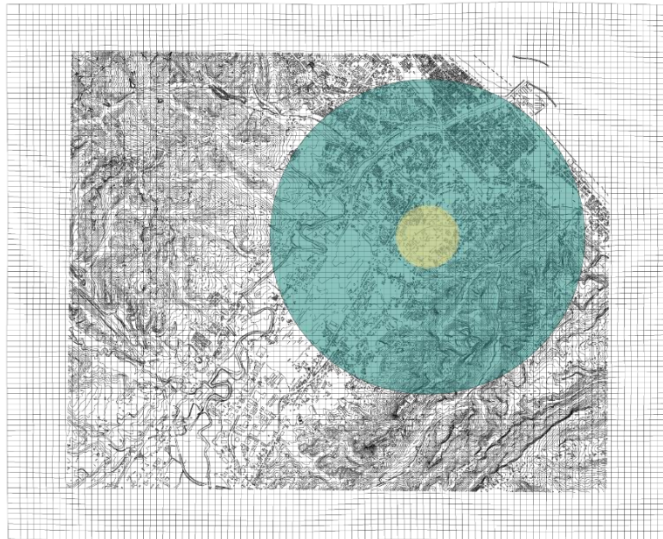


Figure 5.5 - Terrain surface obtained from CTR data

From this surface, a circular portion with a diameter of 6.0 km is "clipped" (green circle in

Figure 5.6), obtaining the surface of the terrain useful for realizing the numerical model. On the other hand, the yellow circle indicates the buildings extracted, thus generating the condition depicted in Figure 5.7a. The polygons of the buildings, then, are extruded downward (Figure 5.7b).



² <https://www.rhino3d.com/>

Figure 5.6 - Circular portion from CTR

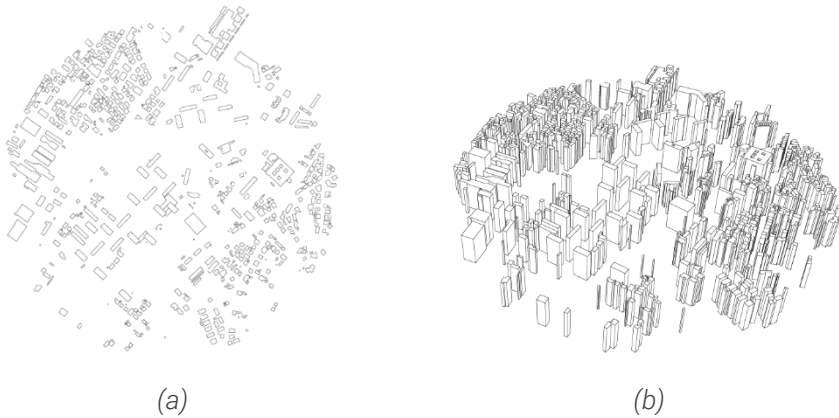


Figure 5.7 - Buildings extracted from CTR: (a) top view and (b) axonometric view

The geometries of the terrain and buildings are merged into a single model where the buildings are perfectly positioned on the terrain surface. The lateral surface is divided into 12 sectors (every 30°) to study the incoming wind from different directions. The top layer is positioned approximately 500.0 m from the ground.

The components of the obtained model are shown in Figure 5.8.

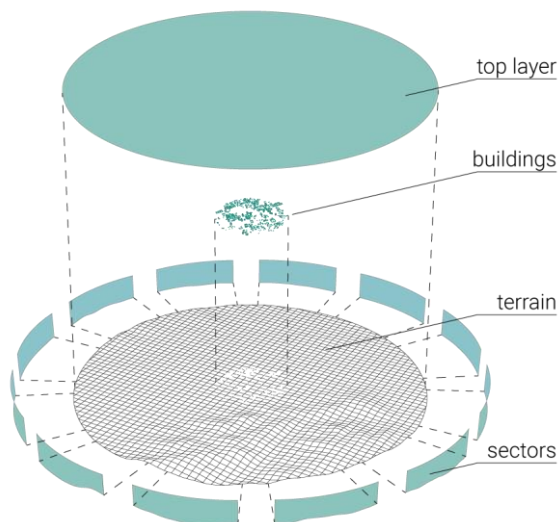


Figure 5.8 - Axonometric view with identification of components

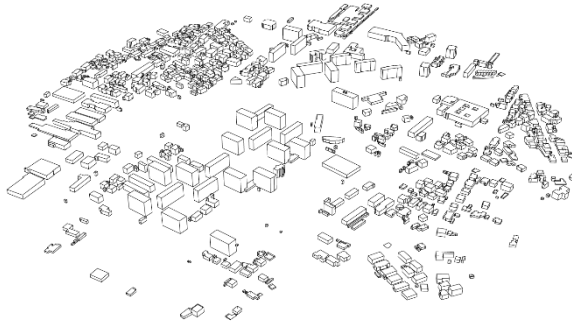


Figure 5.9 - Axonometric view of buildings

5.2 Numerical set-up

The commercial software chosen to perform the microscale analyses is *STAR-CCM+* (acronym of Simulation of Turbulent flow in Arbitrary Regions - Computational Continuum Mechanics) developed by CD-adapco. This software is used to solve fluid flow and heat transfer problems simultaneously.

The 3D model obtained from the CTR data (Figure 5.8) is exported as Parasolid (*.x_t) format and then imported into the CFD simulation software.

Each element is introduced separately to simplify the procedure of assigning boundary conditions (BC) and parameters for the meshing phase.

Moreover, the terrain surface is divided into two parts:

- an outer ring with the largest diameter of 6.0 km and the smallest diameter of 3.0 km,
- an inner circle with a diameter of 3.0 km.

The BCs are assigned as schematized in Figure 5.10. The sectors can be the *Velocity Inlet* or *Pressure Outlet* based on the performed analysis. The building surfaces are set as *No-Slip Wall*. With the *No-Slip* method, relative fluid velocity tangential to the wall is set to zero. Also, the terrain surface is modelled as *No-Slip Wall*; instead, the top surface is set as *Symmetry Plane*. This boundary represents an imaginary plane of symmetry in the simulations.

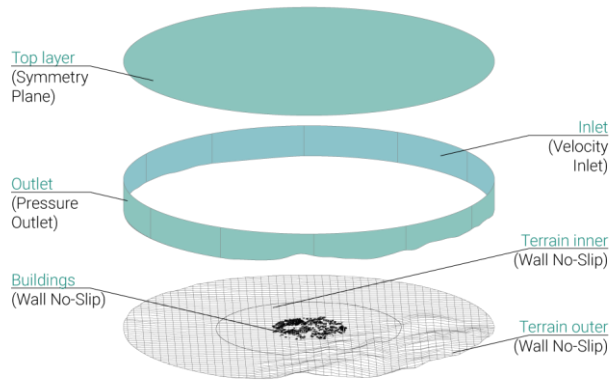


Figure 5.10 - Schematization of the BCs

The following table provides an overview of the adopted BCs.

	Buildings	Terrain inner	Terrain outer	Sectors	Top
BC Type	Wall No-Slip	Wall No-Slip	Wall No-Slip	Velocity Inlet / Pressure outlet	Symmetry Plane

Table 5.1 - BCs type

The *Velocity Inlet* is set with a logarithmic profile according to (Eq. 2.5). The wind direction is managed by the θ parameter, which can assume values of 0° , 90° , 180° or 270° depending on whether the incoming wind comes from the West, South, East, or North respectively. The reference wind speed is 6.0 m/s. The components of the velocity vector are then given by:

$$[U \cos(\theta), U \sin(\theta), 0]$$

The *Pressure Outlet*, on the other hand, is equal to 0.0 Pa.

Regarding the mesh parameters, different mesh size values are assigned for each of the components. These values are obtained after some attempts to achieve both good accuracies in the results and computationally sustainable meshes.

A summary of the parameters used for each element is reported in Table 5.2.

Element	Number of Prism Layers [-]	Prism Layer Stretching [-]	Prism Layer Thickness [m]	Minimum Surface Size [m]	Target Surface Size [m]
Buildings	5	1.5	1.0	2.0	5.0
Terrain in	3	1.5	1.0	5.0	10.0
Terrain out	3	1.5	1.0	15.0	20.0
Top	-	-	-	40.0	60.0
Sectors	3	1.5	1.0	10.0	40.0

Table 5.2 - Mesh parameters

The *number of prism layers* parameter controls the number of cell layers generated within the prism layer on a part surface or boundary. *Prism layer stretching* sets the target growth rate of successive prism layers away from the wall. The *Prism layer thickness* specifies the total thickness of the prismatic cell layers. Finally, the *Minimum surface size* is the absolute minimum size due to refinement.

With the parameters in Table 5.2, a polyhedral mesh is generated. The number of cells is 2,645,737, while the number of vertices is 12,212,485.

For the polyhedral volume mesh, the mesher starts by building a tetrahedral mesh, then combines cells to generate the poly mesh with a lower number of cells.

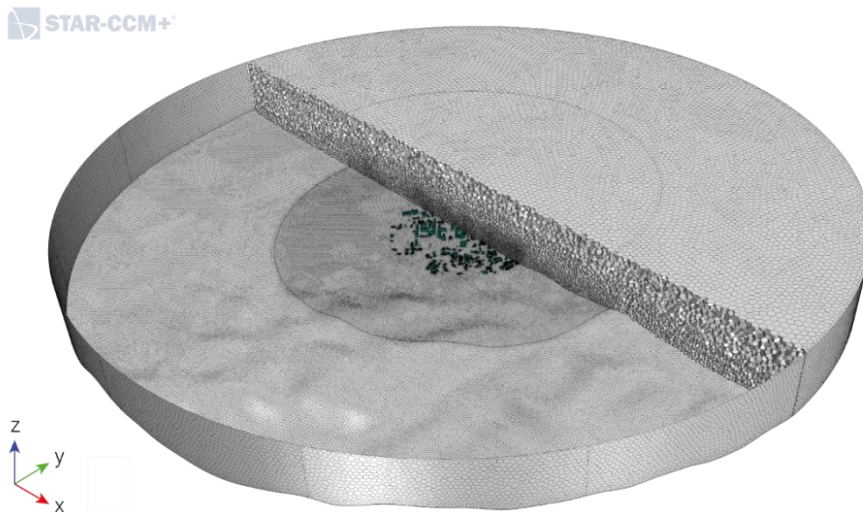


Figure 5.11 - Volume mesh of the numerical model

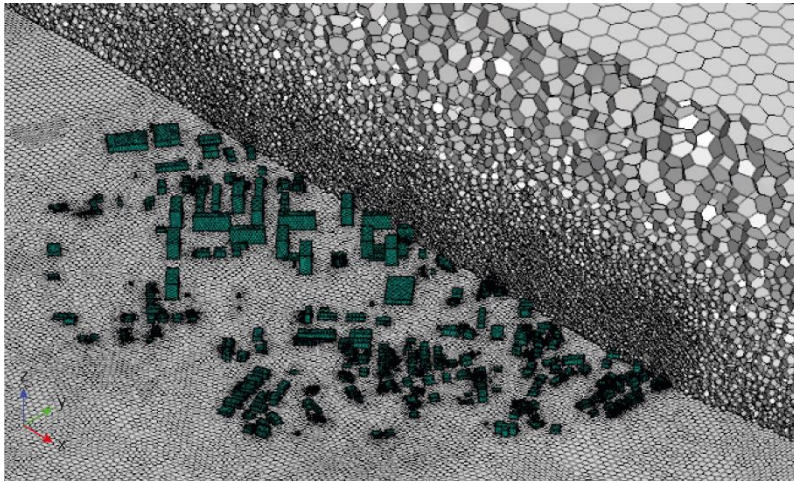


Figure 5.12 - Detailed mesh on buildings

The model selected for the analyses is the unsteady Reynolds-Averaged Navier-Stokes (U-RANS), while the turbulence is considered through the two-equation $k-\epsilon$ model.

5.3 Microscale results

The analyses are conducted along with the four cardinal points. The following schemes summarize the BCs used for each simulation.

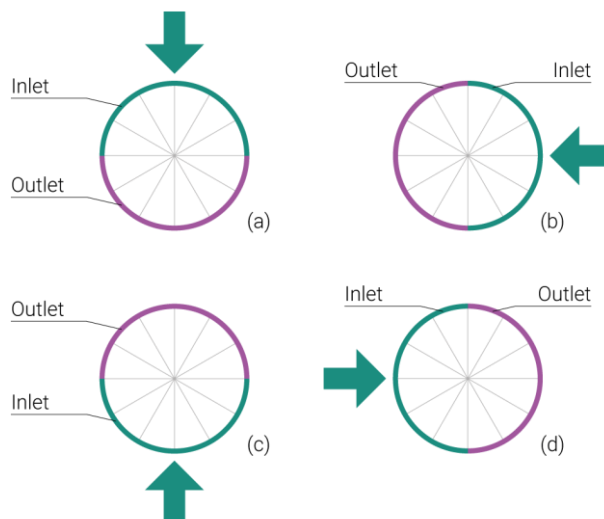


Figure 5.13 - Lateral BCs for North (a), East (b), South (c), and West (d)

5.3 Microscale results

The results obtained from the four analyses are shown from Figure 5.14 to Figure 5.29. The presence of buildings significantly impacts on the motion field. The buildings, in fact, cause a decrease in the wind speed due to the effect that this has on the surfaces of the buildings. For this reason, the area immediately behind the buildings is characterized by a decrease in velocity magnitude (concerning the analyzed direction). On the other hand, pressure levels are higher near the surfaces of the first buildings directly impacted by the incoming wind.

The orography of the terrain also plays a significant role. In all four cases, the wind direction impacts the buildings differently from the starting one, clearly visible from the depicted streamlines.

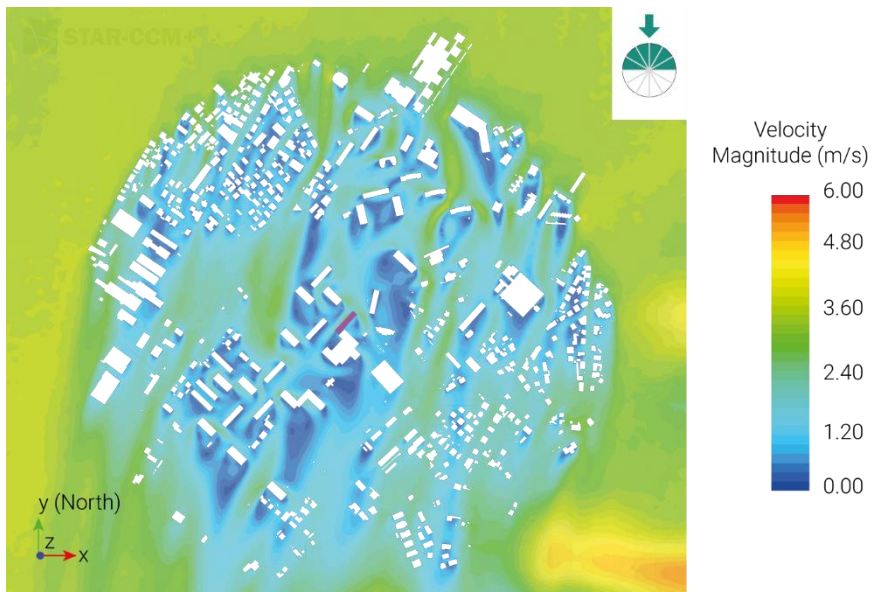


Figure 5.14 – Top view of velocity magnitudes: incoming wind from North (-y)



Figure 5.15 – Top view of pressure levels and streamlines: incoming wind from North (-y)

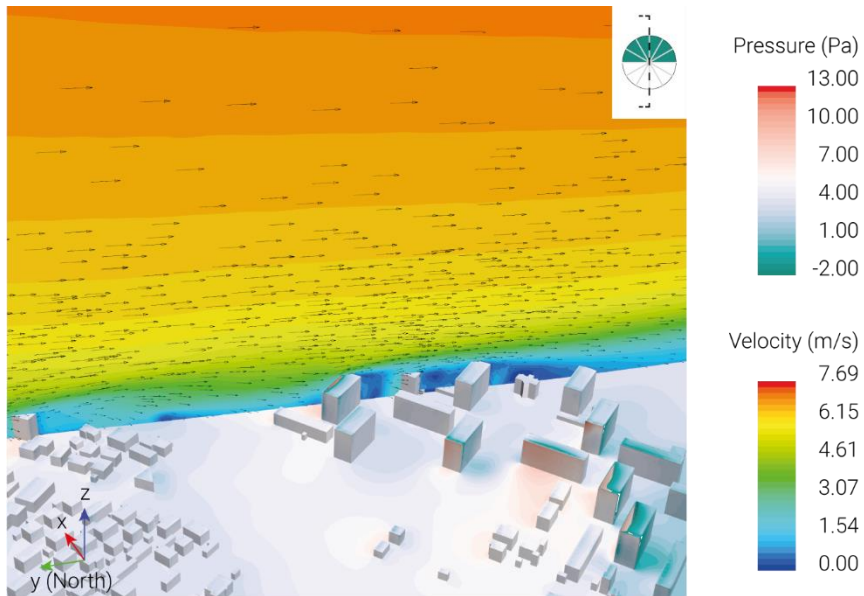


Figure 5.16 - Section view of velocity magnitudes on the reference building and pressure levels: incoming wind from North (-y)



Figure 5.17 - Axonometric view of streamlines in the urban fabric:
incoming wind from North (-y)

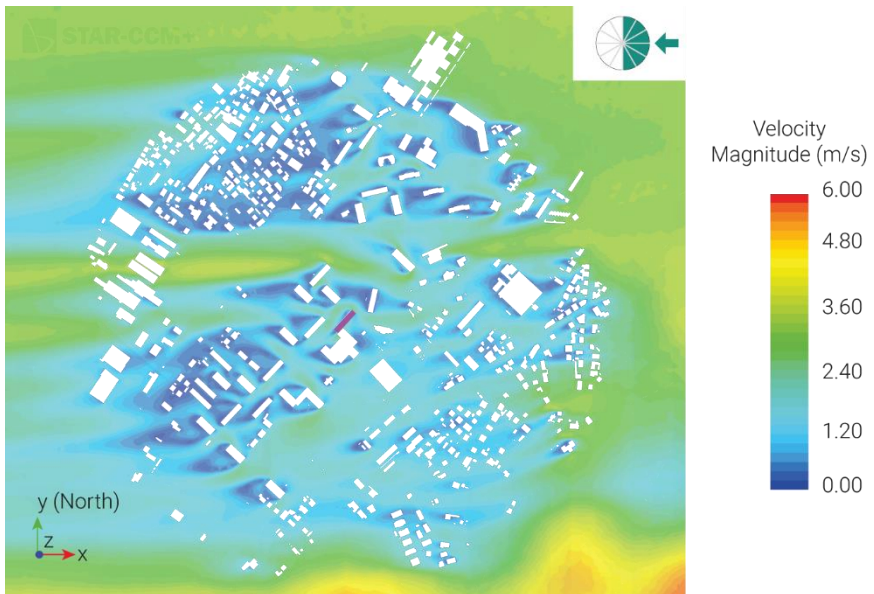


Figure 5.18 – Top view of velocity magnitudes: incoming wind from East (-x)

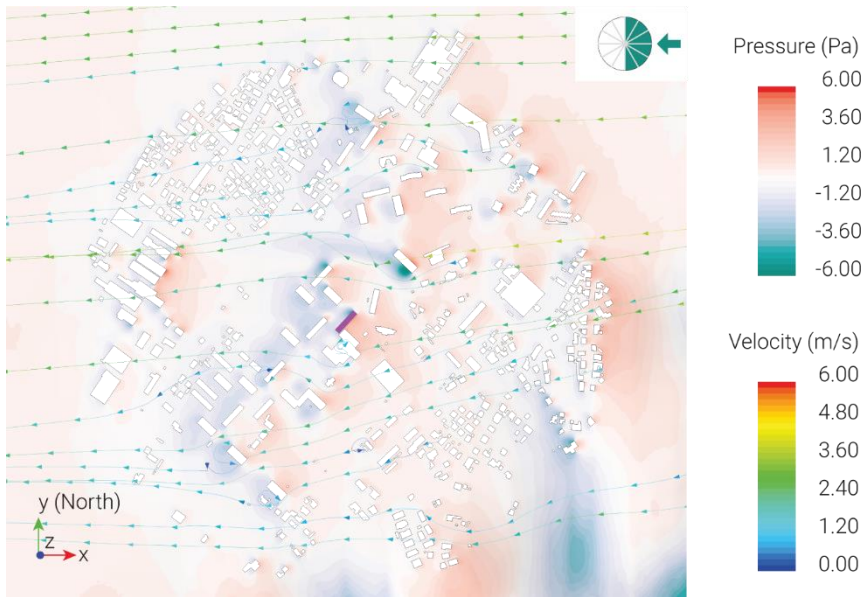


Figure 5.19 – Top view of pressure levels and streamlines:
incoming wind from East (-x)

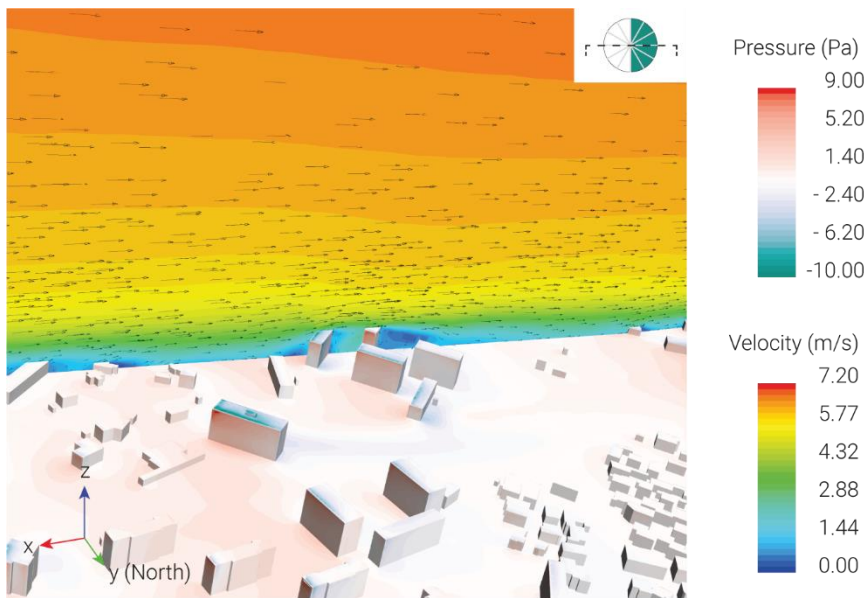


Figure 5.20 - Section view of velocity magnitudes on the reference building and
pressure levels: incoming wind from East (-x)

