Imperial College
London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF MATHEMATICS

BSC RESEARCH PROJECT

Model for Air Quality Prediction from Traffic Data

Author:
Dugelay Eliot

Supervisors:
Dr. Prasun Ray
Prof. Demetrios Papageorgiou

Submitted in partial fulfillment of the requirements for the BSc in Mathematics at Imperial College London

Spring 2024

Abstract

In this project, we develop a model aiming at predicting air pollution levels based on traffic data within a specific area. The model considers both traffic emissions and meteorological conditions, with a particular focus on wind. Our study area is a section of the M25, a major highway surrounding London, England. To model air diffusion, we employ advection-diffusion PDEs. Advection is solved numerically using an explicit upwind scheme, while diffusion using implicit Crank-Nicolson method. Traffic emissions are included as a source term in the PDEs, with wind influencing the advection component. Finally, we analyse the obtained results, compare them against pollution concentration guidelines, and discuss the model's effectiveness.

Plagiarism statement

The work contained in this thesis is my own work unless otherwise stated.

Signature: Dugelay Eliot
Date: June 10, 2024

Contents

List of Figures

Chapter 1

Introduction

This chapter serves as both a motivation for our project and an introduction to key concepts related to air pollution and air transport modelling. Our project's mission is to estimate air quality from traffic emissions.

1.1 Motivation

Air pollution, defined as the contamination by any chemical, physical or biological agent that modifies the natural characteristics of the atmosphere, represents a significant threat to public health and environmental sustainability.
The World Health Organization (WHO)[1] raises the point that 99% of the global population breathe air that exceeds the WHO guideline limits[1] and contains high levels of pollutants. This air pollution, which includes fine particulate matter, contributes to various health issues such as strokes, heart diseases, lung cancer, acute and chronic respiratory diseases. According to WHO, in 2019, outdoor air pollution is estimated to have caused 4.2 million premature deaths worldwide.
It is clear that addressing air pollution is a critical societal challenge that we must tackle to protect both today's humans and future generations.

Estimating air quality is complex as it involves multiple different factors. For instance, there are many different sources of pollution, such as industry, coal burning, household heating or transportation, as outlined in subsection 1.2.2. In addition, chemical interactions between pollutants or meteorological conditions also influence pollution concentrations.
Transportation, among other sectors as detailed in subsection 1.2.2, is often identified as a major pollution source. As an example, in 2020, transport was the largest emitting sector of greenhouse gas emissions in the UK, producing 24% of the country's total emissions[2]; traffic congestion, slow-moving buses and private vehicles contribute to high levels of NO_2 pollution. Especially in urban areas, traffic related air pollution is a big issue, and accurately figuring out its impact is challenging. Lots of regions worldwide are still not equipped with pollution monitors to measure ambient air quality, which is essential to ensure inhabitants' health. Furthermore, transportation emissions contribute to greenhouse gas emissions, which have detrimental environmental effects.

In June 2023, Laura Oporto Lisboa, a student from the University of Bath, published a the-

[1] World Health Organization - Air Pollution
[2] GOV.UK - Transport and environment statistics 2022

sis titled "Hybrid Physical-Statistical Model for Air Quality Prediction from Traffic Data"[2]. In this thesis, they propose a model to forecast air pollution levels within an area of a city based on traffic data and meteorological conditions. While their research focuses on a specific area of Madrid, Spain, they suggest that their model is transferable to other areas. Inspired by their work, our project aims to predict air pollution from traffic data for an area of London, England, using the model developped in the their thesis [2].

1.2 Air pollution

1.2.1 Air pollutants

Air pollutants refer to various substances present in the air that can be harmful to human health and the environment. They are typically divided into two categories: primary and secondary [3]. Primary pollutants are those formed and emitted directly from a source into the atmosphere. Secondary pollutants, on the other hand, are not directly emitted, but form when other pollutants (primary pollutants) react in the atmosphere. The latter are harder to control because they have different ways of synthesising, and their formation is not well understood.

Carbon monoxide (CO), a primary air pollutant, is formed by the incomplete combustion of carbonaceous fuels such as coal, wood, petrol, natural gas and kerosene. It is a poisonous gas that we cannot see, taste or smell. Exposure to CO can cause difficulties breathing, exhaustion, dizziness, and other flu-like symptoms, and can be deadly at high exposure.

Another main primary air pollutant is Nitrogen dioxide (NO_2), a major air pollutant of concern in London. Nitrogen oxides (NO_x), composed from both NO_2 and NO molecules, are produced by combustion sources. Nitrogen dioxide is responsible for most detrimental health effects.

Sulphur dioxide (SO_2), a primary air pollutant, is a highly reactive gas with a pungent and irritating smell. It is formed by the combustion of fossil fuels at power plants and other industrial facilities.

A last main primary air pollutant is Particulate matter (PM). It can be made up of a variety of components including nitrates, sulfates, organic chemicals, metals, soil or dust particles, and allergens. We distinguish PM10 (particles with a diameter $\leq 10\mu m$) and PM2.5 (particles with a diameter $\leq 2.5\mu m$). The first are small enough to pass through the nose and throat and enter the lungs. Once inhaled, these particles can affect the heart and lungs and cause serious health effects. The second are so small they can get deep into the lungs and into the bloodstream. There is sufficient evidence that exposure to PM2.5 over long periods (years) can cause detrimental health effects.

In urban areas, the main secondary air pollutant is Ozone (O_3) at ground level, an unstable, highly reactive gas and a major component of smog. It can be found in two areas of the Earth's atmosphere: in the upper atmosphere and at ground level. While ozone in the upper atmosphere protects us by filtering out ultraviolet radiation from the sun, ozone at ground level is a threat for our health. O_3 is formed through chemical reactions between NO_x and Volatile Organic Compounds (VOCs)[3], in the presence of solar radiation.

[3] Organic chemicals emitted by fuel combustion and industrial processes.

1.2.2 Main sources

Sources of air pollution are multiple. As mentioned before, CO is formed by the incomplete combustion of carbonaceous fuels. Hence, it comes from any source that burns fuel such as cars, trucks, construction and farming equipment, residential heaters and stoves.

Similarly, NO_2 forms from combustion and, so, comes from similar sources, such as motor vehicles, power plants, and industrial facilities.

Particulate matter is emitted from cars and trucks (especially diesel vehicles), fireplaces, woodstoves, agricultural activities and construction sites.

SO_2 is produced by the burning of fossil fuels containing sulfur, such as coal and oil. Major sources include power plants, industrial facilities, and household heating systems that use coal or oil. Additionally, some natural sources, such as volcanic eruptions, can contribute to SO_2 levels in the atmosphere.

As we observe in figure 1.1, road transport often has a significant impact on the formation of pollutants. Especially, it is the main source of NO_x contributing to 40.56% of total emissions. In addition, it has an important impact on CO production, accounting for 24.22% of total emissions. For PM, it contributes approximately 10% of the total emissions.

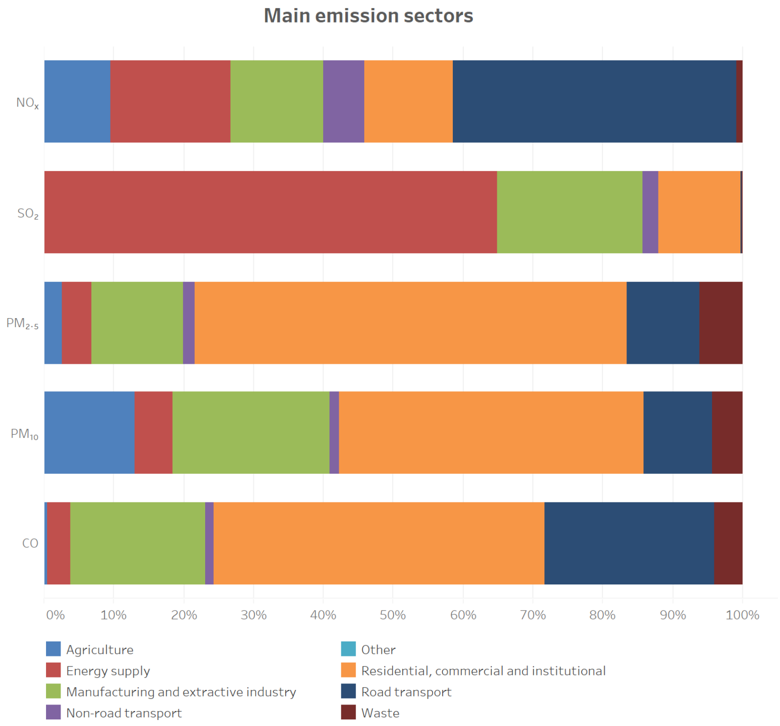


Figure 1.1 Main source sectors of pollutants in 2021[6]

1.2.3 Other factors

Other factors can also influence pollutant concentrations. First of all, on the vertical scale, air quality is affected by the atmospheric boundary layer (ABL)[4] flows and its interaction with obstacles. The thickness of the ABL is not constant and can range from just a few metres to several kilometres, depending on the local meteorology. More precisely, during the day and in the warm season, it tends to be thicker (∼1km), while at night and in the cool season, it tends to be thinner (∼100m). This variation is due to changes in wind speed and the thickness of the air depending on the temperature. Above the ABL is the free atmosphere.

Changes in the height of the ABL have been shown to have significant effects on air pollution levels. For instance, article [7] has demonstrated a significant negative correlation existed between ABL height and PM2.5 concentrations. Similarly, studies such as [8] have shown that diurnal cycles of CO and NO_x are also influenced by the ABL height. A low ABL height and increasing traffic emissions in the morning cause high levels of CO and NO_x concentrations. During the day, the ABL height increases, partly due to rising temperature, leading to a decrease in pollutants concentrations. In the evening, a rapid decrease in the ABL height and a rapid increase in traffic flow (e.g. workers going back home) result in high pollution levels.

As expected, wind also plays a crucial role in the dispersion of pollutants. It can transport pollutants over large distances and influence their distribution in the atmosphere. Therefore, wind direction plays also an important role in determining the trajectory of pollutants emitted from point sources such as industrial facilities or major roadways. This helps identify communities at risk of pollution, even if they are not near a source, as pollutants can be carried downwind.

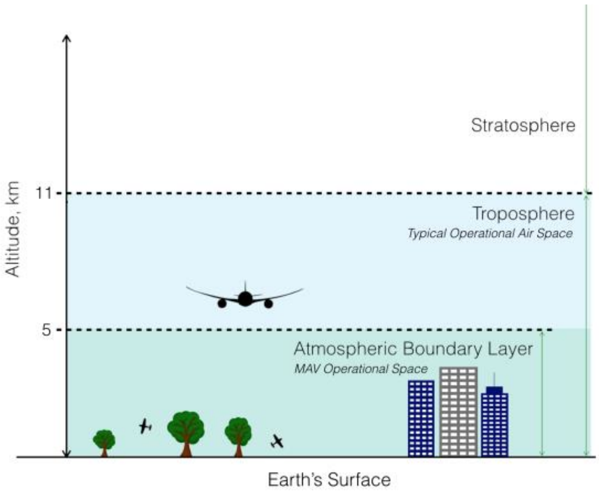


Figure 1.2 Structure of the Troposphere with the Atmospheric boundary layer (ABL)

In the case of London, its geography can lead to periods of intense air pollution. Although there are no mountains near London, it is surrounded by hills that form an air basin, which can trap air pollutants, especially when wind flow is low and temperature inversions occur. Furthermore, most of London's air pollution comes from road transport: traffic congestion, including slow-moving buses and private vehicles contribute to high levels of NO_2 pollution in certain parts of the city.

