A review of Machine Learning CFD-based predictive modelling for real-time forecasting simulations of pollutant dispersion in an urban mesh

Leonardo de Souza Sá

Advisor: D.Sc Fábio Pereira dos Santos

Rio de Janeiro

December 2023

A review of Machine Learning CFD-based predictive modelling for real-time forecasting simulations of pollutant dispersion in an urban mesh

Leonardo de Souza Sá

A bachelor thesis submitted to the Chemical Engineering department of the School of Chemistry at the Federal University of Rio de Janeiro in fulfillment of the requirements for the degree of Chemical Engineer

Author:

Leonardo de Souza Sá

Advisor professor:

Prof. D.Sc Fábio Pereira dos Santos

Thesis Defence Committee:

Prof. PhD Tânia Suaiden Klein

Engr. Gustavo Schiavone

Rio de Janeiro

December 2023

de Souza Sá, Leonardo.
A review of Machine Learning CFD-based predictive modelling for real-time forecasting simulations of pollutant dispersion in an urban mesh / Leonardo de Souza Sá. Rio de Janeiro: UFRJ/EQ, 2023.
xv, 76 p.; il.
(Monografia) - Universidade Federal do Rio de Janeiro, Escola de Química, 2023.
Orientadores: Fábio Pereira dos Santos.
1. CFD. 2. Machine Learning. 3. Pollutants dispersion. 4. Monografia (Graduação UFRJ/EQ). 5. Fábio Pereira dos Santos. I. A review of Machine Learning CFD-based predictive modelling for real-time forecasting simulations of pollutant dispersion in an urban mesh.

This work is dedicated to my grandfather.

"Lead me on my dreams among different time and space. To share hope with nations and believers. To observe with modesty the pure truth. And to reveal prudently the magic and the mystery."

- Varda Carmeli

ACKNOWLEDGEMENTS

I would like to thank Professor Fábio for accepting to be my advisor in this thesis. Big thanks to everybody I met who supported me over these 6+ years of two graduations in engineering, both from inside and outside the university campus.

Resumo da Monografia apresentada à Escola de Química como parte dos requisitos necessários para obtenção do grau de Bacharel em Engenharia Química

Este trabalho tem como objetivo revisar a evolução no estudo da modelagem preditiva com Machine Learning baseada em CFD de forma geral, destacando suas deficiências e a possibilidade de novas melhorias nesta área. Especificamente, esta tese de graduação analisa a base teórica desta técnica conjunta aplicada a simulações de previsão em tempo real de dispersão de poluentes em uma malha urbana. Utilizando a estrutura estabelecida por Ganti e Khare em um artigo publicado em 2020, este trabalho explica a teoria por detrás das técnicas mais utilizadas para fornecer uma visão geral de como se poderia conduzir um tal modelo para prever dispersões de gases em uma cidade. Alguns casos exemplo são simulados utilizando os softwares OpenFOAM® e SimFlow® e os dados gerados por estes são utilizados para realizar a técnica da Regressão do Processo Gaussiano (GPR), com o objetivo de, assim, prever o comportamento do fluxo em menos tempo, melhorando o desempenho e potencialmente reduzindo os custos de simulação.

A review of Machine Learning CFD-based predictive modelling for real-time forecasting simulations of pollutant dispersion in an urban mesh

Leonardo de Souza Sá

December, 2023

Orientadores: Prof. Fábio Pereira dos Santos, D.Sc

Palavras-chave: 1. CFD. 2. Machine Learning. 3. Artificial Inteligence. 4. Pollutants dispersion.

ABSTRACT

This work aims to review the evolution in the study of Machine Learning CFD-based predictive modeling in general, highlighting its deficiencies and the possibility for further improvements in this area. Specifically, this undergraduate thesis analyses the theoretical basis of this conjoint technique applied to real-time forecasting simulations of pollutant dispersion in an urban mesh. Using the framework established by Ganti and Khare in a paper published in 2020, this work explains the theory behind the most-used techniques to provide an overview of how one could conduct a ML CFD-based model to predict gas dispersion in a city. Some toy models are simulated using OpenFOAM® and SimFlow® CFD software and data generated by these are used to perform a Gaussian Process Reduction (GPR) and thus predict the behavior of the flow in less time, improving performance and potentially reducing simulation costs.

A review of Machine Learning CFD-based predictive modelling for real-time forecasting simulations of pollutant dispersion in an urban mesh

Leonardo de Souza Sá

December, 2023

Supervisors: Prof. Fábio Pereira dos Santos, D.Sc

Key-words: 1. CFD. 2. Machine Learning. 3. Artificial Intelligence. 4. Pollutants dispersion.

Contents

Nomenclature

$\mathbf{I}$	Identity tensor
$\mathbf{u}$	Flow velocity
μ	Dynamic viscosity
ν	Cinematic viscosity
$\overline{u}_i$	Time averaged velocity component i
ρ	Fluid density
g	Gravitational acceleration
p	Pressure
u	Velocity vector module
u_i	Velocity vector module in the i direction
u_j	Velocity vector module in the j direction
$2D$	Two-dimensional
$3D$	Three-dimensional
AI	Artificial Inteligence
ANN	Artificial Neural Network
BC	Black Carbon
BH_{std}	Standard deviation of building height
BVN	Bi-variate Normal Distribution

CDS Linear Interpolation

CFD Computational Fluid Dynamics

CM Control mass

CO Carbon monoxide

CV Control volume

DES Detached Eddy Simulation

DMD Dynamic Mode Decomposition

DoE Design of Experiments

DOF Degrees of freedom

ERM Empirical Risk Minimization

FVM Finite Volume Method

GP Gaussian Process

GPR Gaussian Process Reduction or Gaussian Process Regression

$HDDV$ Heavy-Duty Diesel Vehicle

$IARC$ International Agency for Research on Cancer

$IHME$ Institute for Health Metrics and Evaluation

$IROM$ Intrusive Reduced-Order Modelling

LES Large Eddy Simulation

LHS Latin Hypercube Sampling

$LiDAR$ Light Detection and Ranging

Ma Mach number

ML Machine Learning

MVN Multivariate Normal Distribution

$NIROM$ Non-Intrusive Reduced-Order Model

NOx Nitrogen oxides

PCA Principal Component Analysis

PDF Probability Density Function

PN Particulate Number

POD Proper Orthogonal Decomposition

$pPAH$ Polycyclic Aromatic Hydrocarbons

$QUICK$ Quadratic Upwind Interpolation

$RANS$ Reynolds Averaged Navier-Stokes

RBF Radial basis function

Re Reynolds number

RNG Re-Normalization Group

ROM Reduced Order Modeling

RS Response Surface

Sc Schmidt number

SE Squared exponential kernel function

$SRANS$ Steady Reynolds Averaged Navier-Stokes

SRM Structural Risk Minimization

SVD Singular Value Decomposition

UBL Urban Boundary Layer

UCL Urban Canopy Layer

UDS Upwind Interpolation

UPM Polytechnic University of Madrid

$URANS$ Unsteady Reynolds Averaged Navier-Stokes

WHO World Health Organization

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Predictive models have become essential tools to avoid or mitigate tragic events, such as gas leakage from multipurpose industries. Different tools exist nowadays in an effort to construct robust models to prevent this kind of scenario. Between those is Computational Fluid Dynamics (CFD), a technique used in many scientific areas to study the behavior of fluids using numerical methods and strong algorithms that allow a deep understanding of the flow dynamics in a certain space-time frame.

This technique is of high value to the analysis of gas dispersion aiming to avoid the exposure of the population to high concentrations of pollutant molecules. Using not only CFD techniques but also Machine Learning (ML) methods together is essential to speed up the simulation process as well as to understand the scenario. In this sense, this combination might become a useful tool to improve a city management system and thus increase the welfare of its population.

From a technical perspective, conducting simulations repeatedly is vital to assess sensitivity and uncertainties accurately. This underscores the need for a swift algorithm capable of being iterated without excessive time consumption. The element of time assumes critical importance for authorities, especially in swiftly executing emergency plans, such as responding to unexpected occurrences like toxic gas emissions from fuel deposits.

Emergency plans can thus be developed fast and adapted to each specific situation. Furthermore, these models can be used as a comparison with others

already present in the market (e.g.: from American Legislation data, United States Environmental Agency) to provide valuable recommendations like how excessive are the pollution levels and how harmful they are to human health. (MENDIL et al., 2022).

Gas emissions can be very dangerous to human health and affect all countries, developed and underdeveloped. In most cases, the effect is only an olfactory discomfort due to strong odors but can rapidly scale up to acute and chronic respiratory diseases (e.g. lung cancer), heart disease, stroke and problems with the fetus.

In 2017, 68.14% of the Brazilian population lived in places where the mean of the annual concentration of PM2.5 was greater than the limit of $10\mu g/m^3$ suggested by the World Health Organization (WHO). For more information about the part of the population exposed to air pollution levels above WHO's 2017 guideline by country, consult Our World in data website (https://ourworldindata.org/outdoor-air-pollution). PM2.5 is a term created to designate particles with a diameter lower than $2.5\mu m$ that tend to cause negative effects to human health due to their capacity to penetrate the airways and affect the respiratory system.

In 2018, approximately 10% of the global population lived in cities with a total population trespassing 10 million inhabitants in a territorial fraction corresponding only to 0.2% of the Earth's surface. The expectation is that in 2060 68% of the global population will be living in urban territory (UNDESA, 2018). These data show a tendency that human activities are going to be more and more concentrated in a delimited and confined area, a dense urban mesh, which tends to inflate the aforementioned health issues related to air pollution.

Also, a more recent study by the World Health Organization (WHO) showed that ambient air pollution was responsible for the premature death of 4.2 million people around the world in 2019 (WHO, 2022). In Brazil, the most affected age group is of adults over 50 years old (see Table (1.1)), with the number of deaths being concentrated in the most advanced ages (IHME, 2019).

All these data show a concerning situation that only seems to worsen with the concentration of the population in small areas, which highlights the importance of studying the dispersion of pollutants and the air quality of an urban mesh using

Table 1.1: Death distribution caused by external, divided by age group in Brazil, in 2019. Adapted from IHME, Global Burden of Disease, 2019.

Age group	No. of deaths per 100,000 inhabitants
Less than 5 years old	9.10
From 5 to 14 years old	0.09
From 15 to 49 years old	3.47
From 50 to 69 years old	40.51
More than 69 years old	194.58

the most advanced tools to help decelerate this problem.

1.2 Objectives

1.2.1 General Objectives

The main objective of this undergraduate thesis is to review the most recent information available concerning Machine Learning CFD-based predictive modelling of airflow in an urban mesh and test some simple models using the tools presented.

1.2.2 Specific Objectives

The specific objectives are:

1. Comprehend the evolution of the number of publications in ML and CFD, and from their correlation obtain a search method that would allow to go further in the subject;

2. Obtain a list of the most recent studies in Machine Learning CFD-based predictive models to evaluate the strengths and weaknesses of the latest methods;

3. From the results generated in the search, allow anyone to have a major understanding of the subject and thus develop (at least a foundation of) its framework based on the approach of the study presented in Chapter 2;

4. Analyse from recent studies how the prediction of the fluid flow could help infer the pollutant dispersion in an urban mesh.

5. Through simple simulation of the flow around a circular cylinder, prove the efficiency of applying GPR along with CFD for reducing the computational time using toy models;

1.3 Organization of this Work

This text is organized as follows:

- In Chapter 2, a general approach is showcased through an algorithm proposing a Machine Learning CFD-based predictive model based on the work of Ganti and Khare (2020). Each step within the framework succinctly outlines the process of establishing a foundation for creating a robust predictive model. Each stage has its theory detailed individually. First, general considerations are made about an urban mesh, followed by basic concepts in CFD modeling, Reduced Order Modeling (ROM) and Machine Learning (with a special discussion in GPR);

- In Chapter 3, the methodology is presented. The revision of the subject is done based on different current scientific search engines and the application of toy models follow already developed codes in CFD and MATLAB®;

- In Chapter 4, the results produced by the methodology are presented and analyzed. A discussion is developed using the later results and information obtained from the papers searched covering most of the specific objectives stated in the last subsection;

- In Chapter 6, a brief conclusion in this undergraduate thesis gives an overlook of all the work done and presents some future perspectives on the subject;

Chapter 2

Literature Review

In numerical simulations reliant on substantial processing capacity, greater applicability is achieved when the model's time consumption is minimized. Considering the case of pollutant gas dispersion in an urban mesh, it would be very interesting to propose a processing time fast enough to allow authorities to make almost instantaneous and correct decisions concerning the city's management (e.g.: controlling the vehicle traffic disposition) to minimize the concentration of pollutant particles.

The implementation of an ML algorithm becomes a solution to this problem since if set the proper boundary conditions of the system, one can speed up the analysis that often would take much longer with conventional simulation techniques.

2.1 Emulation framework

Developing a robust technique capable of retaining key information of the system dynamics based on a high-fidelity model and accelerating the output result acquisition would be a valuable approach to give this desired rapidness and precision.

In this sense, Ganti and Khare (2020) suggests an emulation framework that consists of four steps resumed in the following subsections. Each stage is detailed to give a general yet solid understanding of how each part works separately.

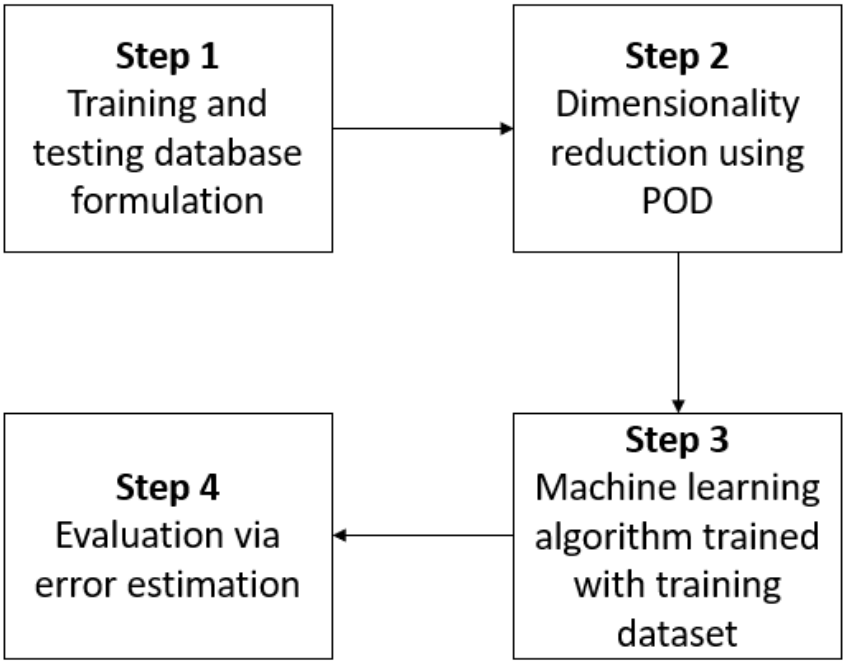

Figure 2.1: Emulation framework using an ML algorithm based on a CFD trained data set to produce an emulated flow field as a good approximation to the high-fidelity model in less time. Adapted from Ganti and Khare (2020).

2.2 Training and testing database formulation - Step 1

2.2.1 Design of Experiments

The Design of Experiments (DoE) is the first stage in the emulation framework and it is an essential part of the process in Figure (2.1). During the DoE the representative points of the dynamic system are chosen to train the ML algorithm later on. These are the starting points that generate the predicted results or emulations.

This stage consists of selecting the points of interest that globally represent the dynamic of the process. As defining a specific point of a complex space-temporal mesh is a fastidious task, a meticulous choice of the starting configuration must be made carefully. So, the preference are methods like the Latin Hypercube Sampling (LHS). More details concerning the application of this technique can be found in Viana (2016).

Usually, one should seek points of the system that represent changes in the dynamic of the flow, such as vortex and/or air re-circulation points. In the specific case of the flow around a circular cylinder, the analysis of a window of interest that covers mostly the immediate downstream points of the flow contains valuable information about the adopted model.

Figure (2.2) depicts the choice of a good window of interest in the flow of air around a circular cylinder of diameter D_0 with a certain Reynolds number.

Figure 2.2: Representation of a window of interest in a flow around a circular cylinder. Adapted from Ganti and Khare (2020).

Being defined in this window of interest, equally spaced points corresponding to the worst-case scenario are chosen. The total number of points covering the entire data space X is a function of the number of intervals N par component K as shown in the following expression:

$$No.\,of\,points = N^K. \tag{2.1}$$

The first block from Figure (2.1) resumes the first stage where the flow of air around a cylinder was taken as reference and the training data set is composed of high-fidelity simulations. In our case, these simulations correspond to CFD simulations. In this thesis, our proposed methodology introduces a comprehensive review process aimed at advancing knowledge within this scientific domain. This approach aims to solidify a distinct model of truth, thereby paving the way for the development of a robust AI tool in the future.

Later on in this undergraduate thesis, it will be demonstrated that these simulations can be easily reproduced (at least for basic scenarios with a few objects) using openSource software such as OpenFOAM® [1]. More specifically, it will

[1]More details at openfoam.com

reproduce the flow of water around a cylinder and the flow of air in a generic urban mesh to illustrate the behavior of velocity fields and their corresponding magnitudes. From these simulations, ML algorithms can be trained to provide accurate results with less computational cost.

2.2.2 Urban Micro-climate

The understanding of the urban micro-climate is fundamental to studying the fluid flow in an urban mesh. It can be divided into two regions: the Urban Boundary Layer (UBL) and the Urban Canopy Layer (UCL). Usually, the dispersion of gases in the macroscopic scale is governed by the turbulent flow in the UBL, while the UCL affects the surroundings of the obstacles. (KADAVERUGU et al., 2019). In Figure (2.3)) one can see the distinction of these two regions.

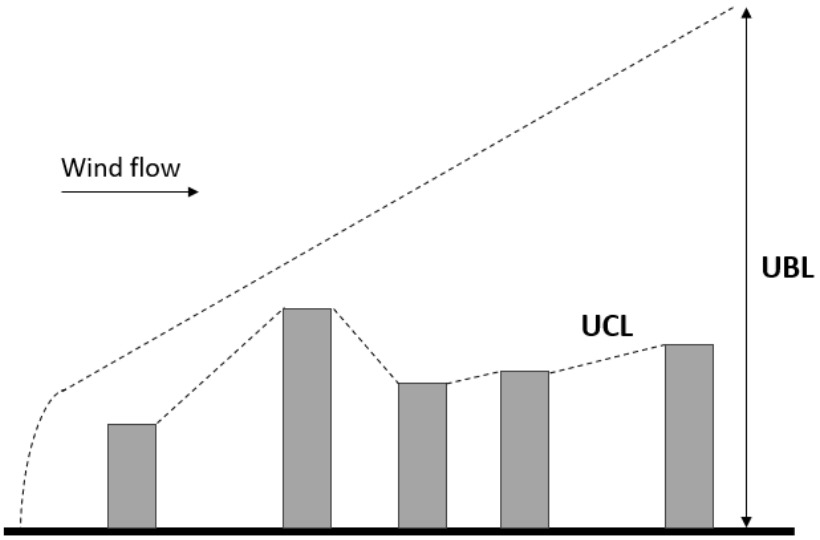

Figure 2.3: UBL and UCL in an urban environment with a longitudinal wind flow. Adapted from Hang et al. (2009).