5.3 Microscale results

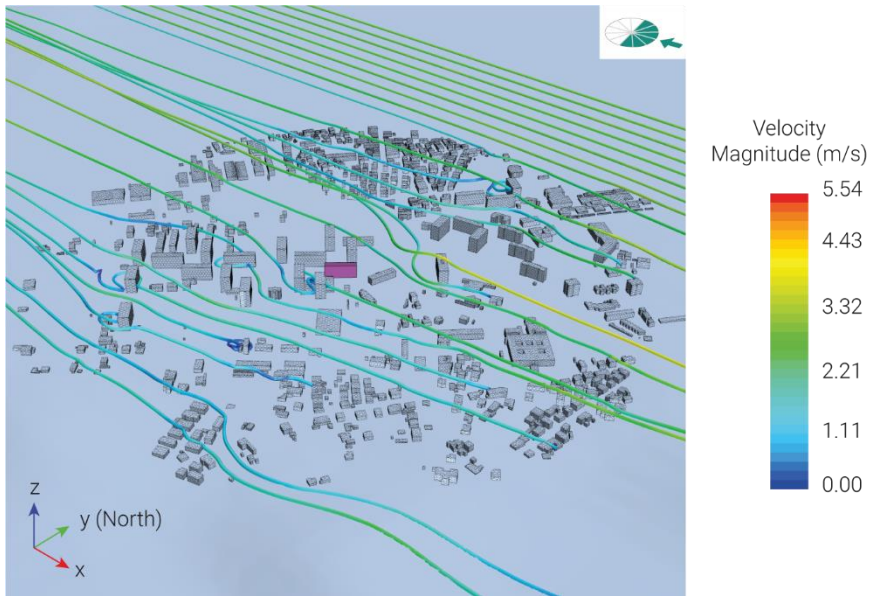


Figure 5.21 - Axonometric view of streamlines in the urban fabric:
incoming wind from East (-x)

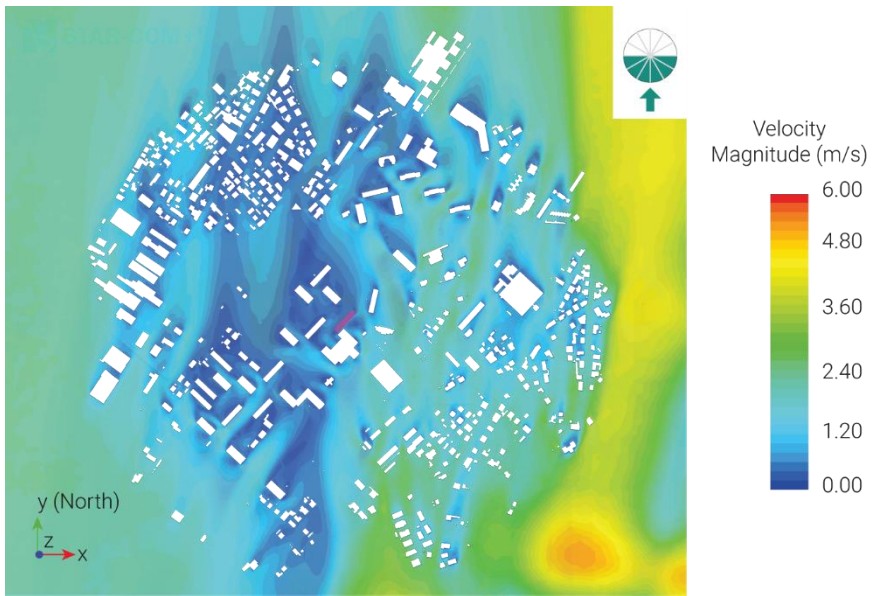


Figure 5.22 – Top view of velocity magnitudes: incoming wind from South (+y)



Figure 5.23 – Top view of pressure levels and streamlines: incoming wind from South (+y)

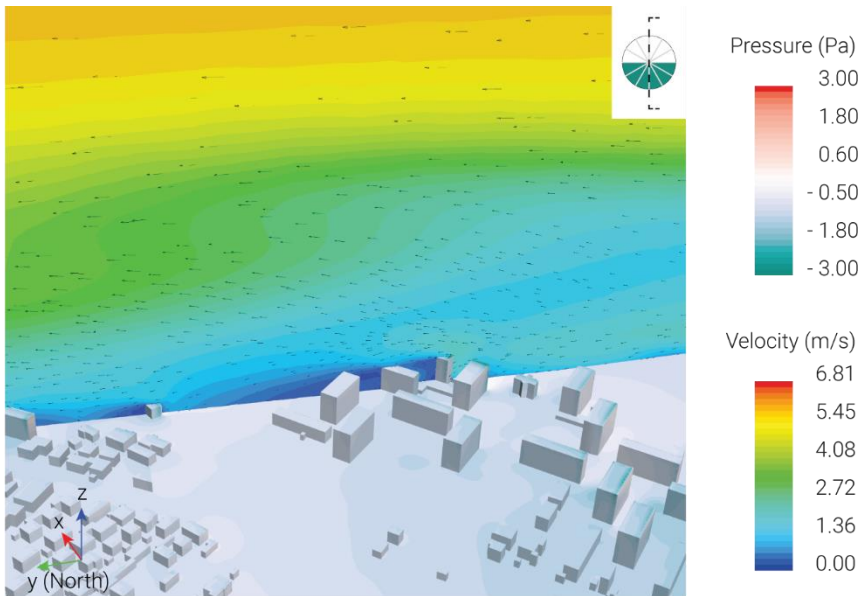


Figure 5.24 - Section view of velocity magnitudes on the reference building and pressure levels: incoming wind from South (+y)



Figure 5.25 - Axonometric view of streamlines in the urban fabric:
incoming wind from South (+y)

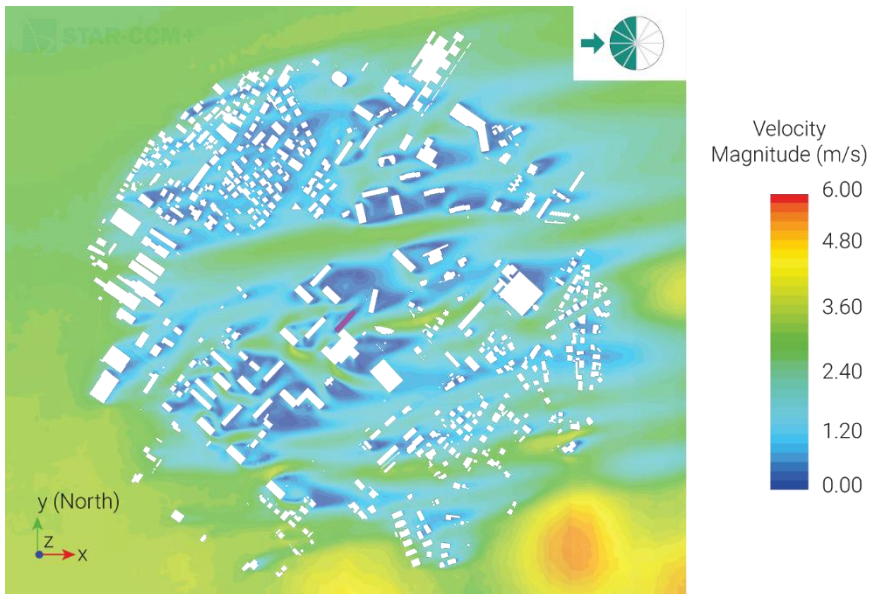


Figure 5.26 – Top view of velocity magnitudes: incoming wind from West (+x)

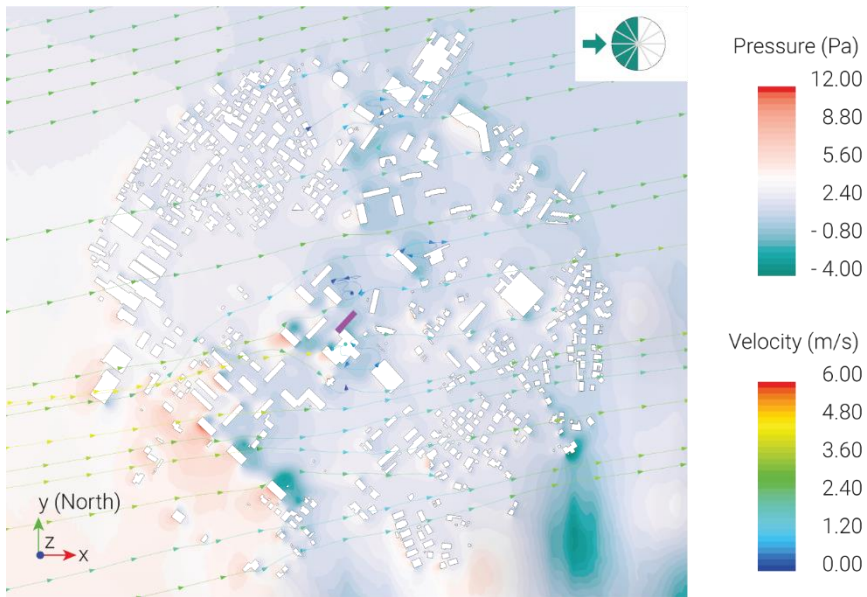


Figure 5.27 – Top view of pressure levels and streamlines: incoming wind from West (+x)

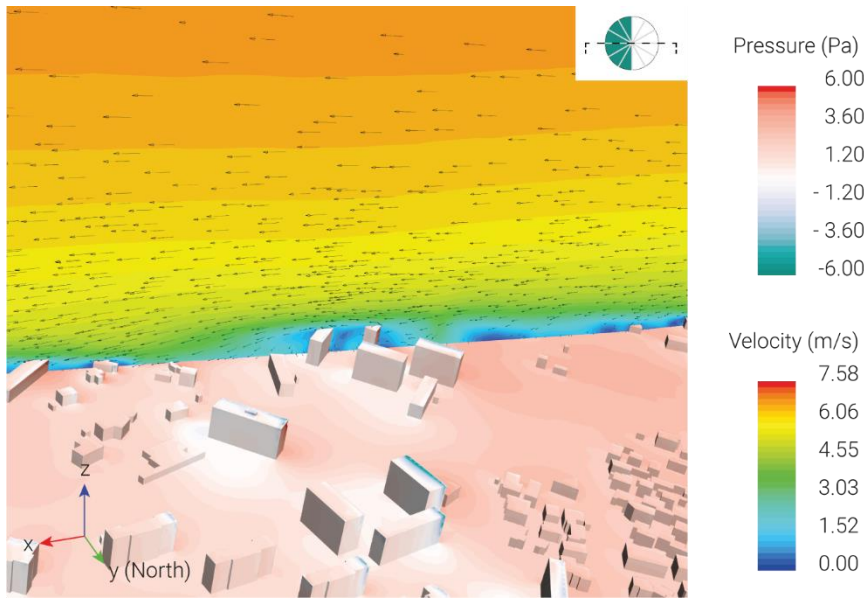


Figure 5.28 - Section view of velocity magnitudes on the reference building and pressure levels: incoming wind from West (+x)

5.3 Microscale results

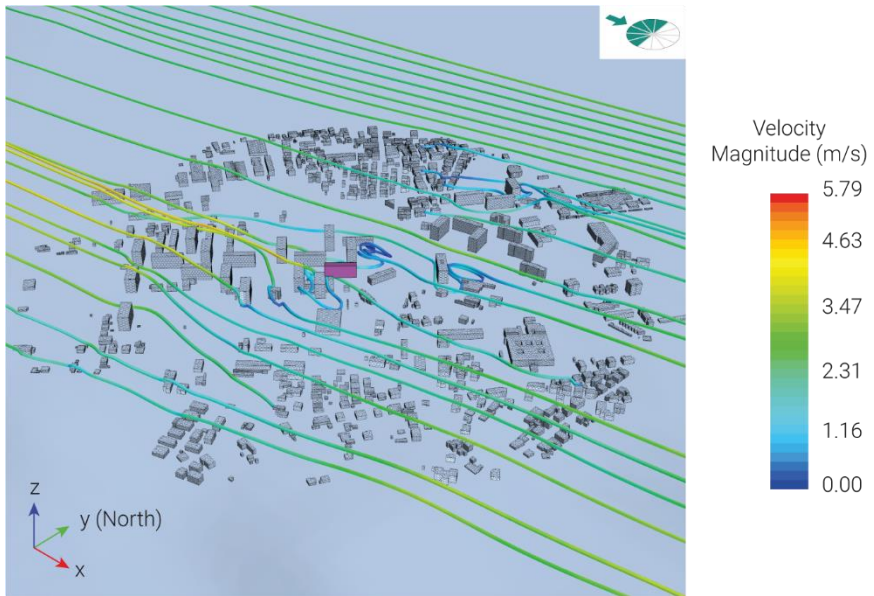


Figure 5.29 - Axonometric view of streamlines in the urban fabric:
incoming wind from West (+x)

6

Coupling procedure

This chapter presents the coupling procedure between the results obtained, respectively, from a numerical weather prediction software and a computational fluid dynamics one.

In the first part, the idea behind this procedure is explained. It evaluates the wind speed and direction at a specific point inside the domain from the information obtained from both mesoscale and microscale analysis. To do this, the numerical domains of each simulation have the same center. Specifically, the domain of the microscale analysis is contained in the smaller domain of the mesoscale one.

In the second part of the chapter, a comparison is provided between the pre and post-coupling results.

All the procedure described in this chapter is implemented in a script written in the *GNU Octave* language (Appendix A).

6.1 Basic principles

The coupling procedure presented in this work calculates the wind speed and direction at a specific point inside the domain from various virtual anemometers.

The anemometers are extracted at the same position and elevations from the terrain for the mesoscale and microscale analysis.

The velocity and direction calculated through this procedure will become the boundary conditions for the building energy simulations on a case study.

The flow charts reported in the following lines are referred to the operations performed, respectively, for the microscale and mesoscale side.

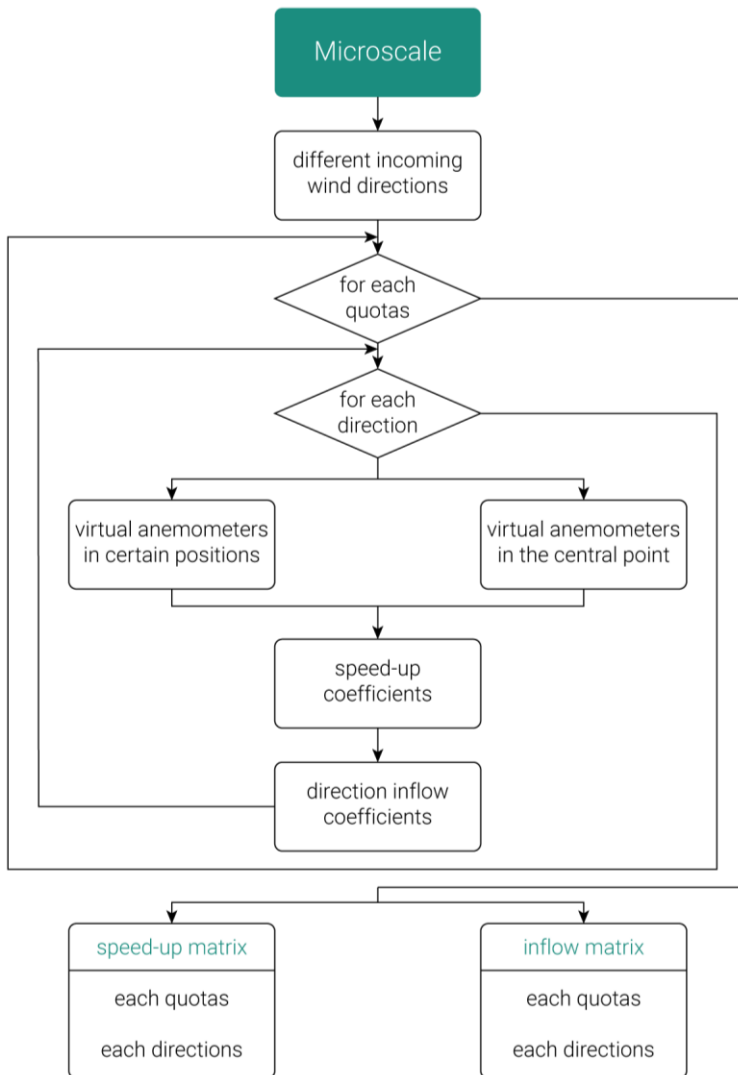


Figure 6.1 - Flowchart microscale side

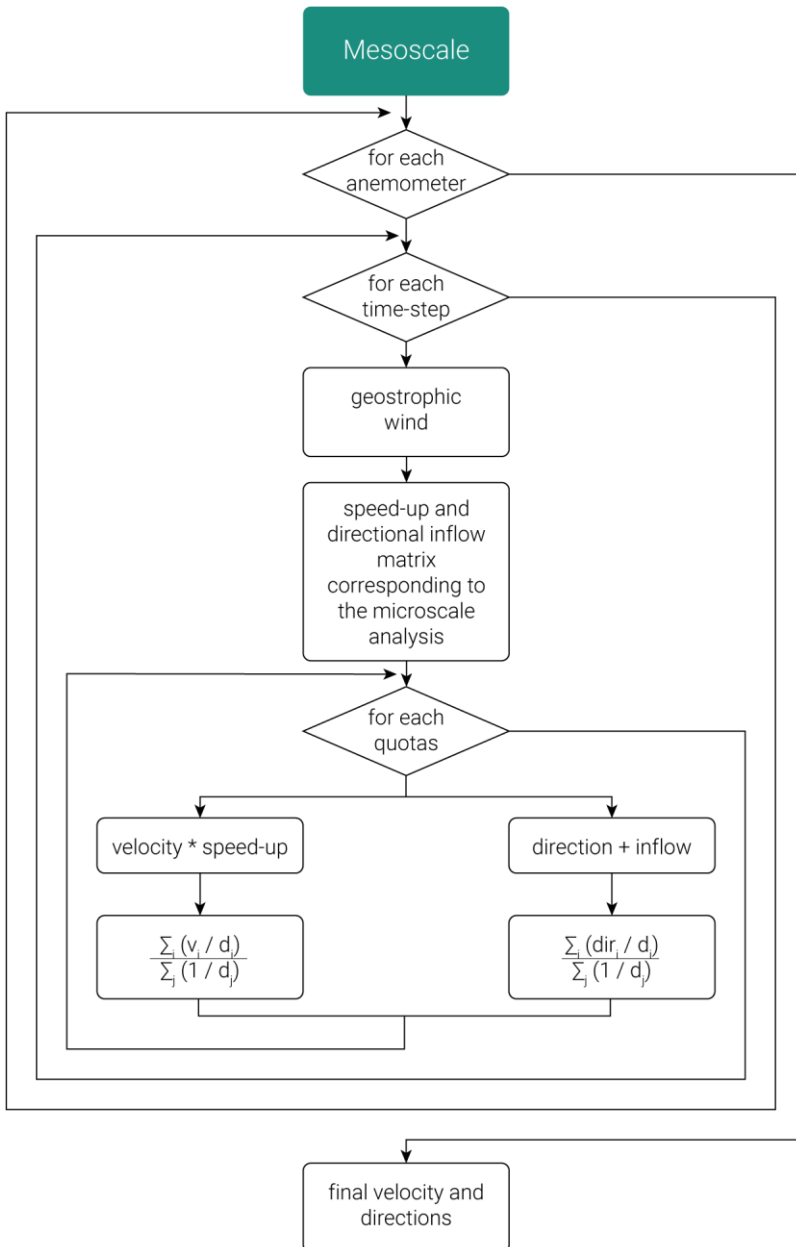


Figure 6.2 - Flowchart mesoscale side

The initial step to be performed for the coupling procedure is the selection of some virtual anemometers within the numerical domain. The location of these anemometers must be the same for both the microscale and mesoscale simulations. The extraction elevations must also be the same for both analyses. The extraction elevations are: 30.0, 51.0, 71.0, 92.0, 122.0, and 163.0 m above ground level.

For every microscale analysis, wind speed and direction information is extracted from each anemometers. This information is also collected at the center of the domain that corresponds with the building on which the energy simulations will be performed.

At this point, both the speed-up coefficient and the directional inflow coefficient are calculated. The first one consists of the ratio between the velocity in the central point and the i -th anemometer. For the second coefficient, instead, the angle between the velocity vector at the central point and the velocity vector of the i -th anemometer is calculated.

These operations provide the speed-up matrix, and the directional inflow matrix for each incoming wind direction analyzed in the present work. Through the speed-up matrix, it is possible to know if there is an acceleration or deceleration of the wind for each extracted anemometers and analyzed altitude. Similarly, by considering the directional inflow matrix, it is possible to see if there are changes in direction for every extracted anemometer (and altitude).

The following steps, graphically represented in the flowchart depicted in Figure 6.2, concern the operations performed on the mesoscale side.

As explained in Chapter 4, the mesoscale analysis is performed over a one-year period obtaining outputs with a 4-minute time-step. Thus, the position and the elevations of the virtual anemometers are the same as those extracted for the microscale analyses.

For each time-step, the geostrophic wind direction (at 204.0 m above ground level), helpful for identifying the specific speed-up and directional inflow matrix previously calculated, is extracted.

The i -th velocity is then multiplied with the corresponding speed-up coefficient and adopted to estimate the final velocity. Since each anemometer is placed at different

Chapter 6

distances from the point of interest, the anemometers positioned close to it will have a more significant influence than those located far away. In particular, the final velocity is calculated by:

$$\frac{\sum_i(v_i/d_i)}{\sum_j(1/d_j)} \quad (\text{Eq. 6.1})$$

where v is the velocity of the i -th anemometer at a given altitude and d is the distance of the anemometer from the center of the domain.

In the same way, the final direction of the velocity vector is also calculated.

$$\frac{\sum_i(\text{dir}_i/d_i)}{\sum_j(1/d_j)} \quad (\text{Eq. 6.2})$$

where dir is the direction of the velocity vector of the i -th anemometer summed with the respective inflow coefficient and d is the distance the anemometer has from the center of the domain.

6.2 Results

The coupling procedure is performed for each mesoscale simulation. Then wind speeds and directions are calculated for the *MM5-MRF*, *MM5-PleimXiu*, and *MM5-BK* analyses. Finally, the wind roses obtained by this procedure are compared with those generated without the coupling approach (Figure 6.3).

In all three cases, there is a significant decrease in wind speed and a change in the direction of the wind occurrence due to nearby buildings. The post-coupling wind roses are more homogeneous than the pre-coupling ones, maintaining a peak in the S-W direction even if it is slightly rotated towards the West.