[4] Lowest part of the atmosphere in direct contact with the surface

1.3 Modelling road traffic pollution

Pollutant dispersion can be modelled by an advection-diffusion-reaction PDEs[10]:

$$\partial_t c_i + \nabla \cdot (\mathbf{w}\, c_i) = D\nabla^2 c_i + R_i - L_i + S_i$$

where $\mathbf{x} \in \mathbb{R}^3$ and $c_i(\mathbf{x}, t)$ represents the concentration of the ith pollutant at position $\mathbf{x}$ and time t. The second term denotes advection[5] by the wind. Additionally, pollutants also disperse in different directions through diffusion, represented by the third term in the PDEs. We integrate reaction R_i and loss L_i terms, as some pollutants, such as NO_x might be higly reactive. Finally, a source term is added in the last term of the PDEs, which, in our case, will be traffic emissions.

1.4 Conditions of the project

In this project, following the approach developped in the thesis [2], we will consider several simplifications. First of all, we will consider traffic as the only source of pollutant emissions. In particular, this means that we will omit any chemical reaction of the pollutants, i.e. the terms R_i and L_i in the PDEs(1.3) will be omitted.

Furthermore, we will assume no flux or deposition of pollutants through the ground, i.e. $\hat{\mathbf{n}} \cdot \nabla c_i = 0$. Additionally, we will assume Neumann boundary conditions on the lateral and top boundaries: $\hat{\mathbf{n}} \cdot \nabla c_i = 0$. This implies that losses of pollutants through the boundaries of the domain arise only from advection by the wind.

Finally, as mentionned in 1.2.3, ABL height influences air pollutant levels. However, taking it into account requires refined meteorological parameters and continuous vertical profiles of air pollutants. Hence, for simplification, we will work only in a 2D space.

1.5 Code

The code developped for this project is available at: Bachelor-Project.git

[5] Transportation of a quantity by the movement of a fluid or medium (in our case the wind).

Chapter 2

Preliminary data exploration

A major part of this project has involved searching for meaningful and suitable data, which is often not as simple as we can think, as many data are either unavailable or unusable. Therefore, a first important step was choosing an appropriate study area with plenty of data to exploit.

2.1 Study area

For this project, a section of the M25 London motorway has been chosen as the study area.

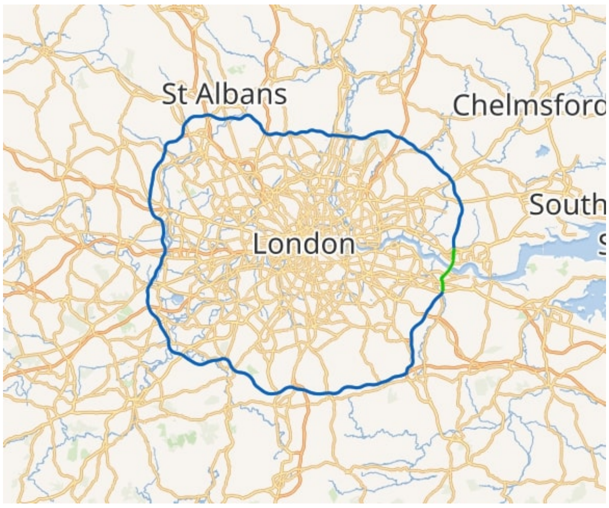


Figure 2.1 Map of the M25

2.1.1 The M25

The M25, or London Orbital Motorway, is a major ring road that encircles most of greater London and is one of the busiest motorways in the UK. When Margaret Thatcher, British Prime Minister at that time, opened the final section in 1986, the M25 was the longest ring road in Europe. The M25 passes through five counties: Buckinghamshire, Essex, Hertfordshire, Kent, and Surrey.
Overall, the project took 11 years to be achieved and used more than two million tonnes of

concrete and 3.5 million tonnes of asphalt[1]. The 188 kilometers (117 miles) motorway cost over £900 million in 1986, worth more than £2.5 billion today. Over 200.000 vehicles a day use the M25 now, about 15% of all UK motorway traffic.
The M25 was one of the first motorway projects to prompt environmental concerns, leading to nearly 40 public inquiries. Today, it remains a central point for environmental protests.

Given its history and environmental concerns, the M25 seemed to be an interesting motorway to analyse.

Figure 2.2 May 11 1979: Residents from Enfield demonstrating against the proposed M25 Link Road.

2.1.2 Choice of a study location

Across the 188 kilometers (117 miles) of the M25, our initial task was to select a smaller, specific study area to focus our analysis on. Our search criteria included identifying a section of the M25 with multiple entrances and exits, a minimal number of adjacent roads, and plenty of recent data available.
After consideration, the section depicted in 2.3 appeared to meet all the criteria. Located in the North-West of London, near St Albans and Bricket Wood, it contains clockwise (resp. anti-clockwise) 3 entrances and 1 exit (resp. 1 entrance and 2 exits), for a total of 4 entrances and 3 exits.
Furthermore, the choice of this location enables us to also address the environmental impacts of the M25 on nearby towns such as Bricket Wood, How Wood and St Albans, in our study area.

2.2 Traffic flow data

National Highways[12] is the government company charged with operating, maintaining and improving England's motorways and major roads. It plays a key role in the English road network, ensuring that the routes are safe, reliable, and efficient. Note that finding a reliable source is essential to be sure to exploit reliable and suitable data. In our case, we can be reasonably confident in the credibility of this source.

National Highways provides a wide range of open datasets to the public, which can be used for various purposes such as research or public information. These datasets include information

[1] Asphalt, a mixture of aggregates, binder and filler, is the surfacing material for over 95% of all UK roads as well as for footpaths, playgrounds, cycle ways and car parks.

on traffic flow, accidents, delays and road conditions over many years.
For this project, we are interested in the traffic flow data[13]. National Highways provides a map[14] showing the position of every traffic scanner and their available data. Looking at our study area, we notice many scanners with plenty of recent data available.

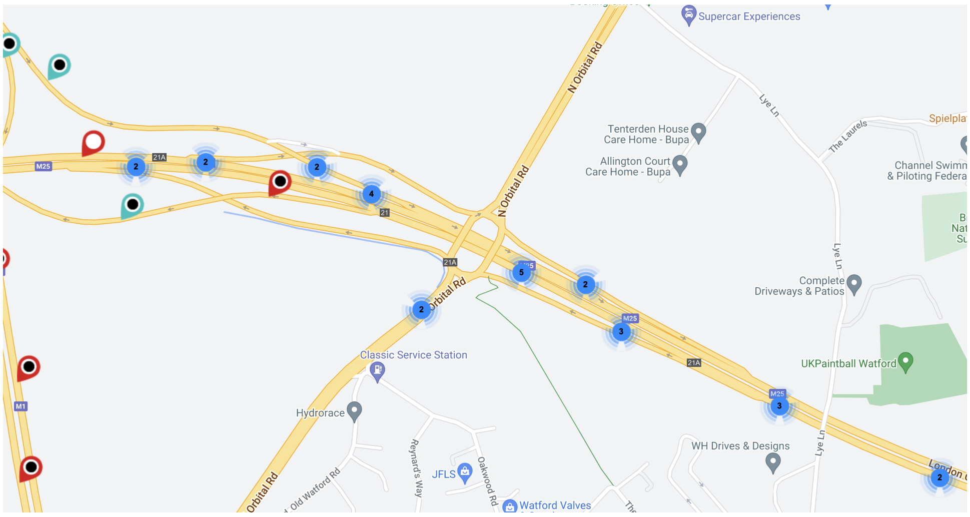


Figure 2.3 Scanners in the study area[14]

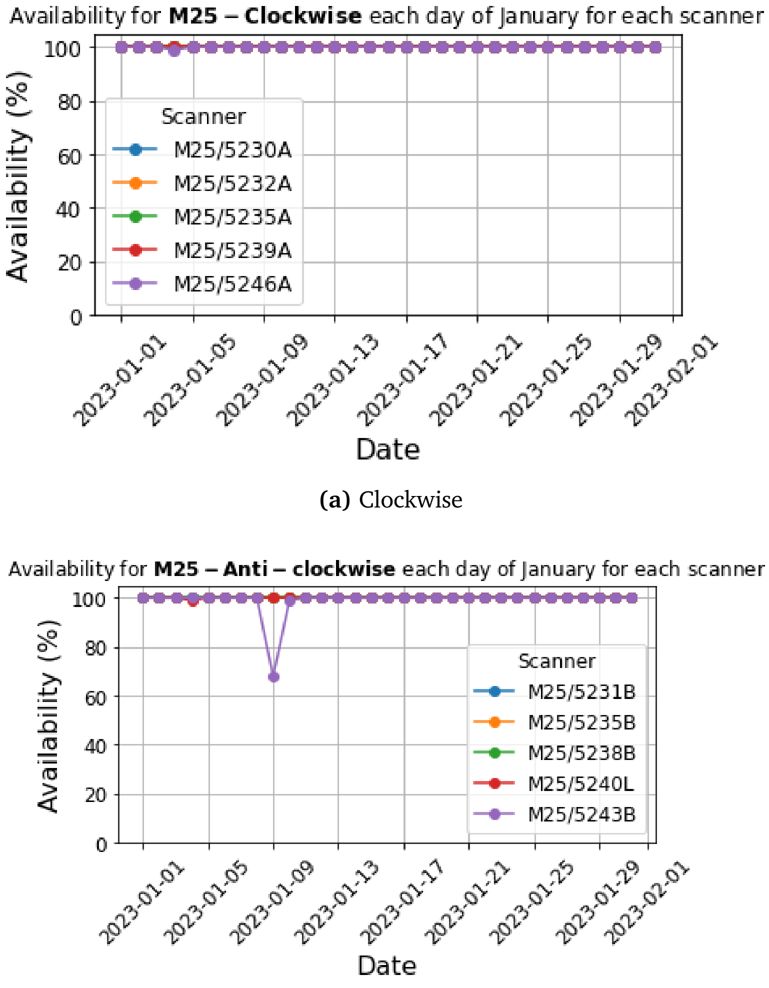
We select two datasets for the clockwise and anti-clockwise directions, respectively. Both contain data recorded every 15 minutes for the entire month of January. Having obtained the data, we can proceed with an exploratory analysis. The data contain:

- Scanner name (e.g. M25/5230A),
- Report date and time, for all days of January 2023 every 15 minutes,
- Length of cars,
- Average speed (mph), which we will use in (3.1),
- Total volume (in vehicles).

In this project, we will not use the variable 'Length of cars'. Nevertheless, it could be interesting to consider it, as it may, for example, be relevant to distinguish between trucks and cars.

First of all, let's examine the data availability. Since, each scan occurs every 15 minutes, we should have 96 scans per day per scanner. We can thus determine the proportion of scans we actually have.
We observe in figure 2.4 that a very good proportion of the data is available. However, for $M25/5243B$, some data appear to be missing on January 9th. When checking the number of missing values, with $.isna()$ in Python, for 'Total Volume', we find that there is 1 missing value for the clockwise dataset and 33 missing values for the anti-clockwise dataset.
Since the number of NA values is very small compared to the total number of scans, a reasonable approach would be to interpolate the missing values using 'a mean' of the available values.