In effect, the concept of urban micro-climate remains of utmost importance to predict the behavior of pollutant gases dispersed in an urban mesh, since it holds essential features to the flow, such as the dimensions and the form of the objects (buildings, roadways, vegetation, waterways, etc.) where the moment and heat transport phenomenons occur.

Through the CFD simulations used as example in this work, it will be observed how the form and the disposition of the objects in an urban mesh directly

influence the speed field and then the creation of accumulation zones of certain pollutants in the urban environment.

In short, the urban micro-climate significantly shapes how pollutants disperse. By understanding wind flow patterns, we can locate problematic areas and predict pollutant behavior. This knowledge is crucial for designing better urban structures and is often achieved through 3D modeling techniques that depict key urban features. In general, the dominant techniques are (Kadaverugu et al., 2019):

1. Photogrammetry;
2. Light Detection and Ranging (LiDAR) (for instance, using the OpenTopography tool available at http://opentopography.org/);
3. algorithm-based simulation.

Numerical simulation is a way to find solutions in huge data problems. It can, adding some boundary conditions (e.g.: some geometrical features due to the UCL) and initial values, predict the physical behavior of a body, a fluid for example. Numerical simulation is indeed a key tool for understanding urban turbulent flows, showing great importance to fields such as atmospheric physics, pollution dispersion and urban planning (XIAO et al., 2019).

The urban mesh is an environment where different geometries are put together in different arrangements and is thus a complex scenario to run numerical simulations, normally being very hardware time-consuming. To ease the amount of calculation, one can assume common geometries to try to create more relevant models but, in reality, an urban mesh can assume any random structural distribution. Either way, the notorious disposition known as street canyons may be the most suitable case of common geometry to analyze, as the structures are normally disposed of in an organized way, which affects the flow behavior. Indeed, in the case of a street canyon, one can have good insights into the flow field and thus the dispersion of pollutants in a populated urban mesh, highlighting the importance of this category of building disposition in a city.

2.2.3 CFD modelling

Few or almost no studies in Latin America, Africa or Asia use CFD according to Toparlar et al. (2017). Even in more developed European countries CFD models are rarely used (THUNIS et al., 2016).

The continents holding the greater part of the global urban population have only a few or zero studies using the CFD tool. For the European case, the Polytechnic University of Madrid (UPM) is currently developing a disruptive AI tool that could help authorities make important decisions regarding the control of atmospheric pollution in urban areas. The Marie Skłodowska-Curie project entitled "Groundbreaking tools and models to reduce air pollution in urban areas" or MODELAIR[2], a conjunct action between different countries has as its objective the development of an AI tool that allows the decision-making and sensitive data collection able to direct the actions of air pollution control in urban areas.

Globally, the interest in researching this subject arises to obtain a high-resolution simulation of a city map able to predict the dispersion of gases while consuming less time and computational resources.

To advance the approach advocated in this thesis, it is imperative to delve into the foundational principles of CFD modeling. This exploration is crucial for a comprehensive understanding of establishing a high-fidelity model, which, in turn, is essential for the subsequent development of a robust ML tool.

2.2.3.1 Fundamental Basis

From the models we focused on, the surrogate models are highlighted as those which have an objective to approximate the results of a desired measure that is not easily computed from a simple mathematical model. For example, the prediction of the behavior of the fluid dispersion in an urban mesh can be done from a Gaussian regression of a database created by high-fidelity CFD simulations, used to create these regressions in a convenient time while keeping the high fidelity/resolution (GANTI and KHARE, 2020).

Examples of surrogate models applied to CFD are the response surface method,

[2]More information available at modelair.eu.

the kriging method, polynomial regression, and ML. The last one is a very useful tool since it can accelerate the computation, improve the accuracy, reduce the complexity and enable new applications in CFD (WU and CHEN, 2023).

The surrogate model (or meta-model, or emulations model or simply emulator) is capable of speeding up the data processing done by the computer from the approximation of important points in the sample space. Figure (2.4) illustrates the basic processing of the model.

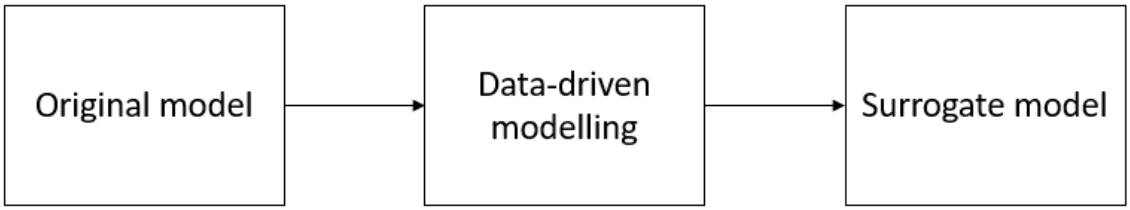

Figure 2.4: Schematic of a surrogate model. Adapted from Kocijan et al. (2022).

It is interesting to notice that this tool can be applied to different situations, from the basic study of a fluid flow around a cylinder to the analysis of gas flow in an urban mesh containing objects of complex geometry.

In this sense, the scientific effort has been to create a model capable of predicting the gas dispersion in a city to be able to dictate where are the high concentration zones using the ML model defined through the CFD-based data training.

The measure of air quality is generally done in monitoring stations associated with research institutes or great urban centers (MEAD et al., 2013) and, besides showing accurate results, they are incapable of treating an urban mesh of great extension in a detailed manner, because this would involve a high cost in constructing new stations. At this point, the creation of an AI tool capable of simulating with fidelity and high spatial-temporal resolution the dispersion of pollutant gases would have a fundamental role in adopting the mitigation measures in an efficient less costly way.

Nevertheless, this is a challenging endeavor due to the multifaceted influences on fluid flow within urban micro-climates. A thoughtful selection of boundary conditions and mesh is essential for accurate computational treatment and numerical simulations.

Furthermore, it is also necessary that one can be able of testing these hypo-

thetical measures that are to be done like: the effect of the plantation of vegetation in an avenue, and the optimization of urban traffic routes to reduce the accumulation of pollutants (KADAVERUGU et al., 2019). All of this adds up to the complexity of the tool yet to be developed, which is of extreme importance to the solution of different cases in a fast and intelligent way. For example, the understanding of the process of jet atomization of a liquid would demand a great computational effort in time and in hardware to produce a simulation of high fidelity that could capture with high-resolution the fluid dynamics in the liquid-gas interface, which makes the process impracticable in a matter of time and cost in the existing simulations, a challenge that could be solved with the implementation of a tool as discussed in this undergraduate thesis.

2.2.3.2 Fluid nature

The first step in predicting the behavior of the fluid in the urban mesh is defining this entity. Although this undergraduate thesis focuses on the behavior of gases the same laws of motion can be extended to liquids.

By definition, the fluid flow is due to the action of external forces into the fluid, surface forces (e.g. pressure, surface tension etc.) and body forces (e.g. gravity, centrifugal forces, electromagnetic forces, etc.).

The study of the velocity field of a fluid will be determinant to the study of pollutant dispersion. The two regimes (laminar and turbulent flow) depend on the Reynolds number, a non-dimensional parameter defined as the fraction between the incident fluid velocity u and a characteristic geometric feature D (e.g. the diameter for a cylinder) with the fluid cinematic viscosity ν:

$$Re = \frac{uD}{\nu}. \tag{2.2}$$

In the following definitions, we will be treating two different types of flow: compressible and incompressible. These are differentiated by the Mach number, which corresponds to the following relation:

$$Ma = \frac{flow\ speed}{speed\ of\ sound\ in\ the\ fluid}. \tag{2.3}$$

For $Ma < 0.3$ the flow is considered incompressible and compressible for $Ma \geq 0.3$.

Also, for the study of the governing equations to estimate pollutant dispersion we will consider only that the fluid flow is composed of Newtonian Fluid. This is important to note since the mathematical formulation of the equations would be affected if the fluid was non-Newtonian.

2.2.3.3 Governing equations in CFD

The focus here will be on developing an integral form for a finite control volume since this is the approach leading to important numerical methods.

The set of general equations which represent the fluid flow are described by the mass, momentum and energy conservation laws of physics. Be the fluid considered a continuum media, its behavior can be described in terms of macroscopic properties: velocity, density, pressure and temperature and its corresponding space and time derivatives. These properties can be calculated by taking a control volume (CV) of the fluid and by assuming that its properties are approximately the mean of the fluid properties. Figure (2.5) represents such CV as a cube of $\delta x \times \delta y \times \delta z$ dimensions according to the oriented coordinate axis $x, y \; and \; z$.

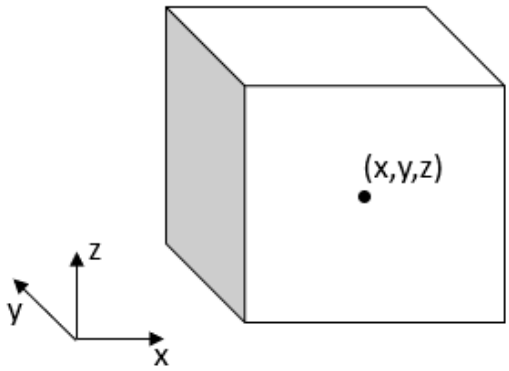

Figure 2.5: CV to determine the properties of a fluid. Adapted from Versteeg and Malalasekera (1995).

We will first develop the differential form of the equations concerning the principle of mass and momentum conservation of the fluid flow. After we will generalize the approach for any ϕ random property of the fluid since they share common features.

The unsteady, three-dimensional mass conservation equation in a compress-

ible fluid for a certain CV and velocity $\boldsymbol{u}$ can be expressed as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0. \tag{2.4}$$

For an incompressible fluid, the density ρ can be considered constant (as for a liquid) and thus the last equation simplifies to:

$$\nabla \cdot (\boldsymbol{u}) = 0, \tag{2.5}$$

or, adopting the Cartesian coordinates:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \tag{2.6}$$

where (u, v, w) corresponds to the Cartesian components (x, y, z) of the velocity.

On the other hand, the momentum conservation equation is derived from the Newton's second law of motion:

$$\frac{d(m\boldsymbol{u})}{dt} = \sum \boldsymbol{f}. \tag{2.7}$$

For a Newtonian fluid, a vector form of the momentum conservation equation is written as:

$$\frac{\partial(\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (p\boldsymbol{u}\boldsymbol{u}) = \nabla \cdot \boldsymbol{T} + \rho \boldsymbol{u}, \tag{2.8}$$

where $\boldsymbol{b}$ are the body forces (per unit mass). In the case of the fluid being Newtonian, $\boldsymbol{T}$ is the stress tensor expressed by:

$$\boldsymbol{T} = -\left(p + \frac{2}{3}\mu \; div \; \boldsymbol{u}\right) \boldsymbol{I} + 2\mu \boldsymbol{D}, \tag{2.9}$$

where p is the static pressure, $\boldsymbol{I}$ is the identity tensor, μ is the dynamic viscosity and $\boldsymbol{D}$ is the deformation tensor expressed as:

$$\boldsymbol{D} = \frac{1}{2}\left[grad \; \boldsymbol{u} + (grad \; \boldsymbol{u})^T\right]. \tag{2.10}$$

In Cartesian coordinates, Equations (2.9) and (2.10) can be noted as follows:

$$T_{ij} = -\left(p + \frac{2}{3}\mu \; \frac{\partial u_j}{\partial x_j}\right) \delta_{ij} + 2\mu D_{ij}, \tag{2.11}$$

$$D_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{2.12}$$

where δ_{ij} is the Kronecker delta function (definition of the Kronecker delta function can be easily found in any book in Mathematical Analysis).

More details on the development of the mass and momentum conservation equation can be found in Ferziger and Perić (2002) "Computational Methods for Fluid Dynamics" used as a source for this theoretical development.

Consider now a general variable ϕ representing any conserved intensive property so that its corresponding extensive property is :

$$\Phi = \int_{\Omega_{CM}} \rho\phi \, d\Omega, \tag{2.13}$$

where Ω_{CM} is the volume occupied by the control mass of the fluid (CM).

The conservative equations of the fluid flow can be expressed as the following equation, which is the starting point for computational procedures in the Finite Volume Method (FVM).

$$\frac{\partial(\rho\phi)}{\partial t} + div(\rho\phi\boldsymbol{u}) = div(\Gamma \, grad\phi) + S_\phi, \tag{2.14}$$

where Γ is the diffusion coefficient.

In essence, Equation (2.14) can be translated by: Rate of increase of ϕ of fluid element + Net rate of flow of ϕ out of fluid element + Rate of increase of ϕ due to diffusion = Rate of increase of ϕ due to sources.

This transport equation is of extreme importance to the FVM. This is a commonly used method in CFD that uses the developed integrated form of this transport equation, which finally gives Equation (2.15) for steady-state processes and Equation (2.16) for time-dependent processes.

$$\int_A \boldsymbol{n} \cdot (\rho\phi\boldsymbol{u}) \, dA = \int_A \boldsymbol{n} \cdot (\Gamma \; grad\phi) \, dA + \int_{CV} S_\phi \, dV, \tag{2.15}$$

$$\begin{aligned} \int_{\Delta t} \frac{\partial}{\partial t} \left(\int_{CV} (\rho\phi) \, dV \right) dt \; + \; \int_{\Delta t} \int_A \boldsymbol{n} \cdot (\rho\phi\boldsymbol{u}) \; dA \; dt = \\ \int_{\Delta t} \int_A \boldsymbol{n} \cdot (\Gamma_\phi \, grad\phi) \; dA \; dt \; + \; \int_{\Delta t} \int_{CV} S_\phi \; dV dt. \end{aligned} \tag{2.16}$$

Further details on the procedure of obtaining the aforementioned equations can be found in Versteeg and Malalasekera (1995), "An Introduction to Computational Fluid Dynamics", on which this part of the thesis is based. It gives a detailed explanation of the process of deriving the transport equation as well as in the FVM.

This last method is commonly used in commercial software. For example, the well-known software ANSYS Fluent® makes use of the finite volume method to solve the governing equations of the fluid flow. In this Section, we tried to expose

just a glimpse of these equations to introduce the reader to what would be used in the core method of problem-solving in the CFD software.

2.2.3.4 Finite Volume Method

FVM is a common method used in CFD simulation programs and it was the one chosen to be summarized in this Section. Other methods like Finite Difference are also relevant and can be easily found in the literature.

The integral form of the conservation equation to a generic quantity ϕ is:

$$\int_S \rho\phi \boldsymbol{v} \cdot \boldsymbol{n}\, dS = \int_S \Gamma\, grad\phi \cdot \boldsymbol{n}\, dS + \int_\Omega q_\phi\, d\Omega, \tag{2.17}$$

which applies to each CV as well as to the whole domain.

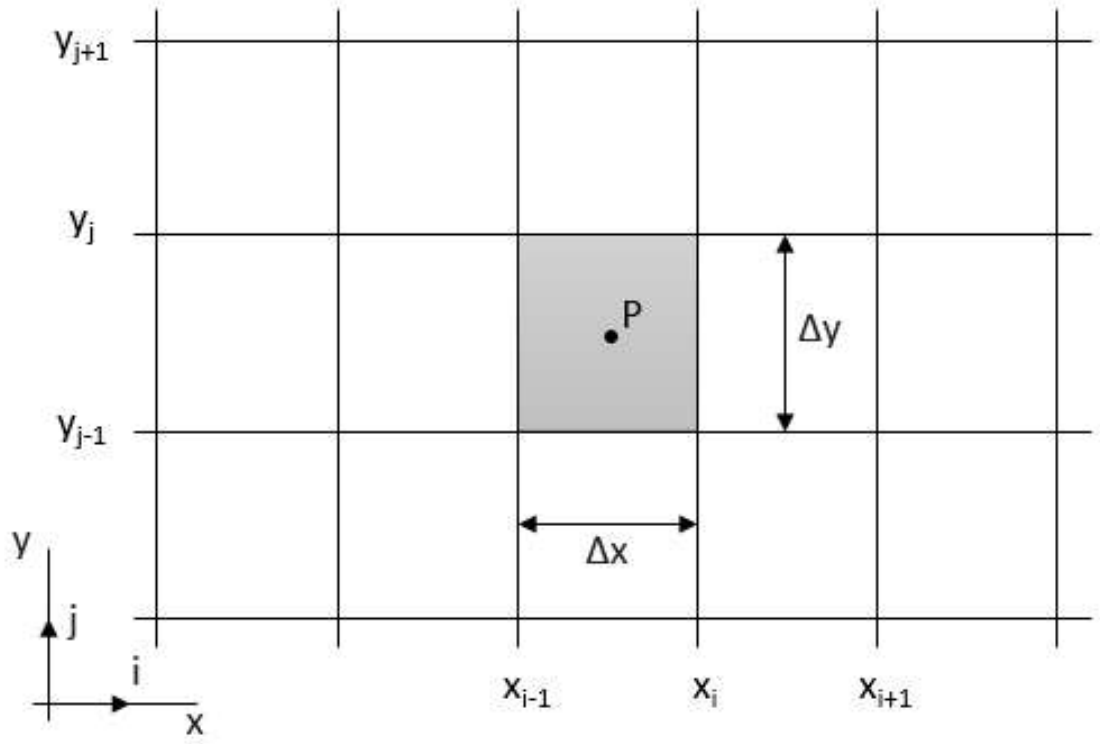

Figure 2.6: Cartesian 2D grid representation of the CV adopted. Adapted from Ferziger and Perić (2002).

To generate an approximate form of the surface and volume integrals to each CV in the domain quadrature formulae are used. We will approximate these integrals for the Cartesian 2D grid but the reasoning can be extended to the Cartesian 3D grid since 2D is only a special case of 3D. To approximate the surface integrals, we

will define the net flux through the CV boundary as follows:

$$\int_S f \, dS = \sum_k \int_{S_k} f \, dS. \tag{2.18}$$

The component f in this notation can be both due to the convective flux $f^c = \rho\phi\boldsymbol{v} \cdot \boldsymbol{n}$ or to the diffusive flux $f^d = \Gamma grad\phi \cdot \boldsymbol{n}$. In the summation in Equation (2.18) we will approximate only for the eastern surface, the others can be calculated similarly. Second-order approximations suitable to this case are the midpoint rule:

$$F_e = \int_{S_e} f \, dS \approx f_e S_e, \tag{2.19}$$

and trapezoid rule:

$$F_e = \int_{S_e} f \, dS \approx \frac{S_e}{2}(f_{ne} + f_{se}), \tag{2.20}$$

where f_e is the value of f at the eastern cell-face center, f_{ne} and f_{se} at the northeast and southeast CV corners. S_e stands for the area of the eastern face. A fourth-order approximation is given by Simpson's rule:

$$F_e = \int_{S_e} f \, dS \approx \frac{S_e}{6}(f_{ne} + 4f_e + f_{se}). \tag{2.21}$$

For approximating the volume integral one can use the second-order accurate approximation:

$$Q_P = \int_\Omega q \, d\Omega \approx q_P \Delta\Omega, \tag{2.22}$$

where q_P is the value of q at the CV center as in Figure (2.6) but in the respective 3D representation and $\Delta\Omega$ is the CV volume. A fourth-order accurate approximation can be done as follows:

$$Q_P = \int_\Omega q \, d\Omega \approx \Delta x \Delta y \left[a_0 + \frac{a_3}{12}(\Delta x)^2 + \frac{a_4}{12}(\Delta y)^2 \frac{a_8}{144}(\Delta x)^2(\Delta y)^2 \right], \tag{2.23}$$

where $q(x, y)$ is the bi-quadratic shape function:

$$q(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy + a_6 x^2 y + a_7 x y^2 + a_8 x^2 y^2. \tag{2.24}$$

To obtain q in different locations, one must make use of interpolation techniques, as we already needed for approximating surface integrals. For the development of common interpolation and differentiation practices, one can consult the work "Computational Methods for Fluid Dynamics" of Ferziger and Peric (2002), which served as a base for this subsection. Some of the techniques discussed are Upwind Interpolation (UDS), Linear Interpolation (CDS) and Quadratic Upwind Interpolation (QUICK), for example.

