Universitat Politècnica de València

Escuela Técnica Superior de Ingeniería del Diseño

KTH Royal Institute of Technology

Department of Mechanics

Aerospace Engineering Degree

Final Degree Project

High-order spectral simulations of the flow in a simplified urban environment

Author:
Pablo Torres Greus

Director:
Dr. Ricardo Vinuesa Motilva

Advisor:
Dr. Sergio Hoyas Calvo

July 10, 2020

Acknowledgement

I would like to dedicate the following lines to express my sincere gratitude the people that, with their help, have contributed to the completion of the present project.

A special mention deserves Dr. Ricardo Vinuesa Motilva, whose help, guidance and constant motivation have been invaluable for the completion of this project. I would also like to thank Dr. Sergio Hoyas Calvo for the help received during the present project and the past three years.

In addition, I would to like to thank Dr. Alvaro Vidal Torreira for the provision of its platform and the help in development of the meshing process. Last but not least, I would like to thank my family for the unconditional support received during all those years.

To all of them, thank you very much.

Abstract

The study of urban flows has been a field of interest in modern fluid mechanics for now more than twenty years. The knowledge of how an airflow behaves in urban canopies presents many different applications, such as urban planning, air quality studies or the prediction of pollutant's propagation. Initially, the methodologies used to study this kind of problems typically involved a partial or full experimental approach. This is rather inconvenient, as it tends to be highly costly. Modern turbulent computational fluid mechanics have enhanced the tools available to study urban turbulent flow as it allows a fully computational approach.

The main objective of the project is to develop a set of tools that allow to systematically solve the flow in a simplified urban environment. The flow simulation will be carried through a "Well-Resolved Large-Eddy Simulation" by means of the Nek5000 routines. Then the flow is analysed using a custom-made statistic toolbox, that allows to obtain the averaged parameters that are typically used in turbulent flow analysis. In addition, the project presents a secondary objetive, which is the development of a series of routines that allow to easily analyse the key parameters of turbulent simulations, such as mesh resolution.

Contents

List of Figures

List of Tables

Part I

Report

1

Introduction

1.1 General concepts

The modern world is characterized by the dominance of urban environment. From the industrial revolution to our times, the vast majority of the world has been shifting from the countryside to cities raising the density of those urban areas. This trend along with the current challenges about sustainability, have put the focus on the study of the vitality of cities. In this way, the discipline of *urban sustainability* emerges as a transversal field, that inevitably requires the symbiotic confluence of engineering, social, political and economic infrastructures [7]. Traditionally the focus was driven towards the socioeconomic topics, as they were factors of high influence on the stability and vitality of urban areas. However, due to the environmental degradation that has our world and in particular the industrialized world, has suffered from now more than a decade, topics related with the environment and its preservation have emerged in both the public opinion and scientific community. It is well known, that sustainability is a broad field in which the study of many different fields are required in order to have a clear image on its inner workings.

From the technological perspective, our world is immersed in an ocean of data. This information overexposure brings important challenges that appear to be interdisciplinary. In fact, the implications reaches all disciplines, from purely technical domains to philosophical and ethical problems. Nevertheless, this data-driven world has opened a completely new application domain for big data techniques as well as the artificial intelligence (AI). Several studies are now developing regarding the implications of enhanced data techniques and artificial intelligence to various fields. The work presented by Vinuesa et al. [34] appraises the implications of AI as well as their impact in the sustainable development goals (SDG) proposed by United Nations. Although the study covers many different topics, it our case it is particularly interesting the appraisal of AI within the frame of SGD 11, i.e. sustainable cities and communities. In such framework, the authors explain, AI could have an enabler effect providing enhanced techniques to overcome the sustainability challenges of our time. In addition to the change of paradigm that AI is inducing, we know from the extended Moore's Law, that computing power is doubled every few years. In this way, computational simulation that used to be out of the feasible time range are now possible to make. For instance, big-size direct numerical simulations or large-eddy simulations can now be solved within feasible time. Thus, the application range of this kind of tools has significantly widen.