Nevertheless, as observed in 2.5, the 'Total Volume' appears to vary significantly depending on the time of the day, following kind of a pattern. Hence, it is important to take this observation into account and compute the mean of data from the same time of day.



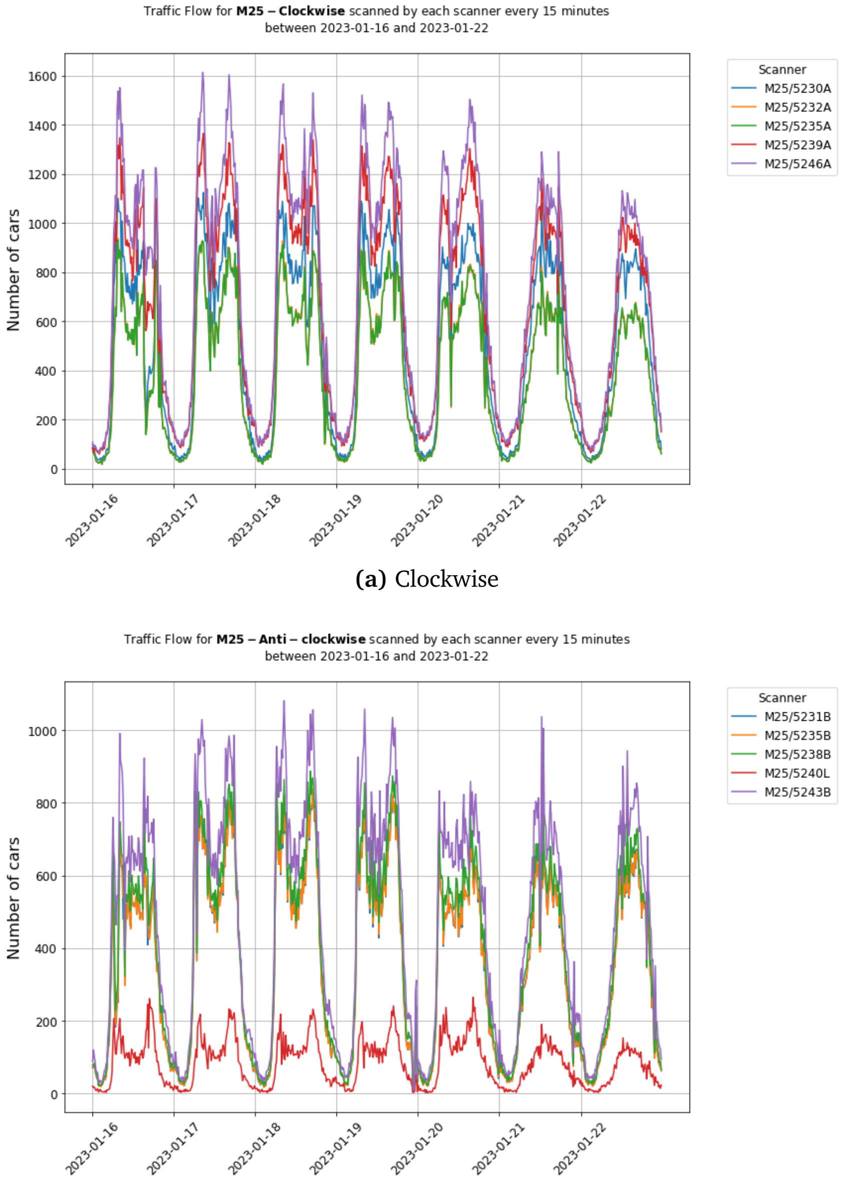
(a) Clockwise

(b) Anti-Clockwise

Figure 2.4 Scanners availability

Furthermore, some values are also missing for the average speed. Since the speed limit is always the same on the M25 (70mph/112kmh), and figure A.1 does not show any discernible pattern, we decide to interpolate the missing values by taking a general mean of all the available average speed data.

Notice the unique behavior of scanner M25/5240L (Anti-Clockwise) in both 2.5 and A.1. It can be explained by the fact that it is located at an exit, while other scanners are positioned on the M25 between entrances and exits. As the exit is positioned after scanner M25/5243B and before scanner M25/5238B, we expect the 'Total Volume' from M25/5240L and M25/5238B to add up to the 'Total Volume' from M25/5243B, which is approximately observed graphically in figure 2.5.



(a) Clockwise

(b) Anti-Clockwise

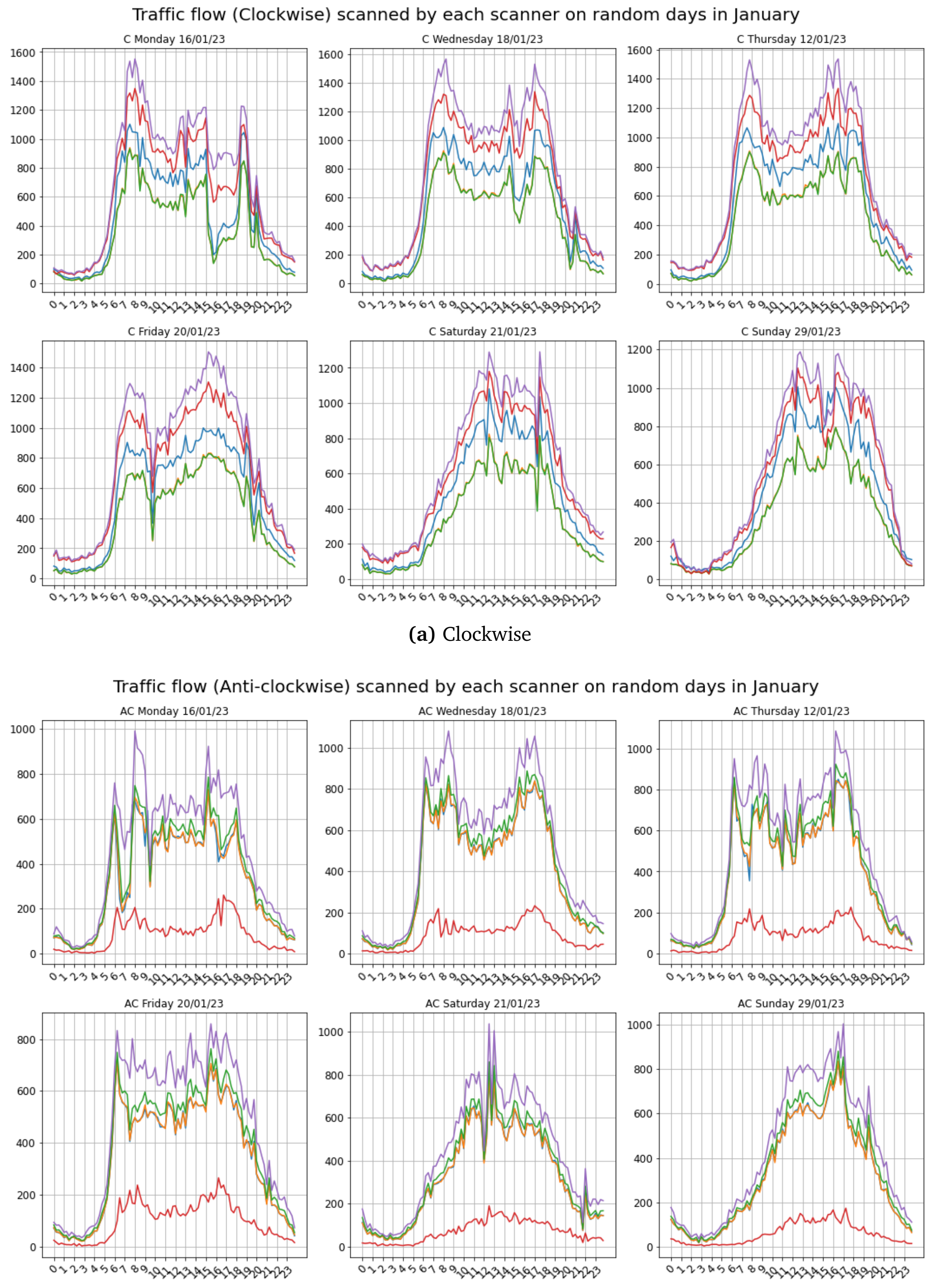
Figure 2.5 Traffic Flow

Next, let's observe the traffic flow in our study area for some specific days in January, see 2.6. First of all, we observe a clear difference between weekdays and weekends. On weekdays, there is a significant increase in traffic between 5am and 8am, peaking around 8am. Then, we observe a small decrease, followed by stagnation until about 5pm, then a small peak after which there is a significant decrease. These tendencies are consistent with typical work hours.

In contrast, during weekends, the increase in traffic in the morning is less pronounced and extends from 6am to noon, with a peak around noon. The traffic then does not decrease significantly until around 6pm, after which it decreases until midnight.

This difference is consistent with the fact that fewer people go to work on weekends, opting instead for afternoon activities.

From these observations, we can conclude that the data appear to be logical and consistent with expected traffic patterns. This indicates that the data are reliable and suitable for further analysis.



(a) Clockwise

(b) Anti-Clockwise

Figure 2.6 Traffic Flow for some days in January

2.3 Wind data

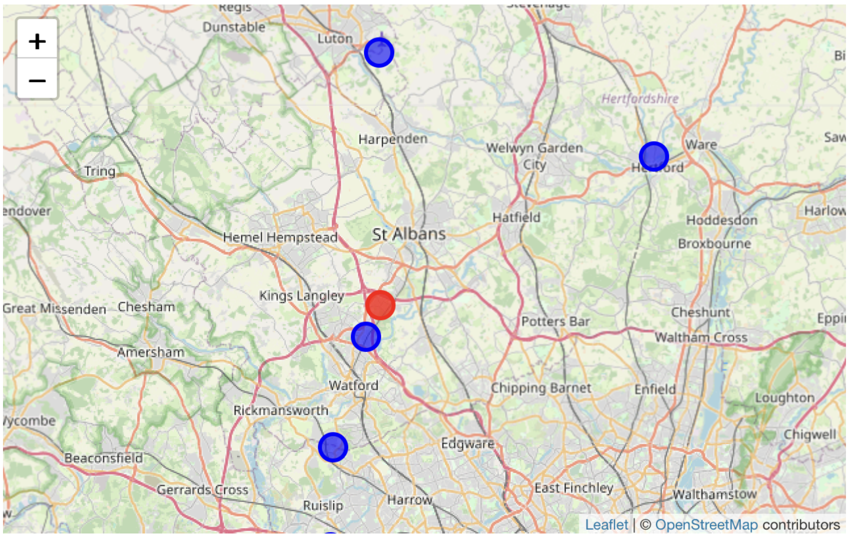


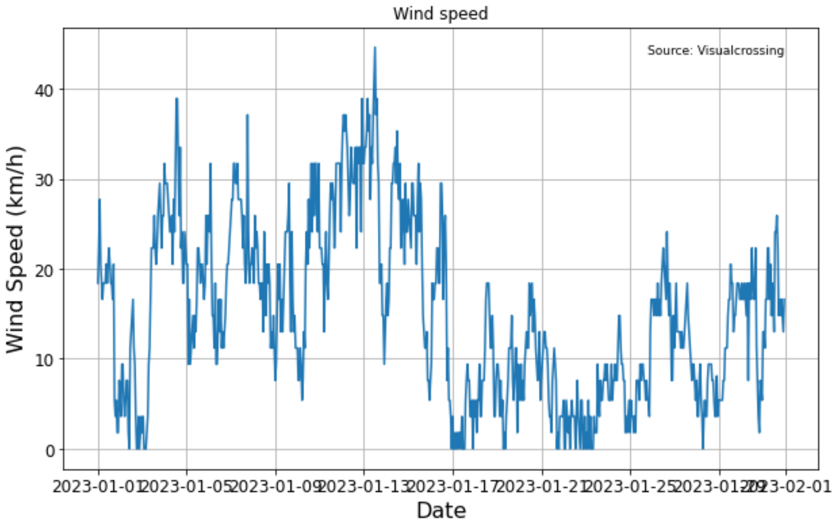
Figure 2.7 Meteorological monitors

Another important factor in our project is the wind. Therefore, finding high-quality wind data is essential as it will play a major role in the diffusion of the pollution in our model.
Visual Crossing[15], founded in 2003, stands out as a premier provider of weather data and enterprise analysis tools for data scientists, business analysts, professionals, and academics. On their website we can find that for nearly 20 years they have been working with some of the largest businesses in the world. Hence, we can reasonably trust the credibility of this source.