2.2.3.5 Turbulence models

Beyond the CFD domain, the analysis of wind flow in urban clusters uniting the conjunct of buildings is realized through two methods: wind tunnel experiments and field analysis. However, as both methods are expensive and complicated, CFD is chosen to facilitate the analysis.

An efficient CFD model with a good approximation depends on the choice of the turbulence model adapted to the context to describe optimally the physics of the flux field. Liu et al. (2016) evaluated the performance of three types while simulating the wind flow around an isolated building (with a 1:1:2 shape):

1. Reynolds Averaged Navier-Stokes (RANS), especially Steady-RANS (SRANS) using models based in the $\kappa - \epsilon$ family;
2. Large Eddy Simulation (LES);
3. Detached Eddy Simulation (DES), which is a hybrid model between URANS (Unsteady Reynolds Averaged Navier-Stokes) and LES.

According to Ferziger and Perić (2002), in engineering applications, it is sufficient to use approximated models that average the unsteadiness of the turbulence, such as the RANS model. For an incompressible flow without body forces, the averaged continuity and momentum equations are:

$$\frac{\partial(\rho\overline{u}_i)}{\partial x_i} = 0\,, \tag{2.25}$$

$$\frac{\partial(\rho\overline{u}_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho\overline{u}_i\overline{u}_j + \rho\overline{u_i'u_j'}) = -\frac{\partial\overline{p}}{\partial x_i} + \frac{\partial\overline{\tau}_{ij}}{dx_j}\,, \tag{2.26}$$

where $\overline{\tau}_{ij}$ are the mean viscous stress tensor components:

$$\overline{\tau}_{ij} = \mu\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i}\right)\,. \tag{2.27}$$

2.2.3.6 RNG $\kappa - \epsilon$ model applied to pollutant dispersion

A recent study by Wu and Chen (2023) can be consulted as a case example of the application of the RNG $\kappa - \epsilon$ model especially to pollutant dispersion. For

a CFD computational domain of buildings of different heights, encircled by other buildings, a normalized pollutant concentration has the following definition:

$$\overline{c}^* = \overline{c}\frac{U_H H^2}{\dot{m}}\,, \tag{2.28}$$

where $\overline{c}^*$ is the dimensionless concentration, $\overline{c}$ is the pollutant concentration, H is the building height, U_H is the reference speed at H and $\dot{m}$ is the source emission rate of pollutants.

In this case, the velocity profile (U), turbulent kinetic energy profile (k) and turbulent dissipation rate profile (ϵ) are defined as:

$$U(z) = \frac{U_*}{\kappa} ln\left(\frac{z+z_0}{z_0}\right)\,, \tag{2.29}$$

$$k(z) = \frac{U_*^2}{\sqrt{C_\mu}}\,, \tag{2.30}$$

$$\epsilon(z) = \frac{U_*^3}{\kappa(z+z_0)}\,, \tag{2.31}$$

where z is the height coordinate, z_0 is the aerodynamic roughness lenght, κ is the von Karman constant, U_* is the atmosferic boundary layer friction velocity and C_μ also a constant.

The solution to the pollutant concentration $\overline{c}$ is the solution of a steady-state time-averaged convection diffusion equation defined as:

$$\frac{\partial(\rho\overline{u}_i\overline{c})}{\partial x_i} = \frac{\partial}{\partial x_i}\left(K\,\frac{\partial\overline{c}}{\partial x_i}\right) + S\,, \tag{2.32}$$

where S is the pollutant source term and K is the mass diffusion coefficient for the concentration defined as:

$$K = D + D_t\,, \tag{2.33}$$

where D is the molecular diffusivity and D_t is the eddy diffusivity defined as:

$$D_t = \frac{\nu_t}{Sc_t}\,, \tag{2.34}$$

where Sc_t is the turbulent Schmidt number, a non-dimensional number ranging from 0.2 to 1.3 in the RANS model.

2.2.3.7 Flow around a cylinder

The simulation of the flow around a circular cylinder uses the numerical simulation based on the solution of the Navier-Stokes equations as explained before.

Beyond the conservation equations, the incompressible flow around a cylinder is governed by the Re number. Indeed, the behavior of the fluid around a cylinder transits in-between different phases, corresponding to different intervals of Re number, as illustrated in Table (2.1) based on Blevins (1990):

Table 2.1: The behavior of a fluid around a cylinder for different values of Reynolds number. Adapted from Blevins (1990) and comments by Buk Júnior (2007).

Case	Reynolds number interval	Description
1	$Re < 5$	Regime of unseparated flow
2	$5 \leq Re \leq 40$	A fixed pair of vortices in wake
3	$40 \leq Re \leq 90$	Regime in which vortex street is laminar
3	$90 \leq Re \leq 150$	Regime in which vortex street is laminar
4	$150 \leq Re \leq 300$	Transition range to turbulence in vortex
4	$300 \leq Re \leq 3.5 \times 10^5$	Vortex street is fully turbulent
5	$3.5 \times 10^5 \leq Re \leq 3.5 \times 10^6$	Laminar boundary layer has undergone turbulent transition and wake is narrower and disorganized
6	$Re \geq 3.5 \times 10^6$	Re-establishment of turbulent vortex street

Note that the methodology of Ganti and Khare (2020) showed positive results for the laminar flow of fluid around a circular cylinder. However, the more complex the studied mesh the more the computational cost and thus the total processing time. The total time that Ganti and Khare (2020) obtained for the jet atomizing case using 32 cores was 33h.

2.3 Dimensionality reduction using POD - Step 2

In the paper written by Ganti and Khare (2020), the second stage corresponds to the dimensional reduction executed using Proper Orthogonal Decomposition (POD), an Intrusive Reduced-Order Model (IROM).

2.3.1 Reduced Order Modelling (ROM)

A flow under a turbulent regime is composed of eddies that can be analyzed to give the most energy-containing part of the fluid behavior using singular value decomposition. The use of only a reduced group of deterministic functions is a way to reduce the computation cost without losing the most important figures of the flow.

This is an efficient way of avoiding long and costing computational calculations and yet correctly approximating the behavior of flow dynamics by considering only its most energy-containing functions. Coherent structures are the regions of flow containing these most important features. As a turbulent flow is composed of eddies with different length scales, considering the ranges with the most of the energy allows us to predict the whole ranges with a good margin. Those ranges are subdivided in energy-containing (permanent) sub-range, inertial sub-range and dissipation sub-range (MASOUMI-VERKI, 2022). Proper Orthogonal Decomposition (POD), the first method developed using this reasoning, was proposed by J.L. Lumley in 1967, entitled "The structure of inhomogeneous turbulent flows."

Reduced-order models (ROMs) reduce the degrees of freedom (DOF) of a complex while maintaining the most energetic modes which contain the most important aspects of the dynamic system (MASOUMI-VERKI, 2022).

Two categories of ROM exist: Intrusive Reduced-Order Model (IROM) and Non-Intrusive Reduced-Order Model (NIROM).

An IROM consists of projecting the governing equations in a reduced-size, physics-based model, normally combined with solvers based on the finite element method. The methods of projecting the system's equation are most commonly the POD technique, Dynamic Mode Decomposition (DMD), greedy algorithms, gappy POD method and discrete empirical interpolation, with the POD-Galerkin method

being the most used approach, although showing some issues related to stability and nonlinear inefficiency, problems for which the implementation of other methods like Petrov-Galerkin projection or discrete empirical interpolation are suggested (MASOUMI-VERKI, 2022).

NIROMs appears as an alternative for solving those issues since it does not require any source code manipulation that might not be available due to patent issues in the IROMs' case, although they have high computational costs during the training stage. However, it is important to note that NIROMs also show some problems since it is based on data and thus could lack physical reliability (MASOUMI-VERKI, 2022).

Table 2.2: Reduction in computing time in recent studies due to the use of ROMs compared to high-fidelity models. Adapted from Masoumi-Verki (2022).

Year	Authors	Reduction factor in computational time
2021	Xiang et al.[66]	226
2021	Xiang et al.[65]	800
2019	Xiao et al.[69]	approx. 10^5

Either way, NIROMs show much better performance when compared to raw CFD models, since they reduce a large amount of the computational time. Table (2.2) resumes some cases present in the literature highlighting this improvement in hardware time.

Since POD remains the dominant technique in the linear dimensionality reduction technique, the next subsection will be an overlook of its theoretical basis.

2.3.2 Proper Orthogonal Decomposition (POD)

The method starts by generating a matrix composed of the velocity field at different time steps:

$$S = [\mathbf{u}^1\mathbf{u}^2\mathbf{u}^3...\mathbf{u}^{N_s}]. \tag{2.35}$$

This matrix is called the snapshots matrix and is rather an advantageous approach compared to independent velocity vectors taken separately. Indeed, taking the velocities of the high-fidelity model at different times and unifying them into

a matrix is said to capture the physical correlations that emerge from the set of velocities, at the same time that produces less data to be trained later on by a neural network (XIAO et al., 2019).

For each n time-step, for a velocity field containing N nodes, each velocity vector in the 3D space can be represented by the following column matrix:

$$\mathbf{u}^n = [u_1^n, u_2^n, ..., u_N^n, v_1^n, v_2^n, ..., v_N^n, w_1^n, w_2^n, ..., w_N^n]^T. \tag{2.36}$$

The POD basis functions are obtained by the Singular Value Decomposition (SVD) of the S matrix:

$$S = U\Sigma V^T. \tag{2.37}$$

The matrix $V \in R^{3N_s \times 3N_s}$ corresponds to the time dynamics of its corresponding components in the $U \in R^{3N \times 3N}$ matrix. The matrix $V \in \Sigma^{3N \times N_s}$ is diagonal and corresponds to an orthogonal set of vectors in which data is incorporated. The values in the diagonal of the matrix represent the singular values obtained by the SVD in order of decreasing magnitude.

Finally, the first N_s columns of U obtained from this operation are the POD basis functions. It is thus obtained a truncated S matrix S_m, known as the closest approximation to the snapshot matrix S in the Frobenius norm. The restriction in the number of basis functions (i.e. taking those functions that correspond to the major part of the energy spectrum) is defined depending on the tolerance η stipulated, so that we can reduce the number of these basis functions to a certain number r of non-zero functions:

$$\frac{\sum_{j=1}^{r} {\sigma_j}^2}{\sum_{j=1}^{N_s} {\sigma_j}^2} \geq \eta. \tag{2.38}$$

The left-side in the previous equation represents the amount of the system's total energy accumulated in the first r POD basis functions σ_j. Further insights can be found in Masoumi-Verki (2022) from which this part is based on.

The problem that arises from this approach is its linearity. Fluid dynamics are normally composed of non-linear features, especially in an urban environment. These features can be difficult to describe with simple POD since this technique computes the optimal linear space in which data will be projected. What happens is that even in simple urban configurations, due to dynamic boundary conditions

and non-linear and advection-dominated flow behavior, the simple POD will not give efficiently and accurately a good approximation of the energy content (MASOUMI-VERKI, 2022). What happens is too many POD modes to capture the major energy content of the flow, which does not optimize the process as one would look to do while applying the POD technique. This is the point when normally one would try to look for a non-linear ROM technique (i.e. NIROM).

2.3.3 Non-linear dimensionality reduction methods

Figure (2.7) illustrates the problem that arises in cases in which data is far from being linear. Principal Component Analysis (PCA), a synonym of POD, would not be able to give a good approximation of the correlated data. Note that a POD will not precisely account for the non-linearity of the data set. In this case, the use of autoencoders solves the problem since it can handle the non-linearity of the system by using nonlinear activation functions.

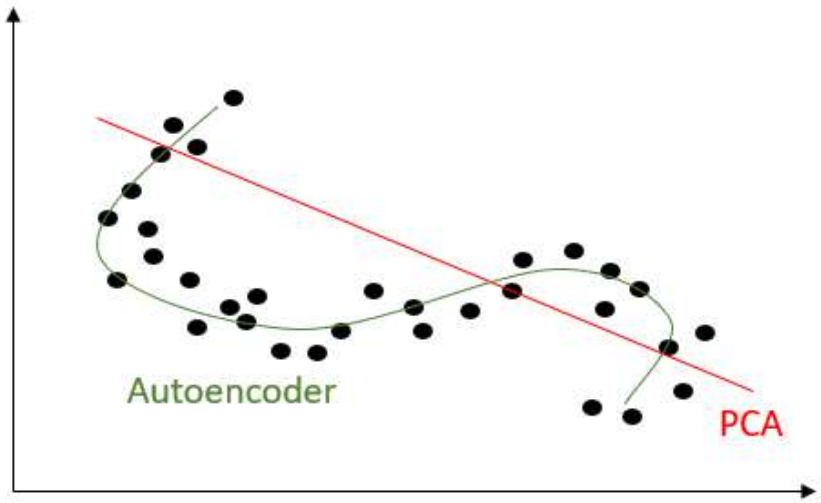

Figure 2.7: The problem of linearity of a data set in a ROM. Adapted from Masoumi-Verki (2022).

In resume, an autoencoder works as a "passage". First, an encoder takes the data from the original data set and transforms it into a dimensional-reduced space called the latent space which will be later approximated again to the original space by the use of a decoder.

The input state $\mathbf{x}$ is transformed by a function f_E into a vector of the latent

space $\mathbf{h}$ using a parameter θ_E:

$$\mathbf{h} = f_E(\mathbf{x}; \theta_E). \tag{2.39}$$

Later on this vector is reconstructed into $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = f_D(\mathbf{h}; \theta_D). \tag{2.40}$$

These parameters are obtained by the minimization of the deviation $\mathcal{L}(\hat{\mathbf{x}}, \mathbf{x})$ of the reconstructed input state $\hat{\mathbf{x}}$:

$$\theta_E{}^*, \theta_D{}^* = \arg\min_{\theta_E, \theta_D} \mathbb{E}_{x \sim P_{data}}[\mathcal{L}(\hat{\mathbf{x}}, \mathbf{x})], \tag{2.41}$$

where the deviation has the following definition:

$$\mathcal{L}(\hat{\mathbf{x}}, \mathbf{x}) = ||\hat{\mathbf{x}}, \mathbf{x}||_2^2. \tag{2.42}$$

One must note that $\mathbf{h} \in R^{N_h}$ is of lower dimension compared to $\hat{\mathbf{x}} \in R^N$, which is expected since the method proposes a reduction in the input data space. Further development of this non-linear method is beyond the scope of this thesis but one can easily find more information about the most common non-linear dimension reduction method called kernel PCA in Vidal et al. (2003).

2.3.4 NIROMs

We take this opportunity to introduce the general aspects of ML applied to our case using as an example the construction of a Non-Intrusive Reduced Order Model for 3D flows in the urban environment based on the work of Xiao et al. (2019) in the paper called "A reduced order model for turbulent flows in the urban environment using machine learning".

It is composed of two steps: Off-line and On-line NIROM. The Offline stage consists of simply finding the POD basis functions and then training a neural network to predict the behavior of the governing equations. Specifically, it consists of, for each POD basis function, producing a function f_j that maps the set of POD coefficients[3] from time level $\boldsymbol{\alpha}^{k-1}$ to the corresponding POD coefficient at the next time level ${\alpha_j}^{k-1}$, as in the following equation:

$${\alpha_j}^k = f_j(\boldsymbol{\alpha}^{k-1}) = f_j(\alpha_1^{k-1}, \alpha_2^{k-1}, ..., \alpha_r^{k-1}), \qquad \forall\, k \in (1, 2, ..., N_s). \tag{2.43}$$

[3]Refer to Section 7.2 in the Appendix for more details on this concept.

To treat all POD basis functions, this procedure is repeated for r times, completing the off-line stage. The next step, the Online stage, consists of running the NIROM as many times as we desire allowing us to predict the POD coefficients at any time within a time step equal to the one in the high-fidelity model:

$$\alpha_j(t + \Delta t) = f_j(\boldsymbol{\alpha}(t)), \qquad \forall j \in (1, 2, ..., r). \tag{2.44}$$

2.4 Machine learning algorithm trained with training dataset - Step 3

The third stage consists of training the ML algorithm through kernel selection and Gaussian Process Regression (GPR). The algorithm trained can thus predict the dynamics of the system for any operating condition within the training bounds and this result is emulated using Galerkin reconstruction (GANTI and KHARE, 2020).

In this third step, the ML occurs through Gaussian Process modelling of the data acquired from the truth model. In order to make GP work correctly, it is interesting that enough points are added and that the kernel functions are chosen adequately, reflecting correctly the dynamics of the process. After the GP, it is of good practice to compare the results obtained with the ones from another model to estimate if the outlets are actually of good approximation.

2.4.1 The importance of ML in forecasting events

Consider an emergency where a toxic gas is emitted from a random source. The speed with which a response is given to a critical situation like this dictates how bad the consequences will be. Having at hand powerful ML tool is game-changing for any decision-making situation.

Indeed, compared to a single CFD-based algorithm, an Artificial Neural Network (ANN) presents superior qualities due to its better efficiency either in time or in less computational resources (MENDIL, 2022). An interesting review of recommendations on how to build ANNs to forecast long-term pollution can be found in Cabaneros et al. (2019) and for short-term in Lauret et al. (2016).

There exist numerous learning algorithms but overall they can be classified into three different categories: supervised, unsupervised, and semi-supervised. The difference between the three categories is based on the degree to which external supervisory information is available.

2.4.2 Mathematical basis

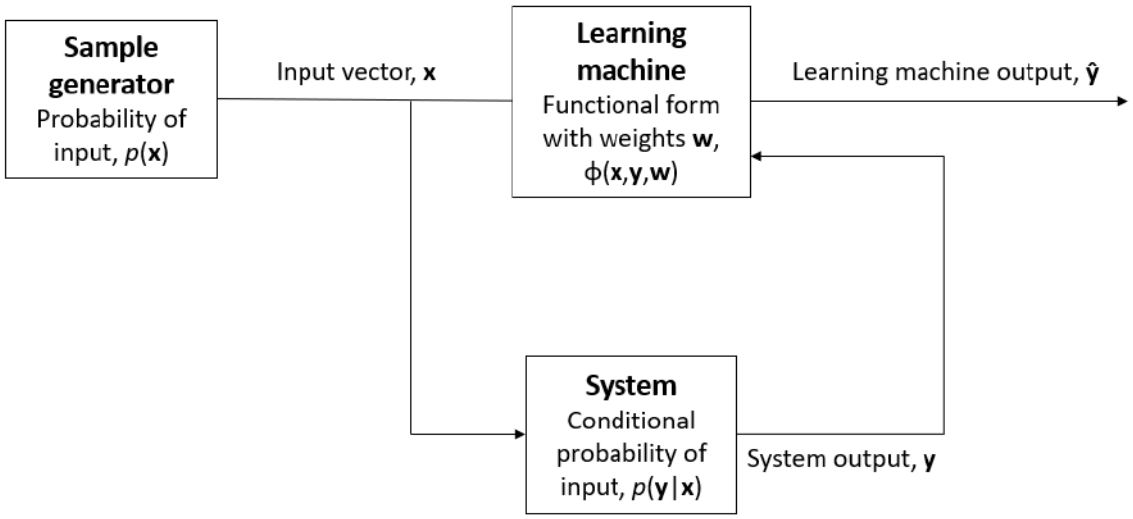

Figure 2.8: The general process of a learning algorithm. Adapted from Brunton et al. (2020).