One of the recently developed areas of study in the sustainability field applied in urban environments, is the urban air quality. This discipline focuses mainly on the creation and propagation of pollutant and how these interact with the environment. Therefore, a fundamental part of this area of study is the flow. Being able to properly characterize the flow in an urban environment is fundamental in the production of quality studies in urban air pollution. It is in this part where modern fluid dynamics can make a significant contribution. The focus will be driven not on the air analysis *per se* but on the study and characterization of the flow in a urban environment. For this matter, several studies have been performed in recent times, applying different approaches on the characterization of the flow.

1.2 Motivation and objectives

The main motivation behind the present is to develop a systematic approach to simulate and study turbulent urban flows. The idea is to gather the tools needed to create and run a turbulent flow simulation in an idealised urban environment. As far as the actual simulations are concerned, our objective is to be able to solve a rather theoretical simulation in a simplified environment to study the flow behaviour. The developed tool, will serve as baseline to more complex tools that might include further analysis related with the sustainability of city, e.g. pollutant dispersion analysis, thermal analysis etc. In addition, we aim to recover and discuss the available tools and studies up to this moment related with urban flows.

In particular, our objective is to develop a large-eddy simulation (LES) over a simplified urban environment such that we can study the processes and factors that are relevant in this kind of flows. In this way, we aim to gather and develop the meshing and solutions tools required to run the simulation as well as design a relevant case to test the afore-mention tools. As far as the case simulation is concerned, we aim to create the geometry and mesh as well as the simulation strategy to be followed. In addition, the tools required to analyse the flow will also be developed such that the final solution can be properly interpreted. Finally, the case simulation will also aim to provide understanding on the behaviour of urban flows.

1.3 Historical perspective

In this section a revision of the current situation of the different studies around the characterization of flows in urban environments is presented. The main objective is to provide a clear presentation on the works relevant to the study here reported. Note that the literature on urban flows is vast, thus not every aspect of the available research will be covered. In this way, the presentation will fundamentally focus on the experimental and large-eddy simulations studies, considering only major application cases.

As far as the structure is concerned, we will be dividing the analysis in two major sections. On the one hand the experimental literature will be covered. On the other hand, the major research lines in numerical simulations will be also introduced, specially large-eddy simulation for being our method of preference.

1.3.1 Experimental studies of urban turbulent flows

The study of flows, as many other physical disciplines, has evolved rapidly in the last decades shifting from a almost pure experimental approach to a numerical strategy of study. Although numerical simulations are gaining ground year after year, one can not ignore the importance of empirical approaches both historically and currently. In this way, it seems consistent to have a clear picture of the current state of experimental studies of urban turbulent flows.

There are two major approaches in the experimental description of urban turbulent flows. On the one hand, there is the probing approach. Generally, it consists on selecting an urban area and installing a series of probes which allow to obtain some characterisation of the flow. On the other hand, there are the wind tunnel studies. Those mainly consist on the manufacturing of scale models that are then tested under some set of conditions inside a wind tunnel. In addition, some studies also present purely empirical characterisations of urban flow.

1.3.1.1 Empirical description of urban turbulent flows

Let us begin with the description of empirical work. Those works focus on the description of the flow's behaviour in urban environments. Although those studies appear to be more qualitative than quantitative, their reading brings a clear understanding on the implications of urban flows, thus their inclusion in the current presentation.

Zajic et al. [42], recovered the work of Oke [22] on the behaviour of idealised urban turbulent flows in order to provide, in addition to experimental measurements, a very neat description of the flow behaviour in urban environments. Their study gathers the flow behaviour in the Central Business District (CBT)[1] of Oklahoma city using atmospheric data. They obtained results on the airflow patterns, stability conditions and turbulence properties on the area, providing evidence on the influence of built environment on the area's thermal effects. Nevertheless, they provide a description on how the flow behaviour is influenced depending on the structure of the urban environments. In fact, they present three distinct flow regimes. On the one hand, there is the isolated roughness regimes which is characterised by a very small interaction between the wake produced by the individual building. This flow regime is found in canopies where the distance between the buildings is large. On the other hand, when the building separation is small, the flow appears to skim over the street canyon. In addition, there is an intermediate case, where wake interaction is found. Zajic et al. [42] present a systematic method to segregate flow regimes. They define a series of aspect ratios and in particular:

$$\frac{1}{\lambda_{hg}} = \left\{ \frac{\text{separation between buildings}}{\text{height}} \right\} = \frac{g}{h} \tag{1.1}$$