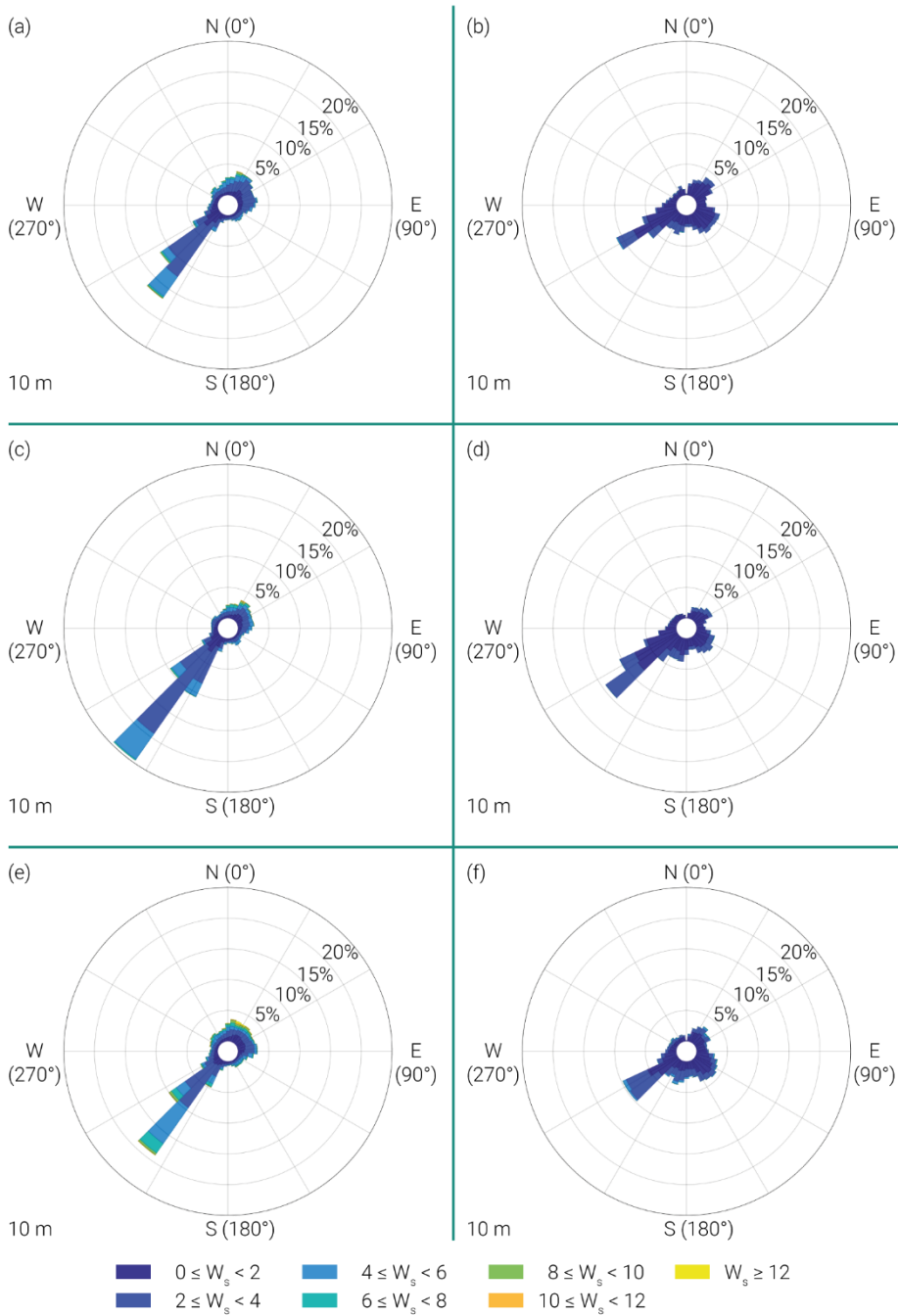


Figure 6.3 - Wind roses for the point (16,16):
 (a) MM5-MRF_pre, (b) MM5-MRF_post, (c) MM5-PleimXiu_pre,
 (d) MM5-PleimXiu_post, (e) MM5-BK_pre, (f) MM5-BK_post

7

Dynamic Energy Simulations

The present chapter is focused on dynamic energy simulations carried on for estimating the energy performance, expressed in terms of energy consumption of a case study, by using climatic files obtained, respectively, before and after the coupling procedure described in the previous section. The choice of performing dynamic analyses is due to more accuracy and better-quality estimations than static or quasi-static simulations. Creating a reliable model for these investigations requires defining various parameters such as building envelope details, thermal properties of materials and operational schedules. These features should be coupled with the suitable dataset's choice, which better represents the building's climate.

7.1 Presentation of the case study

Dynamic Energy simulations are performed to evaluate the impact that local winds have on building energy consumption before and after the coupling procedure. The analyses are carried on by assuming as a case study a reference building located in San Donato district in Pescara. The area is characterized by the Mediterranean climate, with mild winters and hot, sunny summers.

7.1 Presentation of the case study



Figure 7.1 - Location of the case study

The building, owned by ATER (regional enterprise for social housing), was erected in 1983 according to the regulation codes and the construction techniques of the time. It is part of a residential area, and it is selected for being representative of the existing European heritage, considering that 60% of dwellings were built between 1960 and 1990. As shown in Figure 7.2, it has minor architectural value and hosts 48 apartments, of which 12 with a total surface equal to 50.0 m² (A apartment) and 36.0 of 70.0 m² (B apartment). The A apartments, two at every level, are located at the heads of the structure.

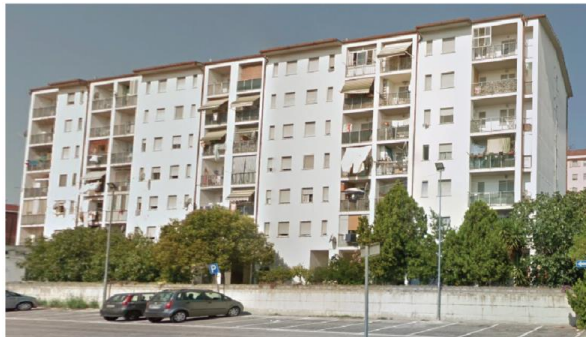


Figure 7.2 - Reference building for the energy analyses

The architectural plans and elevations of the case study are, respectively, depicted from Figure 7.3 to Figure 7.7. The building has seven stories, a total height equal to 21.5 m, and a rectangular shape with 60.0 m and 12.0 m as main dimensions. The inter-storey height is 2.7 m, except for the ground floor that has a 3.5 m height. The

Chapter 7

garages and the entrance are placed on the ground floor, while residential apartments lay on the upper levels.

It is essential to underline that the building envelope does not comply with the energy sector's current standard. The perimeter walls, in fact, does not present any insulation materials but only an air gap enclosed by two brickwork layers, with a total transmittance value equal to $1.46 \text{ W/m}^2\text{K}$. Moreover, single glass windows contribute to the building inadequate performance, especially during the coldest months.

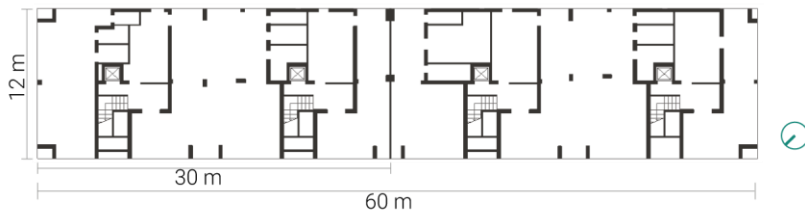


Figure 7.3 - Architectural plan of the ground floor

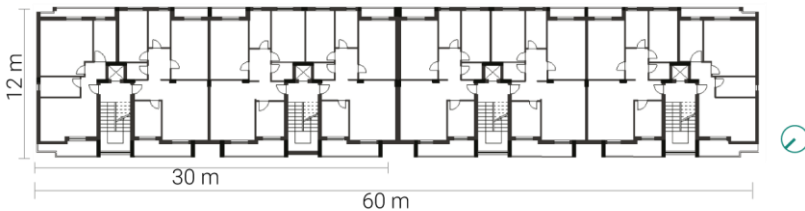


Figure 7.4 - Architectural plan of the typical floor

7.1 Presentation of the case study



Figure 7.5 - South-East elevation



Figure 7.6 - North-West elevation

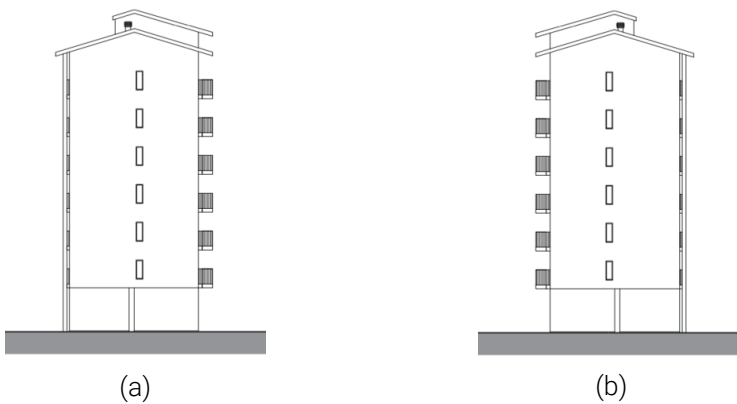


Figure 7.7 - North-East elevation (a) and South-West elevation

7.2 Numerical model and set-up

A preliminary stage for the case study modelling is the definition of the missed information necessary for the energy analyses. Thus, important parameters such as the HVAC systems, the type of window glasses, and the transmittance values for the opaque surfaces are assumed based on the literature review data.

The reference building is subdivided into its thermal zones. The ground floor has four staircases, and each staircase is made of two thermal zones: one for the staircase itself and the other for a storage room. Therefore, the ground floor has eight thermal zones in total. On the other storeys, eight apartments at each level, of which two are A-type (50.0 m²) and six B-type (70.0 m²). The A apartment is composed of five thermal zones: kitchen (5.90 m²), dining room (15.78 m²), bedroom (16.09 m²), bathroom (4.76 m²) and circulation area (6.90 m²). The B apartment has, instead, seven thermal zones: kitchen (8.19 m²), dining room (23.86 m²), bedroom 1 (12.92 m²), bedroom 2 (17.30 m²), bathroom (7.08 m²), a storage room (2.27 m²), and circulation area (10.29 m²). Figure 7.8 depicts the thermal zone localization for the typical floor.

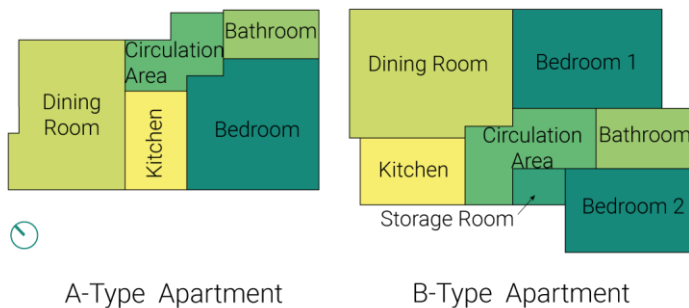


Figure 7.8 - Thermal zones on the typical floor

Based on the collected architectural maps and stratigraphies, the perimetral walls, with a 33.0 cm depth, do not present any insulation layer. In addition, there are 23.0 cm thick walls placed for delimiting the apartment from the staircase and partitions with 11 cm depth for subdividing each apartment into the above-mentioned thermal zones. No detailed information is inserted in the building documentation about glazed surfaces, and single glazing windows with transmittance (U) value

equal to $3.78 \text{ W/m}^2\text{K}$ are selected based on the analysis of the literature review [46]. The main thermals of the building envelope are synthesized in Table 7.1.








Building component	Material (outer to inner)	s [m]	λ [W/mK]	U [W/m ² K]
33cm External Wall 	(a) Lime plaster	0.02	0.80	1.46
	(b) Brickwork	0.13	0.84	
	(c) Air gap	0.08	0.30	
	(d) Brickwork	0.08	0.62	
	(e) Lime plaster	0.02	0.80	
23cm Semi-External Wall 	(a) Lime plaster	0.02	0.80	1.60
	(b) Brickwork	0.13	0.84	
	(c) Air gap	0.03	0.30	
	(d) Brickwork	0.08	0.62	
	(e) Lime plaster	0.02	0.80	
11cm Partitions 	(a) Plaster	0.02	0.16	1.27
	(b) Plasterboard	0.07	0.25	
	(c) Plaster	0.02	0.16	
Glazing 	Single clear glazing	0.006	-	3.78
Interstorey roof 	(a) Cast concrete	0.02	1.13	0.78
	(b) Concrete slab	0.16	0.16	
	(c) Plaster	0.02	0.16	
External roof 	(a) Ceiling tiles	0.02	0.06	0.61
	(b) Cast concrete	0.02	1.13	
	(c) Concrete slab	0.16	0.16	
	(d) Plaster	0.02	0.16	
Pavement 	(a) Cast concrete	0.02	1.40	3.37
	(b) Ceramic	0.02	1.30	

Table 7.1 - Building envelope components

No detailed information is present about the building heating and cooling system. Thus, data are collected from buildings similar to the case study for the construction period and typology. The building heating involves the use of radiators with a seasonal Coefficient of Performance (COP) equal to 0.84 and natural gas as source. For guarantying acceptable inhabitants' comfort condition, splits are installed for the cooling of the inner spaces. The cooling COP is set equal to 1.4, and the source is electricity.

Chapter 7

The temperature setpoints for heating and cooling for all thermal zones are, respectively, 22°C (with 20°C as set-back temperature) and 28°C. The metabolic factor is set to 0.90 for the whole building, and the clothing insulations is fixed at 0.50 clo for the summer season and 1.00 for the winter one. The model infiltration is considered constant and set at 0.70 ac/h. The input parameters which vary according to the activity of the thermal zone are depicted in Table 7.2.

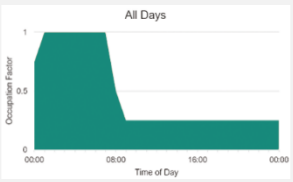


Thermal Zone [-]	Occupation [people/m ²]	Fresh Air [l/s-person]	Illuminance [Lux]	Schedule [-]
Bedroom	0.0229	10	100	
Bathroom	0.0187	12	150	
Kitchen	0.0237	12	300	
Storage room	0.0234	12	100	
During room	0.0169	10	150	

Table 7.2 - Input parameters for the simulations


Thermal Zone [-]	Occupation [people/m ²]	Fresh Air [l/s-person]	Illuminance [Lux]	Schedule [-]
Circulation area	0.0196	10	100	

Table 7.2 (continued) - Input parameters for the simulations

In order to take into account the influence that the neighboring buildings could have on the shading of the case study, the closest structures are inserted and modelled as block components. For the same reason, the presence of the balconies on the north-west façade is considered by placing external profiles, as it can be seen in Figure 7.9.

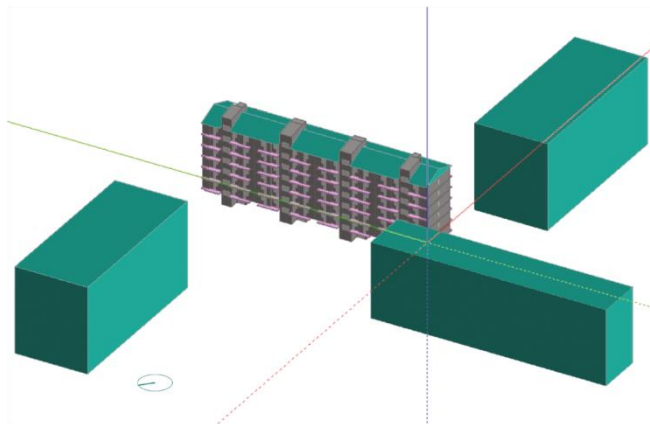


Figure 7.9 - The numerical model of the case study

After the modelling of the case study, a double calibration is adopted for ensuring better quality analyses. First, dynamic simulations are performed for estimating the building natural gas consumption and comparisons are made with data collected from a building similar for typology, localization, and construction period. Then, considerations are done by comparing the obtained outputs with the typical heating consumption of buildings with the same characteristics and age collected from the

TABULA webtool [47]. For brevity, the calibration is not described here, and detailed information can be found in Lops [48].

7.3 Results

The energy performance of the reference building is evaluated in terms of energy consumption by taking into account the annual heating and cooling use of the case study referred to the year 2017 (Chapter 4). The simulations are carried on by using the EnergyPlus solver (Version 8.9).

The comparisons are made among the outputs, subdivided into building and cooling needs, estimated by adopting two different climatic files. First, the energy requirement is calculated considering the selected MM5 models before the coupling procedure. Then, analyses are performed by assuming as input climate conditions the data referred to wind velocity and direction, opportunely calculated on the base of the coupling procedure.

Figure 7.10 and Figure 7.11 depict, respectively, the building heating and cooling requirement.

On the heating side, the results estimated after the coupling procedure underline a general reduction expected for all MM5 models. By considering local wind intensity, in fact, lower heating requirements are needed for ensuring acceptable comfort conditions. The percentage of this reduction is 8 % for all models.

The opposite trend obtained on the cooling side. The presence of neighboring buildings decreases the wind effect on the free cooling of the case study. After the coupling procedure, in fact, higher energy consumptions are expected. The percentages of this increment are 37% for the MM5-MRF, 43% for the MM5-PleimXiu and 56% for the MM5-BK.

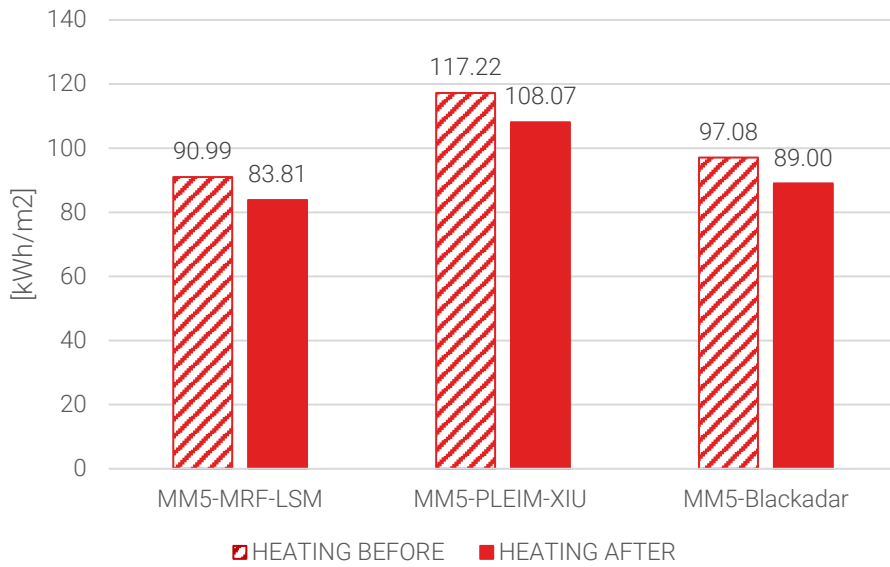


Figure 7.10 - Building heating consumption

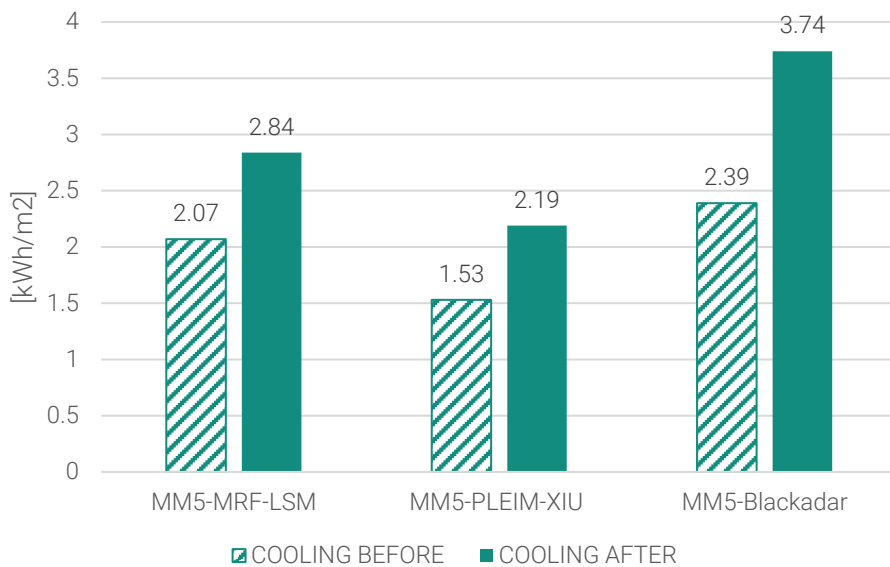


Figure 7.11 - Building cooling consumption

8

Open-source application development

The framework presented in Chapter 3 is also tested by using commercial software (STAR-CCM+). In order to make the whole procedure open source, the commercial software is replaced by an application developed in the Kratos Multiphysics environment. The application allows obtaining a framework completely open-source and, therefore, accessible to anyone.

The realization of the numerical model is a very time-consuming step within CFD analysis. In addition, finding information about a specific area is not always a simple task. For this reason, the datasets used by the developed application are open access. Furthermore, the application allows realizing the numerical model for CFD analyses in a completely automated way starting from some basic information.

The main structure of the above-described application has been developed in collaboration with M.Sc. Simon Wenczowski from the TUM University. It features a modular pipeline³ able to process geodata to obtain the numerical model for CFD simulations.

³ <https://github.com/KratosMultiphysics/Kratos/issues/3959>

Most of the code is developed in *Python*, although some functions are implemented in *C++* to increase its performance.

One of the application's strengths presented in this chapter is its non-reliance on a specific geographic location. As a result, the estimated outputs can be obtained for places all over the world. Thus, the figures below are reported only for illustrative purposes.

This chapter is organized as follows. First, in Section 8.1, the structure of the pipeline and its component modules is given. Next (Section 8.2) presents the datasets used to retrieve the terrain and building geometry information. Then, Section 8.3 reports the strategies adopted to convert the obtained spatial information into two different numerical models (one with the terrain geometry and the other with the building geometry). The two models are necessary to create the final model. It is characterized by a body-fitted numerical mesh with the smallest elements near the buildings and the largest ones far from them. Finally, Section 8.4 explains the selected boundary conditions and Section 8.5 the results obtained from the performed CFD analyses.

8.1 Geodata processing pipeline

The open-source tool⁴ is developed and integrated into the Kratos Multiphysics software package (hereafter Kratos) [49]. Kratos is designed as an Open-Source framework for the implementation of numerical methods to solve engineering problems. It is written in C++ and designed to allow collaborative development by researchers focusing on modularity and performance.

Six separate modules are realized to make the tool as robust as possible, each capable of performing specific operations. In detail, the modules are:

- *geo_data*: it allows obtaining the terrain and building geometry data providing only the coordinates of the center and the radius of the domain to be analyzed.
- *geo_preprocessor*: it performs preliminary operations on the data used for realizing the numerical model.

⁴ <https://github.com/NicolaGermano>

- *geo_importer*: it imports information from different formats such as STL, OBJ, XYZ etc. However, the list can be extended to make the process more versatile.
- *geo_mesher*: it allows computing the volumetric mesh. For this modulus, two different approaches can be followed. The first method is based on calculating the distance to the terrain surface for performing a metric-based re-mesh procedure. This step can be repeated iteratively to increase the mesh quality. The open-source software MMG [50] via API in Kratos is used for this purpose.

On the other hand, the second approach sets the minimum and maximum dimensions of the contour surface elements.

- *geo_building*: it performs operations on the building geometry, such as, for example, positioning within the domain and removing buildings outside a given boundary or below a selected elevation.
- *geo_model*: it allows both to split the lateral surface of the domain to study different incoming wind directions and create boundary conditions following the conventions in Kratos.

8.2 Datasets

The information for the realization of the numerical model is obtained through two different datasets: one for the terrain and one for the buildings. Both databases are managed by the *geo_data* module.

8.2.1 Terrain dataset

GDEM (Global Digital Elevation Model) images obtained through the *ASTER* (Advanced Spaceborne Thermal Emission and Reflection Radiometer) file are used. The *ASTER* instrumentation is onboard *NASA's* Terra spacecraft.

A digital elevation model (*DEM*) is a digital raster representation of ground surface topography or terrain. Each raster cell (or pixel) has a value corresponding to its altitude above sea level.

ASTER-GDEM (Version 3 [51]) files are divided into 1-degree by 1-degree data area tiles. The names of the individual data tiles refer to the latitude and longitude at the

geometric center of the lower left (southwest) corner pixel. GDEM files can be retrieved through the Earthdata website⁵ or API⁶.

Figure 8.1 shows the tile named *ASTGTMV003_N42E014*, related to the area under evaluation (Pescara).



Figure 8.1 - ASTER-GDEM tile of interest area

8.2.2 Building dataset

The geometric model concerning the buildings is obtained through OpenStreetMap (OSM) website⁷ or the *Overpass API*⁸. The OSM provides maps of the world with free content. Geographic data in the OSM is under a free license (the Open Database License). This license makes possible the following operations:

- Copy, distribute, use the database,
- Create works from the database,
- Modify, transform, and develop the database.