As shown in figure 2.7, there are no monitors located precisely within our study area. Hence, a reasonable solution is to take the average of the data from all the nearby available stations.

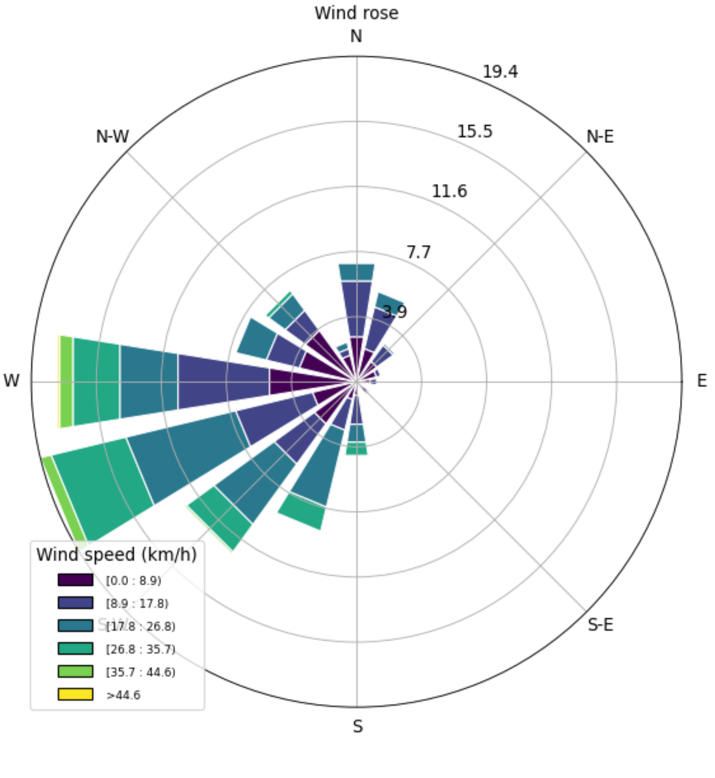
The first figure 2.8 shows the wind speed (in kmh) recorded hourly for the entire month of January. Despite a general trend indicating lower wind speeds during the second half of January, and multiple fluctuations, no discernible pattern is evident from the data.

The second figure 2.9 illustrates the wind direction (and wind speed) for the entire month of January. It is immediately noticeable from it that in majority the wind came from the West or South-West directions. Moreover, the wind speeds seem consistent with those from figure 2.8.



Source: Visualcrossing

Figure 2.8 Wind Speed in January (kmh)



Source: Visualcrossing

Figure 2.9 Wind Rose for the month of January

From these observations, we expect the pollution during the first half of January to be generally more concentrated in the North-East or East regions of our area. During the second half of January, it should be more concentrated along our section of the M25.

2.4 Pollution concentration data

The aim of this work is to estimate pollution concentration from traffic data. When creating a model, an essential part of the process is to verify its credibility by comparing it with known values. However, in our case, no pollution monitor is located in our study area. Nevertheless, the model has already been tested in [2] on the city of Madrid, leading to satisfactory results: "... we showed that we can have very accurate results using simpler traffic data, usually accessible, to predict pollution levels within a city ... Finally, this model is transferable to any city area and, with the appropriate modifications, it can be turned into a forecasting algorithm for a whole city." Hence, we can be quite confident about the performance of this model. It suggests that this model can be used to estimate pollution at locations without pollution monitors.

For exhaustiveness, we still present how we would have proceeded using pollution data from a location not too far from our study area.

Air Quality England (AQE)[16] is a resource for local air quality information and air quality data provision, funded and hosted by *Ricardo Energy & Environment* in 2020. The closest data available are from Hatfield, a small city with about 40,000 inhabitants, approximately 11km North-East from our study area. AQE provides us the hourly concentration of Nitrogen Dioxide (NO_2) in $\mu g/m^3$ for the entire month of January 2023.
It would be interesting to plot traffic flow, wind and pollution on a same figure and compare the trends, peaks and lows.

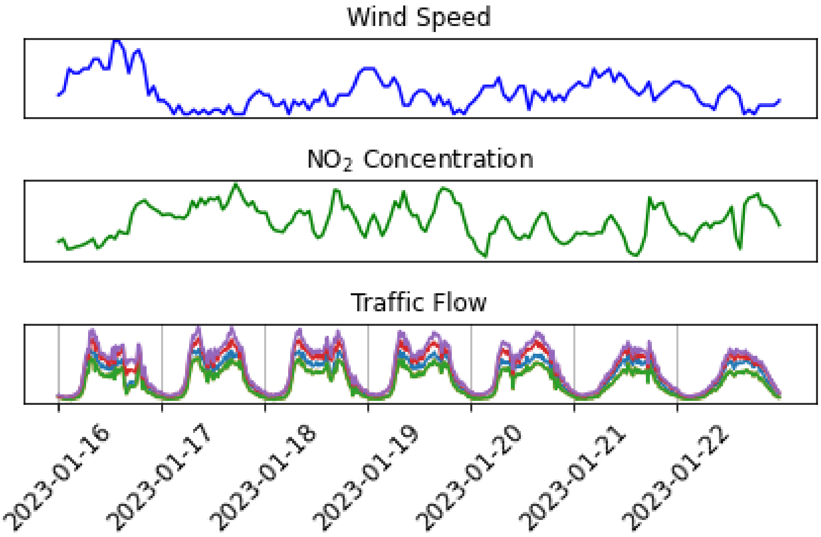


Figure 2.10 Wind Speed, NO_2 Concentration and Traffic Flow during a week in January

In figure 2.10, we observe the trends of wind speed, NO_2 concentration, and traffic flow. Note that we omitted the values/units for simplification, as we are only interested in the trends here. Additionally, we should not treat any comparisons between NO_2 concentration and traffic flow as concrete facts, since their locations are different.
Nevertheless, we can reasonably compare wind speed and NO_2 concentration. As expected, they show kind of opposite trends. When the wind speed is low, NO_2 concentration tends to be high. This observation is clear from 16th January to 17th January. While it becomes less obvious for the rest of the week, we can still notice it at certain times. For example, at the end of 18th January or 19th January, wind speed increases and NO_2 concentration decreases, while at the end of 22th January, wind speed slightly decreases and NO_2 concentration slightly

increases. These qualitative observations should be interpreted cautiously.

2.5 Framework

Having chosen our study area and gathered all the necessary data, our next step is to construct a framework that integrates traffic data and pollution levels into a Partial Differential Equation (PDE) mesh. This framework will allow us to model and analyse how traffic patterns and wind dynamics influence pollution dispersion throughout the area.

2.5.1 From reality to a map

First of all, we need to find a way to export our study area and import it into Python. OpenStreetMap (OSM)[17] appears to be the right tool for this task. It is a free, open geographic database that is updated and maintained by a community of volunteers through open collaboration. They collect data from surveys, aerial imagery and also other freely licensed geodata sources. OSM is commonly used to create electronic maps, assist in humanitarian aid and data visualisation. It uses its own topology to store geographical features, which can then be exported. In our case, we will export only the empty map and define our own geographical features later.
After exporting our study area, we obtain a .zip file containing multiple files. The one we are interested in is the .shp file depicting the roads of our study area. In Python, we use the GeoPandas package to read and exploit the exported files.

To estimate the pollution concentration in our study area, we decide to create a mesh over the map we exported. This mesh will allow us to estimate the pollution concentration in each cell. Hence, we need to define a cell size small enough to be as precise as possible. However, the smaller the cell size, the greater the number of cells, which increases the computing time. The code 2.1 below shows how to create the mesh using the GeoPandas and Shapely packages. Here, we first establish, on a rule of thumb, the number of cells, n_cells. We then calculate their size to ensure an equal number of square cells in both the x and y directions.
Finally, plotting both the map and the mesh gives us figure 2.11.

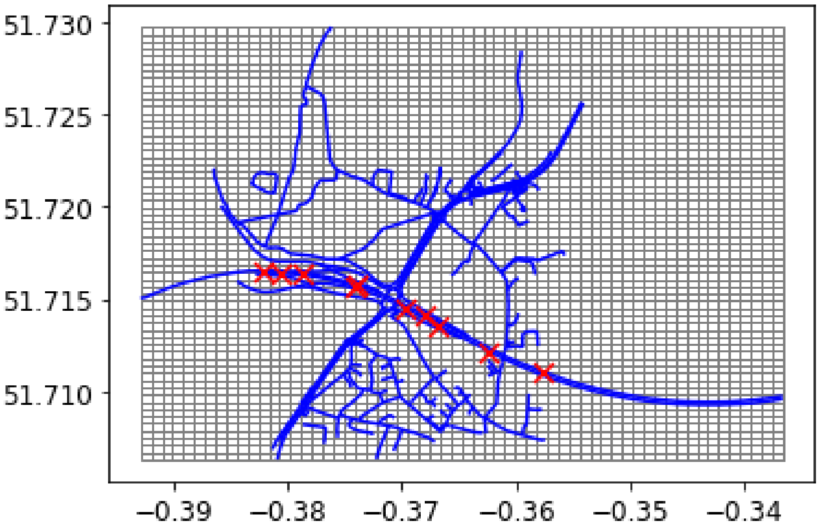


Figure 2.11 Map and Mesh of our study area

Note that OpenStreetMap offers a valuable advantage: it includes longitude and latitude coordinates on our map. This enables us to accurately place the scanners by using their coordinates (indicated on National Highways[14]) and mark their positions with red crosses.

```python
# Get the bounding box of the shapefile
xmin, ymin, xmax, ymax = gdf.total_bounds

# Number and Size of the cells
n_cells = 60
cell_size_x = (xmax - xmin) / n_cells
cell_size_y = (ymax - ymin) / n_cells

# Projection of the grid
crs = "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"

# Create the cells in a loop and stock them in a list grid_cells
grid_cells = []
for i in range(n_cells):
    for j in range(n_cells):
        x0 = xmin + i * cell_size_x
        y0 = ymin + j * cell_size_y
        x1 = x0 + cell_size_x
        y1 = y0 + cell_size_y
        grid_cells.append(shapely.geometry.box(x0, y0, x1, y1))

cell = gpd.GeoDataFrame(grid_cells, columns=['geometry'], crs=crs)
```

Listing 2.1: Python code to create the mesh

2.5.2 From a map to a graph

A street network can be visualised as a finite, connected directed graph. In this representation, intersections correspond to nodes, while the roads between them can be seen as edges; each node and edge having geographical coordinates.