Using the metric presented in Equation 1.1 they can differentiate the afore-stated flow regimes using a series of heuristics,

- For $g/h > 2.5$ the flow is in isolated roughness regime
- For $1.4 < g/h < 2.4$ the flow is in the wake interference regime
- For $g/h < 1.4$ the flow is skimming regime

Note that the afore-mentioned heuristics are the result of the observations and a subsequent idealisation of the flow regimes. In this way, these criteria are a simplification that might differ from the actual behaviour of the flow. It is easy to see that under this criteria the interaction between building wakes in the spanwise direction is not considered. In fact, Zajic et al. [42] treat the urban canopies as two dimensions at this stage of the analysis. One can make the parallelism between

[1] The term CBT is here used tos specify the type of urban environment the study focus on. In this case, the study deals with a tall building area where there is not an important separation between such buildings

this description and the classical way to approach aerodynamic theory, where one starts by understanding the flow over an airfoil in order to extend the theory to three dimensional wings. Figure 1.1 shows the graphical schemes of the afore-mentioned flow regimes.

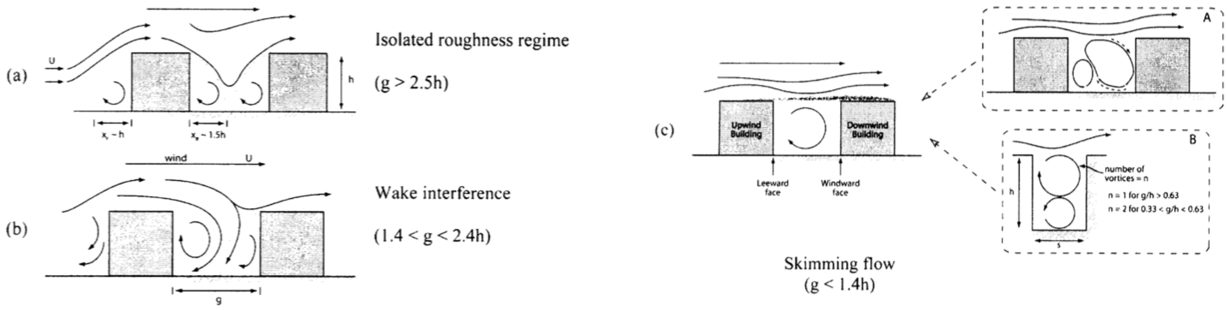


Figure 1.1: Flow regimes in a two dimensional obstacle cluster. Extracted from Zajic et al. [42]

Later on, Zajic et al. [42] treat the effects of three dimensional flow in urban canopies. They very rightly observe that when dealing with a 3D description of the flow, the effects of the vertical side edges become significant. Following the same procedure, they define two additional aspect ratios, this time considering the spanwise and longitudinal dimensions of the building, w and b respectively.

$$\begin{aligned} \frac{1}{\lambda_{hw}} &= \frac{w}{h} \\ \lambda_{hb} &= \frac{h}{b} \end{aligned} \tag{1.2}$$

They observe that for taller building, i.e. building for which $\lambda_{hb} > 1$ "the recirculation bubble behind the building is overwhelmed by the side separation layer" (Zajic et al. [42]). This phenomenon creates an intense turbulence that can destroy the separation bubble.
Moreover, Zajic et al. [42] also treated the influence of non-uniform heights in an idealised urban canopy in two dimensions. Generally speaking, they found that the greater the difference in heights the bigger the vorcity is. Hence, the greater the difference in height the bigger the disturbance of the medium.
To close the descriptive part of their work, Zajic et al. [42] added a study on the effects of the incidence of the flow over a urban canopy. Appraising the descriptive section of the work presented by Zajic et al. [42], one can see the limitations of the empirical description. Despite being a very complete presentation on the flow behaviour in urban surrounding, the methodology appears somehow limited by the idealisation. In fact, such study provides a clear understanding on the flow structure at a macro level. However, questions on the particular structure of the turbulence as well as in application cases remain unanswered, e.g. *how much is the surrounding flow disturbed by the turbulence?*, *how would the disturbance be affected in more complex canopies ?*, etc. This limitation leaves room for the use of numerical simulations that provide a more detailed description on the flow structures.