Figure (2.8) illustrates what is called the learning process, or learning problem. First, the input data is generated from a sample generator of random vectors $\boldsymbol{x}$ from an unknown distribution $P(\boldsymbol{x})$. The input vector $\boldsymbol{x}$ is treated by a supervisor system that gives an output $\boldsymbol{y}$ depending on the condition probability of the input $P(y \mid x)$, also unknown. A learning machine implement a set of functions $f(x, w)$, or $\phi(x, y, w)$, with $w \in W$.

Finally, the learning problem is resumed to a problem of function approximation where one searches for the function in the set implemented that better approximates the supervisor's response (VAPNIK, 1991).

Those approximations are fundamentally stochastic and can be translated into the minimization of a risk function:

$$R(w) = \int L(y, f(x, w))\, dP(x, y), \tag{2.45}$$

where $L(y, f(x, w))$ is the loss or discrepancy between the response y of the supervisor to a given input x and the response $f(x, w)$ provided by the learning machine.

At this stage, as the joint probability distribution $P(x, y) = P(y \mid x)P(x)$ is unknown, the solution to the problem of minimization is also unknown and the only information available is the training set of l independent observations:

$$(x_1, y_1),\ ...,\ (x_l, y_l). \tag{2.46}$$

An alternative to the solution is the so-called induction principle of empirical risk minimization (ERM), which evaluates the uniform convergence and its respective rate of the empirical risk functional $E(w)$ instead of $R(w)$. The uniform convergence of $E(w)$ to $R(w)$ is a necessary and sufficient condition for the consistency of the principle (VAPNIK, 1991).

$$E(w) = \frac{1}{l}\sum_{i=1}^{l} L(y_i, f(x_i, w)). \tag{2.47}$$

The uniform convergence of the empirical function $E(w)$ to the risk functional $R(w)$ over the full set $f(x, w)$, $w \in W$, is defined as:

$$Prob\{sup_{w\in W} \mid R(w) - E(w) \mid > \epsilon\} \to 0 \quad as \quad l \to \infty. \tag{2.48}$$

To evaluate the rate of convergence, the concept of VC-dimension of the set of functions is necessary, but it won't be detailed here. Details on the notion of VC-dimension can be found in the literature.

The bounds for the real case learning with a uniformly good approximation to small $P(w)$ considering:

$$Prob\{sup_{w\in W}\frac{P(w) - \nu(w)}{\sigma(w)} \ > \ \epsilon\}, \tag{2.49}$$

would be approximating for $P(w) << 1$, $\sigma(w) \simeq \sqrt{P(w)}$, which gives the following inequality:

$$Prob\{sup_{w\in W}\frac{P(w) - \nu(w)}{\sqrt{P(w)}} \ > \ \epsilon\} \ < \ (\frac{2le}{h})^h exp\{-\frac{\epsilon^2 l}{4}\}, \tag{2.50}$$

where $\nu(w)$ stands for the empirical risk here.

Then, with probability $1 - \eta$, for all $w \in W$:

$$P(w) \ < \ \nu(w) + C_1(l/h, \nu(w), \eta), \tag{2.51}$$

with confidence interval:

$$C_1(l/h, \nu(w), \eta) = 2\left(\frac{h(ln\ 2l/h + 1 - ln\ \eta}{l}\right)\left(1 + \sqrt{1 + \frac{\nu(w)l}{h(ln\ 2l/h + 1 - ln\ \eta}}\right). \tag{2.52}$$

This mathematical development is based on the work of Vapnik (1991) which can be consulted for further details. For example, it explores the case when l/h is small and a new principle has to be considered for a good approximation: the Structural Risk Minimization (SRM). Also, it treats the problem of local function estimation for both ERM and SRM, knowing that sometimes the set of functions $f(x, w)$, $w \in W$ might only predict well-specified regions of the input space and not necessarily its entirety.

2.4.3 Gaussian Process

The growth of the population is difficult to control and normally gives place to highly populated areas, creating congested and weak air flows that concentrate the pollutants in certain areas (e.g. long street canyons). To assess the dispersal of gases, we already introduced the concept of surrogate models. Although surrogate models are usually combined with DoE methods to improve their reliability, they are mostly chosen for their instant results, which made them gain popularity in many engineering applications. Among those, there are the response surface (RS) model, support vector regression, artificial neural network (ANN), radial basis functions, and Gaussian process (GP) regression.

GPR is a supervised learning method that consists of predicting values based on knowledge incorporated in prior, providing uncertainty measurement over these predictions (WANG, 2020). It involves knowledge of Univariate Normal Distribution, Multivariate Normal Distribution (MVN), kernels, non-parametric modeling, and joint and conditional probability. These concepts will be briefly explained in the following Sections which are based on the work of Wang (2020) and in the majority can be further explored in advanced statistics books.

2.4.3.1 Univariate normal distribution

A random variable X follows a normal distribution if for a mean μ and variance σ^2 it can be represented by a Probability Density Function (PDF) of the form:

$$P_X(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{2.53}$$

The characteristic function for $X \sim N(\mu, \sigma^2)$ has the form:

$$\phi(t) = exp(it\mu - \frac{1}{2}\sigma^2 t^2). \tag{2.54}$$

In this equation, i is the imaginary unit, and $t \in R$ is the frequency parameter of the Fourier transform. In a random normal distribution $\mu = E[X]$ is the mean, and $\sigma^2 = Var(X)$ is the variance. The special case $X \sim N(0, 1)$ corresponds to the standard normal distribution.

2.4.3.2 Multivariate Normal Distribution

In most problems in engineering, we deal with multidimensional parameters and functions. Extending our discussion in Gaussian distribution we now have a D-dimension vector $\boldsymbol{x} = (x_1, x_2, ..., x_D)$. If $\boldsymbol{X} = (X_1, X_2, ..., X_D)$ is a normal random variable, its PDF can be represented as:

$$N(\boldsymbol{x}|\boldsymbol{\mu}, \bar{\bar{\boldsymbol{\Sigma}}}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]. \tag{2.55}$$

As in the univariate case, $\boldsymbol{\mu} = E[\boldsymbol{x}] \in R^D$ is the mean vector, and $\boldsymbol{\Sigma} = cov[\boldsymbol{x}] \in R^D \times R^D$ is the $D \times D$ covariance matrix, with components $\sigma_{ij} = cov(x_i, x_j)$. [4]. The corresponding characteristic function can be written as:

$$\phi(t) = exp(it \cdot \boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\Sigma t} \cdot \boldsymbol{t}). \tag{2.56}$$

Defining the characteristic function for either the uni- and the multivariate case is useful since the distribution of a D-dimensional random variable $\boldsymbol{X}$ is determined uniquely by its characteristic function $\phi(\boldsymbol{t})$. For more development in the uni- and multivariate normal distribution theory in statistics one can consult Bryc (1995).

[4] $|\boldsymbol{\Sigma}|$ represents the matrix product $\boldsymbol{\Sigma} \times \boldsymbol{\Sigma}$

2.4.3.3 Kernel function

A function $k(x, x')$ is called a kernel function when it is solely defined in terms of inner products in the input space (RASMUSSEN, 2006). The most used kernel function for Gaussian processes is called the squared exponential (SE) kernel function, also known as radial basis function (RBF) or simply Gaussian kernel function, represented as:

$$cov(x_i, x_j) = exp\left(-\frac{(x_i - x_j)^2}{2}\right). \tag{2.57}$$

Other functions include exponential kernel function and rational quadratic, for instance. Kernel functions can be understood as assets for acquiring better predictions.

2.4.3.4 Non-parametric modelling

GPR is a non-parametric model, which means that it is described by infinity parameters θ. On the other hand, parametric models, as polynomial regressions have finite parameters:

$$y = \theta_1 + \theta_2 x + ... + \theta_{n+1} x^n. \tag{2.58}$$

Either way, for both cases given a data set $D = [(x_i, y_i)|i = 1, 2, ..., n]$ with n observed points, the model can be trained so that after the training process the predictions f become independent of the observed data set and rely only on the parameters so that new measurements of unobserved data (*) can be done. This can be represented by the following equation:

$$P(f_*|X_*, \theta, D) = P(f_*|X_*, \theta), \tag{2.59}$$

where $P(x|y)$ stands for the conditional probability of an event x to occur when the event y occurs.

2.4.3.5 Gaussian process model

Given the observed data points $\boldsymbol{X} = [x_1, ..., x_n]$, the mean function $\boldsymbol{\mu} = [m(x_1), ..., m(x_n)]$ and $K_{ij} = k(x_i, x_j)$, where k is a positive definite kernel function, the regression function $\boldsymbol{f} = [f(x_1), ..., f(x_n)]$ modeled by a MVN is given by:

$$P(\boldsymbol{f}, \boldsymbol{X}) = N(\boldsymbol{f}|\boldsymbol{\mu}, \boldsymbol{K}). \tag{2.60}$$

GPR is a distribution over functions shaped by $\boldsymbol{K}$, which means that the unobserved data points will also contribute to the future regression function $\boldsymbol{f}_*$. This new distribution depends on the conditional probability $P(\boldsymbol{f}_*|\boldsymbol{f}, \boldsymbol{X}, \boldsymbol{X}_*)$, represented by:

$$\boldsymbol{f}_*|\boldsymbol{f}, \boldsymbol{X}, \boldsymbol{X}_* \sim N(\boldsymbol{K}_*^T \boldsymbol{K} \boldsymbol{f}, \boldsymbol{K}_{**} - \boldsymbol{K}_*^T \boldsymbol{K}^{-1} \boldsymbol{K}_*), \tag{2.61}$$

where $\boldsymbol{K} = K(\boldsymbol{X}, \boldsymbol{X})$, $\boldsymbol{K}_* = K(\boldsymbol{X}, \boldsymbol{X}_*)$ and $\boldsymbol{K}_{**} = K(\boldsymbol{X}_*, \boldsymbol{X}_*)$.

2.5 Error estimation - Step 4

The reconstructed spatial-temporal flow field obtained from the trained ML algorithm must be compared to the corresponding truth model (CFD simulation) at some test points to determine how reliable and accurate the emulation framework is in its entirety. This comparison is done by calculating the L1 error norm between both cases.

L1 error norm can be defined as:

$$error = \frac{1}{n} \sum_{i}^{n} \left(\left| \frac{truth - emulation}{truth} \right| \right)_i . \tag{2.62}$$

Chapter 3

Methodology

The methodology proposed by this undergraduate thesis consists in highlighting the most modern and relevant studies in the two major areas of this review (CFD and ML), combining these two techniques to analyze pollutant dispersion/air quality in an urban mesh. For this first part, the methodology will leverage the research functionalities provided by state-of-the-art research tools. The aim is to curate a collection of recent studies on the emulation framework, encompassing various articles within a specified time frame. These studies will be evaluated based on their relevance through citations and the temporal evolution of publications, presented through organized tables and graphical representations. This approach is intended to deeply understand each facet by harnessing the data from recent papers, thereby paving the way for the thorough development of a novel method.

Also, in the previous Sections, we explored the basic concepts behind CFD and ML. Now, in a more practical approach, it will be performed two simple simulations to illustrate the applicability of all that theory and how the results can actually be of high practical value, especially to determine the pollutant dispersion. The two practical simulations to be explored are:

1. Flow around a circular cylinder;
2. Flow in a random urban mesh.

The whole methodology is divided into three blocks:

1. First, we briefly show the evolution of the publications in ML in the last almost 10 years, as well as the evolution of publications in CFD. This general analysis

is presented as a way to highlight how prominent these two areas are and thus how powerful would it be to combine them to further explore the subject of pollutant dispersion in urban mesh, which we will show is yet in a premature stage;

2. In the following Section, we use an advanced search method through Scopus® [1] to precisely identify relevant publications. Analyzing keywords and platform-generated data offers a comprehensive view of the scientific landscape, aiding further research. A graphic visualization of the specific results obtained in this block is performed using Litmap®;

3. The last block corresponds to the simulations done in the openSource CFD software OpenFOAM® v-2012. The numerical results of the simulations are post-processed through a visualization tool also openSource called ParaView® (version 5.11.1). A Linux Ubuntu subsystem is used as the interface terminal for the realization of the numerical simulation.

3.1 Using Dimensions search platform - Block 1

This first simple search consists of consulting the Dimensions® database to analyze the evolution of publications in CFD and ML. Dimensions® is an accessible online platform and the world's largest linked research database with more than 138 million publications background. The search consists of obtaining general graphs both from the number of publications in the area of ML and CFD (with a focus on air pollution) separately from 2015 and onward. Results of this search using the input "CFD AND (POLLUTANT DISPERSION OR AIR QUALITY)" in the categories "Title and abstract" are shown in Section 4.1.1 starting from 2015. In Section 4.1.2, the results correspond to the term "MACHINE LEARNING" searched in "Title and Abstract" in Dimension®'s database also starting from 2015.

[1] Visit www.scopus.com for more details.

3.2 Advanced search using Scopus® - Block 2

An advanced search in the Scopus® platform by Elsevier® using Boolean operators (e.g. AND, OR and NOT), as well as the specific commands of the tool is an easy way of concentrating data to the most recent studies with relevance to the field of CFD-based ML tools applied to forecasting pollutant dispersion.

For example, a general search in the Scopus® platform using the command TITLE-ABS-KEY("x") gives as a result all the documents in the Scopus® database containing the "x" word in the title of the document, in the abstract or the keywords. To restrict the type of document to articles we use the command DOCTYPE(ar), which will make the search results restricted to articles only.

Table 3.1: CFD and machine learning related articles.

Search query
TITLE-ABS-KEY("CFD" AND "machine learning") AND DOCTYPE(ar)

3.3 Litmap® platform - Block 2

This approach consists of using the online platform Litmap® to construct a graph summarizing, for a certain group of papers, the relation between them in time, the number of citations and the citations in between them.

Figure (3.1) illustrates how Litmap® works if we create a seed based on the reference no.[6] in the Bibliography.

A circle represents an article and is accompanied by the name of the principal author and the year of publication. The circles that are connected by lines represent that there exists citations in-between them.

This is a good approach to better visualize the entirety of papers one is using for their research or even to find new articles connected to the one(s) you have chosen, as the artificial intelligence of Litmap® suggests you if needed. In our case, we will use the 12 articles obtained using the search input in Table (3.2) to create our own customized map.

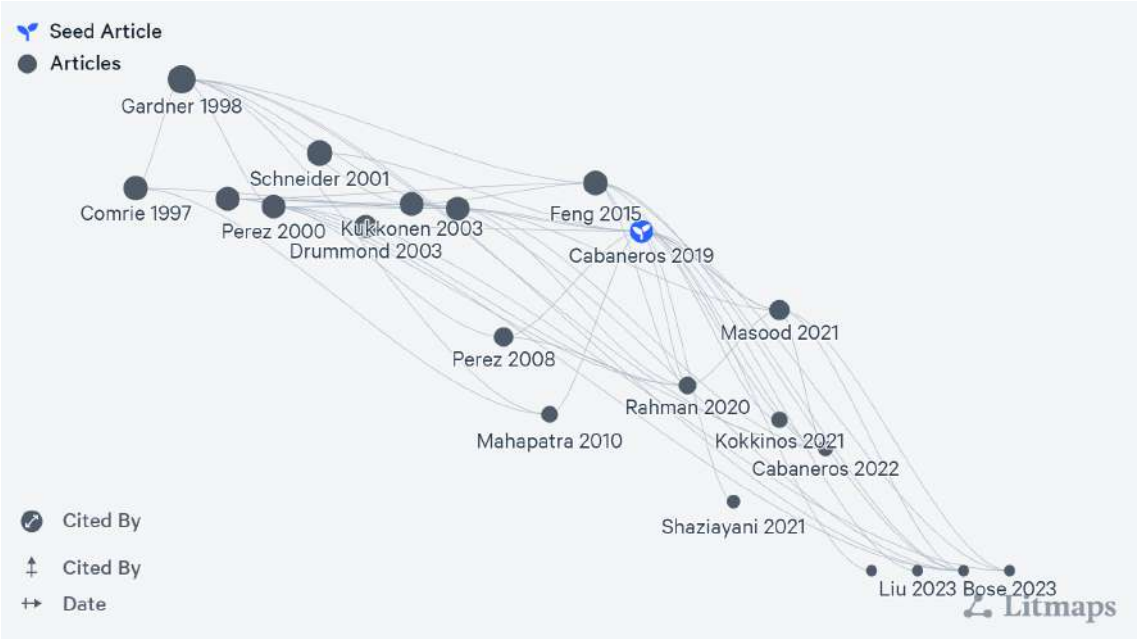

Figure 3.1: Example of using Litmap® platform for a search based on the reference no.[6]. Produced using litmap.com.

Table 3.2: Input for CFD and machine learning-related articles since 2018 relating to pollutant dispersion/air quality.

Search query
TITLE-ABS-KEY(“CFD” AND “machine learning”) AND TITLE-ABS-KEY(“pollutant dispersion” OR “air quality” OR “air pollution”) AND DOCTYPE(ar) AND PUBYEAR AFT 2017

3.4 Toy models - Block 3

3.4.1 GPR Simulation of the flow around a cylinder

The flow of water around a cylinder is chosen as the standard case example using a pre-made algorithm in the OpenFOAM® software to visualize and discuss the results using ParaView® [2] (v. 5.11.1). In sequence, a free version of the simulation CFD software SimFlow® [3] (v. 4.0) expands the last analysis to a generic urban mesh from the simulation using a pre-made algorithm. The previous discussion of

[2]More details about the Paraview® software and download at paraview.org.

[3]More details about the Simflow® software and download at sim-flow.com.

the theoretical aspects of fluid dynamics allows a basic understanding of how the formation of recirculation zones and vortices may be done and aligned with the 3D objects disposition in space can determine a way how the pollutant dispersion occurs and thus the response to be prepared due to this problem.

The simulation begins with the definition of 6 zones in a rectangular simulation region as in Figure (3.2).

Figure 3.2: Divisions of the domain to be meshed.

The fluid used for the flow is water, with a dynamic viscosity of $1 \times 10^{-6}\ m^2/s$ and the cylinder diameter is of $0.05m$. Figure (3.3) describes the domain meshing. The corresponding meshes are related as in Table (3.3) generated using blockMesh. Note that in the proximity of the cylinder, the fragmentation is more intense to capture with fidelity the dynamical behavior of the fluid that changes in this region, essential for the understanding of its mechanics.