To define our graph in Python, we will use the NetworkX package. Our first step involves determining the road segments covered by each scanner. These segments will serve as our edges, and their boundaries as the nodes. Note that the nodes may not necessary coincide with the scanners themselves. Indeed, we define a segment as a part of the road where the traffic volume is constant. Therefore, the nodes will typically be located at entrances and exists, where traffic flow can change. We manually determine these segments and their coordinates using typical geographical tools (e.g. Google Maps). Figure 2.12 illustrates the case for the Clockwise direction. Each scanner's colour corresponds to the segment's colour they cover.
Then, we use NetworkX to build a graph, specifying the longitude and latitude coordinates of the nodes, see figure 2.13 below.

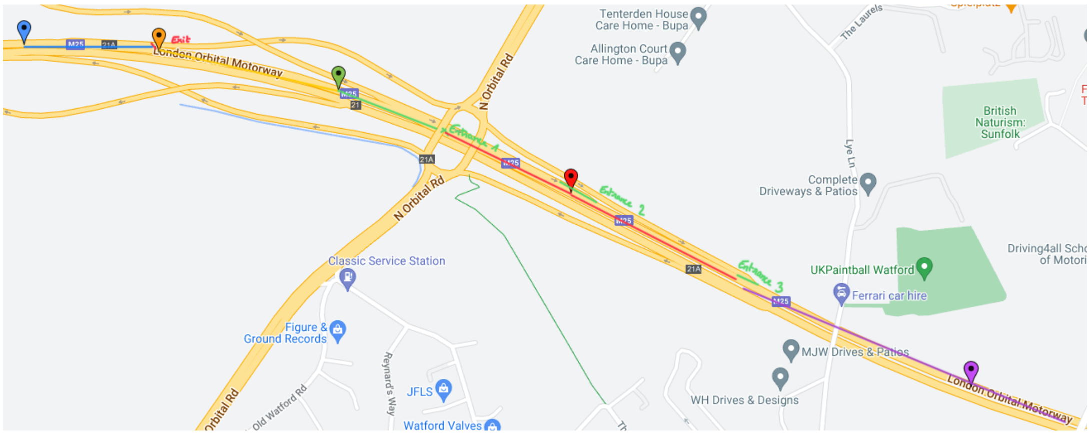


Figure 2.12 Nodes and Edges - Clockwise direction

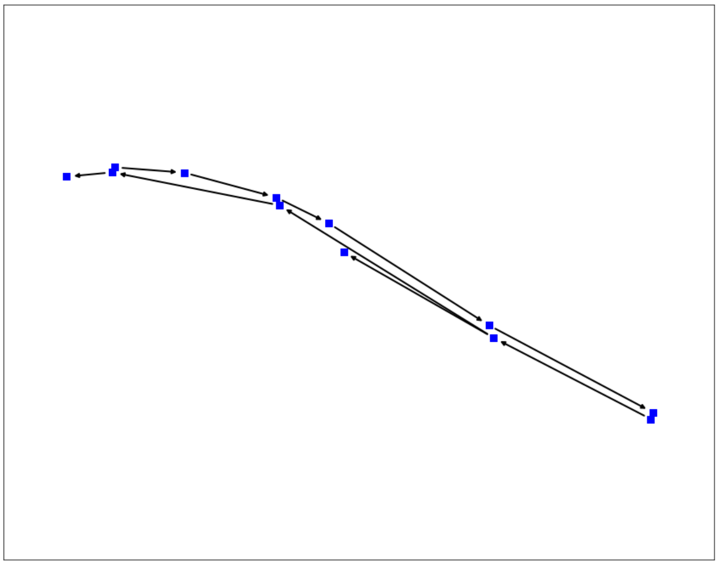


Figure 2.13 Directed graph obtained from the nodes and edges defined above

2.5.3 Distance matrices

Finally, after constructing the graph, we use it to estimate the traffic emissions from traffic flow. Section 3.1 defines the source term of our model, which particularly requires a distance matrix A_i for each edge i of our graph.
Let's fix an i. The distance matrix A_i is a matrix of size (n_{cells}, n_{cells}) constructed as follows: for each cell of the mesh, we determine the length of edge i contained in the cell, and set $(A_i)_{kl}$ to be the length of edge i contained in cell (k, l). More formally,

$$(A_i)_{kl} = length[\{\text{edge } i\} \cap \{\text{cell } (k, l)\}]$$

Hence, we obtain a set of $|E|$ matrices of size (n_{cells}, n_{cells}), where E is the set of edges of our graph. The code A.1 shows how to implement the distance matrix in Python. It is important to note that these matrices will be very sparse, as edges often only intersect a few cells compared to the total number of cells. This observation is significant as it will allow us to save considerable computational time for later computations.

Compute distance between two geographical points

In subsection 2.5.3, we compute the length of the intersection between edge i and cell (k, l), which is similar, considering the size of the cells, to calculating the distance between the two geographical intersection points of edge i and the boundaries of cell (k, l). Specifically, we need to determine the great-circle distance[2].

To compute this distance, we could reasonably opt for the Euclidean distance, assuming a 2D space due to the small size of the cells, or predefined Python functions. However, for completeness, we present a formula known as the Haversine formula[18], which allows us to compute the great-circle distance between two geographical points on Earth using their latitude and longitude. The formula is given as follows:

Algorithm 0: Haversine Formula, see A.2 for Python code.

```
$\Delta\text{lon} \leftarrow \text{lon}_2 - \text{lon}_1$
$\Delta\text{lat} \leftarrow \text{lat}_2 - \text{lat}_1$
$a \leftarrow \sin^2(\Delta\text{lat}/2) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2(\Delta\text{lon}/2)$
$c \leftarrow 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$
$\text{radius\_earth} \leftarrow 6371$                    ▷ Radius of the Earth in kilometers
$\text{great\_circle\_distance} \leftarrow \text{radius\_earth} \cdot c$
```

[2] The shortest distance between two points on the surface of a sphere.

Chapter 3

Model

In this chapter, we explain how we can use traffic data, such as traffic flow and vehicles speed, to obtain pollution emission estimations. Then, we implement a PDE model to simulate pollution dispersion according to wind data, and traffic emissions as the source.

3.1 Estimate traffic emissions from traffic flow

This section explains how we can estimate traffic pollutant emissions from the available traffic data, such as traffic flow and vehicles speed. We define the source term to be of the form:

$$\mathbf{S} = \sum_{i=1}^{m} \gamma \rho_i \mathbf{A}_i \quad \in \mathbb{R}^{N_x \times N_y}$$

where γ is a constant emission factor, ρ_i is the density of vehicles in edge i, and $A_i \in \mathbb{R}^{N_x \times N_y}$ is the distance matrix computed as in 2.5.3.
The density of vehicles in edge i is computed, from the traffic flow and the average speed, as follows:

$$\rho_i = \frac{Q_i}{U_i}, \qquad \text{with} \quad Q_i = \frac{'TotalVolume[\#veh \textit{ detected in 15mins}]'}{0.25[h]}, \tag{3.1}$$

where, ρ_i is the density of vehicles in edge i in vehicles per meter, Q_i the traffic flow in edge i in vehicles per hour, U_i the average speed in meters per hour in edge i.

Hence, **S** is computed as above for every 15 minutes interval of every day of January 2023.

3.2 Advection-Diffusion equation

We model the pollutant dispersion by an advection-diffusion source equation:

$$c_t + \nabla \cdot (\mathbf{w}\, c) = D\nabla^2 c + \mathbf{S} \tag{3.2}$$

where $\mathbf{x} \in \mathbf{R}^2$ are the coordinates in the 2D-plan, $\nabla = (\partial_x, \partial_y)$ is the nabla operator, $\mathbf{w} = (w_x, w_y)$ are the wind components in the x and y directions, $c(x, y, t)$ is the pollutant concentration at a specific place at time t, and **S** is the source term as in 3.1.

Let's have a look at the space and time dependencies of the terms in this equation:

- $\mathbf{c}(t) = c(x, y, t)$ is the pollutant concentration at time t. Hence, it depends on space and time.
- $\mathbf{w} = (w_x, w_y)$ are the wind components in the x and y directions. The wind is updated every hour and is supposed to be uniform on the area of study (given the small size of the area of study).
- $\mathbf{S} \in \mathbb{R}^{N_x \times N_y}$ is the source term as computed in 3.1, where we have values for every 15 minutes of every day of January 2023.

The second term '$\nabla \cdot (\mathbf{w}\, c)$' represents the transport of the pollutant by the wind (advection). The third term '$D\nabla^2 c$' corresponds to the transport of the pollutant by diffusion, where D the diffusivity (also called diffusion coefficient), assumed to be the same in the x and y directions. Notice that in our model we do not consider any chemical reactions or other sources than traffic emissions.

3.3 Numerical solution

This section demonstrates a way of solving numerically the advection-diffusion source equation defined in 3.2. Indeed, this kind of equation can only rarely be solved by hand. Most often, their solution is approximated numerically by computers, using several methods.
In our case, we will solve the PDE (3.2) using a similar method to the one developed in the thesis [2].

First of all, we denote the domain of our problem Ω, which is a square region of size $N_x \times N_y$, with equal size cells. We denote the grid spacing by Δx and Δy in the x and y directions, respectively. Furthermore, we index the cells by i and j. In doing this, we can approximate the value of c using discrete values, which we denote by c_{ij}.

To solve the PDE (3.2) numerically, we first write it in the form:

$$c_t = \mathcal{A}(c) + \mathcal{D}(c) + \mathcal{S}(c)$$

where $\mathcal{A}(c), \mathcal{D}(c)$ and $\mathcal{S}(c)$ denote the advection, diffusion and source term operators. Then, the advection term is solved explicitly with an upwind scheme. Next, the diffusion term is solved implicitly with a Crank-Nicolson scheme. Finally, the source term is added explicitly.

3.3.1 The Advection Problem

As said above, we aim to solve the advection problem explicitly with an upwind scheme. We consider:

$$c_t = -(w_1 c_x + w_2 c_y)$$

where c_x and c_y are the partial derivatives of c, and as before, w_1 and w_2 are the wind components in the x and y directions respectively. Then, the upwind scheme discretises the advection terms in a way to take into account the direction of the wind flow. The idea is to look at the values c at neighboring cells in the direction of the wind flow, to approximate its derivative in that direction.

Applying this idea, for $w_1 > 0$, we find in the x direction a one-side finite difference approximation for $w_1 c_x$:

$$w_1 c_x \approx w_1 \frac{c_{i,j} - c_{i-1,j}}{\Delta x}$$

where $c_{i,j}$ is thus the value of c at the current cell (i, j) and $c_{i-1,j}$ the value at cell $(i - 1, j)$. Conversely, if $w_1 < 0$, then:

$$w_1 c_x \approx w_1 \frac{c_{i+1,j} - c_{i,j}}{\Delta x}$$

where, here, $c_{i+1,j}$ is the value of c at the cell $(i + 1, j)$. Similarly in the y direction we find:

$$w_2 c_y \approx \begin{cases} w_2 \frac{c_{i,j} - c_{i,j-1}}{\Delta y}, & \text{if } w_2 > 0 \\ w_2 \frac{c_{i,j+1} - c_{i,j}}{\Delta y}, & \text{if } w_2 < 0 \end{cases}$$

For simplicity, we set Dirichlet boundary conditions $c = 0$ to the two boundaries where the wind is coming from.