Other studies provide a wider description of those flows choosing different scales in the study. While Zajic et al. [42] focused on a rather small scale in the urban environment, i.e. analysing the urban canopy, other studies focus on wider scales, treating the flow at regional or city scale. Britter and Hanna [2] precisely distinguish four scales in the study of urban flows. Table 1.1 gathers the scales presented by Britter and Hanna [2].

Scale	Length
Regional	up to 100 or 200 km
City	up to 10 or 20 km
Neighbourhood	up to 1 or 2 km
Street	up to 100 or 200 m

Table 1.1: Scales of the study and the correspondent scale length. Adapted from Britter and Hanna [2]

The different scales respond both to the flows behaviour and the methodology available to describe it. Once again, the study presents, in addition to the flow description, both experiments and modelling, but for the moment let us focus exclusively in the flow description.

Regional and city scales Britter and Hanna [2] consider the regional scale as an vast area that is principally affected by the urban area, i.e. the city scale. Although the area *per se* can not be considered [2], the flow behaviour at a city level has an impact at the regional scale and thus it is worth considering the interaction between those two scales.
The flow description is actually focused at the **city scale** . The city scale is defined by Britter and Hanna [2] as the diameter of the average urban area, i.e. the area over which flow variations can be averaged out. The city area is characterised for having large obstacles and hence a large drag force. The authors then introduce the average obstacle height following the averaging approach that characterises the area in question. Britter and Hanna [2] describe three major sublayer in the city scale.

1. **Inertial sublayer**: It is defined as the area in which the boundary layer has integrated the perturbations introduced by the obstacles. In this way, the layer can be considered as a pseudo-free stream layer and thus it's possible to apply the standard atmospheric models. In fact, this is actually a well-known assumption in theoretical aerodynamics where the effects induced by the flow perturbation are neglected in the region far from the obstacle. Note that the inertial sublayer is placed at the outer band. Figure 1.2 presents a schematic representation of the different sublayers.

2. **Urban canopy sublayer**: This is the layer where the flow is directly affected by the obstacles. In this way, a given point in the flow is affected by the presence of a local obstacle, modifying its trajectory.

3. **Roughness sublayer**: This layer contains the urban sublayer and it is extended to meet the inertial sublayer. It corresponds to a transient band, where the flow progressively integrates the perturbations introduced in the urban canopy sublayer. As in any transitional problem the idealisation of this particular area is limited.

[2] The characterisation of a flow over a region of 100 to 200 km appears to be an arduous task. Taking into account the interaction of the elements in such an area, thermal variations etc. one realises that implementing a model at that scale would very probably lead to significant inefficiencies. Note that models do exist at those level and even higher levels, e.g. atmospheric models. However, such models focus on the macro level rather than in an actual characterisation of the flow behaviour.