Each element within the OSM can be represented through four different types:

⁵ <https://search.earthdata.nasa.gov/search/>

⁶ <https://earthdata.nasa.gov/collaborate/open-data-services-and-software/api/earthdata-search-api>

⁷ <https://www.openstreetmap.org>

⁸ https://wiki.openstreetmap.org/wiki/Overpass_API

Chapter 8

- *node*: it indicates both point objects such as trees or traffic lights and, in a simplified way, commercial activities, places of interest etc.,
- *way*: it is a set of unclosed points and can describe a road, river, path, etc.,
- *polygon*: it is a closed line and can define a building, lake, park, etc.,
- *relation*: it is a set of the previous elements. It can be defined as a container used to describe complex objects. For buildings with complex geometry, the relationships are of the Multipolygon type. An example is shown in Figure 8.2 where the insides polygon represents the interior patio of the building.

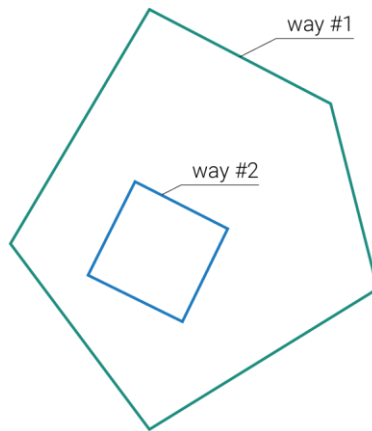


Figure 8.2 - Multipolygon in OSM. way_1: outer perimeter,
way_2: inner perimeter

Considering the open-source nature of OSM, some areas may be not sufficiently mapped, and, therefore, some buildings are not present. In this case, the buildings can be inserted manually into the OSM through the dedicated section.

Figure 8.3 shows the area analyzed in this work and the zones in which some buildings are included are highlighted in red.

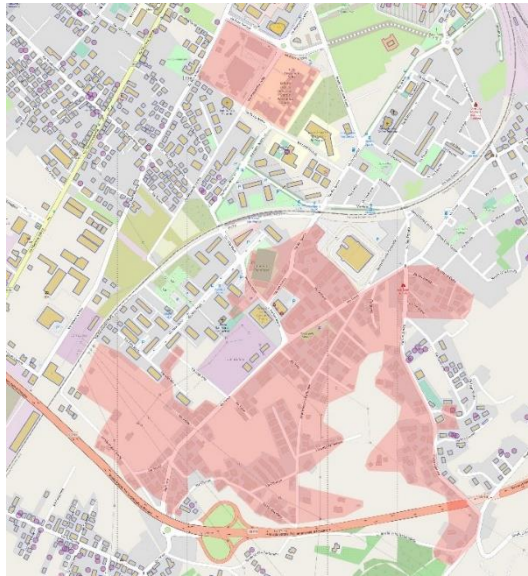


Figure 8.3 - Area analyzed in this work: pre-existing buildings (in yellow) and the areas where buildings are added (in red)

8.3 Numerical model

The numerical model, generated for performing CFD analyses, is created using information referred to both terrain and building datasets. The operations necessary for the realization of the final numerical model are described in the following subsections.

8.3.1 Terrain model

Data obtained through the Earthdata portal can be converted into 3D files using the *DEMto3D*⁹ plugin within the *QGIS* (Quantum Geographic Information System) software. This tool is designed for the conversion of *DEM* files into 3D printing files.

Tests conducted on small models (with a resolution of a few hundred meters) produce satisfactory results. The tool, however, is strongly limited because it does not allow the conversion of large areas of a few square kilometers useful for realizing numerical models of urban quarters.

⁹ <https://demto3d.com/>

Chapter 8

For this reason, an automated procedure is created (within the `geo_data` module) for downloading and converting the *DEM* file into the *OBJ* format.

After setting the bounding box of the area of interest, the *ASTER-GDEM* tiles in which the domain falls are downloaded (Figure 8.4 left) via Earthdata API and the tiles are cut according to the coordinates of the bounding box (Figure 8.4 right). Subsequently, the *DEM* file is converted into an *OBJ* file, thus obtaining a 3D terrain model (Figure 8.5).

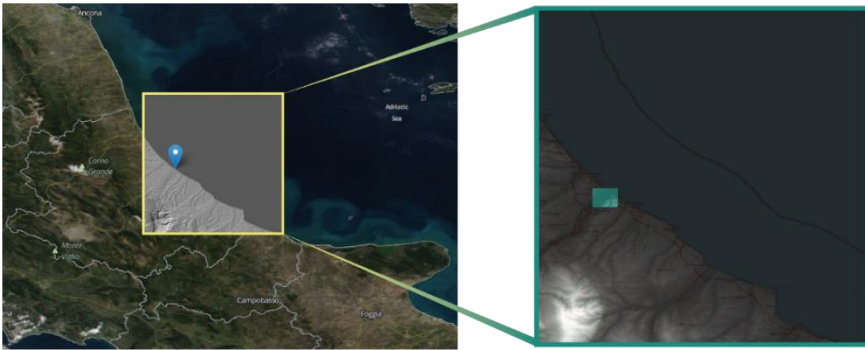


Figure 8.4 - ASTER-GDEM tile in which the domain falls (left), and the bounding box of the domain (right)

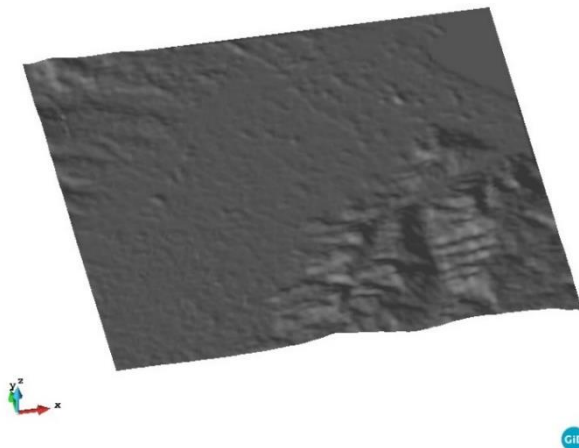


Figure 8.5 - 3D model of the terrain

A specific condition can appear when the area of interest is situated on several tiles. The worst situation is when it is at the intersection of 4 different tiles (Figure 8.6).

In this case, therefore, it is necessary to merge the tiles and, afterwards, crop the domain of interest.

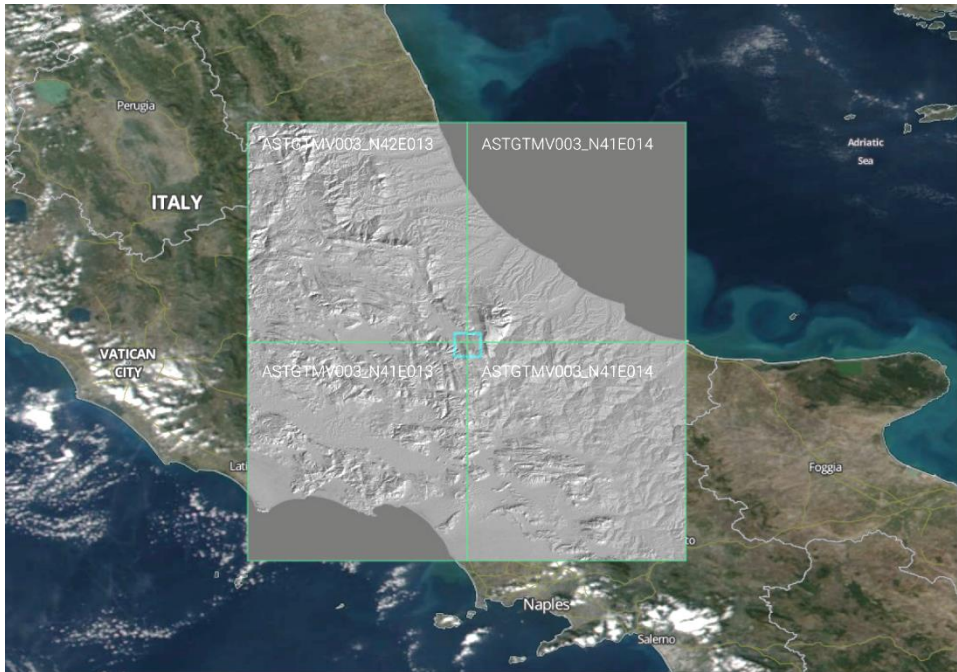


Figure 8.6 - Domain positioned at the intersection of 4 tiles

The obtained OBJ file is imported using the `geo_importer` module.

With the `geo_preprocessor`, however, a circular portion of the terrain surface is cropped and divided into three different areas:

- A central circle with the real orography, where the buildings are placed,
- An intermediate annular portion with only the real orography (without the buildings),
- An outer annular portion, subject to a smoothing procedure, is modeled with a simplified orography with extreme nodes located at zero elevation.

This configuration avoids a discontinuity in the incoming wind speed profile.

The smoothing procedure consists of changing the z-coordinate of the nodes according to the following equation:

$$z_i = (z_i - z_{min}) \beta + z_{min} \quad (\text{Eq. 8.1})$$

where z_i is the coordinate of the i -th node, z_{min} is the minimum elevation of the domain, and β is a reductive coefficient that can take values between 0.0 and 1.0. It is calculated as:

$$\beta = 1 - \frac{d_i - r_{ground}}{r_{boundary} - r_{ground}} \quad (\text{Eq. 8.2})$$

where d_i is the distance of the i -th node from the center of the domain, r_{ground} is the inner radius of the annular portion with simplified orography, and $r_{boundary}$ is the radius of the domain.

The result of the operations just described is a cloud of points that is converted into a 2D mesh using the *Triangle* library (available in Python).

In Figure 8.7 it is shown the scheme of the points mentioned above.

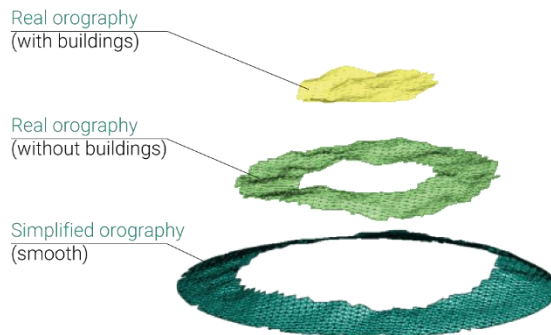


Figure 8.7 - Portions of the terrain surface

8.3.2 Buildings model

The *geo_data* module allows, in addition, to download (via the *Overpass* API) and convert the data provided by the *OSM* into three-dimensional models.

Data via the *Overpass* API are downloaded by providing the bounding box coordinates of the area of interest and the type of data to be extracted. In the

specific case, only the information about the buildings is downloaded and stored in a *GeoJSON* file.

GeoJSON is an open format that is used to store spatial geometries whose attributes are described through the JavaScript Object Notation. Possible geometries are points, lines, polygons, and multiple collections of these types. The typical structure of a *GeoJSON* file is like the following:

```
1   {
2     "type": "way",
3     "id": 333362603,
4     "bounds": {
5       "minlat": 42.4424610,
6       "minlon": 14.2067793,
7       "maxlat": 42.4429295,
8       "maxlon": 14.2073653
9     },
10    "nodes": [
11      3404740665,
12      3404740666,
13      3404740667,
14      3404740668,
15      3404740665
16    ],
17    "geometry": [
18      {"lat": 42.4429295, "lon": 14.2072623},
19      {"lat": 42.4428541, "lon": 14.2073653},
20      {"lat": 42.4424610, "lon": 14.2068823},
21      {"lat": 42.4425326, "lon": 14.2067793},
22      {"lat": 42.4429295, "lon": 14.2072623}
23    ],
24    "tags": {
25      "building": "apartments",
26      "building:levels": "6"
27    }
28  }
```


Chapter 8

Information regarding the height of buildings can be provided either as number of floors or as total height. In the first case, the full height is calculated as the product of the number of floors and the storey height, which, by default, is set at 3.0 m.

Within the `geo_data` module, the `GeoJSON` file is converted into an `OBJ` format. To do this, first, each building is processed iteratively, and the ground footprint geometry is created; then, it is extruded upwards, obtaining the 3D model. However, this procedure has some limitations:

- tapered buildings are not correctly represented.
- In the case of Multipolygon, only the external perimeter is considered (buildings with internal courtyards are not well modelled).
- Adjacent buildings are combined into a single building, and the joint surface is removed.
- Overlapping buildings are combined into a single building.
- Since the community uploads the information, it may happen that some buildings do not have height information. In this case, a default height is set to create the 3D model of the building.

The result of the procedure just described is shown in Figure 8.8 and Figure 8.9.



Figure 8.8 - 3D model of the buildings obtained through the developed procedure
(Top view - city of Pescara)

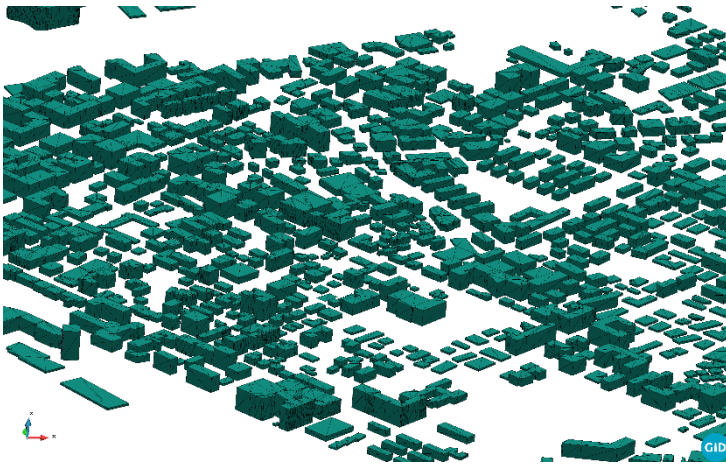


Figure 8.9 - 3D model of the buildings obtained through the developed procedure
(Axonometric view - city of Pescara)

As mentioned in the previous section, the buildings cover only the central portion of the final numerical model. For this reason, a function (in *geo_building* module) is implemented to remove structures outside the given perimeter.

Before merging the geometry of the buildings with one of the terrain, an additional step is necessary to place the buildings at the correct elevation. The OSM, in fact, does not provide the elevation information and, therefore, the buildings are all located on a zero level.

Thanks to the potential provided by Kratos, a function that can calculate the distance (on the z-axis) that each building has with the geometry of the ground is developed within the *geo_building* module. This function allows to shift the building at the exact altitude. The outline of what has just been described is shown below:

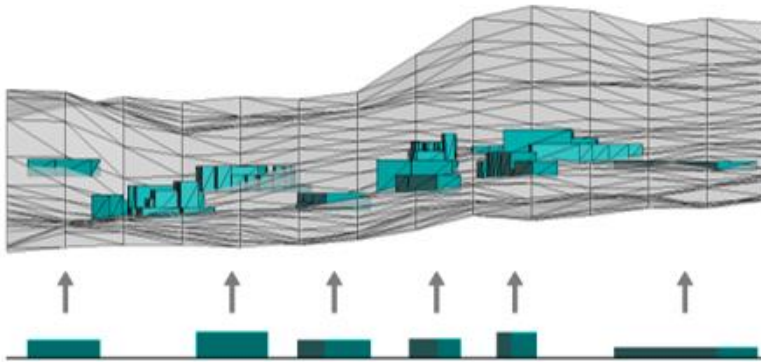


Figure 8.10 - Shifting of buildings on the terrain surface

8.3.3 Final model

The operations described in the previous subsection allow the generation of the 3D model of a portion of a city located anywhere in the world (fully automatically). However, the geometry of the buildings and the terrain are still “separated”, as reported in Figure 8.11 by using different colours.

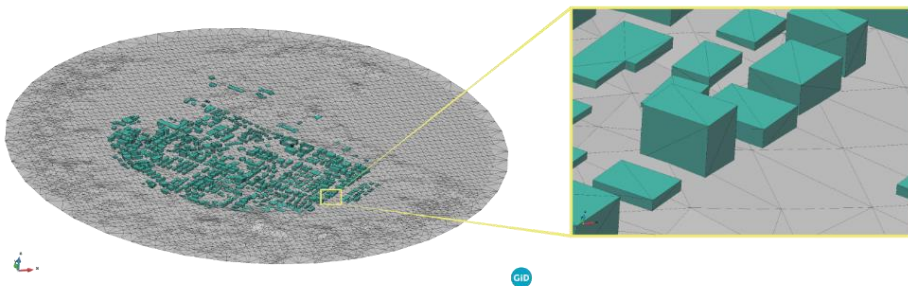


Figure 8.11 - Axonometric view of the terrain surface and building geometries

Two different procedures are implemented in the `geo_mesher` module to obtain the final numerical model.

The first method involves the creation of an initial coarse volumetric mesh in which the geometries of the buildings are inserted. Then, the distance field is calculated

with the level-zero on the building's surface through an iterative process. Finally, the refinement process is executed with the *MMG*¹⁰ library (via API in Kratos).

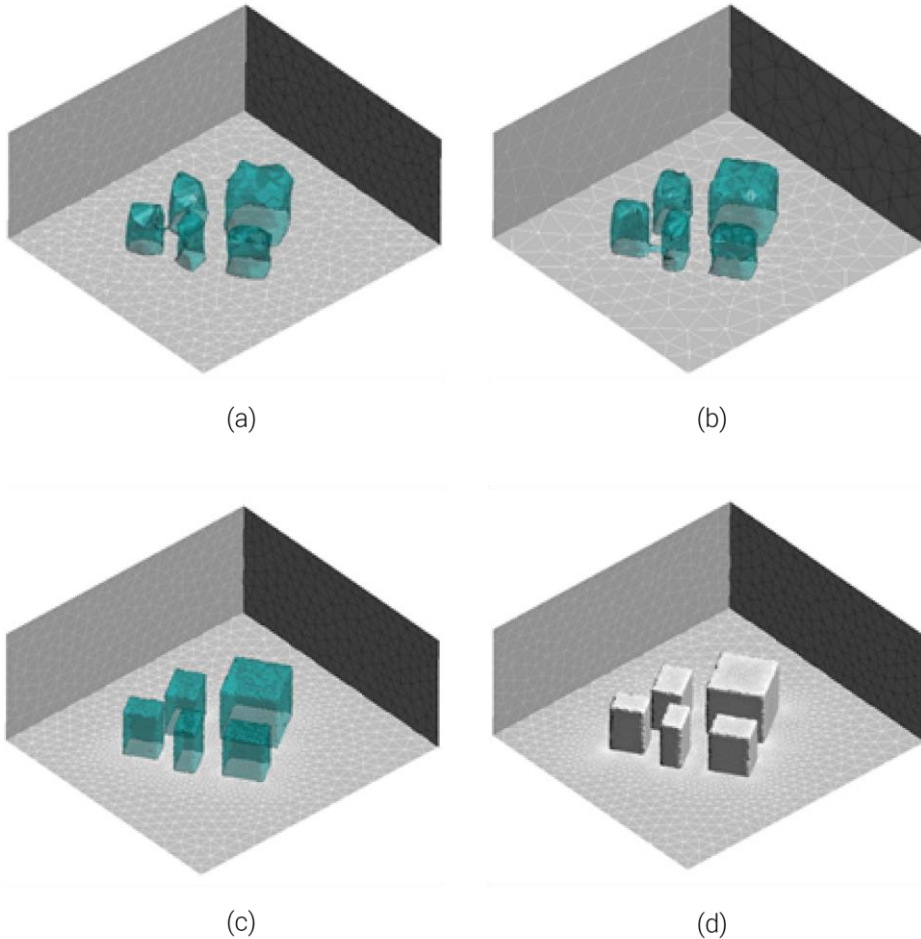


Figure 8.12 - Refinement mesh procedure: (a) distance field step 1, (b) distance field step 2, (c) distance field step 3, and (d) result

Figure 8.12 shows a small test carried out on a group of buildings to evaluate the effectiveness of the meshing process. The iterative refinement is performed to improve the quality of the mesh as shown from Figure 8.12a to Figure 8.12c. After some iterations, elements with negative distance are removed. The result is a body-

¹⁰ <https://github.com/MmgTools/mmg>

Chapter 8

fitted mesh (Figure 8.12d) where smaller elements are placed near the buildings while coarser elements are located away from them.

Good results are also obtained in a complex test case. Figure 8.13 depicts the result achieved on many buildings with complex shapes. The section on a building (Figure 8.14) underlines the different sizes of the elements (smaller near the surface of the building and larger away from the building).

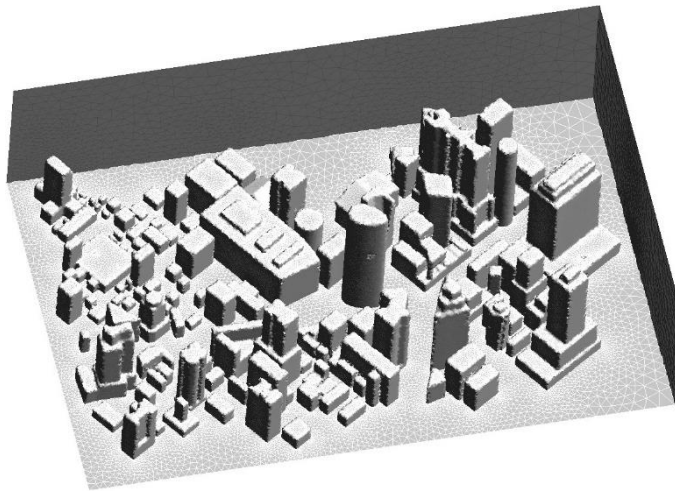


Figure 8.13 - Large scale test with refinement process: axonometric view

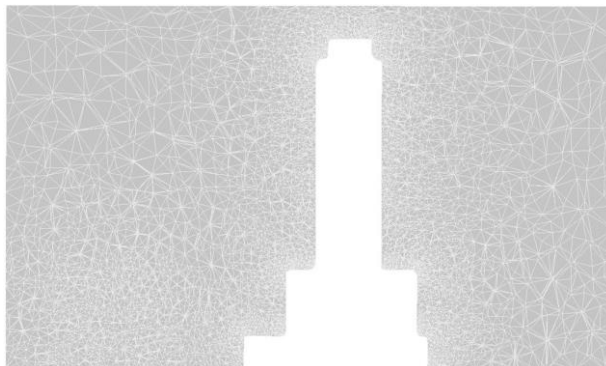


Figure 8.14 - Large scale test with refinement process: section on a building

The procedure just described allows to obtain a good quality mesh but with very high computational efforts. This problem can be solved by adopting the parallel

version of the used re-mesh tool (*ParMMG*¹¹). At the present stage, however, the parallel version is under development. For this reason, a different procedure is created and followed to generate a numerical model by combining terrain and building geometries.

The first step performed with this new procedure involves drilling the terrain surface with the building footprints (Figure 8.15). This operation allows the creation of new nodes on the terrain surface to insert the geometry of the buildings (Figure 8.16). Figure 8.17 shows the terrain surface before and after the "drilling" step focusing on the inserted nodes.