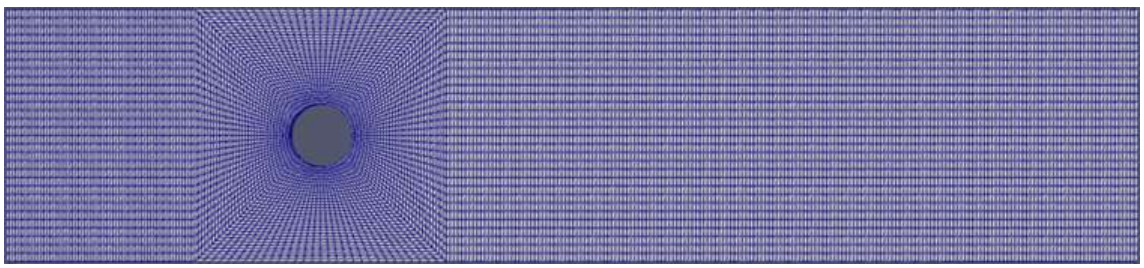

Figure 3.3: Meshing process of the flow domain. Produced using ParaView®.

Table 3.3: Meshing sizes by region.

Region	Mesh
1	60×90
$2, 3, 4\ and\ 5$	90×90
6	180×90

Note that for choosing the mesh domain, a study of the mesh convergence should be done in advance. As this analysis would extend the discussion in this work, the mesh domain chose by the author of the simulation was kept the same (see ref.[51]). One can consult Roger (1996) for more information about this type of analysis. The results of this simulation are depicted in Figure (4.5).

3.4.2 Simulation of flow in an urban mesh

In this case, we have another simulation interesting to have an idea of the behavior of how would be a pollutant dispersion in a random urban mesh. The geometrical domain is depicted in Figure (3.4).

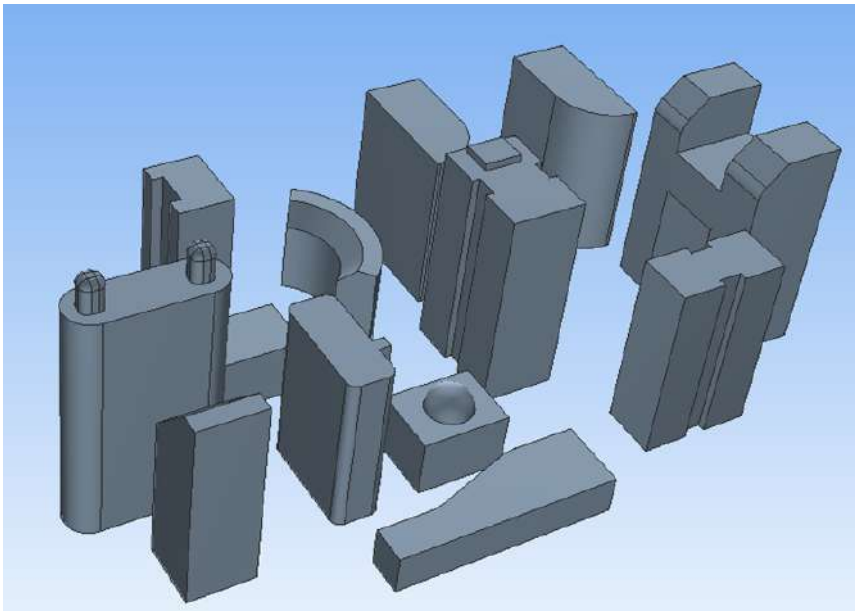

Figure 3.4: The 3D domain of the urban mesh. Produced using SimFlow®.

The simulation procedure can be found in detail in the SimFlow® tutorial "Wind around Buildings" from where it was reproduced[4]. The result of this simulation is showcased in Section 4.4.2.

[4]help.sim-flow.com/tutorials/buildings.

3.4.3 GPR for the simulation of flow around a cylinder using MATLAB®

We will once again proceed with the simulation of the flow over a circular cylinder, but now the objective is to analyse and most importantly predict the behavior of the flow velocity over time using a GPR.

In practice, we chose this simulation because that would be a simulation of an accident scenario where a fluid is leaked from some source and flows over some object (e.g. a residential building with a cylindrical shape) changing its velocity pattern and consequently the concentration pattern. Of course, this is a gross simplification of a real-case leakage because there would be many other affecting parameters for the flow like crosswind, multiple buildings/objects, vegetation and waterways absorption of the fluid and, so on. Anyway, this work looks only to illustrate how the approach would work and then has some importance.

3.4.3.1 Coding with MATLAB®

The machine used for running the simulation is of personal use, with a Intel(R) Core(TM) i5-9300H CPU 2.40GHz 64-bit operational system with 16.0GB of RAM.

Results obtained from ParaView® can be extracted in the format of an Excel table to feed the Gaussian algorithm responsible for approximating with accuracy the results.

There are many different libraries available for proceeding with a GPR but with basically the same algorithm. Some of them include GPY, MATLAB®, and others. Here we will evaluate the code for MATLAB®. Before starting to develop the algorithm in MATLAB®, we must keep in mind a few important theoretical bases already discussed in GPR, as stated below.

Gaussian Process Regression Models are non-parametric kernel-based probabilistic models. Considering a training set:

$$\{(x_i, y_i); i = 1, 2, ..., n\}, \tag{3.1}$$

where $x_i \in R^d$ and $y_i \in R$ are points from an unknown distribution. The GPR model can, based on this training set, for a new value x_{new} predict the result of y_{new}

with good accuracy.

In our case, we will try to prove the accuracy of the model by training the GPR with some few points in a 2D water flow around a cylinder. The pair (x_i, y_i) corresponds to a time-space vector $\boldsymbol{x_i}$ and a velocity magnitude y_i. The time-space vector corresponds to the position in the flow relatively to the XYZ coordinates in time t. In our case the position is fixed and only time varies and consequently the velocity magnitude.

$$\boldsymbol{x_i} = (x_i, y_i, z_i, t_i). \tag{3.2}$$

For more information about the MATLAB® coding and other information in GPR and MATLAB® one can consult the website in reference no.[16]. This website served as reference for the information in this Section.

3.4.3.2 GPR example using fitrgp function in MATLAB®

The generic algorithm in MATLAB® for developing a GPR using a random function as example is as follows:

```
gprMdl1 = fitrgp(x\_observed, y\_observed)
x=linspace(0,10)'
[ypred1,-,yint1]=predict(gprMdl1,x)
fig=figure
fig.Position(3)=fig.Position(3)*2
nexttile
hold on
scatter(x_observed, y_observed1,'r') % Observed data
    points
fplot(@(x) x.*sin(x),[0,10],'--r')  % Function plot of x*
    sin(x)
plot(x,ypred1,'g')                  % GPR predictions
patch([x;flipud(x)], [yint1(:,1);flipud(yint1(:,2))],'k',
    'FaceAlpha',0.1); % Prediction intervals
hold off
title('GPR Fit of Observations')
```

```
14 legend({'Obs','g(x)','GPR predictions','95% pred.int.'},'
   Location','best')
```

We can compute the predicted responses and 95% prediction intervals using the fitted models as in lines 2 and 3. We thus need to resize a figure to display two plots in one figure (lines 4 and 5). We can compare the good predictions with the original curve containing all points from the CFD training and finally create a 1-by-2 tiled chart layout starting in line 5.

The lines below show an example of GPR application for a "x * sin(x)" curve with some observed points. The execution of these lines produces Figure (3.5).

```
1 #True curve
2 fun = @(x) x.*sin(x)
3 xx  = linspace(0,10,100)'
4 yy  = fun(xx)
5
6 #Observation points
7 xd = [1,3,5,6,7,8]'
8 yd = fun(xd)
9
10    #(Fit a GP model. Initialize 'Sigma' to a small value
        . A GP estimates its parameters by maximizing the
        marginal log likelihood. Depending on the data,
        the marginal log likelihood can have multiple
        local optima corresponding to different
        interpretations of the data. Initializing 'Sigma'
        to a small value discourages the high noise
        variance interpretation of the data.)
11
12 gp = fitrgp(xd,yd,'KernelFunction','squaredexponential','
   sigma',0.1,'verbose',1);
13
14 #Plot
15 figure(1); clf
```

```
16 plot(xx,yy, 'r-.')
17 hold on
18
19 [ypred,~,yint] = predict(gp,xx)
20
21 plot(xx,ypred, 'g-')
22 plot(xx,yint(:,1),'k-')
23 plot(xx,yint(:,2),'m-')
24
25 plot(xd,yd,'ro')
26
27 legend('f(x) = x.*sin(x)','GPR predictions','Lower 95\%
    interval','Upper 95\% interval','Observations','
    Location','Best')
```

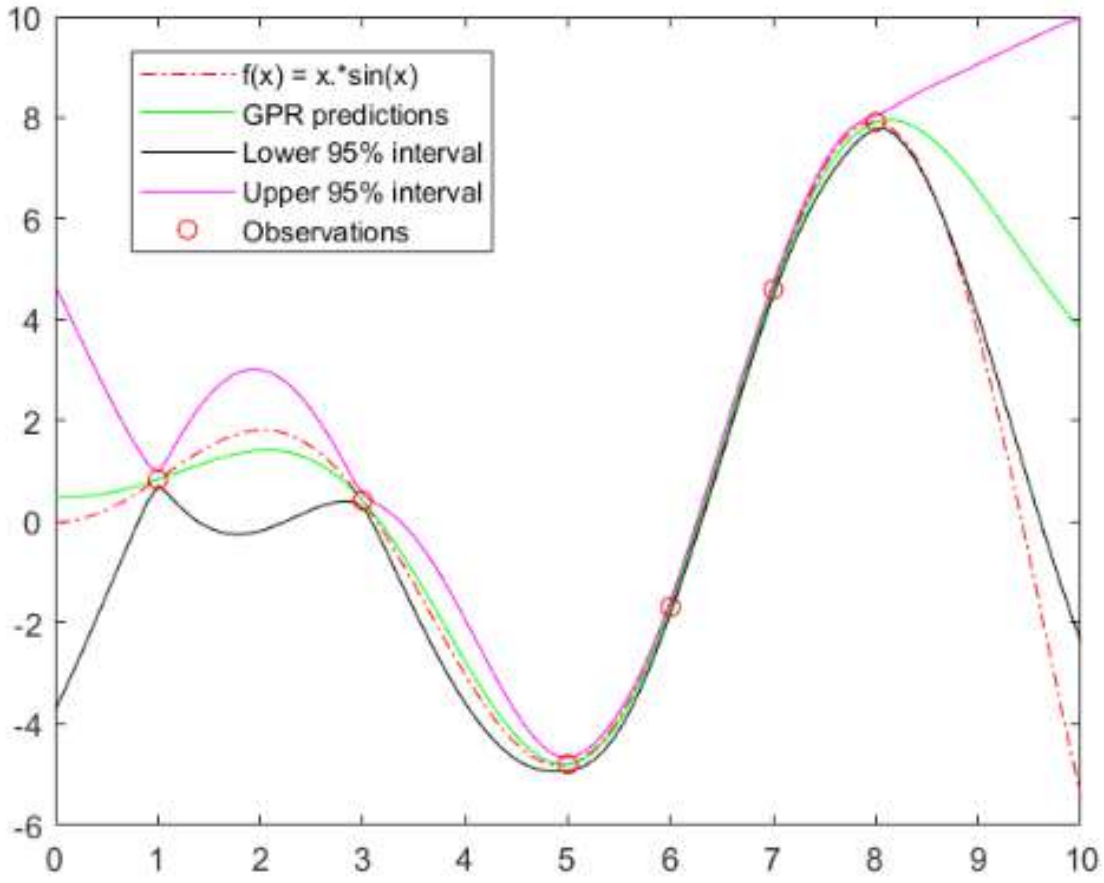

Figure 3.5: Example of a GPR using the function $x * sin(x)$.

3.4.3.3 Plotting over a line

Using the "Filters" toolbar in Paraview®, located in "Data Analysis", we can use the functionality "Plot Over Line" to obtain pairs of (x_i, y_i) for different points in a fixed-line. The following paragraphs contains a suggestion of a similar approach developed in the last subsection but now adapted to the plot of data over a line.

Note that in resonance with the objectives of this undergraduate thesis, which is predicting the behavior of a gas flow in an urban mesh, we could select four representative lines that one studying the dispersion of gases in a city would be interested in. They give the regions with the most important features of the flow and thus helps authorities take actions based on this information towards the population's health and welfare.

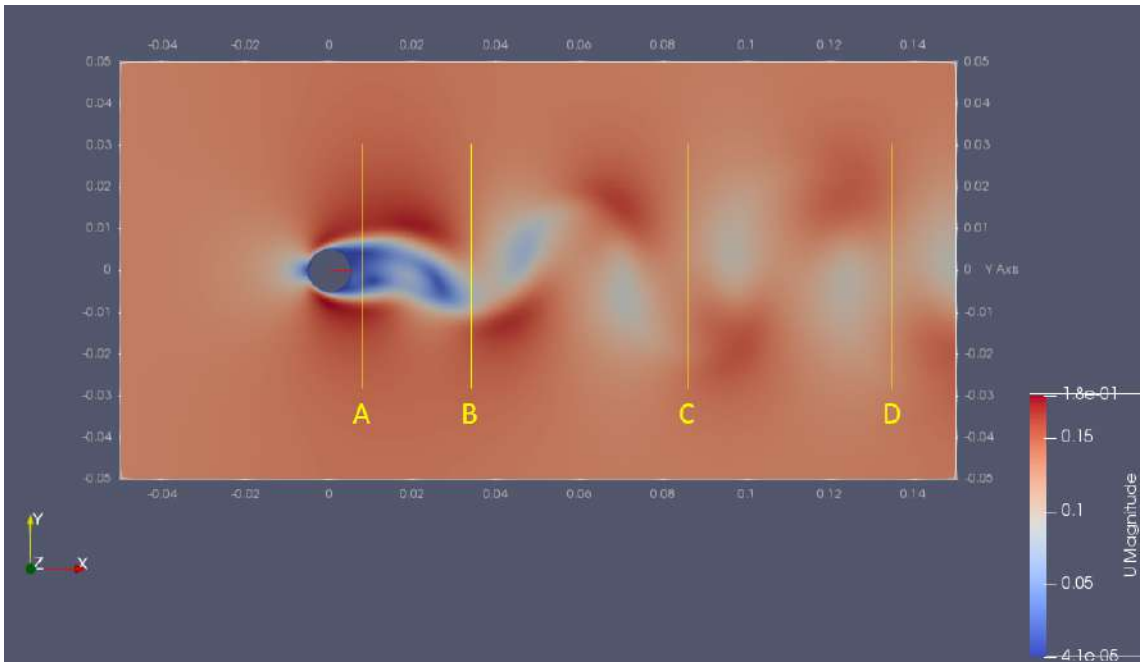

Figure 3.6: Screenshot of t=5s for the simulation using Simflow®. Note the regions covered by lines A, B, C, and D representing zones with important flow figures.

Line A would correspond to points perpendicular to the flow in the immediate region after the obstacle (e.g. a residential building), line B the region close to the obstacle when the vortex starts to unfold, line C a medium distance from the obstacle, and line D a long distance measurement. With these four points, we can infer how the dispersion would affect smaller object (as pedestrians) that would have

near zero effect over the flow characteristics but would be affected by it.

We could divide each line into 10 equally spaced subsections, for instance. In each point connecting these sections as well as in the boundaries, we could take the vectors (x_i, y_i) from our CFD simulation. As we set in SimFlow® our simulation time to 5 seconds, with a time-stepping of 1 millisecond, we take note of each 0.10 section, which will give, for each XYZ position, a total of 50 different times giving 50 different velocity magnitudes, which is our y_i.

To simplify, we could proceed with the analysis of a GPR for three different points in line A: the two extremes and the middle point. These observation points vary over time and thus originate three different GPR predictions to be plotted in a graph of velocity magnitude U versus time in a given position (x, y, z) in the Cartesian plane.

For each data set, the GPR model can be fitted using the "fitrgp" function in MATLAB®. To simplify even more, we apply this model for the middle point in Line A in Figure (3.6), solely to not extend too much the discussion but yet have an insight into the pros of using fitrgp function to predict flow features faster and accurately. The other points as well as points in the other lines can be obtained similarly.

Applying the before-mentioned code for Line A, choosing some few data from the spreadsheet extracted from Paraview® after the simulation using Simflow®, we have the following code lines, with the results shown in Figure (4.8) in Chapter 4.

```
% GPR

      xx = [-0.00926;-0.00866;-0.00636;-0.00394;-0.00186;
      -0.0011;-0.0006;0.00038;0.00122;0.00394;0.00588;
      0.0073;0.01];

      yy = [0.165268583;0.159939235;0.100765951;
      0.028153305;0.008155402;0.016071;0.019629658;
      0.022873396;0.022377322;0.009639;0.040527163;
      0.089619082;0.160516745];

```

```
12      % Fit a GP model. Initialize 'Sigma' to a small
           value. A GP estimates
13      % its parameters by maximizing the marginal log
           likelihood. Depending
14      % on the data, the marginal log likelihood can have
            multiple local
15      % optima corresponding to different interpretations
            of the data.
16      % Initializing 'Sigma' to a small value discourages
            the high noise
17      % variance interpretation of the data.
18      gp = fitrgp(xx,yy,'KernelFunction',"
           squaredexponential","OptimizeHyperparameters","
           auto","Sigma",0.1,'verbose',0);
19
20      % Plot.
21      figure(1); clf
22      hold on;
23
24      [ypred,~,yint] = predict(gp,xx);
25
26      plot(xx,ypred, 'g-');
27      plot(xx,yint(:,1),'k-');
28      plot(xx,yint(:,2),'m-');
29      hold off;
30
31      legend('GPR predictions','Lower 95% interval','
           Upper 95% interval','Location','Best');
```

3.4.3.4 Plotting data over time in a fixed point

Now, instead of choosing points over a line for a fixed time, we will proceed with the GPR for the fixed point $(0.01, 0, 0)$ with velocity magnitude varying over

time. The regression is based on some few points from the data acquired during 2s of CFD simulation with a 0.001s time-step using Simflow®. The code used for the regression can be found below. Results are shown in Section 4.4.3.2.

```
%Data acquired from simulation using Simflow for the
   period of 0-2s usign 0.001s time-step
    xx=[0.1;0.15;0.2;0.25;0.3;0.35;0.4;0.45;0.5;0.55;
    0.6;0.65;0.7;0.75;0.8;0.85;0.9;0.95;1;1.05;1.1;
    1.15;1.2;1.25;1.3;1.35;1.4;1.45;1.5;1.55;1.6;
    1.65;1.7;1.75;1.8;1.85;1.9;1.95;2];
    yy=[0.100296012;0.088955045;0.066730802;
    0.052009614;0.050477718;0.051;0.046647615;
    0.036124784;0.019906029;0.014616429;0.040311289;
    0.058249464;0.050990195;0.029966648;0.028600699;
    0.051107729;0.060373835;0.044011362;0.053851648;
    0.021289669;0.053150729;0.064288413;0.043518846;
    0.009638444;0.030149627;0.061611687;0.061;
    0.035062943;0.005247857;0.043301386;0.06670832;
    0.053190695;0.023648467;0.019126944;0.057070132;
    0.066287254;0.044070965;0.010259142;0.036031236];

    % Fit a GP model. Initialize 'Sigma' to a small
       value. A GP estimatesits parameters by
       maximizing the marginal log likelihood.
       Depending on the data, the marginal log
       likelihood can have multiple local optima
       corresponding to different interpretations of
       the data. Initializing 'Sigma' to a small value
       discourages the high noise variance
       interpretation of the data.
    gp = fitrgp(xx,yy,'KernelFunction',"
       squaredexponential","OptimizeHyperparameters","
       auto","Sigma",0.1,'verbose',0);
```

```
19
20      %Plot the discrete data for visualization
21      figure(1); clf
22      scatter(xx,yy,"filled")
23
24      hold on;
25      [ypred,~,yint] = predict(gp,xx);
26
27      plot(xx,ypred, 'g-');
28      plot(xx,yint(:,1),'k-');
29      plot(xx,yint(:,2),'m-');
30      legend('Data','GPR predictions','Lower 95% interval
           ','Upper 95% interval','Location','Best');
31      axis([0 2 -0.01 0.12]);
32      hold off;
```

Chapter 4

Results and Discussion

Through the last chapters, it was evaluated many conclusions obtained from recent studies that reinforce the correlation between the urban mesh geometrical disposition (as well as other elements such as vehicle flow and vegetation) and the flow behavior. From the results of these studies, it was highlighted the existence of re-circulation zones between buildings, which was supposed to accumulate a more concentrations of pollutants, as it happens similarly in the case of high-traffic longitudinal roads.