Finally, we proceed to a forward finite difference in time to compute $c^n_{i,j}$ at each discrete time t_n at cell (i, j), where we set $t_n = n\Delta t$, with Δt a time step. For instance, consider a wind of 225°, i.e. coming from the south-west (SW), with a wind speed of $w > 0$. Notice that we have 225°= $\frac{5\pi}{4} rad$, also a wind rose and a trigonometric circle are not oriented the same (on a wind rose 0° is at the north and is defined clockwise), so, one can check that:

$$w_1 = w\, cos(\frac{5\pi}{4} + \frac{\pi}{2}) > 0 \qquad w_2 = -w\, sin(\frac{5\pi}{4} + \frac{\pi}{2}) > 0$$

Then, the forward finite difference in time is done as follows:

$$\frac{c^{n+1}_{i,j} - c^n_{i,j}}{\Delta t} = -w_1 \frac{c^n_{i,j} - c^n_{i-1,j}}{\Delta x} - w_2 \frac{c^n_{i,j} - c^n_{i,j-1}}{\Delta y}$$

$$c^{n+1}_{i,j} = c^n_{i,j} + w_1 \frac{\Delta t}{\Delta x} c^n_{i-1,j} + w_2 \frac{\Delta t}{\Delta y} c^n_{i,j-1} - (w_1 \frac{\Delta t}{\Delta x} + w_2 \frac{\Delta t}{\Delta y}) c^n_{i,j}$$

Next, we set= $\alpha = w_1 \frac{\Delta t}{\Delta x}$ and $\beta = w_2 \frac{\Delta t}{\Delta y}$, giving:

$$c^{n+1}_{i,j} = c^n_{i,j} + \alpha c^n_{i-1,j} + \beta c^n_{i,j-1} - (\alpha + \beta) c^n_{i,j} \tag{3.3}$$

Notice, we could write (3.3) in matrix form. Indeed, first of all, reshape the matrix of concentration $\mathbf{c} \in \mathbb{R}^{N_x \times N_y}$ into a vector:

$$\mathbf{c} = (c_{1,1}, c_{2,1}, \ldots, c_{N_x,1}, c_{1,2}, \ldots, c_{N_x,2}, \ldots, c_{1,N_y}, \ldots, c_{N_x,N_y})^T \in \mathbb{R}^{N_x \cdot N_y \times 1}$$

Then, by doing this we can write (3.3), as:

$$\mathbf{c}^{n+1} = (I + A)\mathbf{c}^n$$

where,

$$A = \begin{bmatrix} -(\alpha+\beta) & 0 & 0 & & & \cdots & 0 \\ \alpha & -(\alpha+\beta) & 0 & & & \cdots & 0 \\ 0 & \alpha & -(\alpha+\beta) & & & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & & & \vdots \\ \beta & 0 & & \ddots & \ddots & & \vdots \\ 0 & \beta & & & & & \vdots \\ \vdots & \vdots & \ddots & & & & \\ 0 & \cdots & & \beta & \cdots & \alpha & -(\alpha+\beta) \end{bmatrix} \in \mathbb{R}^{N_x \cdot N_y \times N_x \cdot N_y}$$

In other words,

$$A = diag([\beta, -(\alpha+\beta), \alpha], [k=-N_x, k=0, k=-1])$$

Similarly for the other three cases:

- $w_1 > 0, w_2 < 0 : A = diag([\beta, (-\alpha+\beta), -\alpha], [k=-N_x, k=0, k=1])$
- $w_1 < 0, w_2 > 0 : A = diag([-\beta, (\alpha-\beta), \alpha], [k=N_x, k=0, k=-1])$
- $w_1 < 0, w_2 < 0 : A = diag([-\beta, (\alpha+\beta), -\alpha], [k=N_x, k=0, k=1])$

3.3.2 The Diffusion Problem

The next step is to solve the diffusion problem:

$$c_t = D(c_{xx} + c_{yy})$$

We implicitly approximate its solution using a Crank-Nicolson scheme:

$$\begin{aligned}\frac{c_{i,j}^{n+1} - c_{i,j}^{n}}{\Delta t} = \frac{D}{2(\Delta x)^2}&\left[(c_{i-1,j}^{n+1} + c_{i+1,j}^{n+1} - 2c_{i,j}^{n+1}) + (c_{i-1,j}^{n} + c_{i+1,j}^{n} - 2c_{i,j}^{n})\right]\\ &+ \frac{D}{2(\Delta y)^2}\left[(c_{i,j-1}^{n+1} + c_{i,j+1}^{n+1} - 2c_{i,j}^{n+1}) + (c_{i,j-1}^{n} + c_{i,j+1}^{n} - 2c_{i,j}^{n})\right]\end{aligned}$$

Set $\alpha = \frac{D}{2(\Delta x)^2}$ and $\beta = \frac{D}{2(\Delta y)^2}$ and rearrange:

$$\begin{aligned}c_{i,j}^{n+1} - \frac{\alpha}{2}(c_{i-1,j}^{n+1} + c_{i+1,j}^{n+1}) - \frac{\beta}{2}(c_{i,j-1}^{n+1} + c_{i,j+1}^{n+1}) + (\alpha+\beta)c_{i,j}^{n+1} = c_{i,j}^{n} &+ \frac{\alpha}{2}(c_{i-1,j}^{n} + c_{i+1,j}^{n}) + \frac{\beta}{2}(c_{i,j-1}^{n} + c_{i,j+1}^{n})\\ &- (\alpha+\beta)c_{i,j}^{n}\end{aligned}$$

which we can express in matrix form:

$$(I - B)\mathbf{c}^{n+1} = (I + B)\mathbf{c}^{n} \tag{3.4}$$

where,

$$B = \begin{bmatrix} -(\alpha+\beta) & \alpha/2 & 0 & \cdots & \beta/2 & \cdots & 0 \\ \alpha/2 & -(\alpha+\beta) & \alpha/2 & & & \ddots & 0 \\ 0 & \alpha/2 & -(\alpha+\beta) & \ddots & & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & & \beta/2 \\ \beta/2 & 0 & & \ddots & \ddots & & \vdots \\ 0 & \beta/2 & & & & & \vdots \\ \vdots & \vdots & \ddots & & & & \alpha/2 \\ 0 & \cdots & & \beta/2 & \cdots & \alpha/2 & -(\alpha+\beta) \end{bmatrix} \in \mathbb{R}^{N_x \cdot N_y \times N_x \cdot N_y}$$

Or, in other words:

$$B = diag([\beta/2, \alpha/2, -(\alpha+\beta), \alpha/2, \beta/2], [k=-N_x, k=-1, k=0, k=1, k=N_x])$$

Notice (3.4) is an equation of the form $A\mathbf{x} = \mathbf{b}$ which can be solved using multiple methods (e.g. least squares, pseudoinverse, with *scipy.linalg.solve()* in Python, ...), see section 5.3.
Here also we set the Dirichlet boundary conditions $c = 0$, on the two boundaries the wind is coming from, and Neumann boundary conditions $\hat{\mathbf{n}} \cdot \nabla c = 0$ on the outgoing boundaries.

3.3.3 The Source Term

As a final step, we add the source term to the solution:

$$\mathbf{c}^{n+1} = \mathbf{c}^n + \mathbf{S}$$

Python Code

Recall that wind data is updated every hour, so the advection and diffusion matrices can be computed only every hour to save computational time.

```python
# Loop over the days
for i, day in days.items():

    # Loop over the 15 minutes intervals
    for t in range(96):

        # Determine wind and compute Diffusion and Advection matrices hourly
        if t % 4 == 0:
            # Determine wind in x and y directions
            # Notice at how a wind rose is oriented compared to a trigonometric circle
            w1 = w_speed * np.cos(w_dir + np.pi/2)
            w2 = - w_speed * np.sin(w_dir + np.pi/2)

            #Diffusion term
            alpha = D_diff * dt / dx_meters**2
            beta = D_diff * dt / dy_meters**2

            B = sp.diags([beta/2, alpha/2, -(alpha+beta), alpha/2, beta/2], [-n_cells, -1, 0, 1, n_cells], shape=(n_cells*n_cells, n_cells*n_cells))

            #Advection term
            alpha = w1 * dt / dx_meters
            beta = w2 * dt / dy_meters

            if w1 > 0: # Four different cases
                if w2 > 0:
                    A = sp.diags(...)
            ...

        # Algorithm
        k_end = round(900 / dt)
        for k in range(k_end):
            c_diffusion, _ = sp.linalg.cg(B_minid, B_id @ c0)
            c_advection = A_id @ c_diffusion
            c_new = c_advection + dt * source_term_day.iloc[t]['S(t)']
            # Apply Neumann boundary conditions
            c_new = apply_neumann_bc(c_new, n_cells)
            c0 = c_new

        c_history[i].append(c_new)
```

Listing 3.1: Extract of the Python code for the numerical solution of the advection-diffusion equation 3.2.

Chapter 4

Results

For our project, we will consider the CO and NO_2 concentrations. Recall that in our model, we only take into account emissions for traffic, the advection of the pollutants by the wind and their diffusion.

As a rule of thumb and after numerous simulations, we set the number of grid cells per direction to be 60. The initial concentration $c0$ is set to zero. Additionally, after several simulations, we set the time step $dt = 0.25s$. The time step must be chosen to ensure the convergence and stability of our model.

We now need to set the different parameters required for our computations. First of all, the effective diffusion coefficient D, assumed to be the same in both x and y directions, must range between 3-30m^2/s, according to L. Oporto Lisboa's thesis[2]. We decide to set $D = 10m^2/s$. Secondly, we need to determine the emission factors γ necessary for the source term, see 3.1, for both CO and NO_2. The EMEP/EEA Air Pollutant Emission Inventory Guidebook 2019[19] provides multiple emission factors for various pollutants, depending on the size of the vehicle or the type of fuel. The European Union established emissions standards (Euro 4/5/6) to regulate the amount of pollutants emitted by vehicles. The adoption of Euro 5 and Euro 6 standards, established in 2009 and 2014 respectively, has led to significant reductions in vehicle emissions, contributing to improve air quality in urban areas. Given the timeframe, it is reasonable to assume that the vehicles from our traffic data are Euro 5 or 6 vehicles. We will consider the approximation of regular diesel passenger cars. Referring to the tables provided in [19], we set $\gamma_{CO} = 0.016mg/m$ (similar to the value used in L. Oporto Lisboa's thesis[2]), and $\gamma_{NO_2} = 150\mu g/m$.

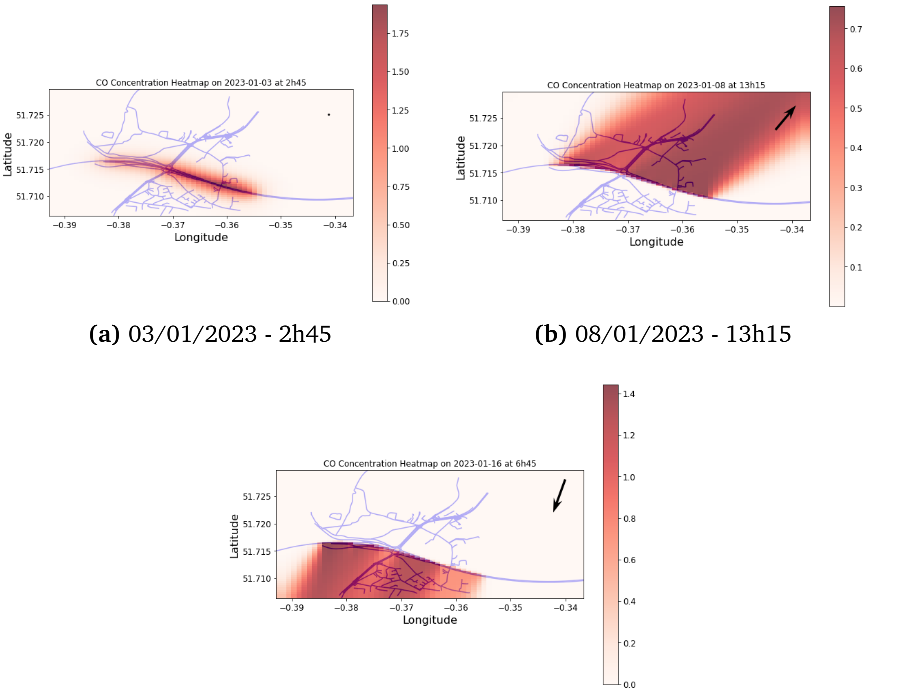
4.1 Heatmaps of CO and NO_2 concentrations

The first results plots for our project, given the nature of pollutants, are presented as heatmaps.