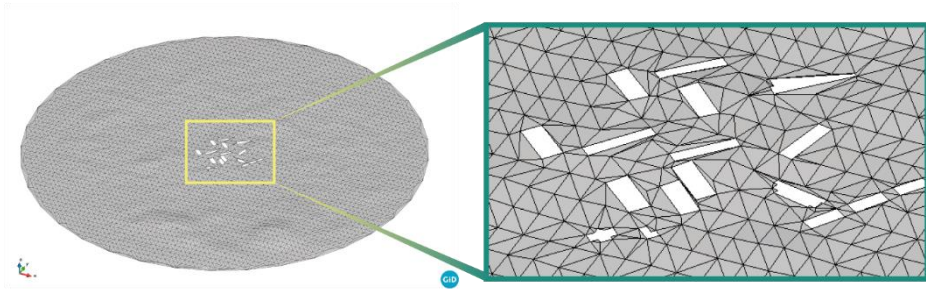


Figure 8.15 - Terrain surface with the building footprints

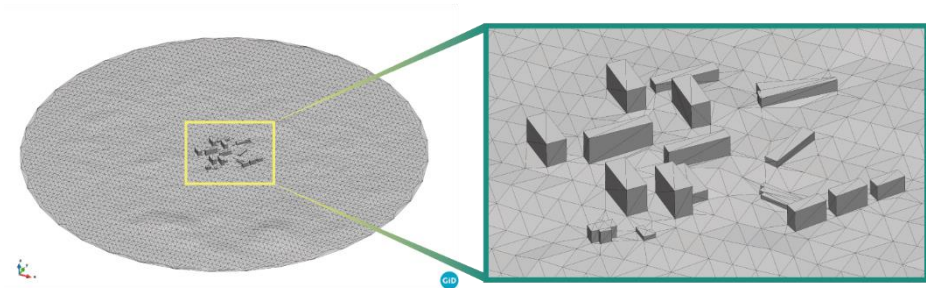


Figure 8.16 - Terrain surface with the building geometries

¹¹ <https://github.com/MmgTools/ParMmg>

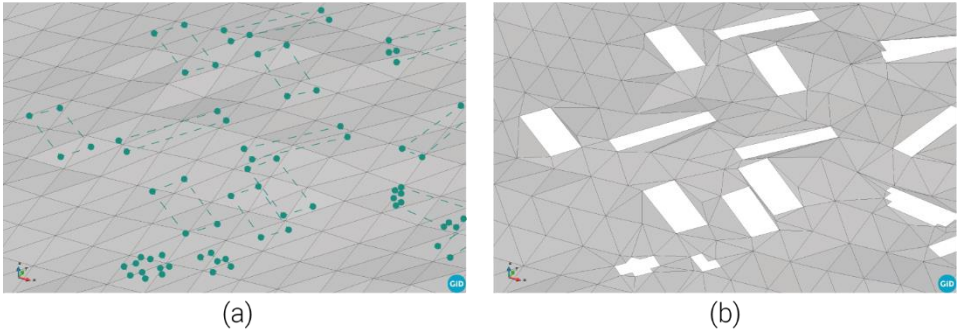


Figure 8.17 - (a) terrain surface before the "drilling" procedure with the new building nodes highlighted, (b) terrain surface after the "drilling" procedure

The next step is the creation of the lateral and upper surface of the domain. Then, the minimum and maximum dimensions that the elements can reach are set for each surface. This step is schematized in Figure 8.18.

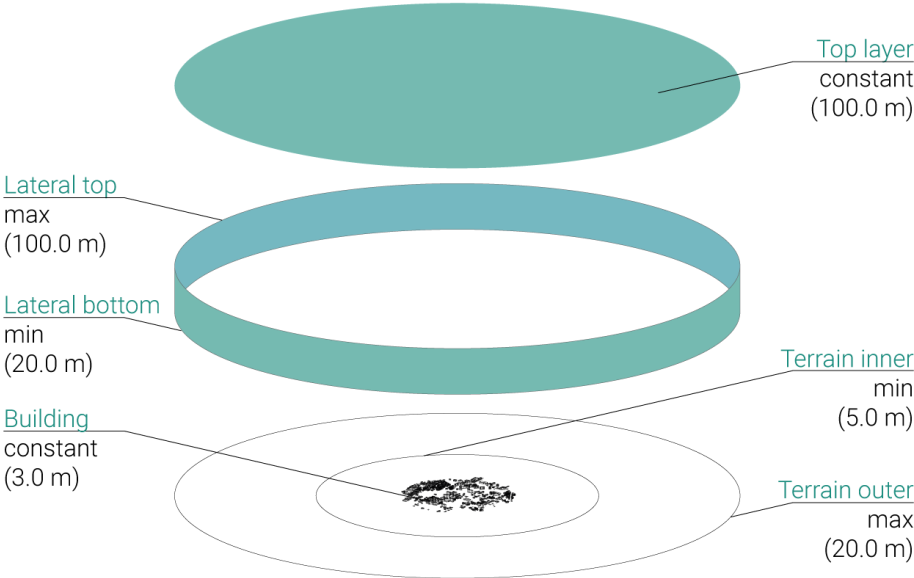


Figure 8.18 - Size of elements for each mesh. In parentheses the default values

Subsequently, the tetrahedral mesh is created through *TetGen* (available with the *MeshPy*¹² library in Python). Figure 8.19 reports the final mesh of a test conducted on a few buildings. The domain has a diameter of 2.0 km and a height of 300.0 m.

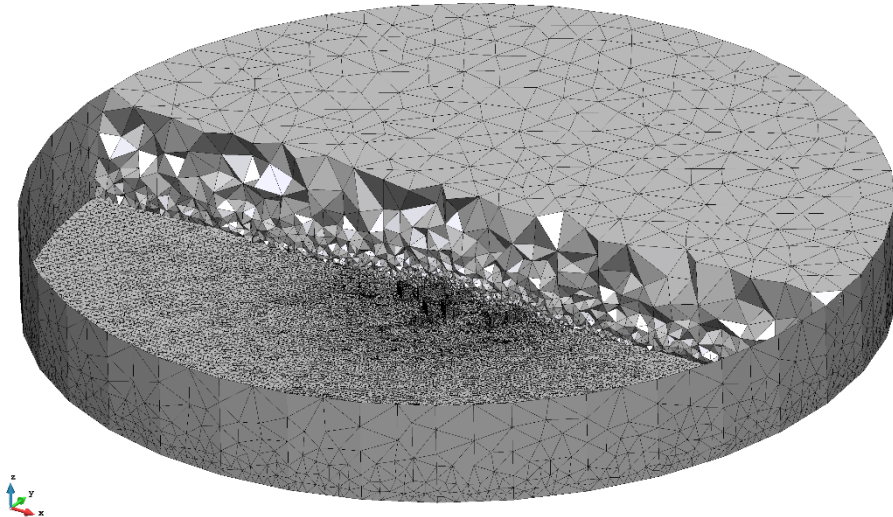


Figure 8.19 - Tetrahedral mesh on a test case: axonometric view

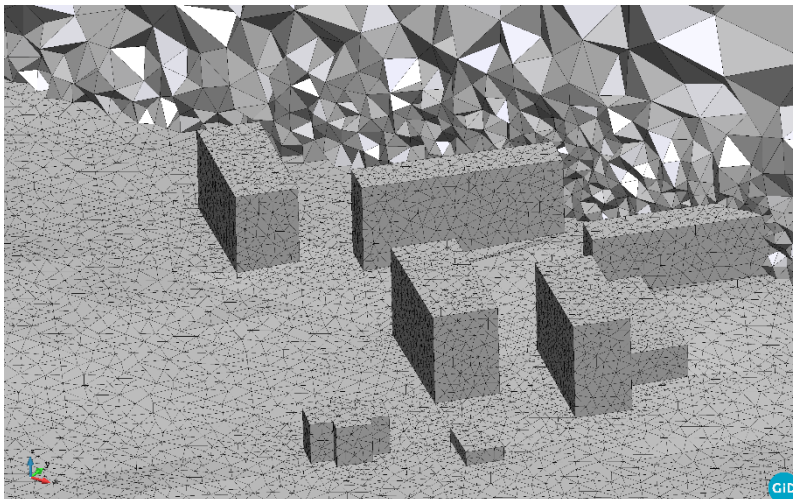


Figure 8.20 - Tetrahedral mesh on a test case: detail view of buildings

¹² <https://documen.tician.de/meshpy/>

Chapter 8

As noted in Figure 8.19 and Figure 8.20, the element sizes change in different domain regions. It is due to the dimensions previously set.

In the same way, the area analyzed with the commercial software (Chapter 5) is also generated with the developed application. The substantial difference lies in the fact that the model is created in a completely automatic way.

To obtain the result reported below, it is necessary to provide only the coordinates of the center of the domain (42.442679 latitudes and 14.207077 longitudes), its diameter and height, and the diameter of the portion of the buildings.

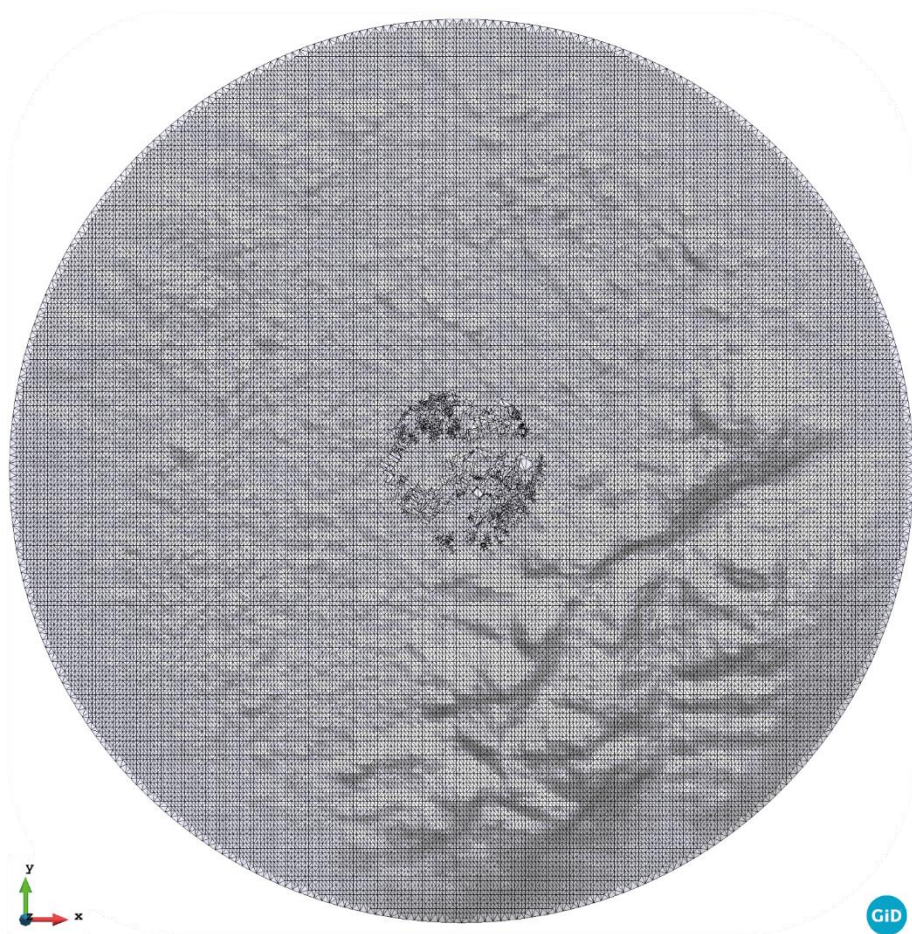


Figure 8.21 - Numerical model generated with the developed application: terrain and building surfaces (top view)

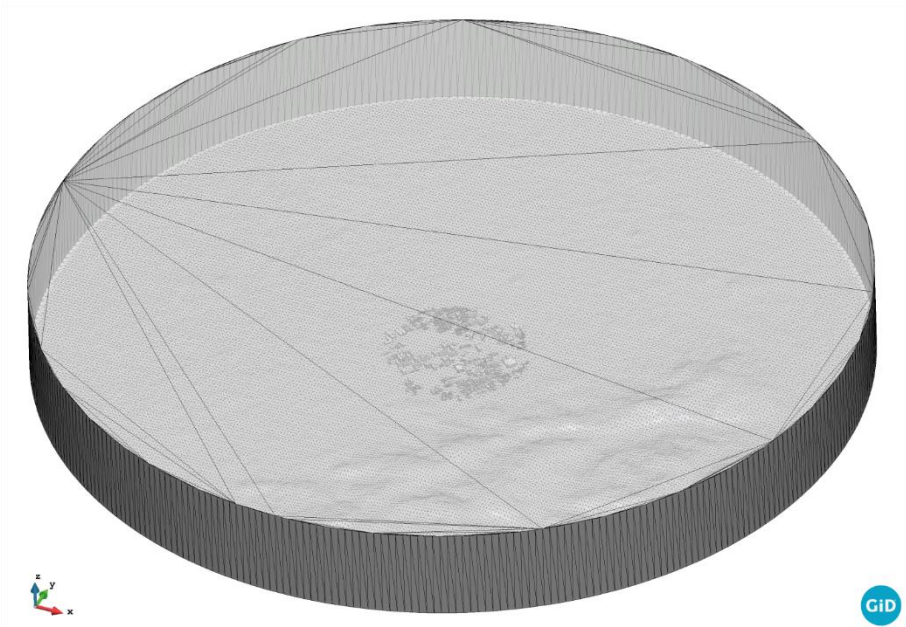


Figure 8.22 - Numerical model generated with the developed application: boundary surfaces with the top surface in transparency (axonometric view)

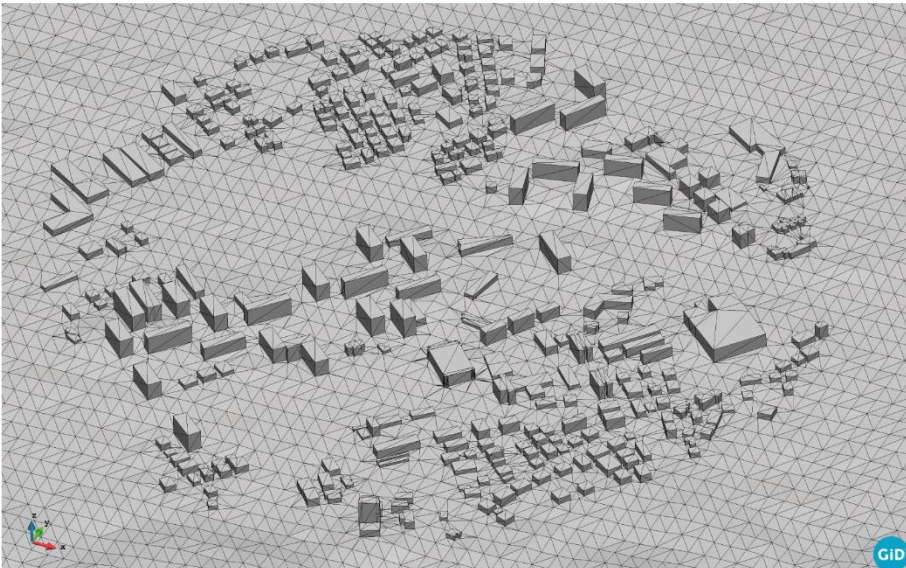


Figure 8.23 - Numerical model generated with the developed application: detail on building surfaces (axonometric view)

Chapter 8

To create the volumetric mesh, the minimum and maximum dimensions of the elements for each region are set. This information is indicated in the table reported below:

Region	Inner Radius [m]	Outer Radius [m]	Height [m]	Minimum Surface Size [m]	Maximum Surface Size [m]
Buildings	-	500.0	-	3.0	3.0
Terrain inner	-	1,000.0	-	5.0	5.0
Terrain middle	1,000.0	2,000.0	-	5.0	10.0
Terrain outer	2,000.0	3,000.0	-	10.0	20.0
Top	-	3,000.0	-	100.0	100.0
Lateral	-	-	500.0	20.0	100.0

Table 8.1 - Size of regions and meshes

where the *Terrain inner* is the central circle where the buildings are located, the *Terrain middle* is the intermediate annular portion without buildings, and the *Terrain outer* is the outer annular portion subject to the smoothing procedure.

A tetrahedral mesh is generated on the base of the parameters reported in Table 8.1. The total number of nodes is 1,379,229, the total number of triangles is 635,948, and one of tetrahedra is 7,764,938.

Triangular elements are the outer faces of the tetrahedra positioned on the external surfaces of the domain. On the triangular elements, the boundary conditions are appropriately set.

Figure 8.24, Figure 8.25 and Figure 8.26 depict, respectively, the axonometric view of a portion of the generated model, a zoom on the buildings and a detail on the building mesh. Lastly, a section is shown in Figure 8.27 where different elements' sizes relative to the buildings' surface can be seen.

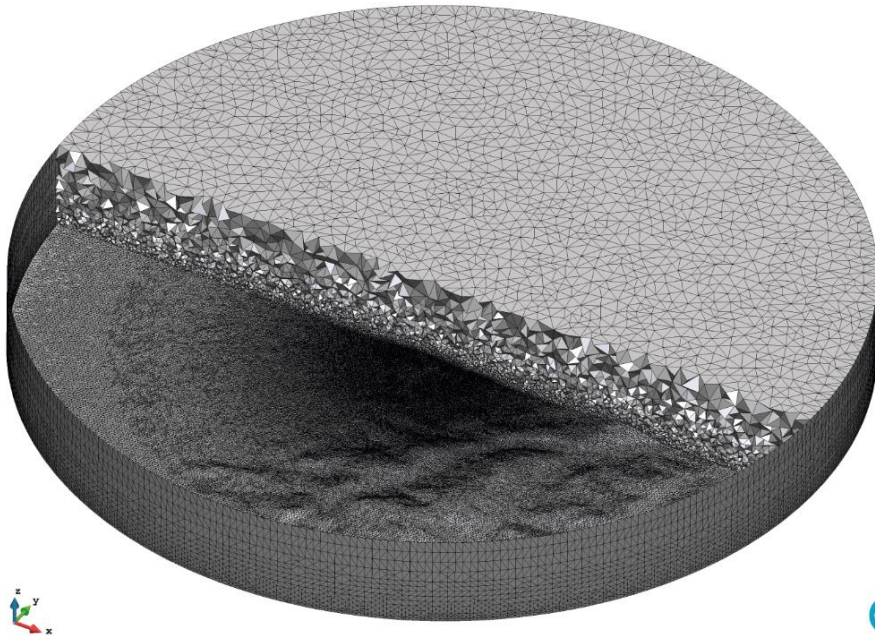


Figure 8.24 - Tetrahedral mesh of the area under study generated with the developed application: axonometric view

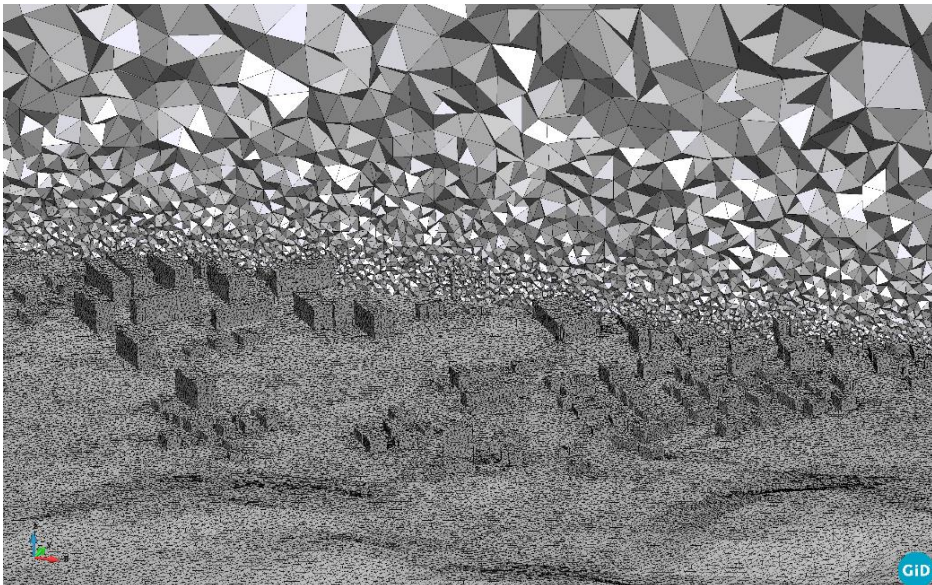


Figure 8.25 - Tetrahedral mesh of the area under study generated with the developed application: axonometric view of the buildings

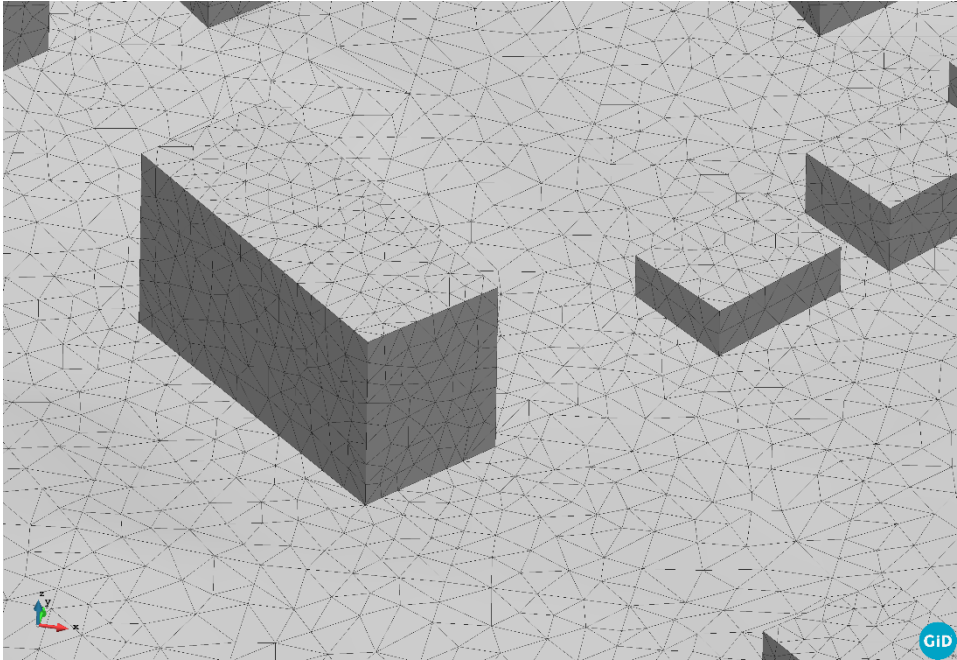


Figure 8.26 - Tetrahedral mesh of the area under study generated with the developed application: detailed mesh on the buildings

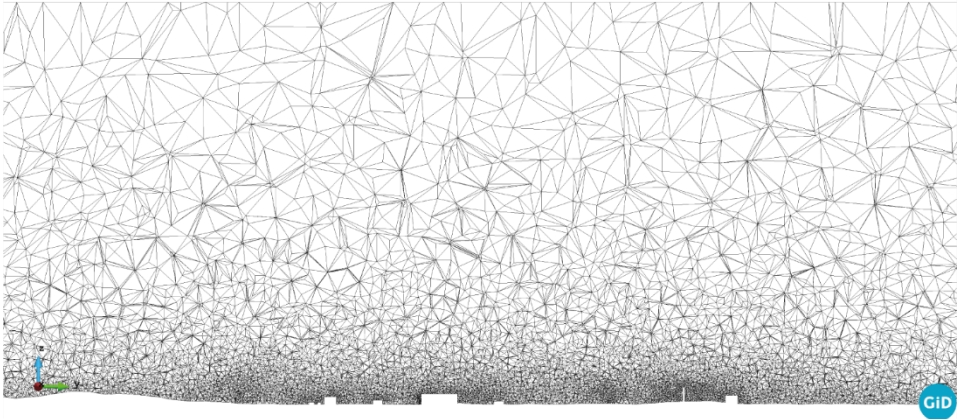


Figure 8.27 - Tetrahedral mesh of the area under study generated with the developed application: section on the buildings

8.4 Boundary conditions

Before assigning BCs, the lateral surface is divided into several sectors. The default value is 12, but different values can be set.