As we described the emulation framework proposed by Ganti and Khare's using a high-fidelity CFD model to forecast in real-time the dispersion of pollutant gases using machine learning techniques, with the search methodology proposed in Chapter 3, one can obtain sufficient background to further develop any essential part of this subject.

4.1 Dimensions platform

4.1.1 Publications in CFD

A more general search at Dimensions.ai using the input "CFD AND (POLLUTANT DISPERSION OR AIR QUALITY)" in the categories "Title and abstract" gives us an overview of the effort that has been put into this area of study, represented by an augmentation in the number of articles since 2015 even though the quantity of papers is yet low showing that there is still space for growth.

Indeed, the total number found was, since 2015, 455 according to the search tool Dimensions.ai[1]. The choice of using the platform Dimensions® is due to its accessibility since it has a free version. This would give an alternative to those who are not willing to pay in a very low stage of research for well-known platforms such as Scopus® but still want to have an overview about any scientific topic.

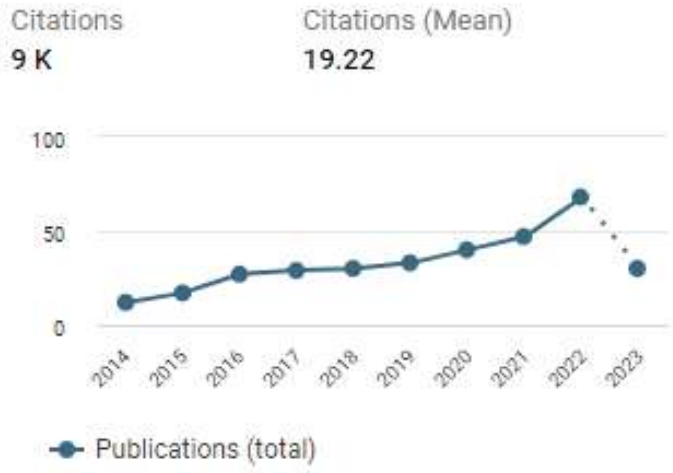

Figure 4.1: Citations since 2015 using the given input. Extracted from Dimensions.ai.

4.1.2 Publications in ML

The use of machine learning in many scientific areas has exploded in the past few years as a very promising tool. In Figure (4.2), one can observe how the number of publications has been constantly increasing, passing over 1.4 million publications since 2015. Once again the term "MACHINE LEARNING" was searched in "Title and Abstract" in Dimension®'s database.

4.2 Advanced search using Scopus®

The first search in articles containing the words CFD and machine learning gives a total number of 597 documents found. The corresponding result, by year, is presented in Figure (4.3).

This being stated, it can be created a new and better search entry (always in Scopus®) taking into account only articles from the last 5 years (since 2018). This

[1]Visit www.dimensions.ai for more details.

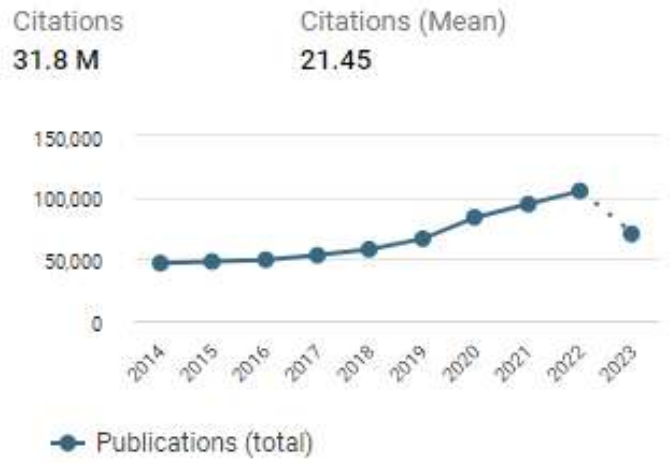

Figure 4.2: Number of publications in the area of Machine Learning since 2015. Extracted from Dimensions.ai.

Table 4.1: CFD and machine learning related articles.

Search query	Result
TITLE-ABS-KEY("CFD" AND "machine learning") AND DOCTYPE(ar)	597

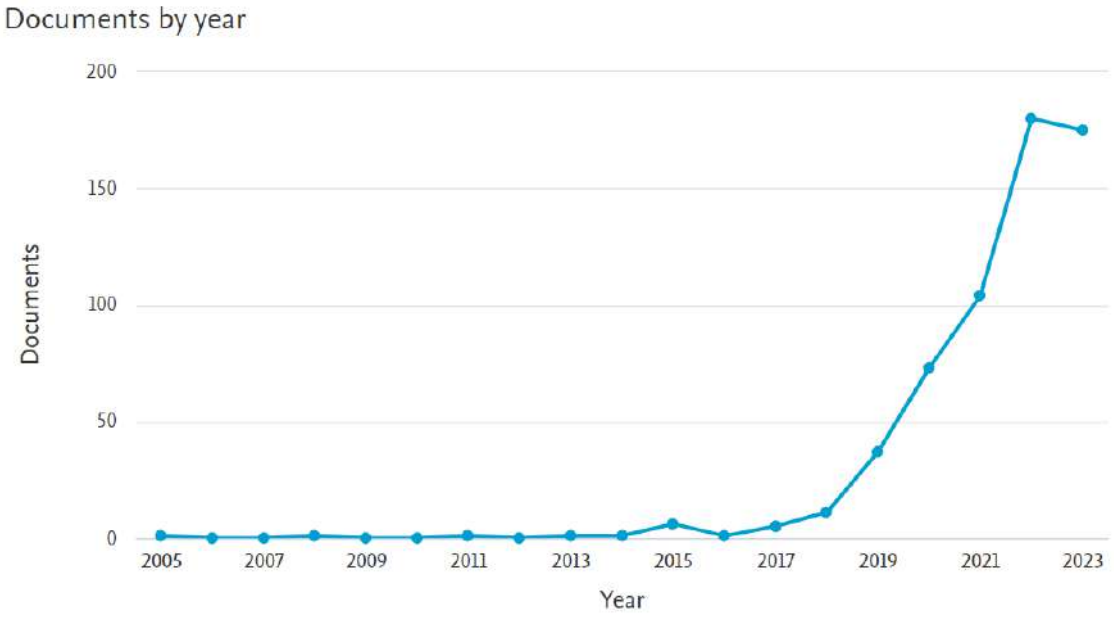

Figure 4.3: Number of articles, per year, corresponding to the search query in Table (4.1).

will help narrow the search to the most recent scientific knowledge. The result is obtained and shown in Table (4.2).

As expected by the past figure, the number is still high since the trend of

Table 4.2: CFD and machine learning-related articles since 2018.

Search query	Result
TITLE-ABS-KEY("CFD" AND "machine learning") AND DOCTYPE(ar) AND PUBYEAR AFT 2017	580

this study is recent. Now to focus on the subject of our interest we will add a new restriction that is to the scenario of CFD-based machine learning techniques applied to pollutant dispersion/air quality according to the entry query in Table (4.3).

Table 4.3: CFD and machine learning-related articles since 2018 relating to pollutant dispersion/air quality.

Search query	Result
TITLE-ABS-KEY("CFD" AND "machine learning") AND TITLE-ABS-KEY("pollutant dispersion" OR "air quality" OR "air pollution") AND DOCTYPE(ar) AND PUBYEAR AFT 2017	12

Now the result is extremely reduced to only 12 articles been published since 2018, which highlights the importance of the present thesis of fomenting new studies in this area. These 12 articles are listed in Table (4.6).

Research by hand was done in Google Scholar using the Keywords for the search: CFD, pollutant dispersion, and urban, which gives us the Table (4.4) and Table (4.5). These tables provide an extensive list of recent studies that suits well this subject area and that would be of great base to develop more studies in the area of CFD-based ML techniques applied to pollutant dispersion in urban areas. The articles obtained using this approach highlight the significance of this research area. These served as reference throughout the thesis, aiding in a deeper comprehension of the subject. All utilized information is duly listed in the Bibliography.

Again, an interesting result to be observed is depicted in the Figure (4.3). Note how the interest in this research area has grown exponentially in the last five years compared to the beginning of the decade, when almost nothing was published correlating CFD and machine learning techniques.

In fact, a more general analysis of the graphs in Sections 4.1.1 and 4.1.2 shows that both machine learning and CFD techniques in air quality have been constantly

growing in the past almost 10 years.

This suggests a reason for the evolution observed in Figure (4.3). One can note that, since 2015, when machine learning started growing at a fast pace, a tendency in the publication of articles published using CFD-based machine learning techniques has appeared, suggesting that with the maturity of new ML techniques arriving, new methods were integrated to optimize current CFD models and also develop new ones.

4.3 Customized Litmap®

The 12 articles found by using the search query in Table (4.3) cover multiple areas, like Environmental Science, Engineering, Physics and Astronomy, etc. These papers are related in Table (4.6) in the end of the document and are represented by the map in Figure (4.4).

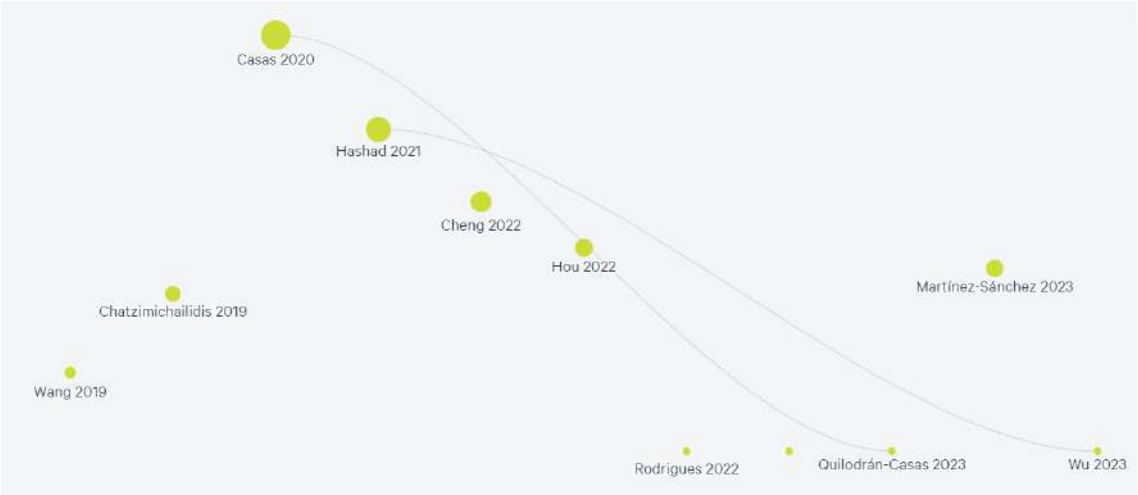

Figure 4.4: Litmap® of the 12 articles found.

The circle between "Rodrigues 2022" and "Quilodrán-Casas 2023" corresponds to Wai and Yu (2023). Note how the correlation that corresponds to the line connecting the dots is scarce and shows how the recent studies haven't been focusing in one single area, and mostly that it hasn't been many developments yet, which contributes to the importance of this review and propitiating new papers to be written.

4.4 Toy models

4.4.1 GPR Simulation of the flow around a cylinder

We can observe the time evolution of the fluid velocity around the cylinder, as well as the formation of vortices in Figure (4.5), where we represent the instantaneous capture of the simulation proposed in the corresponding methodology part for different time-steps.

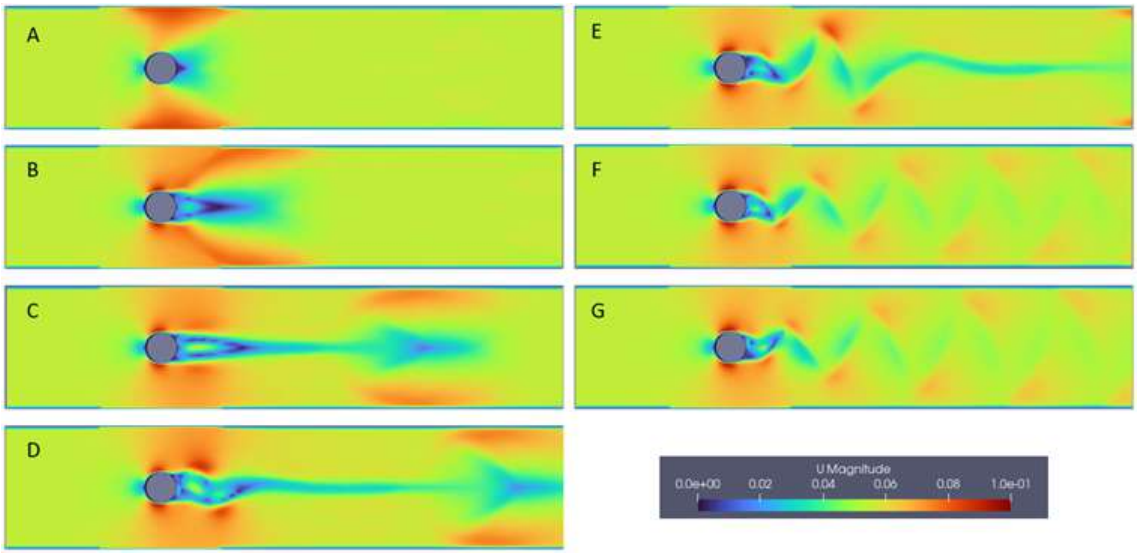

Figure 4.5: Simulation of flow around a cylinder for different time steps: 100 (A), 500 (B), 1500 (C), 2000 (D), 2500 (E), 4500 (F), 6500 (G). U Magnitude is shown in m/s. Produced using OpenFOAM®.

Note how the velocities are less intense for the region after the cylinder, where the vortices are formed, as predicted by the theory discussed previously. If we take this time as an idealization of the flow of polluted air around a building, we see that there is a tendency to concentrate the pollutant after the cylinder where we have lower velocity. This concentration is even more intensified if there is a second obstacle that blocks the propagation of the vortices, creating a kind of bubble that accumulates and is not dispersed. These ideas are essential for the discussion that will be developed later.

The files used to run this simulation can be found in Panchigar et al. (2022).

4.4.2 Simulation of flow in an urban mesh

Figure (4.6) is a snapshot from SimFlow® software after running the simulation of flow in an urban mesh. The overall visualization of it confirms the last conclusions about how simulating the flow of wind around buildings as it is in this case can give an idea of how the pollutant dispersion would occur. Note how in-between buildings the field velocity is weaker and thus collaborates to the formation of accumulation zones where pollutants can be concentrated, augmenting the negative effects on the population who permeates these zones.

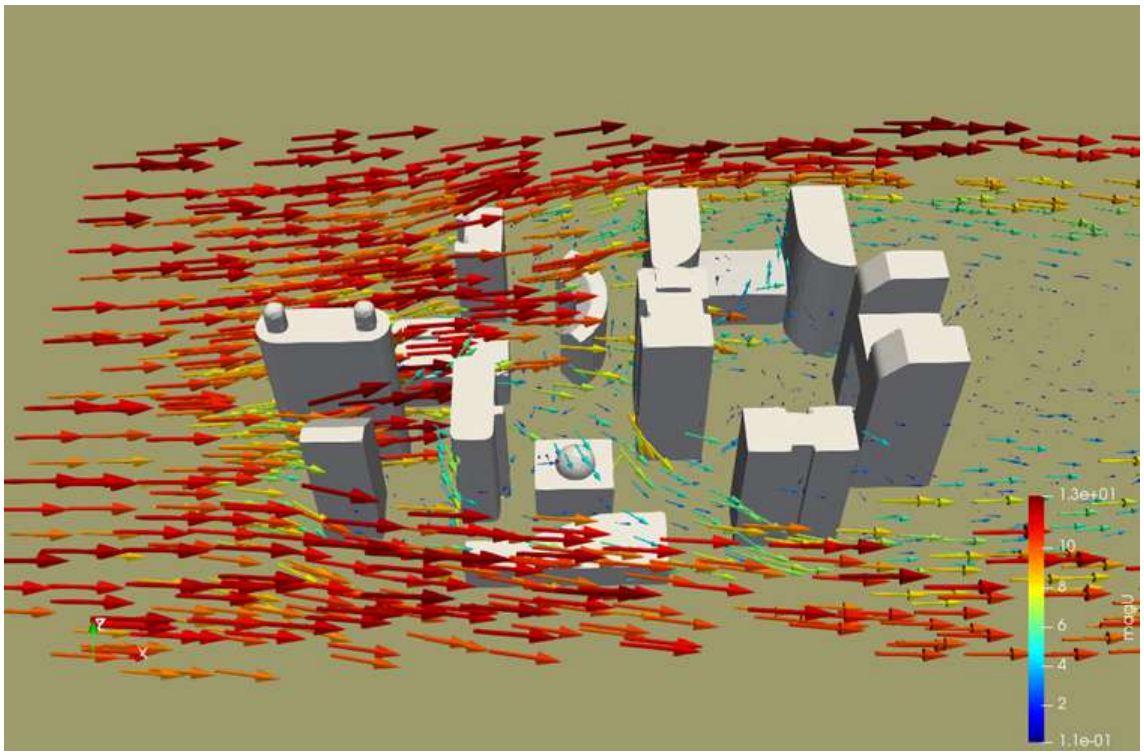

Figure 4.6: Simulation of wind flow in a generic urban mesh obtained using SimFlow®. The speed field is represented with arrows. The corresponding magnitudes are given in the right bottom of the image.

4.4.3 GPR for the simulation of flow around a cylinder using MATLAB®

4.4.3.1 Plotting over a line

In MATLAB®, the first step was fitting the GPR model to the data set. Line A extremes in Figure (3.6) are $(0.01, 0.03, 0)$ and $(0.01, -0.03, 0)$. Thus the middle point is $(0.01, 0, 0)$. Paraview® gives us the following U versus y graph corresponding to the velocity magnitude over the entire line. Note the canyon corresponding to low-velocity values for the points that are shadowed by the cylinder. The value for the velocity magnitude in the middle point is 0.0222573m/s at 5s.

The other two points corresponding to the extremas of line A can be obtained similarly and easily, as any other $U \times t$ function in any of the four lines (A,B,C or D).