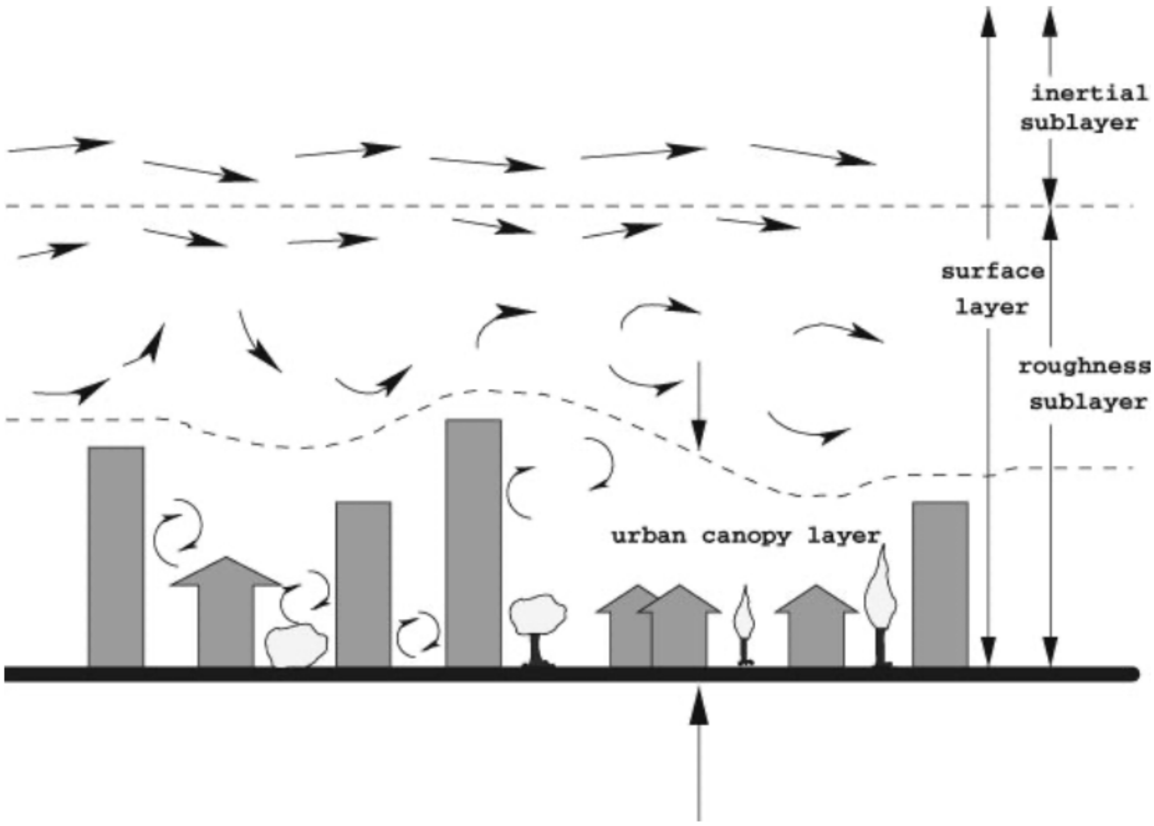


Figure 1.2: City scale sublayer scheme. Extracted from Britter and Hanna [2].

Neighbourhood and street scales In this part of Britter and Hanna [2] describe the flow at a local level. The neighbourhood scale consists on a series of arbitrarily distributed obstacles. Those canopies are treated as idealised geometries such that both their description and testing is feasible. The neighbourhood scales actually corresponds to the description presented by Zajic et al. [42], where the authors present how the spacing in the obstacles affects the resulting flow. Once again, the concepts of skimming and isolated roughness regimes arise as an idealisation of flow behaviour. As far as the street scale is concerned the authors assess this smaller scale from the application's perspective. In fact, Britter and Hanna [2] present how, at such a small scale, minor local variable elements such as traffic or pedestrian are affected and affect the flow. In this way, the characterisation of the flow under those conditions appears to be significantly harder as the variability of the cases has increased. Once again, the encountered difficulties leave room for the implementation of computational methodologies. However, Britter and Hanna [2] advert that the computational approaches still require a important idealisation and thus have to be properly appraised.

1.3.1.2 Open-environment testing of urban turbulent flows: Full-scale and reduced models

Urban flow experiments can be generally divided in wind tunnel and full-scale. The objective of this part and that one that follows is to expose and compare both methodologies while appraising the current work on both areas. Full-scale probing distinguishes itself from other methodologies for being carried in a significantly less controlled environment. This approach, is valuable precisely for that reason. By testing in the actual environment, one could expect the results to be more trustworthy than the ones obtained in control conditions. However, full-scale testing has its own challenges, as data gathered in uncontrolled environments tends to carry noise and inconsistencies. In this way, some of the available literature is simply oriented to study the suitability of full-scale testing, comparing it with other experimental or numerical methodologies, in the frame of a given application.