The BCs are assigned according to the settings reported in Table 8.2.

	Buildings	Terrain inner	Terrain middle	Terrain outer	Sectors	Top
BC Type	Wall No-Slip	Wall No-Slip	Wall No-Slip	Wall No-Slip	Velocity Inlet / Pressure outlet	Wall Slip

Table 8.2 - BCs assigned

The definition of the BCs is handled automatically by the `geo_model` module. Once the analysis to be performed is set (wind from North, East, South or West), the lateral surfaces are fixed as *Velocity Inlet* or *Pressure Outlet*, respectively.

The building and terrain surfaces are set as *Wall No-Slip*, while the top surface is defined as *Wall Slip*.

All the information about BCs is enclosed in a *JSON* file. An example is reported below:

```

57 | "boundary_conditions_process_list": [
58 |   {
59 |     "Parameters": {
60 |       "model_part_name":
61 |       "FluidModelPart.Buildings"
62 |     },
63 |     "kratos_module":
64 |     "KratosMultiphysics.FluidDynamicsApplication",
65 |     "process_name": "ApplyNoSlipProcess",
66 |     "python_module": "apply_noslip_process"
67 |   },
68 |   {
69 |     "Parameters": {
70 |       "model_part_name":
71 |       "FluidModelPart.TopModelPart"

```

```

69         },
70         "kratos_module":
"KratosMultiphysics.FluidDynamicsApplication",
71         "process_name": "ApplySlipProcess",
72         "python_module": "apply_slip_process"
73     },
74     {
75         "Parameters": {
76             "model_part_name":
"FluidModelPart.BottomModelPart"
77         },
78         "kratos_module":
"KratosMultiphysics.FluidDynamicsApplication",
79         "process_name": "ApplyNoSlipProcess",
80         "python_module": "apply_noslip_process"
81     },
    . . .
274 ]

```

8.5 Results of CFD analysis

The numerical model, obtained with the above-described application, is tested by performing a CFD simulation. In the present analysis, an incoming wind from the North direction is set, and the obtained results are depicted from Figure 8.28 to Figure 8.31.

The results show a very similar trend to those obtained with the microscale commercial software. Specifically, in Figure 8.28, the velocity magnitude result is shown; in Figure 8.29, the pressure field with streamlines is reported, in Figure 8.30, the velocity vectors on a longitudinal section on the reference building are depicted, and in Figure 8.31, the streamlines on the neighborhood under investigation are plotted.

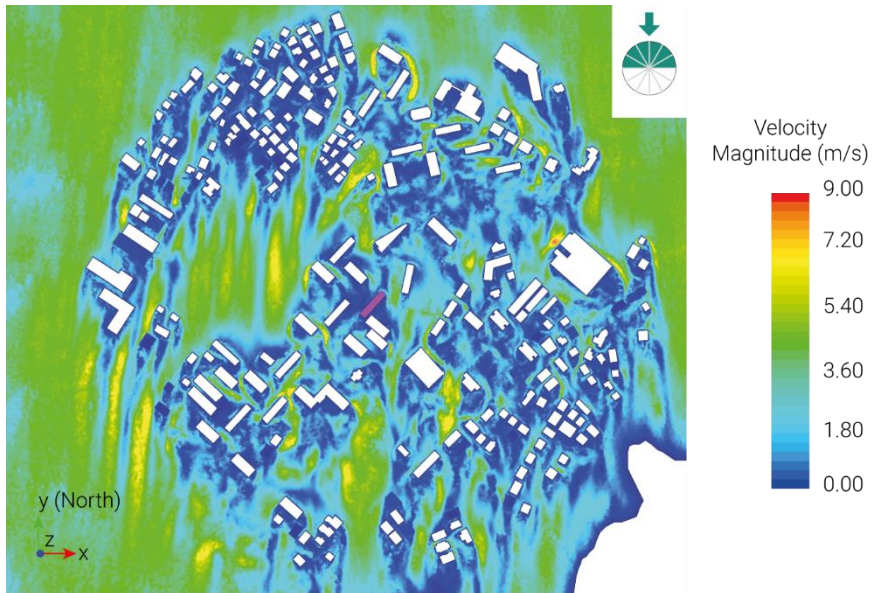


Figure 8.28 - Top view of velocity magnitudes: incoming wind from North (-y)

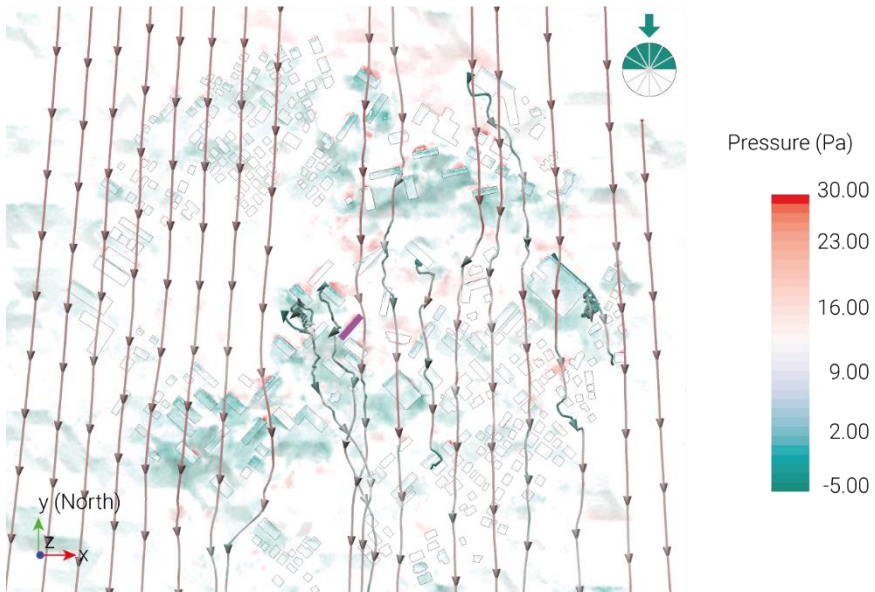


Figure 8.29 - Top view of pressure levels and streamlines: incoming wind from North (-y)

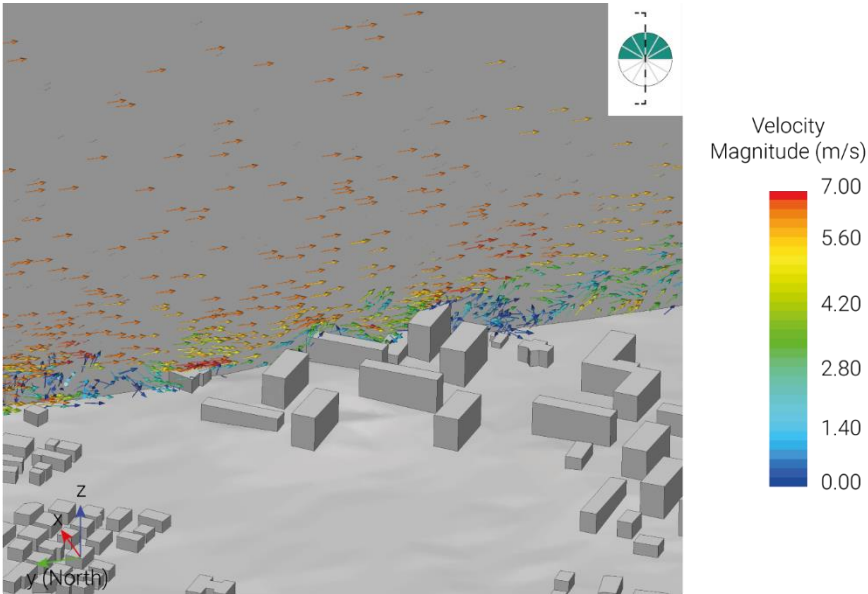


Figure 8.30 - Section view of velocity magnitudes on the reference building: incoming wind from North (-y)



Figure 8.31 - Axonometric view of streamlines in the urban fabric: incoming wind from North (-y)

9

Comparison between open-source and commercial software

In this chapter, numerical models and results obtained with open-source and commercial software are compared. The comparison is necessary since the application in the open-source environment is developed to make the whole framework completely license free. The goal is to conduct CFD analysis and then, subsequently, energy analysis on buildings trying to involve the largest number of people by breaking down as many barriers as possible. The numerical model for the microscale analysis in the open-source environment (KratosMultiphysics) are generated in a fully automated way by merging the information of two different datasets (one for the terrain and one for the buildings). The model for the analysis with the commercial software (STAR-CCM+), on the other hand, is obtained from the data provided by the regional cartography.

In the first part of the chapter the comparisons between the two numerical models are reported and, subsequently, the analysis with the incoming wind from North is reported. Finally, some considerations about the two software are reported.

9.1 Numerical model comparison

The numerical models, obtained from completely different data, produced a roughly similar result. The major difference lies in the fact that the numerical model with open-source software (Figure 9.1a) is generated with a time and operator effort that is not comparable to that obtained with commercial software (Figure 9.1b).

On the commercial side there is the necessity to find the cartographic data of the area of interest, join the different tiles (Chapter 5) and then process the information with a 3D modeling software to obtain the result shown below (Figure 9.1b).

Completely different discussion for the open-source side. In this case, in fact, it is necessary to provide only the coordinates of the center of the domain and information about the size of the area of interest and the mesh.

The time required to obtain the result obviously depends on the dimension of the domain and the size of the desired mesh. In the case under consideration, the diameter is 6.0 km, the height about 500 m and the mesh size is as shown in Table 8.1. The result shown in the figure is obtained after a time of about 10 minutes which is not comparable to that used for the commercial software.

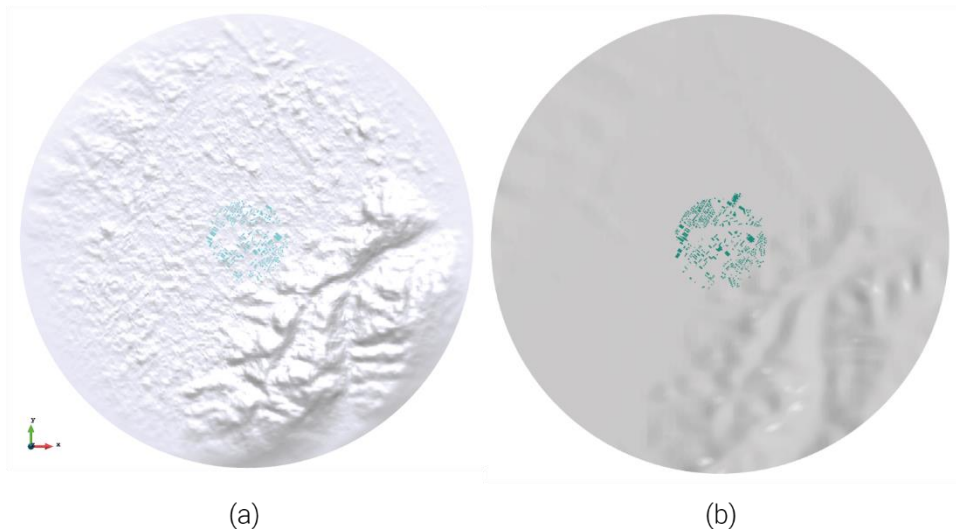


Figure 9.1 - Numerical model of terrain and buildings obtained with open-source software (a) and commercial software (b): top view

9.1 Numerical model comparison

In Figure 9.2 are reported the axonometric views of the two obtained domains and in Table 9.1 are reported the data relative to the number of elements in the two models.

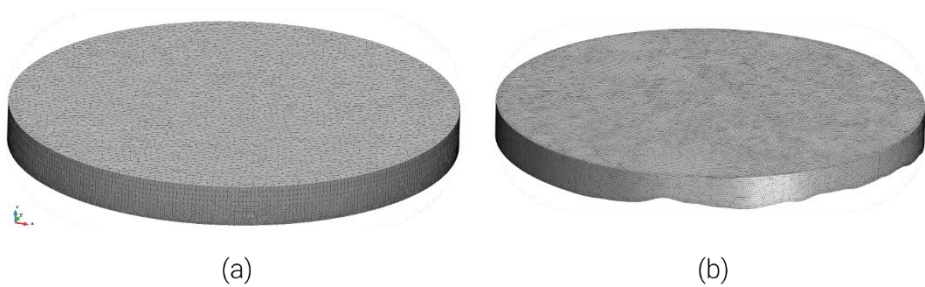


Figure 9.2 - Numerical model obtained with the open-source software (a) and with the commercial software (b): axonometric view

	Open-source software	Commercial software
Nodes	1,379,229	12,212,485
Triangles	635,948	491,214
Polyhedra	-	2,645,737
Tetrahedra	7,764,938	-

Table 9.1 - Number of mesh elements

The following is a comparison of the building geometry obtained with the open-source software (Figure 9.3a) and the one obtained with the commercial software (Figure 9.3b). The data from the regional mapping (Figure 9.3b) is more detailed than that obtained from the OSM database (Figure 9.3a). The major limitation of the OSM data lies in the fact that some cities are not fully mapped. This problem can be solved by inserting the missing geometries from the dedicated section.

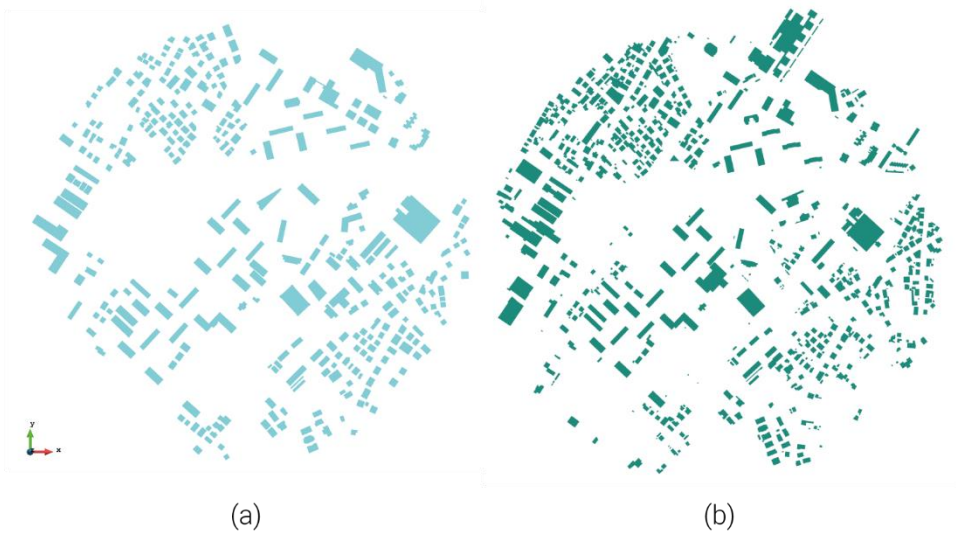


Figure 9.3 - Buildings obtained through the open-source (a) and commercial (b) software: top view

In the following image, instead, the buildings located in the center of the domain are shown in detail. The meshes obtained with the two software are very similar to each other.

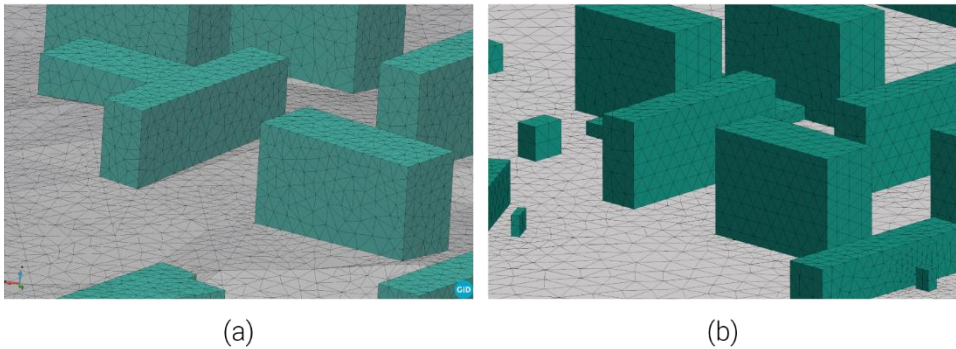


Figure 9.4 - Detail on buildings located in the center of the domain: open-source (a) and commercial (b) software

9.2 Results comparison

The results of the CFD analyses obtained with the two software are illustrate below. Figure 9.5 and Figure 9.6 show the top views of the velocity field over the analyzed neighborhood obtained with the open-source and commercial software, respectively. As noted in the figures, the results are very similar to each other.

As further evidence of this, streamlines obtained with open-source and commercial software respectively are presented in Figure 9.7 and Figure 9.8. The similarities between the two results are even more pronounced here.

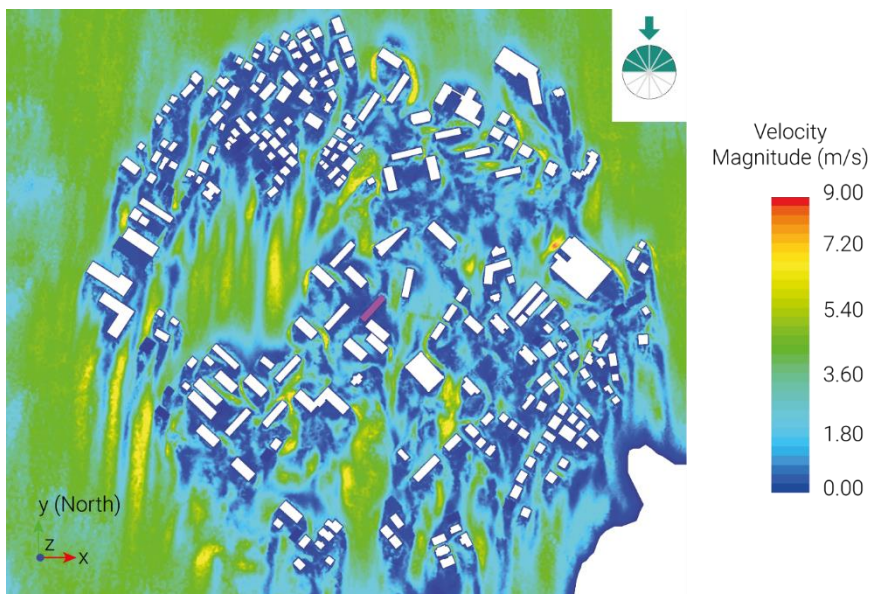


Figure 9.5 - Velocity magnitude for microscale analysis with open-source software: incoming wind from North (-y)

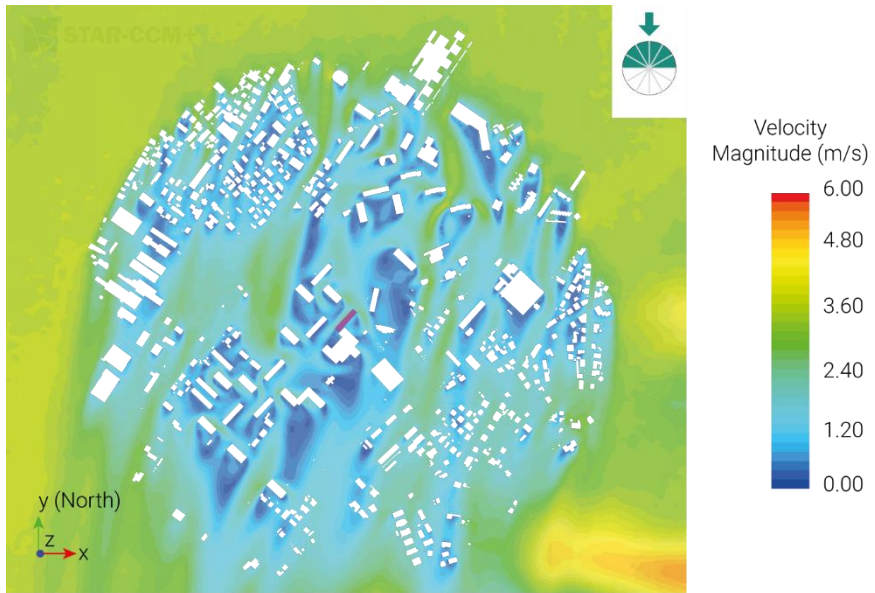


Figure 9.6 - Velocity magnitude for microscale analysis with commercial software:
incoming wind from North (-y)

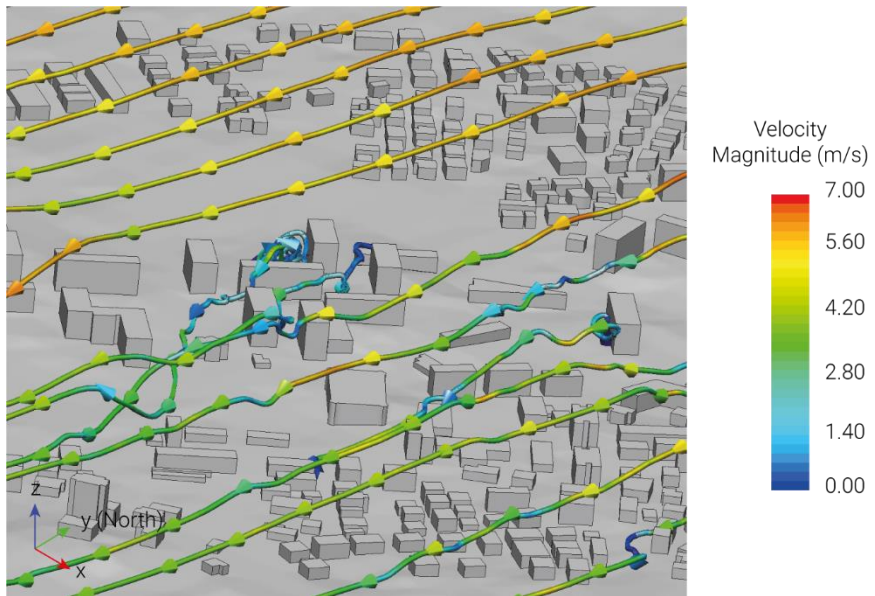


Figure 9.7 - Streamlines in the urban area obtained with the open-source software:
incoming wind from North (-y)

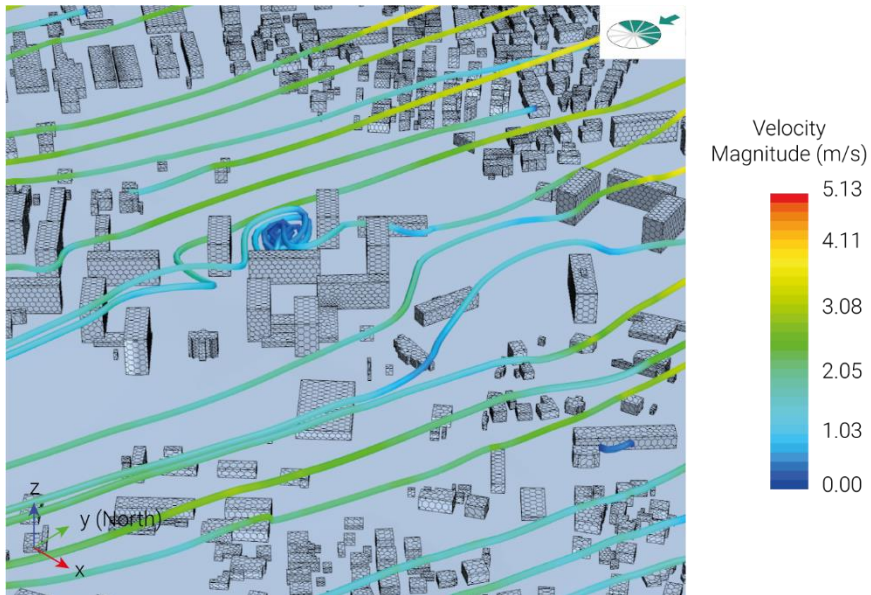


Figure 9.8 - Streamlines in the urban area obtained with the commercial software: incoming wind from North (-y)

9.3 Considerations

In Table 9.2 some considerations regarding the two software programs with which the microscale analyses are conducted.