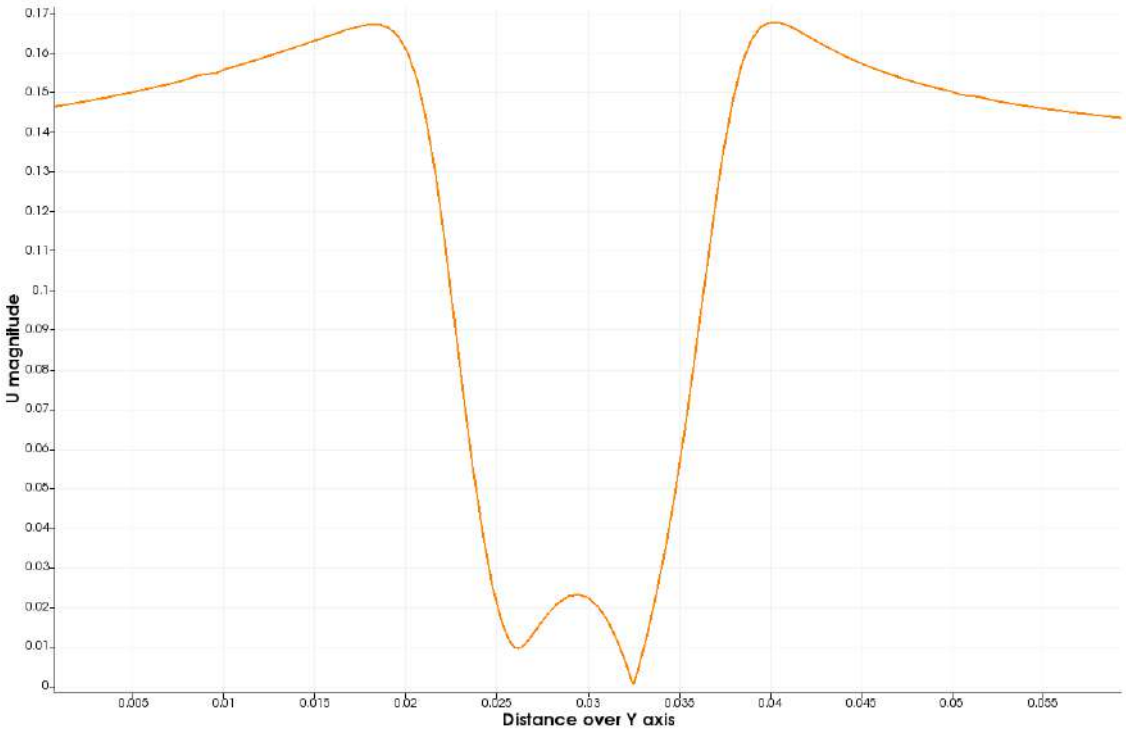

Figure 4.7: Screenshot from Paraview® of U magnitude versus position in line A.

Figure (4.8) is the application of the code discussed in the methodology using GPR to try to predict Figure (4.7).

Note the correspondence of the model. GPR makes a good predictions if compared to the values of the true curve as well as values from the raw data in

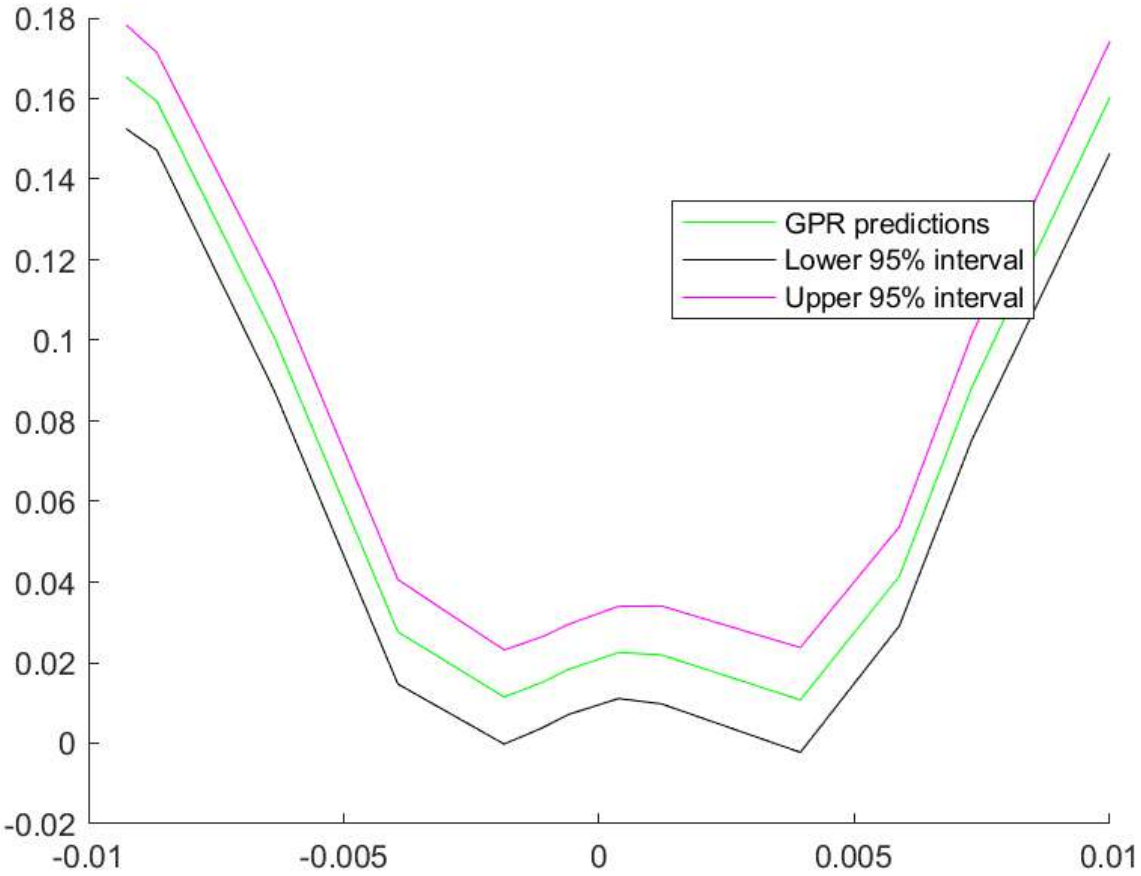

Figure 4.8: GPR fit of the data provided previously.

Figure (4.7). An error estimative should be performed in order to estimate how "good" is the prediction.

4.4.3.2 Plotting data over time in a fixed point

A simulation of the flow over a cylinder as described in the previous Section was performed using a time-step of 0.0001s in SimFlow® for an interval of 5s. The total simulation time was 1998s (or 33.3 minutes), producing the true curve in Figure (4.9).

In order to shorter the simulation time and yet reproduce the aspects of Figure (4.9), the same simulation was performed again but now using a larger time-step of 0.001s for the first 2s in Simflow®. With this configuration, the total time spent in the simulation was of 122s. Using the data to train a GPR model using the fitrgp function in MATLAB® provides a good regression for this period (Figure (4.10)) that can be extended until 5s observing the periodicity of the vortex shedding. The total time in MATLAB® using the code below was 20.3745s. Adding both

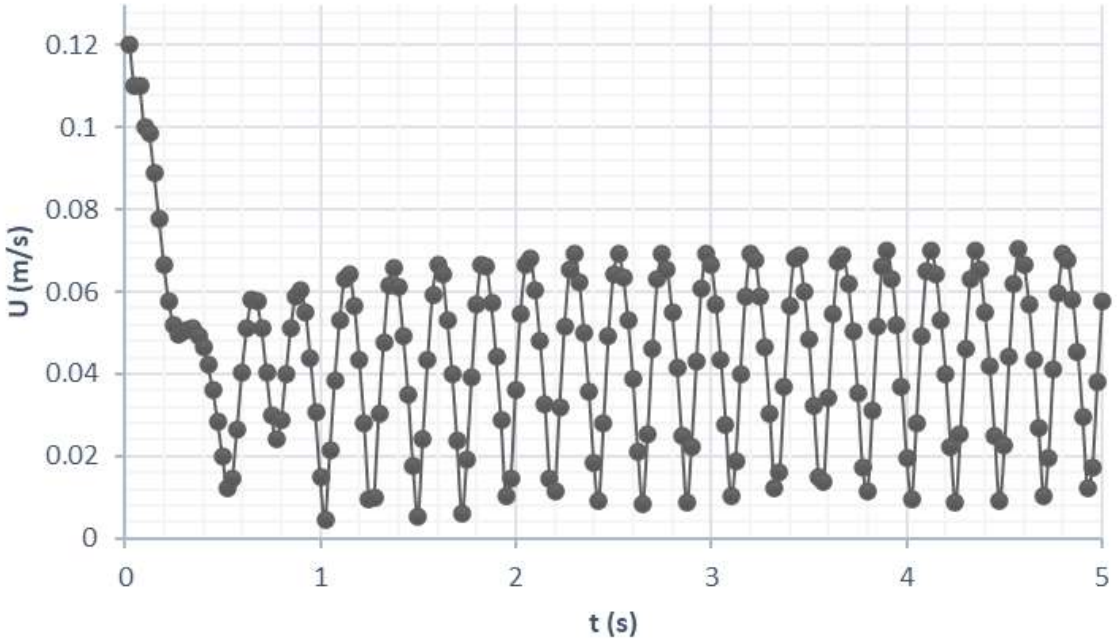

Figure 4.9: Velocity magnitude over a period of 5s for the simulation of the flow around a cylinder using a 0.0001s time-step in Simflow®.

simulations, the final time required for the regression of data was approximately 142s wich is only 7.11% of the time we spent to acquire the data in Figure (4.9).

4.5 Predicting the pollutant dispersion in an urban mesh

At this stage, there is sufficient information to highlight some points concerning the pollutant dispersion in an urban mesh and then analyse from recent studies how the prediction of the fluid flow could help infer about the urban mesh special case.

4.5.1 The case of a street canyon

A recent study by Kwak et al. (2018) analyzed the concentration of common urban atmospheric pollutants (NOx, BC, pPAH and PN) in urban areas using mobile monitoring and CFD modeling. This article showed that in street canyons (streets of great length generally composed of several lanes, tall buildings around them, with poor ventilation and a large concentration of emitting vehicles), there is an important

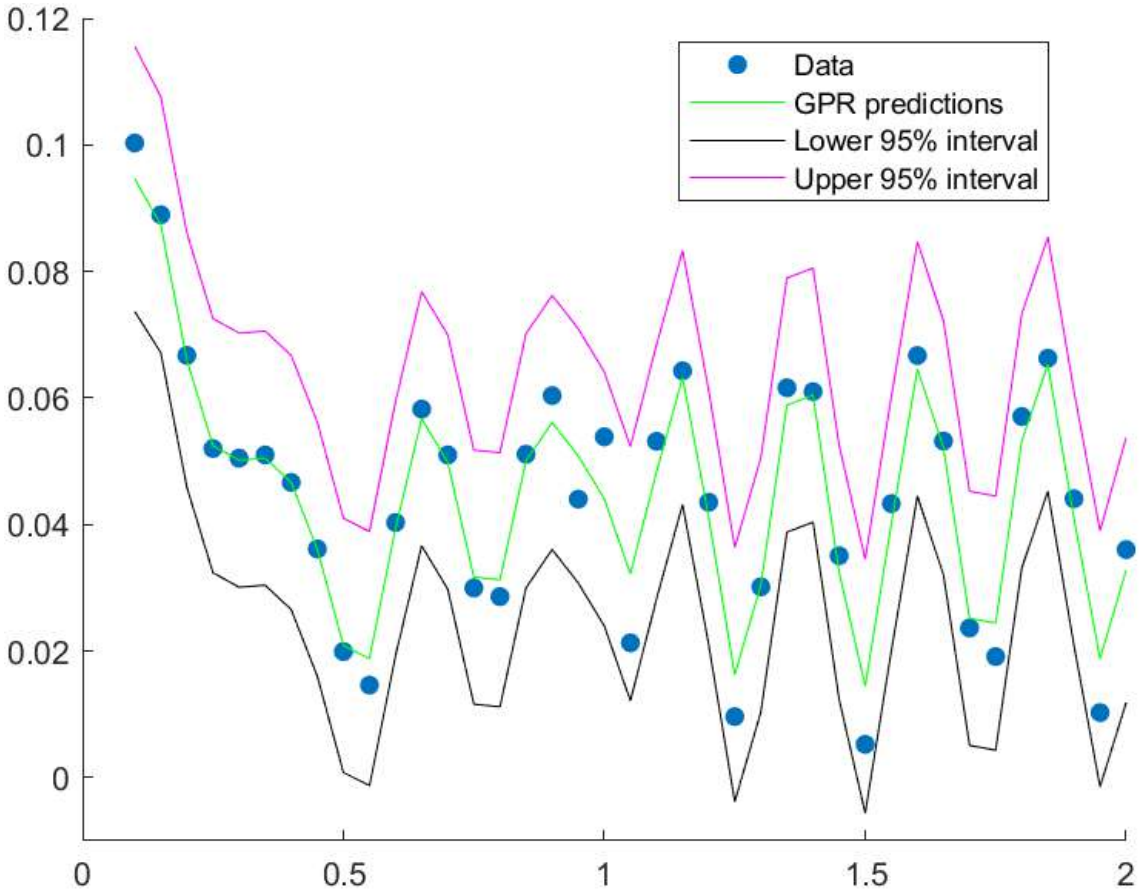

Figure 4.10: GPR simulation in MATLAB® using Data acquired from CFD simulation using Simflow® for the period of 0-2s using a 0.001s time-step.

correlation between poor air quality and the presence of heavy-duty diesel vehicles (HDDVs). This correlation shows the relevance of this study since those HDDV vehicles are diesel exhausted, classified as Group 1 carcinogens by the International Agency for Research on Cancer (IARC). In fact, heavy vehicles emitting diesel are largely responsible for the emission of pollutants that are later dispersed into the urban mesh (50% of particle number (PN), and 60% of particulate matter emissions).

The study uses the RANS model, one of the most common (and already discussed previously in Chapter 2), with the re-normalization group (RNG) k-ϵ turbulence closure scheme to simulate the gas dispersion in a street canyon in the Seoul (South Korea) metropolitan area.

The high concentration of tall buildings in the entourage of the main roadway is responsible for enclosing the flow in the longitudinal path, with few concurrent streets. This study can then be interesting to help us understand how the dispersion of pollutants works in an urban mesh, as this is usually composed of a set of street

canyons, with some isolated vegetation regions (e.g. squares), and a large volume of vehicle traffic.

Therefore, in addition to airflow simulations around buildings, the study of street canyons is complementary and helps understand what types of measures could be taken by authorities to better manage urban traffic to avoid exposure of their population to high volumes of polluting gases. For example, it is known that along the street canyon, the airspeed has low magnitude because there are no exit routes for it since we are surrounded by tall commercial and residential buildings. On the contrary, at signs of road intersections, there are side winds that accelerate the local wind and thus contribute to the dispersion of pollutants. So, a simple solution one could think of is to avoid the construction of buildings too close to each other in a way to allow ventilation between the buildings that break this longitudinal flow in the street canyon and consequently disperse the polluting gases that concentrate and affect the population's health.

Proposing an urban geometry that would allow strong winds to push away the pollutants from its source and thus from pedestrians and residences along the street canyon would be a good approach to mitigate the pollutant dispersion in an urban mesh that affects its inhabitant's health. However, the first study in Table (4.6) shows that this would only spread the pollutants to nearby living blocks and thus enlarge the exposure area. In fact, the pollutants can have a wide range of impact of more than 300m from the source. For example, the furthest diffusion distance carbon monoxide (CO) can reach from a road source is between 100m and 200m. Thus, spreading the gas would not be a real solution but would displace the issue to other areas. A more effective alternative is proposed by Wu and Chen (2023): guiding the pollutants into the UCL, lowering the concentration in the pedestrian zones. The study proposes a ML approach to find an optimized block configuration based on a data set from CFD simulations. The use of ML is known to have sped-up the CFD simulations by 800 times. The modification of the block morphology changes the pathway of pollutants not only for the adjacent buildings but also the inner buildings in the block (considering only perpendicular winds from the street canyon). Adding a building of a certain height affects the pollutant concentration by the formation of accumulation zones in the street canyon.

In consequence, the study of the UCL and understanding how block morphology and block layouts contribute to pollutant dispersion remain important. An often ignored morphological indicator, the standard deviation of building height (BH_{std}), for example, is an important parameter to analyze the changes in mean flows in horizontal and vertical directions and consequently the pathways and rates of pollutant transport. Indeed, urban morphology is an important parameter and can not be ignored since it affects the airflow patterns, turbulence intensity and ventilation performance in urban blocks and street canyons (Wu and Chen, 2023).

Many empirical studies do not show a singular linear relationship between BH_{std} and pollutant concentration. In fact, for a block of buildings facing a main road of a street canyon, Wu and Chen (2023) suggest that a height of $33m$ for buildings facing the street canyon would be the minimum for mitigating the spread of gases into the building block. It also suggests as a general method that tall buildings facing the main road and small buildings inside the block would be an optimal configuration for preventing a threatening concentration of traffic-related air pollutants inside the block.

One can conclude then that a lot of discussion and research can still be produced in this area to give further insights and better options for urban planning and improvement of urban human health. It also answers partially the specifics objective listed in Chapter 1 while it opens a door to new reflections on the subject.

4.5.2 Turbulence models and computing time

As discussed in Chapter 2, there are different possibilities for turbulence models to describe the wind flow. As in the articles cited before, using different turbulence models affects directly the computational time and thus must be considered in the analysis of any case study.

Liu et al. (2016) compare also the effects in the variation of different computational parameters such as grid resolution, discretization time-step and sampling time to observe the impact of these in the computational time for the simulation of the wind flow in an isolated curbstone in a 1:1:2 scale representing a building in an urban mesh adopting the parameters of Table (4.7).

This analysis, beyond being interesting for the evaluation of the compared

computational cost of the models, is important to see how the wind flow would behave in an urban mesh when meeting a building surface and, thus, how the pollutants flow would behave.

Table 4.4: List of relevant publications in CFD related to air pollution in recent years.

Ref.	Year	Title
[63]	2023	System Coupled GIS and CFD for Atmospheric Pollution Dispersion Simulation in Urban Blocks.
[47]	2023	CFD Evaluation of Ventilation and Pollutant Dispersion Withindiscontinuity Urban Streets.
[41]	2023	CFD simulation of pollutant dispersion in a street canyon: Impact of idealized and realistic sources.
[35]	2023	Bi-objective optimization of traffic assignment with air quality consideration via CFD-based surrogate model.
[40]	2022	Air Pollution Dispersion Modelling in Urban Environment Using CFD: A Systematic Review.
[20]	2022	Investigation of O3–NOx–VOCs chemistry and pollutant dispersion in street canyons with various aspect ratios by CFD simulations.
[60]	2022	Evacuation route optimization under real-time toxic gas dispersion through CFD simulation and Dijkstra algorithm.
[31]	2021	CFD simulation of flow fields and pollutant dispersion around a cubic building considering the effect of plume buoyancies.
[70]	2021	CFD simulations of wind flow and pollutant dispersion in a street canyon with traffic flow: Comparison between RANS and LES.
[4]	2021	CFD modelling: The most useful tool for developing mesoscale urban canopy parameterizations.
[69]	2021	CFD-based analysis of urban haze-fog dispersion—A preliminary study.
[29]	2021	Application of Improved CFD Modeling for Prediction and Mitigation of Traffic-Related Air Pollution Hotspots in a Realistic Urban Street.