4.1.1 CO Concentration

Figure 4.1 illustrates heatmaps of computed CO concentration for three different days and times in January. The source of CO is clearly defined by the section of the road from which our traffic data come from, as it is the only considered source in our model. Next, the movement of the pollutant aligns well with the wind direction (and speed), indicated by the black arrow in the top right corner. The dynamics of the pollutant appear realistic.

In the case of CO concentration, we also decide to present the results as a GIF[20]. This dynamic visualisation allows us to observe the changes in pollution levels over several consecutive times and days, providing a clearer and more intuitive understanding of how pollution varies and disperses with time.



(a) 03/01/2023 - 2h45

(b) 08/01/2023 - 13h15

(c) 16/01/2023 - 6h45

Figure 4.1 Heatmaps of the CO concentration for some days in January.

4.1.2 NO_2 Concentration

Figure A.2 illustrates similar observations for NO_2 heatmap as those noted for CO.

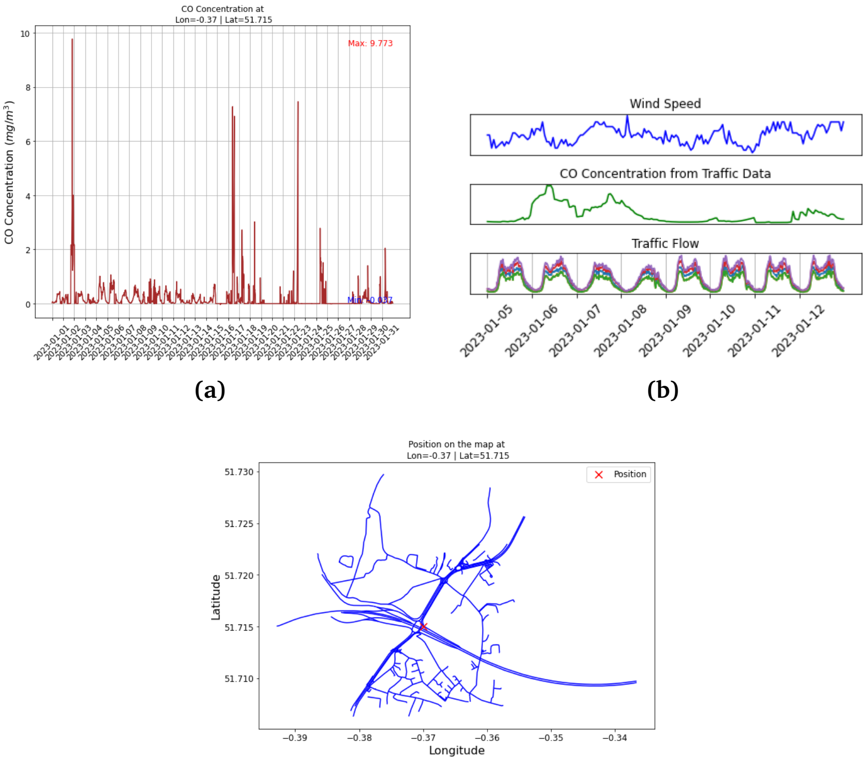
4.2 Plots for specific locations

Another way of presenting the results is by choosing a specific location on the map and observing the pollution levels over time at this fixed position. To achieve this, given a position (latitude, longitude), we identify the closest cell on the grid and use the pollutant concentration of this cell as the value for the given position.

4.2.1 CO Concentration

Figure 4.2 **(a)** shows the CO concentration (mg/m^3) at Latitude=51.715, Longitude=-0.37, on the section of the M25 (see **(c)**), for the entire month of January. We immediately observe three significant peaks that are much higher than the average trend. These are clearly outliers, likely due to model errors, which will be discussed further in Chapter 5. Nevertheless, overall, the

values range between 0 and 1 mg/m^3, which is a reasonable and realistic range for CO concentration, as can be seen in L. Oporto Lisboa's thesis [2] and public data sources.
The purpose of **(b)** is to verify any expected relationships between wind, traffic and CO concentration. We observe that when the wind is high, the CO concentration tends to be low and vice versa. This can be observed on both 6th and 7th January, where the wind increases (CO concentration decreases) and decreases (CO concentration increases), respectively.



(a)

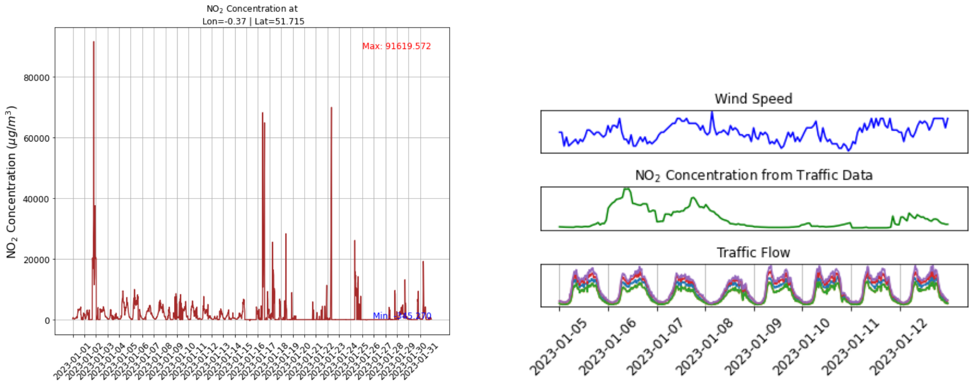
(b)

(c) Location

Figure 4.2 CO Concentration over time at a fixed location.

4.2.2 NO$_2$ Concentration

As expected, the trends in figure 4.3 for NO$_2$ concentration are similar to those in figure 4.2 for CO concentration at the same fixed location. Indeed, notice that in our model, the differentiation between pollutants is limited to their emission factor γ in the source term. However, for NO$_2$ we observe overall too high values (compared to usual NO$_2$ concentrations), almost reaching 10^5$\mu g/m$, which are clearly unrealistic. This suggests that the model may be missing crucial characteristics about the behaviour of NO$_2$, or there could be errors in the parameters. Nevertheless, some values do seem to be in a reasonable range (below 50$\mu g/m^3$). We will further discuss the previous observations in Chapter 5.



(a)

(b)

Figure 4.3 NO$_2$ Concentration over time at a fixed location.

Chapter 5

Discussion

The purpose of this chapter is to discuss the results obtained in Chapter 4.

5.1 Heatmaps analysis

First of all, the heatmaps 4.1 and A.2 demonstrates a good correlation between pollution levels and the wind conditions. The polluted areas around our section of the M25 are consistent with the wind speed and direction. Observe on both **(a)** plots, when wind is low (no advection), the diffusion of the pollutant around the road. In addition, pollution levels are always higher near the road, which is anticipated since it is the only source of emissions in our model.

Looking at the GIF[20] for CO concentration, we observe overall good dynamics. The movements of the pollution by advection are in good accordance with the wind speed and direction. Furthermore, at times when the wind is low (no advection), we can see the diffusion of pollution around the road, which is as expected from our model. Nevertheless, we sometimes observe unexpected behaviours: the whole area getting polluted (whole map getting coloured), harsh transitions between consecutive times or days, and seldom suspiciously high levels of pollution.

Multiple reasons could be the origin of these unexpected behaviours. First of all, since they occur either at random times or during a range of time, we could consider that our traffic data present some outlier values at these times. We must examine the data we have at these times to verify their consistency. Another possible cause could be the wind data. Indeed, these unwanted behaviours often occur when the wind is very low or in a specific cardinal direction, implying the wind component in the x or y direction is 0. These very low or even null values for the wind components could lead to computational errors.

5.2 Graphs analysis

Figures 4.2 and 4.3 present the model's results in another way: pollutant concentration at a fixed location over time. As mentionned in the previous Chapter 4, while, overall the trends are consistent and more or less coincide with the wind patterns, there are some outliers in both CO and NO_2 concentrations. While reasonable peaks could be explained by unexpected events on the M25 such as traffic jams, accidents or protests, peaks of this magnitude are more likely errors. As with the heatmaps, the causes of these outliers can be multiple: inconsistent data, implementation or computational errors. Additionally, these high concentration levels could also be explained by the choice of the emission factors γ_{CO} and γ_{NO_2}. Indeed, we chose a unique value for both pollutants, while as explained in [19], emission factors depend on numerous characteristics: type of vehicle, fuel (Petrol or Diesel), size of the vehicle, model year, Euro 4/5/6 criteria, etc. Therefore, setting a unique value for γ is a significant simplification that can lead to a lack of accuracy.

Another simplification we made is setting a constant value for the effective diffusion coefficient D. Nevertheless, there is no reason why D should remain constant. In reality, it likely depends on the pollutant and the meteorological conditions. These simplifications impact the accuracy and realism of our model.

Finally, recall that for simplification we decided to not consider chemical reactions among the gases. However, it is worth noting that NO_2 is a highly reactive gas, and its concentration in the atmosphere is significantly influenced by these reactions, which we did not account for. This simplification could explain the important number of high concentration values observed for NO_2.

5.3 Algorithm 3.1 errors

Recall that in our model 3, we need to solve a problem of the form $A\mathbf{x} = \mathbf{b}$, with $A = (I - B)$, $\mathbf{x} = \mathbf{c}^{n+1}$ and $\mathbf{b} = (I + B)\mathbf{c}^n$. However, since B is not necessarily invertible, we need to employ specific methods to solve it. First of all, it is worth noting that B is real, symmetric and sparse (pentadiagonal matrix). In addition, it can be easily verified that almost always B is positive definite (e.g. in Python, by checking directly that all eigenvalues are positive, or by using Cholesky decomposition). Hence, there are many efficient methods available in Python to solve such problem $Ax = b$ efficiently, where A is real, symmetric, sparse and positive definite.

The package Scipy (scipy.sparse.linalg[21]) gathers many of these methods:

- Direct methods: *spsolve*, *spsolve_triangular* (for A triangular);
- Iterative methods: *bicg* (Biconjugate Gradient), *cg* (Conjugate Gradient), *gmres* (Generalized Minimal Residual), etc.

After several simulations, we decide to use the Conjugate Gradient (*cg*) method, as our matrix B (as well as $I + B$ and $I - B$ under some conditions) meets the conditions required for this method: hermitian, positive definite and sparse. The CG method provides a very good computational time (∼1.80s/it, where an iteration represents a 15-minute time step), compared to more than 8s/it using *spsolve*. For a detailed explanation of the CG method, we refer the reader to [22].