The retrieval of geospatial data is certainly easier for the application developed with the open-source environment. The regional cartography is often difficult to find, and when it is not available it is necessary to refer to other datasets making the procedure much more tedious. The datasets used for the open-source application, on the other hand, are available almost everywhere in the world. The terrain data are obtained from NASA DEMs and the building data from OSM which has a very active community. According to the statistics¹³, there are:

- 7.8 million registered users
- ~4.5 million map changes/day
- 1.6 million different user contributors.

¹³ https://www.openstreetmap.org/stats/data_stats.html

Chapter 9

Adding missing buildings (if needed) is possible in both environments even if with different procedures.

As widely reported above, the time for generating the two models is not comparable given the automation process developed for the open-source software. This procedure does not require a high level of experience with 3D modeling software; however, an experienced operator is required for the settings regarding CFD analysis.

Finally, the reliability of the results obtained in both cases is very high since these are two software widely used in research.

	Open-source software	Commercial software
geodata retrieval	✓	✗
geodata availability	✗ ✓	✗
add missing data	✓	✓
time consuming (numerical model)	✓	✗
requires expert operator	✗ ✓	✓
reliable results	✓	✓

Table 9.2 - Final considerations

10

Conclusions

The present thesis proposes a methodology suitable to study local wind flows within an urban area. The presented solution provides information about airflows considering both meteorological and environmental conditions in different urban contexts. The information obtained from these analyses is used as input in energy applications improving their accuracy. To test the proposed framework, mesoscale, microscale, and energy simulations were conducted to evaluate the energy consumption of a building located in Pescara (Central Italy).

10.1 Major results

Based on a multiscale and totally open approach, the proposed framework was largely tested and validated by comparing the estimated results with values obtained from the literature review.

The complete versatility of the procedure allows the generation of models by using licence-free software, replicable all over the world and with a minor user effort. Moreover, by adopting a conceived framework, various simulations could be carried on in a more agile way compared to the current state.

Chapter 10

On the basis of these considerations, the approach would make possible to fill the gap in numerical modelling of urban areas, experimented between the developed regions of the world with the other countries.

In addition, the approach, being based on the coupling method, permits the interaction between the mesoscale and microscale results, ensuring higher accuracy in the estimations of local urban phenomena. Finally, working on different scales, the method takes into account various aspects related to them.

The results obtained after the coupling procedure show variability in terms of wind velocity and direction. The plotted wind roses, in fact, underline the effect that the surrounding buildings have on local fluxes inside urban areas. In the test case analyzed, the presence of the nearby buildings significantly reduces the velocity magnitude (31 - 48%) and increases the orientation towards the west direction ($\sim 30^\circ$).

Moreover, the adoption of higher quality inputs allows better estimations in the case of dynamic building energy simulations. In fact, more accurate building energy requirements were calculated by considering climatic data obtained from the coupling procedure. In detail, the expected cooling consumption increases around 37-56%, whereas the heating need decreases approximately 8%.

Once the framework has been validated, the whole procedure was made fully open by introducing a developed application that permits the automatic generation of numerical models for microscale analyses. The comparison between the two approaches underlines the capability of the open-source software in predicting wind fluxes of the same entity of those obtained by adopting the commercial procedure.

10.2 Future works

Future developments of the research should be focused on various aspects. Microscale analyses could be carried on by considering multiple wind directions. In fact, by increasing the number of incoming winds, better estimations are expected with consequent benefits on dynamic building energy simulations.

A second aspect that should be considered is related to the accuracy of the generated 3D model by improving the building modelling. For example, the

presence of structures with patio or with different heights is not correctly described by the current version of the open-source application. Moreover, the possibility of introducing custom buildings should be considered for allowing the use of the framework from engineers and architects since the design stage.

Finally, interesting evaluations would be related to the use of the whole framework all over the world for testing its capability and reliability by assuming different case studies at various scales, from the urban quartier to the single building.

A

Appendix

The script developed to automatize the coupling procedure is reported below. The code is written by using the programming language for scientific computing called *Octave*.

```
1 close all; clear all; clc;
2 myFolder = "import_file";
3 % check if the folder "import_file" exists
4 if ~isfolder(myFolder)
5     errorMessage = sprintf('Error: The following folder does not
6         exist:\n%s\nPlease specify a new folder.', myFolder);
7     uiwait(warndlg(errorMessage));
8 end
9 %%%%%%%%%%%%%%%
10 %%    CFD    %%
11 %%%%%%%%%%%%%%%
12 % CFD outputs
13 file_CFD = fullfile(myFolder + "\file_CFD", "*.csv");
```

```

14   CFD_files = dir(file_CFD);
15
16   CFD_tables = cell(length(CFD_files),1);
17
18   % virtual anemometer
19   n_anemometer = 8; % number of virtual anemometer
20   anem_speedup = cell(n_anemometer,1); % we store anemometer
    speed-up for each sector
21   anem_delta_th = cell(n_anemometer,1); % we store anemometer
    delta theta for each sector
22
23   % each CFD files with a loop
24   for i = 1:length(CFD_files)
25       baseFileName = CFD_files(i).name; % name file
26       fullFileName = fullfile(CFD_files(i).folder, baseFileName);
    % name file and path
27       % i-th table into CFD_tables
28       CFD_tables{i} = readtable(fullFileName, "ReadVariableNames",
    true);
29
30       pos_R = CFD_tables{i}(1, 5:7); % position of the Reference
    building
31       pos_R = pos_R{:,,:}; % we convert the table into an array
32
33       % we read the velocity directions (i, j) of the Reference
    building
34       dir_R = CFD_tables{i}(1, 2:3); % velocity direction in 2D
    (i,j)
35       dir_R = dir_R{:,,:}; % we convert the table into an array
36
37       % we save the Velocity Magnitude of the Reference point
38       vm_R = CFD_tables{i}.Velocity_Magnitude_m_s(1);
39
40       % we add a column with the speed-up ratio
41       CFD_tables{i}.SpeedUp = vm_R ./
    CFD_tables{i}.Velocity_Magnitude_m_s;
42
43       % we initialize an array for the geom distance
44       dist = zeros(size(CFD_tables{i}, 1), 1);
45

```

Appendix A

```
46     % we initialize an array for the angles between vectors
47     delta_theta = zeros(size(CFD_tables{i}, 1), 1);
48
49     % we calculate the distance and delta_theta of all points
50     % from the Reference
51     for j = 1:size(CFD_tables{i}, 1)
52         pos_j = CFD_tables{i}(j, 5:7);
53         pos_j = pos_j{:,:};
54         dist(j) = distance(pos_R, pos_j);
55
56         dir_j = CFD_tables{i}(j, 2:3);
57         dir_j = dir_j{:,:};
58         delta_theta(j) = ang2vector(dir_R, dir_j);
59     end
60
61     % we add the column "Distance" into the table_CFD
62     CFD_tables{i}.Distance_m = dist;
63
64     % we add the column "Delta_Theta" into the table_CFD
65     CFD_tables{i}.Delta_Theta = delta_theta;
66
67     % we populate the anem_speedup with the speedup of each
68     % sector
69     for k = 1:n_anemometer
70         anem_speedup{k} = [anem_speedup{k}
71             CFD_tables{i}.SpeedUp(k+1)];
72         anem_delta_th{k} = [anem_delta_th{k}
73             CFD_tables{i}.Delta_Theta(k+1)];
74     end
75 end
76
77 disp("CFD done");
78
79 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
80 %%      MM5      %%
81 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
82
83 % we read the MM5 outputs
84 file_MM5 = fullfile("../00_file_MM5", '*.txt');
```

```

80  MM5_files = dir(file_MM5);
81
82  MM5_tables = cell(length(MM5_files),1);
83
84  % we read each MM5 files with a loop. It depends on the number
      of anemometers
85  for i = 1:length(MM5_files)
86      baseFileName = MM5_files(i).name; % name file
87      fullFileName = fullfile(MM5_files(i).folder, baseFileName);
          % name file and path
88
89      % we assign the i-th table into MM5_tables
90      MM5_tables{i} = readtable(fullFileName, "ReadVariableNames",
          true);
91
92      % we compute the sector for each time-step (we read the
          geostrophic wind)
93      % table #1 = anemometer 1
94      % table #2 = anemometer 2; etc etc
95
96      % geostrophic wind direction
97      dir_geostr = MM5_tables{i}.dir_0204m_deg;
98
99      % we initialize an array for the sectors
100     sect_array = zeros(size(dir_geostr));
101
102     % we initialize an array for the speed-up
103     speedup_array = zeros(size(sect_array));
104
105     % we initialize an array for the delta_theta
106     delta_th_array = zeros(size(sect_array));
107
108     % we populate "sect_array" with the corresponding sector
          number
109     for k = 1:size(dir_geostr)
110         n_sector = sectors_4_dir(dir_geostr(k));
111         sect_array(k) = n_sector;
112
113         speedup_array(k) = anem_speedup{i}(n_sector);

```


Appendix A

```
114
115     delta_th_array(k) = anem_delta_th{i}(n_sector);
116 end
117
118 % we add the Sector column with the information from
119     sect_array
120 MM5_tables{i}.Sector = sect_array;
121
122 % we add the Speedup column with the corresp. speed-up of
123     each sector in this anemometer
124 MM5_tables{i}.Speedup_0030m = speedup_array;
125
126 % we compute the Vmod = speed_up * speed
127 MM5_tables{i}.Velocity_MOD = MM5_tables{i}.Speedup .*
128     MM5_tables{i}.speed_m_s_;
129
130 % we compute V/dist
131 MM5_tables{i}.velocity_on_dist = MM5_tables{i}.Velocity_MOD
132     ./ dist(i+1); % (i+1) because dist have also the distance
133     from R to R
134
135 % we add the Delta Theta column with the corresponding
136     delta_theta of
137 % each sector in this anemometer
138 MM5_tables{i}.Delta_Theta = delta_th_array;
139
140 % we add the direction mod. We sum the direction with the
141     delta theta for each time-step
142 MM5_tables{i}.Dir_MOD = MM5_tables{i}.dir_deg_ +
143     MM5_tables{i}.Delta_Theta;
144
145 for c = find(MM5_tables{i}.Dir_MOD >= 360)
146     MM5_tables{i}.Dir_MOD(c) = MM5_tables{i}.Dir_MOD(c) - 360;
147 end
148
149 for c = find(MM5_tables{i}.Dir_MOD <= -360)
150     MM5_tables{i}.Dir_MOD(c) = MM5_tables{i}.Dir_MOD(c) + 360;
151 end
152
153 % we compute direction/dist
```

```
146     MM5_tables{i}.dir_on_dist = MM5_tables{i}.Dir_MOD ./
        dist(i+1); % (i+1) because dist have also the distance
        from R to R
147
148     end
149     disp("MM5 done");
150
151     % we initialize an array with the same length of the
        MM5_tables
152     sum_V_on_dist = zeros(length(MM5_tables{1}).DATA, 1);
153     sum_Dir_on_dist = zeros(length(MM5_tables{1}).DATA, 1);
154     sum_inverse_distance = 0;
155
156     for i = 1:n_anemometer
157         sum_V_on_dist = sum_V_on_dist +
            MM5_tables{i}.velocity_on_dist;
158         sum_Dir_on_dist = sum_Dir_on_dist +
            MM5_tables{i}.dir_on_dist;
159         sum_inverse_distance = sum_inverse_distance + (1 /
            dist(i+1));
160     end
161
162     v_fin = sum_V_on_dist / sum_inverse_distance;
163     dir_fin = sum_Dir_on_dist / sum_inverse_distance;
164
165     disp("done");
```

B

Appendix

The *Python* script, adopted for the creation of various modules of the pipeline, is here reported.

```
1 import KratosMultiphysics as Kratos
2 import KratosMultiphysics.GeodataProcessingApplication as
  KratosGeo
3 import KratosMultiphysics.FluidDynamicsApplication
4 if (Kratos.IsDistributedRun()):
5     import KratosMultiphysics.mpi
6
7 from
  KratosMultiphysics.GeodataProcessingApplication.geo_importer import GeoImporter
8 from
  KratosMultiphysics.GeodataProcessingApplication.geo_mesher import GeoMesher
9 from
  KratosMultiphysics.GeodataProcessingApplication.geo_preprocessor import GeoPreprocessor
```

```

10     from
        KratosMultiphysics.GeodataProcessingApplication.geo_build
        ing import GeoBuilding
11     from KratosMultiphysics.GeodataProcessingApplication.geo_model
        import GeoModel
12     from KratosMultiphysics.GeodataProcessingApplication.geo_data
        import GeoData
13
14     import math
15     import os
16     import time
17
18     # Parameters
19     with open("data/json/ProjectParameters.json", 'r') as
        parameter_file:
20         parameters =
            KratosMultiphysics.Parameters(parameter_file.read())
21         print("parameters read!")
22
23     with open("data/json/RemeshingParameters.json", 'r') as
        parameter_file:
24         remeshing_parameters =
            KratosMultiphysics.Parameters(parameter_file.read())
25         print("remeshing_parameters read!")
26
27     print("extract_center: ",
        parameters["problem_data"]["extract_center"].GetBool())
28
29     num_test =
        parameters["problem_data"]["problem_name"].GetString()
30
31     # we create a new folders for this test
32     if not os.path.exists("cfdata/test_{}".format(num_test)):
33         os.mkdir("cfdata/test_{}".format(num_test))
34     if not
        os.path.exists("cfdata/test_{}/gid_file".format(num_test)):
35         os.mkdir("cfdata/test_{}/gid_file".format(num_test))
36     if not
        os.path.exists("cfdata/test_{}/gid_file/Binary".format(
        num_test)):
37         os.mkdir("cfdata/test_{}/gid_file/Binary".format(num_test))

```

Appendix B

```
38     if not
39         os.path.exists("cfd_data/test_{}/gid_file/Ascii".format(num_test)):
40         os.mkdir("cfd_data/test_{}/gid_file/Ascii".format(num_test))
41     if not
42         os.path.exists("cfd_data/test_{}/stl_file".format(num_test)):
43         os.mkdir("cfd_data/test_{}/stl_file".format(num_test))
44     if not
45         os.path.exists("cfd_data/test_{}/analysis_file".format(num_test)):
46         os.mkdir("cfd_data/test_{}/analysis_file".format(num_test))
47     if not
48         os.path.exists("cfd_data/test_{}/mdpa_file".format(num_test)):
49         os.mkdir("cfd_data/test_{}/mdpa_file".format(num_test))
50     if not
51         os.path.exists("cfd_data/test_{}/metric".format(num_test)):
52         os.mkdir("cfd_data/test_{}/metric".format(num_test))
53
54     if not
55         os.path.exists("cfd_data/test_{}/AsterGDEM".format(num_test)):
56         os.mkdir("cfd_data/test_{}/AsterGDEM".format(num_test))
57     if not
58         os.path.exists("cfd_data/test_{}/OSM".format(num_test)):
59         os.mkdir("cfd_data/test_{}/OSM".format(num_test))
60
61     # domain parameters
62     r_domain =
63         parameters["problem_data"]["r_domain"].GetDouble()
64         # meters
65     r_buildings =
66         parameters["problem_data"]["r_buildings"].GetDouble()
67         # meters
68     height =
69         parameters["problem_data"]["height"].GetDouble()
70         # height of the domain (in meters)
71     num_sector =
72         parameters["problem_data"]["num_sector"].GetInt()
73         # number of sectors
```

```

59     circ_division =
        parameters["problem_data"]["circ_division"].GetInt() #
            number of divisions along the circle
60
61     data = GeoData(r_domain=r_domain)
62
63     # ┌───────────────────────────────────────────────────────────────────────────────────┐
64     # │                                     TERRAIN                                     │
65     # └───────────────────────────────────────────────────────────────────────────────────┘
66     imp_terrain = GeoImporter()
67     mesher = GeoMesher()
68
69     # download ASTER-GDEM
70     lat = parameters["problem_data"]["latitude"].GetDouble() #
        latitude of the building of interest
71     lon = parameters["problem_data"]["longitude"].GetDouble() #
        longitude of the building of interest
72     file_aster = "cfd_data/test_{}/AsterGDEM".format(num_test)
73     file_dem = data.DownloadAsterGDEM(file_aster, lat, lon)
74
75     # compute bounding box for the terrain
76     data.ComputeBbox(lat, lon, r_domain)
77
78     # crop ASTER-GDEM
79     file_crop = data.CropAsterGDEM(file_dem)
80
81     # ASTER-GDEM to OBJ
82     terrain_obj = data.AsterGDEMtoOBJ(file_crop)
83
84     imp_terrain.ObjImport(terrain_obj)
85     mesher.SetGeoModelPart(imp_terrain.GetGeoModelPart()) # we
        pass the ModelPart from imp_terrain to mesher
86     # GiD file
87     mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Ascii
        /terrain".format(num_test), "GiD_PostAscii")
88     mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Binar
        y/terrain".format(num_test), "GiD_PostBinary")
89     # MDPa file
90     mesher.WriteMdpaOutput("cfd_data/test_{}/mdpa_file/terrain".fo
        rmat(num_test))

```

Appendix B

```
91
92 # setup some variables
93 mesher.x_center = imp_terrain.x_center
94 mesher.y_center = imp_terrain.y_center
95 mesher.r_boundary = r_domain
96 mesher.r_ground = 2/3 * r_domain
97 mesher.r_buildings = r_buildings
98 mesher.height = height
99
100 mesher.extract_center()
101
102 # [-----]
103 # |           BUILDINGS           |
104 # [-----]
105
106 imp_building = GeoImporter()
107
108 # compute bounding box for the buildings
109 data.ComputeBbox(lat, lon, r_buildings)
110
111 # download of buildings from OpenStreetMap and conversion into
112 # JSON format
113 file_json =
114     "cfd_data/test_{}/OSM/osm_from_API.json".format(num_test)
115 geojson = data.DownloadBuildingsOSM(file_json)
116 # JSON to OBJ
117 building_obj =
118     "cfd_data/test_{}/OSM/osm_from_API.obj".format(num_test)
119 data.GeoJSONtoOBJ(geojson, building_obj)
120
121 imp_building.ObjImportBuildings(building_obj,
122     "BuildingModelPart") # (obj_file_name_input,
123     name_model_part, change_coord)
124
125 building_model_part = imp_building.GetGeoModelPart()
126
127 building = GeoBuilding()
128 building.SetGeoModelPart(building_model_part)
129
130 # GiD file
```

```

125     building.CreateGidControlOutput("cfd_data/test_{}/gid_file/Asc
        ii/buildings".format(num_test), "GiD_PostAscii")
126     building.CreateGidControlOutput("cfd_data/test_{}/gid_file/Bin
        ary/buildings".format(num_test), "GiD_PostBinary")
127     # MDPa file
128     mdpa_out_name =
        "cfd_data/test_{}/mdpa_file/buildings".format(num_test)
129     building.WriteMdpaOutput(mdpa_out_name)
130
131     building.DeleteBuildingsOutsideBoundary(mesher.x_center,
        mesher.y_center, mesher.r_buildings)
132     # GiD file
133     building.CreateGidControlOutput("cfd_data/test_{}/gid_file/Asc
        ii/buildings_cleaned".format(num_test), "GiD_PostAscii")
134     building.CreateGidControlOutput("cfd_data/test_{}/gid_file/Bin
        ary/buildings_cleaned".format(num_test),
        "GiD_PostBinary")
135
136     # building.ShiftBuildingOnTerrain(mesher.GetGeoModelPart())
137     building.ShiftBuildingOnTerrain(mesher.center_model_part)
138
139     # GiD file
140     building.CreateGidControlOutput("cfd_data/test_{}/gid_file/Asc
        ii/buildings_shifted".format(num_test), "GiD_PostAscii")
141     building.CreateGidControlOutput("cfd_data/test_{}/gid_file/Bin
        ary/buildings_shifted".format(num_test),
        "GiD_PostBinary")
142     # MDPa file
143     mdpa_out_name =
        "cfd_data/test_{}/mdpa_file/buildings_shifted".format(num
        _test)
144     building.WriteMdpaOutput(mdpa_out_name)
145
146     mesher.MeshHole(building.GetGeoModelPart(),
        circ_division=circ_division,
147                     num_sector=num_sector)
148
149     mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Ascii
        /ForoTerreno".format(num_test), "GiD_PostAscii")
150     mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Binar
        y/ForoTerreno".format(num_test), "GiD_PostBinary")
151
152     mesher.AddBuildingsOnTerrain(building.GetGeoModelPart())

```


Appendix B

```
153 mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Ascii
    /ForoTerrenoEdifici".format(num_test), "GiD_PostAscii")
154 mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Binar
    y/ForoTerrenoEdifici".format(num_test), "GiD_PostBinary")
155
156 metric_file =
    "cfd_data/test_{}/metric/mtr_file.mtr".format(num_test)
157
158 mesher.BuildMesh3D(metric_file)
159 # GiD file
160 mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Ascii
    /01_Mesh_cylinder".format(num_test), "GiD_PostAscii")
161 mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Binar
    y/01_Mesh_cylinder".format(num_test), "GiD_PostBinary")
162
163 # we populate the sub model part sectors
164 mesher.MeshSectors(num_sector)
165 # GiD file
166 mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Ascii
    /02_Mesh_final".format(num_test), "GiD_PostAscii")
167 mesher.CreateGidControlOutput("cfd_data/test_{}/gid_file/Binar
    y/02_Mesh_final".format(num_test), "GiD_PostBinary")
168 # MDPA file
169 mdp_out_name =
    "cfd_data/test_{}/mdpa_file/02_Mesh_final".format(num_tes
    t)
170 mesher.WriteMdpaOutput(mdp_out_name)
171
172 # JSON parameters useful to run the analysis
173 # we update MDPA file name
174 parameters["solver_settings"]["model_import_settings"]["input_
    filename"].SetString(mdp_out_name)
175
176 model = GeoModel(parameters)
177 # fluid
178 model.PartsFluid(sub_model_name="Parts_Fluid")
179 # no slip
180 model.NoSlip(sub_model_name="SKIN_ISOSURFACE")
181 # slip
182 model.Slip(sub_model_name="TopModelPart")
183 model.Slip(sub_model_name="BottomModelPart")
```

```
184 # inlet/outlet
185 direction = 4 # from North
186 n_sectors = 12
187 model.Inlet_Outlet(n_sectors=n_sectors,
188                   dir_in=direction,
189                   v_in=6.0,
190                   base_smp="LateralSector_")
191
192 # store json in a file
193 json_name =
194     "cfd_data/test_{}/analysis_file/ProjectParameters.json".f
195     ormat(num_test)
196 with open(json_name, "w") as outfile:
197     outfile.write(model.cfd_param.PrettyPrintJsonString())
198
199 print("\nTEST ", num_test, "END\n\n")
```

C

Appendix

To show the potentialities of the open-source application (Chapter 8) the numerical models, obtained automatically, of two different Italian cities are reported in the following lines.