Vita et al. [40] present a full study on the assessment of pedestrian distress in urban environments comparing full-scale experimental techniques with wind-tunnel approaches as well as numerical methodologies. Their study is focused at a street level, thus conditions are somewhat constrained, in the sense that no strong variability in the environmental conditions is expected apart from the wind variation which precisely what's being tested.
As far as the experimental setup is concerned, Vita et al. [40] distributed eight sonic anemometers over the streets at a two meters distance from the ground. In addition to those probes, a reference anemometer is permanently working. This reference probe was placed at the roof of a 62 meters height tower on a 10 meters mast with the purpose of reading the baseline conditions in the undisturbed zone. Both reference and testing probes where configured to record three-dimensional data. Note, that the testing zone was selected to be specially gusted, precisely because the objective was to assess the different methodologies used in pedestrian distress studies.
The full-scale testing aimed to obtain data on mean flow speed. In addition, the data is also used to help characterising the flow. Once the data recovered, it was compared with the results of wind-tunnel experiments and numerical simulations.
The mean wind-speed data presented a good qualitative agreement with the wind-tunnel data, both having the same trendline over the measurement positions. However, the wind-tunnel data didn't lie within the standard deviation range found in the full-scale measurements with exception of some measurement position. This mismatch suggests that either the full-scale measurements or the wind-tunnel experiments have a major error. The authors appear to decline for the full-scale data, considering the wind-tunnel a more limited tool. Later on Vita et al. [40] compare the results with numerical simulations (RANS and LES) and deduce that the wind-tunnel data fail to reproduce the full-scale results particularly in the recirculation region. In conclusion, the authors clearly present the full-scale results as the main verification tool in the study. In this way, the limitations appear to lie on the other methodologies. However, Vita et al. [40] also discuss the limitations of the study in the environment considered. From the simulations, they found that the velocity streamlines present a rather complex flow. Thus, it's seem plausible that the efficacy of using eight uniformly distributed probes over a straight line will inevitably have its limitations in terms of flow characterisation. In addition, mean velocity measurements are, as a magnitude, limited to describe the behaviour of the flow. Hence the need of introducing numerical simulations. Recapitulating, the study presented by Vita et al. [40] provides a clear picture on the suitableness of the available methods and their relations in term of results. The fundamental conclusion exposed by the authors suggests the integration of a multi-method approach for maximising the validity and understanding of a given problem study.

The previous paragraph focused on the use of full-scale testing as "sanity check" for further studies. However, literature is found on studies that rely on full-scale testing as sole methodology. Hirose et al. [13] presented a project was to study wind-induced natural ventilation in cities and how those are affected by the surroundings urban flows. The study fully relies on a outdoor approach that although was not full-scale it is still related to the afore-mentioned concepts precisely for being an outdoor experiment. The experimental setup consisted in a 512 cubical blocks matrix where each block had a height of 1.5 meter. The total dimensions of the models were $100 \times 50\ m^2$. Note that the site was oriented such that wind typically flows in the length-wise direction.