Table 4.5: Continuation of Table (4.4).

Ref.	Year	Title
[38]	2020	An investigation into the effects of green space on air quality of an urban area using CFD modeling.
[48]	2020	Analysis of transport methodologies for pollutant dispersion modelling in urban environments.
[10]	2020	A novel approach to simulate pollutant dispersion in the built environment: Transport-based recurrence CFD.
[25]	2019	High Resolution Urban Air Quality Modeling by Coupling CFD and Mesoscale Models: a Review.
[36]	2019	Street canyon ventilation and airborne pollutant dispersion: 2-D versus 3-D CFD simulations.
[13]	2019	How parked cars affect pollutant dispersion at street level in an urban street canyon? A CFD modelling exercise assessing geometrical detailing and pollutant decay rates.
[18]	2019	Urban areas parameterisation for CFD simulation and cities air quality analysis.
[27]	2018	On-Road Air Quality Associated with Traffic Composition and Street-Canyon Ventilation: Mobile Monitoring and CFD Modeling.

Table 4.6: The 12 articles found by using the search query in Table (4.3). Cit. stands for number of citations. Elaborated based on Scopus®.

Ref.	Year	Title	Cit.
[64]	2023	Optimizing block morphology for reducing traffic pollutant concentration in adjacent external spaces of street canyons: A machine learning approach	0
[43]	2023	A data-driven adversarial machine learning for 3D surrogates of unstructured computational fluid dynamic simulations	0
[32]	2023	Data-driven assessment of arch vortices in simplified urban flows	2
[57]	2023	Application of a Machine Learning Method for Prediction of Urban Neighborhood-Scale Air Pollution	0
[45]	2022	Unobtrusive Cardio-Respiratory Assessment for Different Indoor Environmental Conditions	0
[23]	2022	Prediction and optimization of thermal comfort, IAQ and energy consumption of typical air-conditioned rooms based on a hybrid prediction model	6
[8]	2022	BIM-supported sensor placement optimization based on genetic algorithm for multi-zone thermal comfort and IAQ monitoring	8
[14]	2021	Bim and data-driven predictive analysis of optimum thermal comfort for indoor environment	17
[21]	2021	Designing roadside green infrastructure to mitigate traffic-related air pollution using machine learning	15
[42]	2020	A Reduced Order Deep Data Assimilation model	28
[7]	2019	Implicit definition of flow patterns in street canyons-recirculation zone-using exploratory quantitative and qualitative methods	3
[58]	2019	Wind field reconstruction for the dispersion modeling of accidental chemical spills on complex geometry	1

Table 4.7: Mesh arrangement description and computing time costs of SRANS, LES and DES cases. Adapted from Liu and Niu (2016).

Case	Mesh nos. (million)	Min grid size (m)	Model	Computing time (h)
$RNG-2$	2.60	0.001	$RNG\ \kappa-\epsilon$	4.8
$LES-2$	4.80	0.0005	LES	84.0
$DES-2$	4.80	0.0005	$DDES$	84.0
$DES-9$	3.58	0.0005	$DDES$	67.2

Chapter 5

Conclusion

In Chapter 1, it was highlighted the importance of the simultaneous work of CFD and ML methods to provide fast results in the prediction of pollutant gas dispersion in an urban mesh and how this would be of high value for any authorities responding to emergency scenarios as well as to the increase of the population's welfare. This necessity comes from alerting global scenario of pollution/air quality in many countries from all continents, including Brazil.

In this undergraduate thesis, It was reviewed recent papers on CFD-based machine learning algorithms for predicting pollutant dispersion. It was used Ganti and Khare's (2020) approach as the basis, summarizing its four steps and delving into each step's theory independently. It was also analyzed fundamental concepts such as Urban Micro-climate, governing equations in CFD, common turbulence models, FVM, ROM, and ML (specially GPR with applications in MATLAB®). This comprehensive examination not only provided an understanding of Ganti's approach but also offered insights for those interested in constructing their own framework.

In Chapter 3 the methodology based on the research in current scientific database platforms (Dimensions®, Scopus® and Litmap®) was performed. Also GPR examples showcased the benefits in time of applying ML to our study case. In the following Chapter 4, it was discussed the evolution of recent works in CFD, ML and both of them together applied to the field of pollutant dispersion, when we realized that the number of papers from the last 5 years is not very large, indicating that there is still a huge space for improvement in this research area. Nonetheless, we still were able to analyze with these results how the formation of accumulation

zones and the urban mesh morphology could affect pollutant dispersion especially in a street canyon and its surroundings.

From the 12 papers we found in the combination of the subjects, not all of them were necessarily applied to an urban mesh, which narrows, even more, the information available in the scientific community to give insights into urban planning improvement. This suggests that further studies are necessary yet to understand deeply the dynamics of pollutant dispersion in urban meshes. These studies could also open ways to other fields that would improve even more the analysis, such as thermal comfort.

Chapter 6

Bibliography

[1] BLEVINS, R. D. **Flow-induced vibration.** 2.ed. New York: Van Nostrand Reinhold. 1990.

[2] BRUNTON, S. L.; NOACK, B. R.; KOUMOUTSAKOS, P. **Machine Learning for Fluid Mechanics.** Annual Review of Fluid Mechanics. 52:1, 477-508. 2020.

[3] BRYC, W. **Normal distribution: characterizations with applications.** Lecture Notes in Statistics. 100. Springer, Berlin. 1995.

[4] BUCCOLIERI, R.; SANTIAGO, J. L.; MARTILLI, A. **CFD modelling: The most useful tool for developing mesoscale urban canopy parameterizations.** Build. Simul. 14, 407–419. 2021.

[5] BUK JUNIOR, L. **Estudo numérico do escoamento ao redor de um cilindro fixo:** dissertação apresentada à escola politécnica da universidade de São Paulo para a obtenção do título de mestre em engenharia mecânica. 62 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Departamento de Engenharia Mecânica, Universidade de São Paulo, São Paulo. 2007.

[6] CABANEROS, S.M.; CALAUTIT, J.K.; HUGHES, B.R. **A review of artificial neural network models for ambient air pollution prediction.** Environ. Model. Software. 119, 285–304. 2019.

[7] CHATZIMICHAILIDIS, A. E. et al. **Implicit Definition of Flow Patterns in Street Canyons—Recirculation Zone—Using Exploratory Quantitative and Qualitative Methods.** Atmosphere. 10, no. 12: 794. 2019.

[8] CHENG, J. C. P. et al. **BIM-supported sensor placement optimization based on genetic algorithm for multi-zone thermal comfort and IAQ monitoring.** Building and Environment. Volume 216, 108997. 2022.

[9] Demographia World Urban Areas: 14th Annual Edition. Demographia. 201804. 2018.

[10] DU, Y.; BLOCKEN, B.; PIRKER, S. **A novel approach to simulate pollutant dispersion in the built environment: Transport-based recurrence CFD.** Building and Environment. Volume 170, 106604. 2020.

[11] FERZIGER, J. H.; PERIĆ, M. Computational Methods for Fluid Dynamics. 3rd Berlin: Springer. 2002.

[12] **Frobenius Norm.** Available at: https://mathworld.wolfram.com/FrobeniusNorm.html. Access date: 03 November 2023.

[13] GALLAGHER, J.; LAGO, C.; **How parked cars affect pollutant dispersion at street level in an urban street canyon? A CFD modelling exercise assessing geometrical detailing and pollutant decay rates.** Science of The Total Environment. Volume 651, Part 2, Pages 2410-2418. 2019.

[14] GAN, V. J. L. et al. **BIM and Data-Driven Predictive Analysis of Optimum Thermal Comfort for Indoor Environment.** Sensors. 21, no. 13: 4401. 2021.

[15] GANTI, H.; KHARE, P. **Data-Driven Surrogate Modeling of Multiphase Flows Using Machine Learning Techniques.** Computers & Fluids, 211, 104626. 2020.

[16] Gaussian Process Regression Models. Published at: https://www.mathworks.com/help/stats/gaussian-process-regression-models.html. Access date: 28 October 2023.

[17] GOLUB, G. H.; VAN LOAN, C. F. **Matrix Computations**. 3rd ed. Baltimore, MD: Johns Hopkins. 1996.

[18] GRAZIA BADAS, M. et al. **Urban areas parameterisation for CFD simulation and cities air quality analysis.** International Journal of Environment and Pollution. 66:1-3, 5-18. 2019.

[19] HANG, J.; SANDBERG, M.; LI, Y. **Age of air and air exchange efficiency in idealized city models.** Building and Environment - BLDG ENVIRON. 44. 1714-1723. 2019.

[20] HANG, J. et al. **Investigation of O3–NOx–VOCs chemistry and pollutant dispersion in street canyons with various aspect ratios by CFD simulations.** Building and Environment. Volume 226, 109667. 2022.

[21] HASHAD, K. et al. **Designing roadside green infrastructure to mitigate traffic-related air pollution using machine learning.** Science of The Total Environment. Volume 773, 144760. 2021.

[22] Horizon Europe. **Groundbreaking tools and models to reduce air pollution in urban areas.** Available at: cordis.europa.eu/project/id/101072559. Access date: 17 July 2023.

[23] HOU, F. et al. **Prediction and optimization of thermal comfort, IAQ and energy consumption of typical air-conditioned rooms based on a hybrid prediction model.** Building and Environment. Volume 225, 109576. 2022.

[24] **How to use the gaussian process regression function in matlab 2015b?** Available at: https://www.mathworks.com/matlabcentral/answers/254981-how-to-use-the-gaussian-process-regression-function-in-matlab-2015b. Access date: 03 November 2023.

[25] KADAVERUGU, R. et al. **High Resolution Urban Air Quality Modeling by Coupling CFD and Mesoscale Models: a Review.** Asia-Pacific J Atmos Sci, 55, 539–556. 2019.

[26] KOCIJAN, J. et al. **Surrogate modelling for the forecast of Seveso-type atmospheric pollutant dispersion.** 2022.

[27] KWAK, K.-H. et al. **On-Road Air Quality Associated with Traffic Composition and Street-Canyon Ventilation: Mobile Monitoring and CFD Modeling.** Atmosphere. 9, 92. 2018.

[28] LAURET, P. et al. **Atmospheric dispersion modeling using Artificial Neural Network based cellular automata.** Environ. Model. Software. 85, 56–69. 2016.

[29] LAURIKS, T. et al. **Application of Improved CFD Modeling for Prediction and Mitigation of Traffic-Related Air Pollution Hotspots in a Realistic Urban Street.** Atmospheric Environment. Volume 246, 118127. 2021.

[30] LIU, J.; NIU, J. **CFD simulation of the wind environment around an isolated high-rise building: An evaluation of SRANS, LES and DES models.** Building and Environment, 96, 91–106. 2016.

[31] MA, H. et al. **CFD simulation of flow fields and pollutant dispersion around a cubic building considering the effect of plume buoyancies.** Building and Environment. Volume 208, 108640. 2022.

[32] MARTINEZ-SANCHEZ, Á. et al. **Data-driven assessment of arch vortices in simplified urban flows.** International Journal of Heat and Fluid Flow. Volume 100, 109101. 2023.

[33] MASOUMI-VERKI, S. et al. **A review of advances towards efficient reduced-order models (ROM) for predicting urban airflow and pollutant dispersion.** Building and Environment. Volume 216, 108966, ISSN 0360-1323. 2022.

[34] MEAD, M.I. et al. **The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks.** Atmos. Environ. 70, 186–203. 2013.

[35] MEI, D.; LIU, C.-H. **Bi-objective optimization of traffic assignment with air quality consideration via CFD-based surrogate model.** Sustainable Cities and Society. Volume 91, 104425. 2023.

[36] MEI, S.-J. et al. **Street canyon ventilation and airborne pollutant dispersion: 2-D versus 3-D CFD simulations.** Sustainable Cities and Society. Volume 50, 101700. 2019.

[37] MENDIL, M. et al. **Hazardous atmospheric dispersion in urban areas: A Deep Learning approach for emergency pollution forecast.** Environmental Modelling & Software. Volume 152, 105387. 2022.

[38] MORADPOUR, M.; HOSSEINI, V. **An investigation into the effects of green space on air quality of an urban area using CFD modeling.** Urban Climate. Volume 34, 100686. 2020.

[39] PANCHIGAR, D. et al. **Machine learning-based CFD simulations: a review, models, open threats, and future tactics.** Neural Comput & Applic. 34, 21677–21700. 2022.

[40] PANTUSHEVA, M. et al. **Air Pollution Dispersion Modelling in Urban Environment Using CFD: A Systematic Review.** Atmosphere, 13, 1640. 2022.

[41] QIN, P.; RICCI, A.; BLOCKEN, B. **CFD simulation of pollutant dispersion in a street canyon: Impact of idealized and realistic sources.** E3S Web of Conf. 396, 02042. 2023.

[42] QUILODRAN CASAS, C. et al. **A Reduced Order Deep Data Assimilation model.** Physica D: Nonlinear Phenomena. Volume 412, 132615. 2020.

[43] QUILODRAN-CASAS, C.; ARCUCCI, R. **A data-driven adversarial machine learning for 3D surrogates of unstructured computational fluid dynamic simulations.** Physica A: Statistical Mechanics and its Applications. Volume 615, 128564. 2023.

[44] RASMUSSEN, C. E.; WILLIAMS, C. K. I. **Gaussian Processes for Machine Learning**. The MIT Press. 2006.

[45] RODRIGUES, M. C. J.; POSTOLACHE, O.; CERCAS, F. **Unobtrusive Cardio-Respiratory Assessment for Different Indoor Environmental Conditions.** IEEE Sensors Journal. vol. 22, no. 23, pp. 23243-23257. 2022.

[46] ROGER, P. **Handbook of Computational Fluid Mechanics.** London: Academic Press. 1996.

[47] SIN, C. H. et al. **CFD Evaluation of Ventilation and Pollutant Dispersion Withindiscontinuity Urban Streets.** 2023.

[48] TEE, C.; NG, E.Y.K.; XU, G. **Analysis of transport methodologies for pollutant dispersion modelling in urban environments.** Journal of Environmental Chemical Engineering. Volume 8, Issue 4, 103937. 2020.

[49] THUNIS, P. et al. **Overview of current regional and local scale air quality modelling practices: assessment and planning tools in the EU.** Environ. Sci. Pol., 65, 13–21. 2016.

[50] TOPARLAR, Y. et al. **A review on the CFD analysis of urban microclimate.** Renew. Sust. Energ. Rev. 80, 1613–1640. 2017.

[51] Tutorial by Asmaa Hadane. Available at https://github.com/AsmaaHADANE/Youtube-Tutorials.

[52] UN Department of Economic and Social Affairs. UNDESA. 2018.

[53] VAPNIK, V. **Principles of risk minimization for learning theory.** In Proceedings of the 4th International Conference on Neural Information Processing Systems (NIPS'91). Morgan Kaufmann Publishers Inc., San Francisco, CA, 1991, USA, 831–838. 1991.

[54] VERSTEEG, H. K.; MALALASEKERA, W. **An Introduction to Computational Fluid Dynamics: The Finite Volume Method.** Prentice Hall, Harlow. 1995.

[55] VIANA, F. A. C. **A Tutorial on Latin Hypercube Design of Experiments.** Qual. Reliab. Engng. Int. 32, 1975– 1985. 2016.

[56] VIDAL, R.; MA, Y.; SASTRY, S. **Generalized Principal Component Analysis (GPCA)**. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2003.

[57] WAI, K.-M.; Yu, P. K. N. **Application of a Machine Learning Method for Prediction of Urban Neighborhood-Scale Air Pollution.** International Journal of Environmental Research and Public Health. 20, no. 3: 2412. 2023.

[58] WANG, B.; QIAN, F.; ZHONG, W. **Wind field reconstruction for the dispersion modeling of accidental chemical spills on complex geometry.** Chinese Journal of Chemical Engineering. Volume 27, Issue 11, Pages 2712-2724. 2019.

[59] WANG, J. **An Intuitive Tutorial to Gaussian Processes Regression.** 2020.

[60] WANG, J. et al. **Evacuation route optimization under real-time toxic gas dispersion through CFD simulation and Dijkstra algorithm.** Journal of Loss Prevention in the Process Industries. Volume 76, 104733. 2022.

[61] WEISS, J. **A Tutorial on the Proper Orthogonal Decomposition.** AIAA Aviation Forum, Dallas, Texas, United States. 17–21 June 2019.

[62] World Health Organization. **Ambient (outdoor) air pollution.** 19 Dec. 2022. Available at: www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health. Access date: 17 July 2023.

[63] WU, Q. et al. **A System Coupled GIS and CFD for Atmospheric Pollution Dispersion Simulation in Urban Blocks.** Atmosphere. 14, 832. 2023.

[64] WU, Y.; CHEN, H. **Optimizing block morphology for reducing traffic pollutant concentration in adjacent external spaces of street canyons: A machine learning approach.** Building and Environment. Volume 242, 110587. 2023.

[65] XIANG, S. et al. **Fast simulation of high resolution urban wind fields at city scale.** Urban Clim. 39, 100941. 2021.

[66] XIANG, S. et al. **Non-intrusive reduced order model of urban airflow with dynamic boundary conditions.** Build. Environ. 187, 107397. 2021.

[67] XIAO, D. et al. **A reduced order model for turbulent flows in the urban environment using machine learning.** Building and Environment. Volume 148, Pages 323-337. 2019.

[68] XIAO, D. et al. **Machine learning-based rapid response tools for regional air pollution modelling.** Atmos. Environ. 199, 463–473. 2019.

[69] ZHANG, Y. et al. **CFD-based analysis of urban haze-fog dispersion—A preliminary study.** Build. Simul. 14, 365–375. 2021.

[70] ZHENG, X.; YANG, J. **CFD simulations of wind flow and pollutant dispersion in a street canyon with traffic flow: Comparison between RANS and LES.** Sustainable Cities and Society. Volume 75, 103307. 2021.

Chapter 7

Appendix

7.1 Frobenius norm

The Frobenius norm is a matrix norm of an $m \times n$ matrix A defined as:

$$||A||_F = \sqrt{\Sigma_{i=1}^{m} \Sigma_{j=1}^{n} |a_{ij}|^2}. \tag{7.1}$$

(GOLUB and VAN LOAN, 1996)

7.2 POD definitions

Let $\boldsymbol{u'}(\boldsymbol{x}, t)$ be the fluctuating velocity vector in a flow, defined as:

$$\boldsymbol{u'} = \boldsymbol{U} - \bar{\boldsymbol{U}}, \tag{7.2}$$

where $\boldsymbol{U} = (U, V, W)$ is the velocity vector, $\bar{\boldsymbol{U}}$ its temporal mean, $\boldsymbol{x} = (x, y, z)$ the position vector and t time. According to WEISS (2019), POD is the decomposition of the random vector field $\boldsymbol{u'}(\boldsymbol{x}, t)$ into a set of deterministic spatial functions $\boldsymbol{\Phi_k(x)}$, also known as the POD (spatial) modes, modulated by random time coefficients $\alpha_k(t)$ so that:

$$\boldsymbol{u'}(\boldsymbol{x}, t) = \sum_{k=1}^{\infty} \alpha_k(t) \boldsymbol{\Phi_k(x)}. \tag{7.3}$$