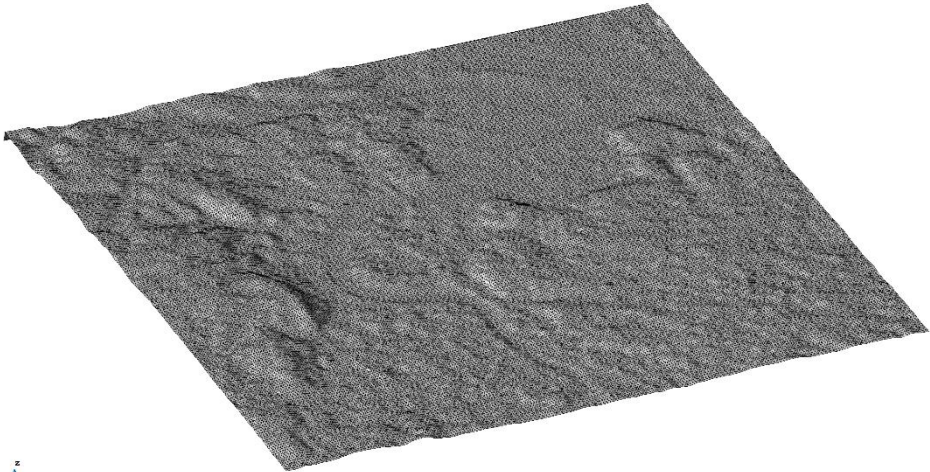
The first area is located in the city of Rome (Italy) and the center of the domain coincides with the Auditorium Parco della Musica "Ennio Morricone" (latitude 41.9280, longitude 12.4735). Figure C.1 shows the OSM map from which the building geometry information is derived, while Figure C.2 presents a satellite image of the selected area. First, the geometry of the terrain (Figure C.3) and buildings (Figure C.4) is generated, and then they are combined into a single numerical model (Figure C.7). The potential of the application is clearly visible by comparing an aerial photo of the chosen area (Figure C.5) and a respective three-dimensional representation obtained (Figure C.6).



Figure C.1 - OSM of Rome (Italy)



Figure C.2 - Satellite image of Rome (Italy)



GID

Figure C.3 - 3D terrain of Rome (Italy)



GID

Figure C.4 - 3D building in an area located in Rome (Italy)



Figure C.5 - Photo of an area in Rome (Italy)

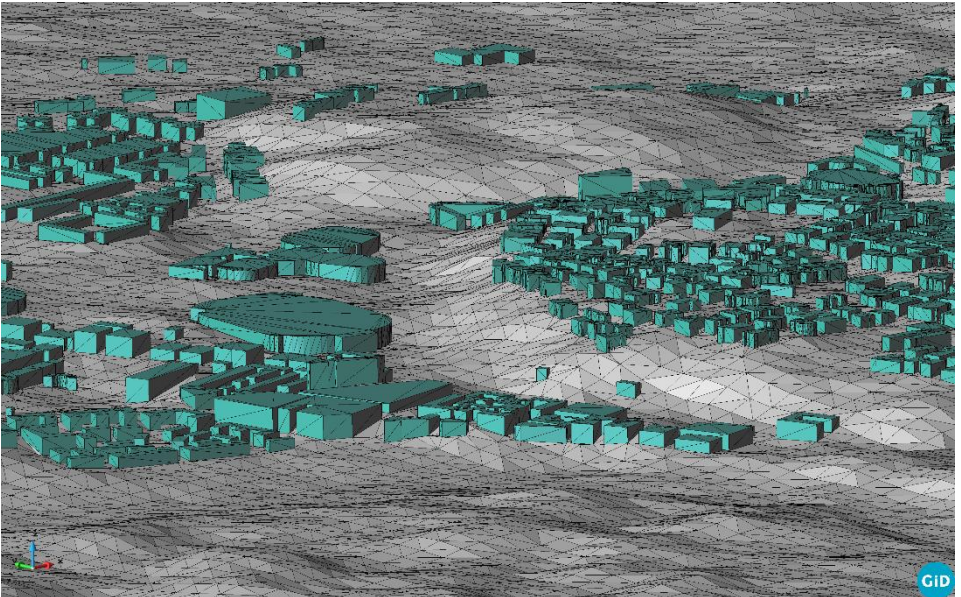


Figure C.6 - Detail of terrain surface and 3D buildings: Rome (Italy)

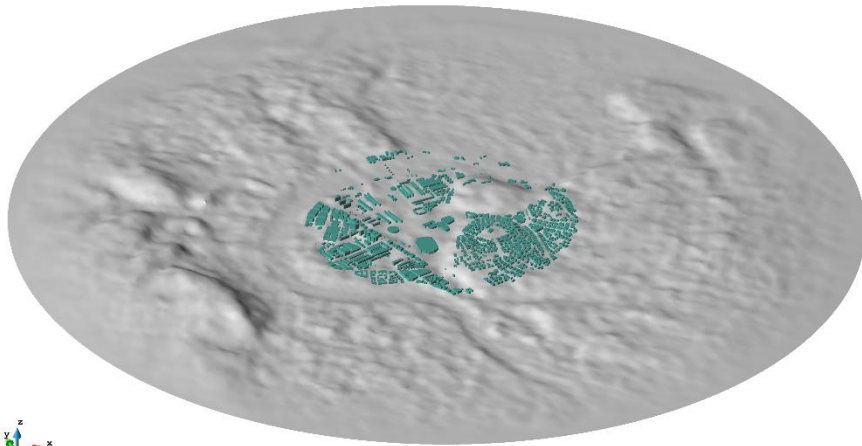


Figure C.7 - Terrain surface and 3D buildings: Rome (Italy)

The second example that is reported concerns the city of San Giovanni Rotondo (Italy). The center of the domain is located at latitude 41.70803 and longitude 15.72813. Also in this case, the OSM map (Figure C.8) and the respective satellite image (Figure C.9) are reported. The terrain surface of the analyzed area and the automatically generated buildings are shown, respectively, in Figure C.10 and Figure C.11, while the combined numerical model is shown in Figure C.14. Finally, an aerial photo (Figure C.12) and 3D model (Figure C.13) of the same area are shown to appreciate the quality of the result obtained.



Figure C.8 - OSM of San Giovanni Rotondo (Italy)

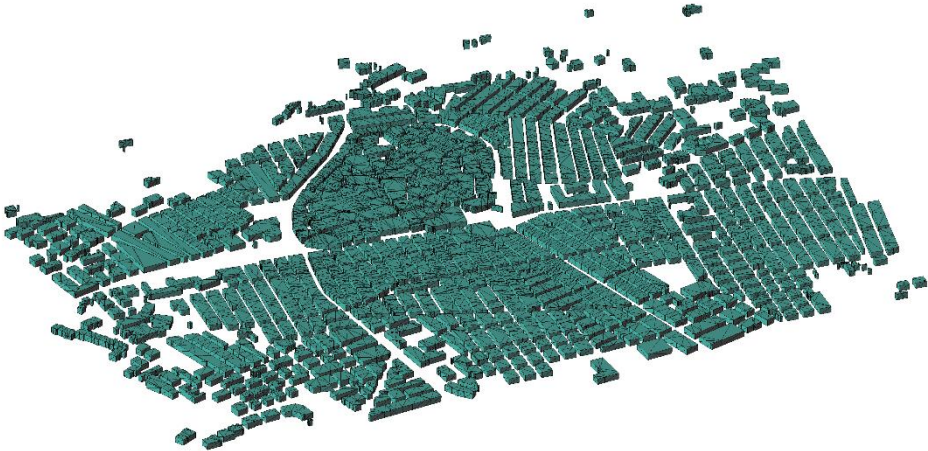


Figure C.9 - Satellite image of San Giovanni Rotondo (Italy)



GID

Figure C.10 - 3D terrain of San Giovanni Rotondo (Italy)



GID

Figure C.11 - 3D building in an area located in San Giovanni Rotondo (Italy)



Figure C.12 - Photo of an area in San Giovanni Rotondo (Italy)

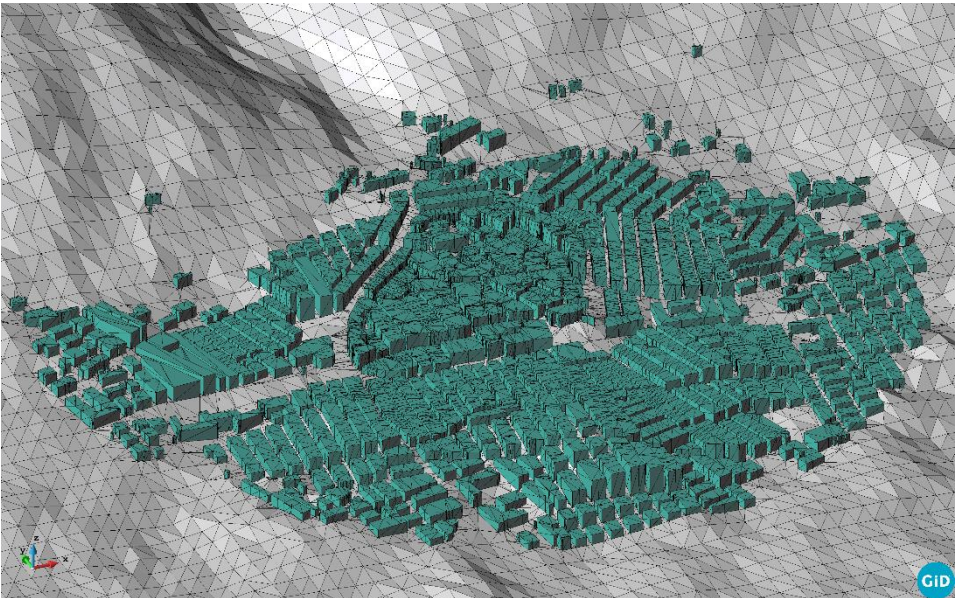


Figure C.13 - Detail of terrain surface and 3D buildings: San Giovanni Rotondo (Italy)

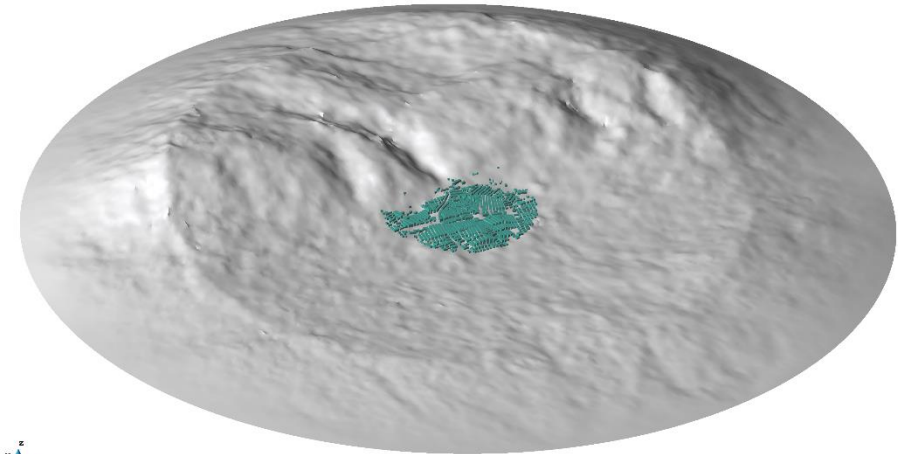


Figure C.14 - Terrain surface and 3D buildings: San Giovanni Rotondo (Italy)

Reference

- [1] WMO (2020). United in Science 2020. 1–20. [Online]. Available: https://public.wmo.int/en/resources/united_in_science
- [2] Vicedo-Cabrera A. M. *et al.* (2021). The burden of heat-related mortality attributable to recent human-induced climate change. *Nature Climate Change*. 11: 492–500. <https://doi.org/10.1038/s41558-021-01058-x>
- [3] Karanasiou A. *et al.* (2021). Short-term health effects from outdoor exposure to biomass burning emissions: A review. *Science of the Total Environment*. 781: 146739. <https://doi.org/10.1016/j.scitotenv.2021.146739>
- [4] Oke R. T. (1987). *Boundary Layer Climates*. London. 2nd ed. <https://doi.org/10.4324/9780203407219>
- [5] Howard L. (1820). *The climate of London*. London. Volume II
- [6] Department of Economic and Social Affairs (2015). *Population Division World Urbanization Prospects - The 2014 Revision*.
- [7] Blocken B. (2015). Computational Fluid Dynamics for urban physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations. *Building and Environment*. 91: 219–245. <https://doi.org/10.1016/j.buildenv.2015.02.015>
- [8] Toparlar Y. *et al.* (2015). CFD simulation and validation of urban microclimate: A case study for Bergpolder Zuid, Rotterdam. *Building and Environment*. 83: 79–90. <https://doi.org/10.1016/J.BUILDENV.2014.08.004>
- [9] Glossary of Meteorology. <https://www.ametsoc.org/ams/index.cfm/publications/glossary-of-meteorology/>
- [10] Toparlar Y., Blocken B., Maiheu B., and van Heijst G. J. F. (2017). A review on the CFD analysis of urban microclimate. *Renewable and Sustainable Energy Reviews*. 80: 1613–1640. <https://doi.org/10.1016/j.rser.2017.05.248>
- [11] Jeanjean A., Buccolieri R., Eddy J., Monks P., and Leigh R. (2017). Air quality affected by trees in real street canyons: The case of Marylebone neighbourhood in central London. *Urban Forestry and Urban Greening*. 22: 41–53. <https://doi.org/10.1016/j.ufug.2017.01.009>
- [12] Jones W. P. and Launder B. E. (1972). The prediction of laminarization with

- a two-equation model of turbulence. *International Journal of Heat and Mass Transfer*. 15: 301–314. [https://doi.org/10.1016/0017-9310\(72\)90076-2](https://doi.org/10.1016/0017-9310(72)90076-2)
- [13] Tominaga Y. et al. (2008). AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings. *Journal of Wind Engineering and Industrial Aerodynamics*. 96: 1749–1761. <https://doi.org/10.1016/J.JWEIA.2008.02.058>
- [14] Shih T. H., Liou W. W., Shabbir A., Yang Z., and Zhu J. (1995). A new k- ϵ eddy viscosity model for high reynolds number turbulent flows. *Computers & Fluids*. 24: 227–238. [https://doi.org/10.1016/0045-7930\(94\)00032-T](https://doi.org/10.1016/0045-7930(94)00032-T)
- [15] Yakhot V. and Orszag S. A. (1986). Renormalization group analysis of turbulence. I. Basic theory. *Journal of Scientific Computing* 1986 1:1. 1: 3–51. <https://doi.org/10.1007/BF01061452>
- [16] Souch C. and Grimmond S. (2006). Applied climatology: Urban climate. *Progress in Physical Geography*. 30: 270–279. <https://doi.org/10.1191/0309133306pp484pr>
- [17] Rizwan A. M., Dennis L. Y. C., and Liu C. (2008). A review on the generation, determination and mitigation of Urban Heat Island. *Journal of Environmental Sciences*. 20: 120–128. [https://doi.org/10.1016/S1001-0742\(08\)60019-4](https://doi.org/10.1016/S1001-0742(08)60019-4)
- [18] Moonen P., Defraeye T., Dorer V., Blocken B., and Carmeliet J. (2012). Urban Physics: Effect of the micro-climate on comfort, health and energy demand. *Frontiers of Architectural Research*. 1: 197–228. <https://doi.org/10.1016/j.foar.2012.05.002>
- [19] Haupt S. E. et al. (2015). First Year Report of the A2e Mesoscale to Microscale Coupling Project. <https://doi.org/10.13140/RG.2.2.21572.01927>
- [20] Kadaverugu R., Sharma A., Matli C., and Biniwale R. (2019). High Resolution Urban Air Quality Modeling by Coupling CFD and Mesoscale Models: a Review. *Asia-Pacific Journal of Atmospheric Sciences*. 55: 539–556. <https://doi.org/10.1007/s13143-019-00110-3>
- [21] Miao Y., Liu S., Zheng H., Zheng Y., Chen B., and Wang S. (2014). A multi-scale urban atmospheric dispersion model for emergency management. *Advances in Atmospheric Sciences*. 31: 1353–1365. <https://doi.org/10.1007/s00376-014-3254-9>
- [22] Miao Y., Liu S., Chen B., Zhang B., Wang S., and Li S. (2013). Simulating urban flow and dispersion in Beijing by coupling a CFD model with the WRF model. *Advances in Atmospheric Sciences*. 30: 1663–1678. <https://doi.org/10.1007/s00376-013-2234-9>

- [23] Zheng Y., Miao Y., Liu S., Chen B., Zheng H., and Wang S. (2015). Simulating flow and dispersion by using WRF-CFD coupled model in a built-up area of Shenyang, China. *Advances in Meteorology*. 2015. <https://doi.org/10.1155/2015/528618>
- [24] Tewari M. et al. (2010). Impact of coupling a microscale computational fluid dynamics model with a mesoscale model on urban scale contaminant transport and dispersion. *Atmospheric Research*. 96: 656–664. <https://doi.org/10.1016/j.atmosres.2010.01.006>
- [25] Blocken B., Janssen W. D., and 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. *Environmental Modelling and Software*. 30: 15–34. <https://doi.org/10.1016/j.envsoft.2011.11.009>
- [26] Li S., Sun X., Zhang S., Zhao S., and Zhang R. (2019). A study on microscale wind simulations with a coupled WRF-CFD model in the chongli mountain region of hebei province, China. *Atmosphere*. 10: 1–22. <https://doi.org/10.3390/ATMOS10120731>
- [27] Carvalho D., Rocha A., Santos C. S., and Pereira R. (2013). Wind resource modelling in complex terrain using different mesoscale-microscale coupling techniques. *Applied Energy*. 108: 493–504. <https://doi.org/10.1016/j.apenergy.2013.03.074>
- [28] Bilal M., Birkelund Y., Homola M., and Virk M. S. (2016). Wind over complex terrain – Microscale modelling with two types of mesoscale winds at Nygårdsfjell. *Renewable Energy*. 99: 647–653. <https://doi.org/10.1016/j.renene.2016.07.042>
- [29] Erba S., Causone F., and Armani R. (2017). The effect of weather datasets on building energy simulation outputs. *Energy Procedia*. 134: 545–554. <https://doi.org/10.1016/J.EGYPRO.2017.09.561>
- [30] Lupato G. and Manzan M. (2019). Italian TRYs: New weather data impact on building energy simulations. *Energy and Buildings*. 185: 287–303. <https://doi.org/10.1016/J.ENBUILD.2018.12.001>
- [31] Bhandari M., Shrestha S., and New J. (2012). Evaluation of weather datasets for building energy simulation. *Energy and Buildings*. 49: 109–118. <https://doi.org/10.1016/J.ENBUILD.2012.01.033>
- [32] Chiesa G. and Grosso M. (2015). The Influence of Different Hourly Typical Meteorological Years on Dynamic Simulation of Buildings. *Energy Procedia*. 78: 2560–2565. <https://doi.org/10.1016/J.EGYPRO.2015.11.280>
- [33] Ciobanu D., Eftimie E., and Jaliu C. (2014). The Influence of Measured/simulated Weather Data on Evaluating the Energy Need in

- Buildings. *Energy Procedia*. 48: 796–805.
<https://doi.org/10.1016/J.EGYPRO.2014.02.092>
- [34] Montelpare S., D'Alessandro V., Lops C., Costanzo E., and Ricci R. (2019). A Mesoscale-Microscale approach for the energy analysis of buildings. *Journal of Physics: Conference Series*. 1224. <https://doi.org/10.1088/1742-6596/1224/1/012022>
- [35] Daftry S., Hoppe C., and Bischof H. (2015). Building with drones: Accurate 3D facade reconstruction using MAVs. in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 3487–3494
<https://doi.org/10.1109/ICRA.2015.7139681>
- [36] Biljecki F., Ledoux H., and Stoter J. (2016). An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*. 59: 25–37. <https://doi.org/10.1016/j.compenvurbsys.2016.04.005>
- [37] Kolbe T. H. (2009). Representing and Exchanging 3D City Models with CityGML., in *3D Geo-Information Sciences*, J. Lee and S. Zlatanova, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg. 15–31.
https://doi.org/10.1007/978-3-540-87395-2_2
- [38] Over M., Schilling A., Neubauer S., and Zipf A. (2010). Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany. *Computers, Environment and Urban Systems*. 34: 496–507.
<https://doi.org/10.1016/J.COMPENVURBSYS.2010.05.001>
- [39] Olsen B. T. (2018). Mesoscale to microscale coupling for determining site conditions in complex terrain. <https://doi.org/10.11581/00000036>
- [40] Grell G. A., Dudhia J., and David S. (1994). A description of the fifth-generation Penn State/NCAR Mesoscale Model (MM5). *University Corporation for Atmospheric Research*. <https://doi.org/10.5065/D60Z716B>
- [41] Dudhia J. et al. (2005). PSU/NCAR Mesoscale Modeling System Tutorial Class Notes and User's Guide: MM5 Modeling System Version 3.
- [42] Hong S.-Y. and Pan H.-L. (1996). Nonlocal Boundary Layer Vertical Diffusion in a Medium-Range Forecast Model. *Monthly Weather Review*. 124: 2322–2339.
[https://doi.org/10.1175/1520-0493\(1996\)124<2322:NBLVDI>2.0.CO;2](https://doi.org/10.1175/1520-0493(1996)124<2322:NBLVDI>2.0.CO;2)
- [43] Pleim J. E. and Chang J. S. (1992). A non-local closure model for vertical mixing in the convective boundary layer. *Atmospheric Environment. Part A. General Topics*. 26: 965–981.
- [44] Xiu A. and Pleim J. E. (2001). Development of a land surface model. Part I: Application in a mesoscale meteorological model. *Journal of Applied Meteorology*. 40: 192–209. <https://doi.org/10.1175/1520->

[0450\(2001\)040<0192:DOALSM>2.0.CO;2](#)

- [45] Blackadar A. K. (1976). Modeling the Nocturnal Boundary Layer. *American Meteorological Society*. 46–49.
- [46] Corrado V., Ballarini I., Corgnati S. P., and Talà N. (2014). Building Typology Brochure -Italy Fascicolo sulla Tipologia Edilizia Italiana.
- [47] UNI EN 15251:2008 Criteri per la Progettazione dell'Ambiente Interno e per la Valutazione della Prestazione Energetica degli Edifici, in Relazione alla Qualità dell'Aria Interna, all'Ambiente Termico, all'Illuminazione e all'Acustica.
- [48] Lops C. (2020). INTEGRATED SOLUTIONS FOR THE ENERGY AND SEISMIC RETROFIT OF EXISTING BUILDINGS.
- [49] Dadvand P., Rossi R., and Oñate E. (2010). An object-oriented environment for developing finite element codes for multi-disciplinary applications.17. <https://doi.org/10.1007/s11831-010-9045-2>
- [50] Dapogny C., Dobrzynski C., and Frey P. (2014). Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*. 262: 358–378. <https://doi.org/10.1016/j.jcp.2014.01.005>
- [51] NASA, METI, AIST, Japan Spacesystems, and U.S. Japan ASTER Science Team (2018). ASTER Global Digital Elevation Model V003. *NASA EOSDIS Land Processes DAAC*. <https://doi.org/https://doi.org/10.5067/ASTER/ASTGTM.003>