In practice the Conjugate Gradient method can be unstable with respect to even small perturbations. The accuracy and speed of the method depend on the condition number $\kappa(A)$ of the matrix A. Hence, if the matrix B is ill-conditioned (i.e. large $\kappa(B)$) then the solutions obtained may not be accurate enough, or even makes the model unstable. Various methods exist to improve the conditioning of the system, such as preconditioning, which aims to reduce κ, thereby improving the convergence properties of the method.

Chapter 6

Conclusion

All in all, the model we developped demonstrated reasonable results while being accessible, providing clear and fast results. Furthermore, it can be easily adapted to any area with sufficient and consistent data. The flexibility and efficiency of the model make it a valuable tool for studying air pollution from traffic emissions in various locations.

Nonetheless, as discussed in 5, the model could be improved in several ways. First of all, the choice of the parameters γ and D. Instead, of defining a fixed value for both, we could estimate them statistically, derive them from further data analysis. In addition, the algorithm 3.1 could be improved with more accurate methods to ensure stability and improve accuracy. Furthermore, other models could be developped to include additional sources of pollution, changes in the ABL height and to account for chemical reactions.

Finally, the CO concentration values obtained from our model appear to respect the World Health Organization (WHO)[1] guidelines. These limits are set to prevent critical health effects from exposure to CO. Concentrations exceeding these limits can pose significant health risks, especially to vulnerable populations such as people with heart conditions, pregnant women, and young children:

- 1-hour mean: 30mg/m^3
- 8-hour mean: 10mg/m^3

From these results, we can conclude that our section of the M25 does not significantly represent a critical source of CO. However, it is essential to remember that this is just a small portion of the entire M25. Transport, in general, remains a significant threat to both health and the environment due to the pollution it generates. Various measures are being implemented, including the promotion of electric vehicles, improvements of the infrastructure and restrictions on vehicles. Specifically for the M25, efforts are being made to improve safety and reduce pollution, as highlighted in [23].

Appendix A

Appendix

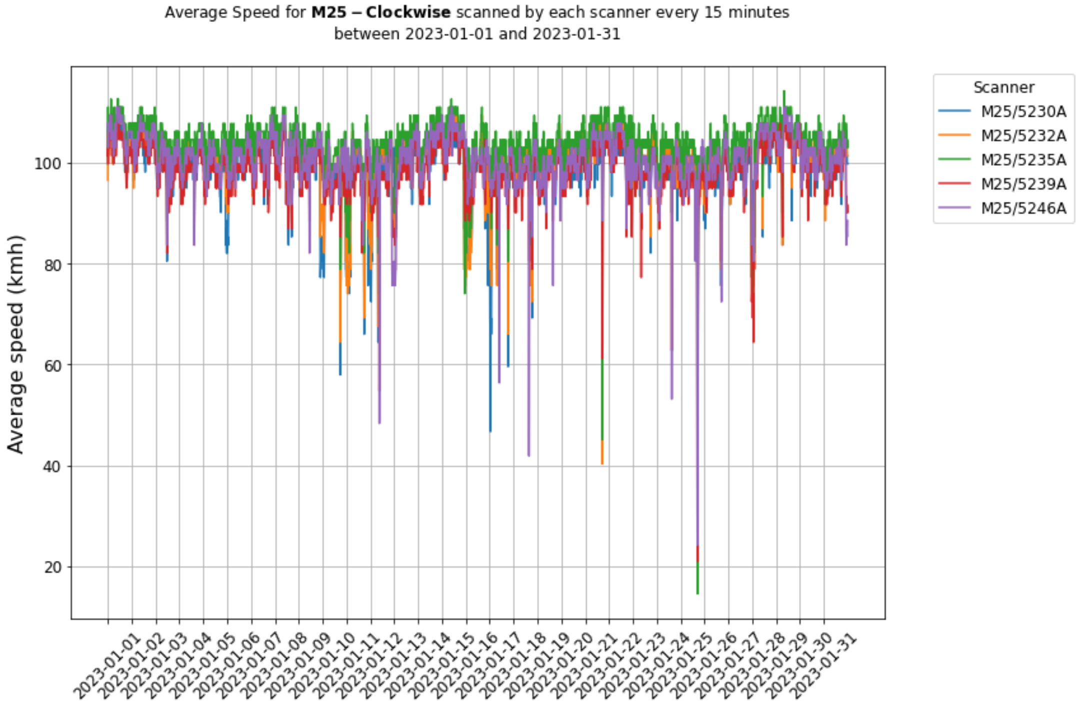


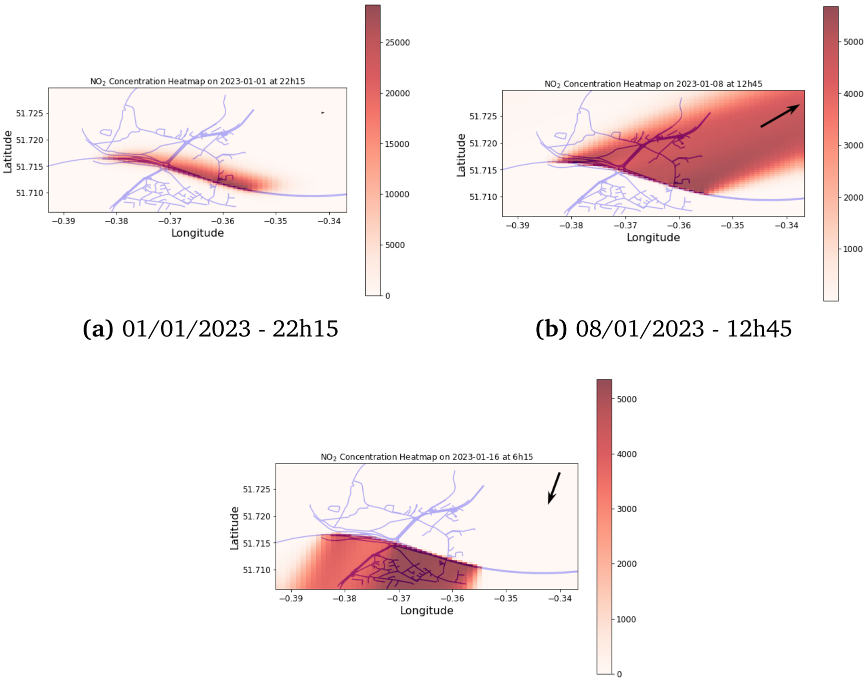
Figure A.1 Clockwise Average Speed (kmh)

```python
# Initialize a distance matrix with zeros of size (num_cells, num_cells, num_edges)
num_cells = n_cells
num_edges = len(G.edges)
distance_matrix = np.zeros((num_cells, num_cells, num_edges))

# Loop through each edge in the graph
for i, (source, target) in enumerate(G.edges):
    source_x, source_y = pos_coords.loc[pos_coords[’POS’] == source, [’Longitude’, ’Latitude’]].values[0]
    target_x, target_y = pos_coords.loc[pos_coords[’POS’] == target, [’Longitude’, ’Latitude’]].values[0]

    # Loop through each cell in the grid
    k = 0
    l = 0
    for j, cell in enumerate(grid_cells):
        cell_geom = cell
        # Check for intersection with the cell
        if cell_geom.intersects(shapely.geometry.LineString([(source_x, source_y), (target_x, target_y)])):
            intersection = cell_geom.intersection(shapely.geometry.LineString([(source_x, source_y), (target_x, target_y)]))
            if intersection.length > 0:
                # Compute the distance for the portion of the edge within the cell, using haversine_distance()
                portion_length = haversine_distance(intersection.coords[0][1], intersection.coords[0][0], intersection.coords[1][1], intersection.coords[1][0])
                distance_matrix[k, l, i] = portion_length
        k += 1
        if k >= num_cells:
            k = 0
            l += 1
```

Listing A.1: Python code to construct the distance matrices 2.5.3.

```python
def haversine_distance(lat1, lon1, lat2, lon2):
"""
Calculate the great circle distance between two points on Earth using their longitude and latitude.

Parameters:
    lat1, lon1 (float): Latitude and Longitude of point 1 (in degrees).
    lat2, lon2 (float): Latitude and Longitude of point 2 (in degrees).

Returns:
    distance (float): Distance between the two points in kilometers.
"""

# Convert latitude and longitude from degrees to radians
lat1, lon1, lat2, lon2 = map(radians, [lat1, lon1, lat2, lon2])

# Haversine formula
dlon = lon2 - lon1
dlat = lat2 - lat1
a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
c = 2 * atan2(sqrt(a), sqrt(1-a))
radius_earth = 6371  # Radius of the Earth in kilometers
distance = radius_earth * c

return distance
```

Listing A.2: Python code to compute the great circle distance between two points on Earth using their longitude and latitude 2.5.3.



(a) 01/01/2023 - 22h15

(b) 08/01/2023 - 12h45

(c) 16/01/2023 - 6h15

Figure A.2 Heatmaps of the NO$_2$ concentration for some days in January.

Bibliography

[1] World Health Organization (WHO), 1948.

[2] Laura Oporto Lisboa. *Hybrid physical-statistical model for air quality prediction from traffic data*. PhD thesis, University of Bath, June 2023.

[3] Envira. Primary and secondary pollutants: these are the most dangerous, January 2022.

[4] Clarity. How measuring different types of air pollutants creates a more holistic picture of air pollution, December 2021.

[5] NSW Health. Particulate matter (PM10 and PM2.5), November 2020.

[6] European Environment Agency (EEA). Air pollution in Europe: 2023 reporting status under the National Emission reduction Commitments Directive, June 2023.

[7] Y. Xiang, T. Zhang, J. Liu, L. Lv, Y. Dong, and Z. Chen. Atmosphere boundary layer height and its effect on air pollutants in Beijing during winter heavy pollution. *Elsevier*, 215: 305–316, 2019.

[8] X. Tie, S. Madronich, G. Li, Z. Ying, R. Zhang, A. R. Garcia, J. Lee-Taylor, and Y. Liu. Characterizations of chemical oxidants in Mexico City: A regional chemical dynamical model (WRF-Chem) study. *Elsevier*, 41(9):1989–2008, 2007.

[9] William H. Brune. The atmospheric boundary layer is your home, Spring 2024.

[10] Wikipedia. Convection–diffusion equation, 2024.

[11] Lynn Rusk. In pictures: A history of the M25 motorway and how it was built. *London World*, March 2024.

[12] Company Government. National Highways, 2015.

[13] Company Government. National highways, Traffic Flow Open Data, 2024.

[14] Company Government. National highways, Traffic Flow Map, 2024.

[15] Visual Crossing, 2003. Provider of weather data and enterprise analysis tools.

[16] Ricardo Energy & Environment. Air Quality England (AQE), 2020. Resource for local air quality information and air quality data provision.

[17] Steve Coast. OpenStreetMap (OSM), 2004. Open geographic database updated and maintained by a community of volunteers via open collaboration.

[18] Simon Kettle. Distance on a sphere: The Haversine Formula, 2017.

[19] EMEP/EEA. Air pollutant emission inventory guidebook 2019, 2019.

[20] GIF - Heatmap of CO concentration.

[21] Scipy - Sparse linear algebra. Methods to solve linear problems.

[22] Indranil Ghosh. Introduction to Mathematical Optimization with Python, 2021.

[23] J. Panons and P. Media. Good progress on M25 works, says highways agency. *BBC News*, May 2024.