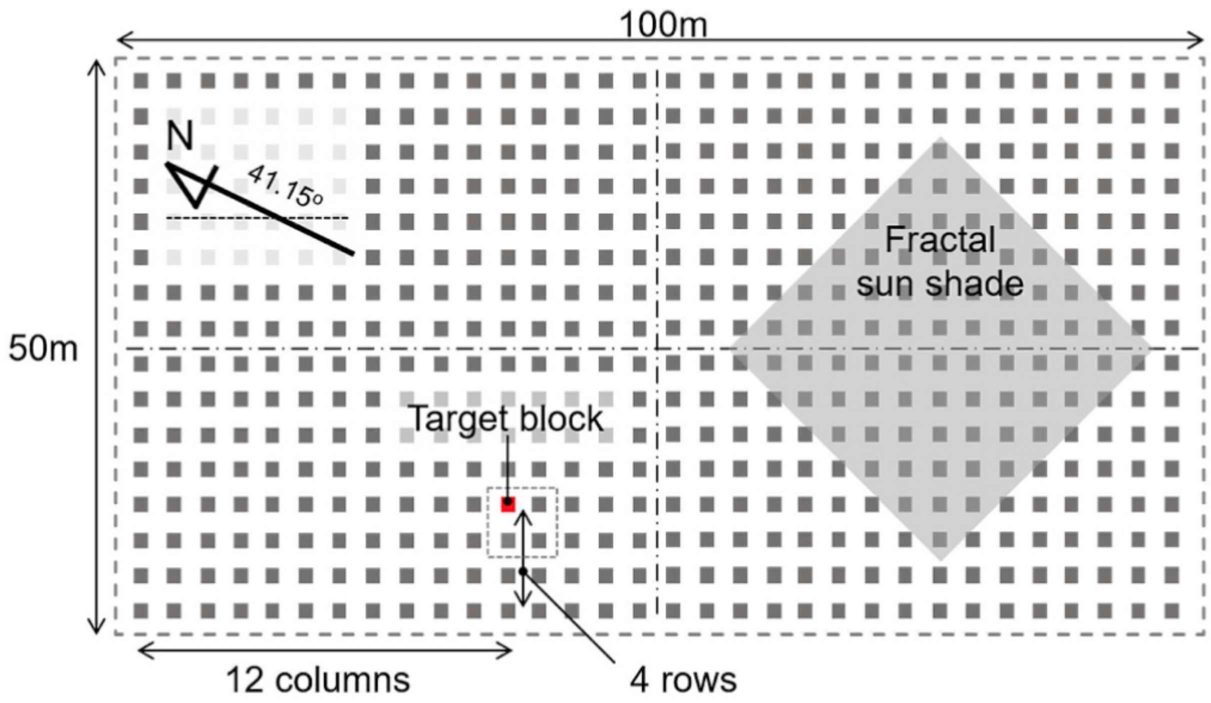


Figure 1.3: Experimental site scheme. Extracted from Hirose et al. [13].

The packing density of the site is roughly at 25%. The data acquisition system was composed of 700 sonic anemometers equiped with a TR90-T probe. Those were installed at half-height distance from the top of the block. In addition, acrylic plates with pressure tabs were also installed in the northwestern and southeastern faces of the block.

Using the afore-mentioned setup two dataset were collected. From the analytical perspective, the study rest on two major axes. On the one hand, the study of approaching flow conditions, is characterised for using the well-known statistical descriptions of turbulent flows. On the other hand, the study revolved around the relation between the pressure difference and the velocity. Note, that both axes incorporate the use of time-averaged statistics as principal metric to describe the flow. More precisely, Hirose et al. [13] examined the probability distribution of the velocity at the horizontal wind direction by means of the streamwise velocity magnitude, the velocity's standard deviation as well as the velocity range, i.e. minimal and maximal velocities, under both southeastern and northwestern winds. As far as the relation between pressure and velocity is concerned, their relation was examined by means of the pressure coefficient, derived for every positions using the least squares method. Then, the results were plotted in terms of the pressure difference as a function of the streamwise velocity, combining both data points with regressed lines. Analysing the results in terms of the pressure coefficient, Hirose et al. [13] observed that the pressure coefficient values increased with height in the upper-half positions while remain constant in the lower-half of the block. Regarding the spanwise direction, the authors also observed that the pressure coefficient was higher at the edges than it was at the block's centre. Regarding the afore-mentioned observations, the authors conclude that "a stable vortex with very low wind speed might be generated

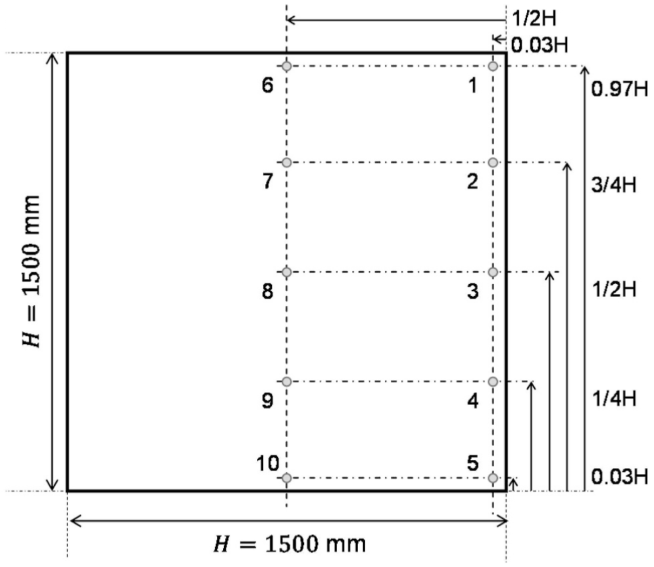


Figure 1.4: Acrylic plates disposition. Extracted from Hirose et al. [13].

in the cavities between two blocks and the approaching flows over the blocks might only skim the air at the upper parts of the cavity" (Hirose et al. [13]). In fact, the flow regime described by the authors corresponds to the now well-known, skimming flow regime, introduced through the work of Zajic et al. [42], in §1.3.1.1. Moreover, from the sole observation of the pressure distribution, the authors concluded that no pressure scale effects were found under the considered setup.
An additional part of the study focused on the analysis of temporal variation in both wind speed and pressure coefficient on specifically targeted blocks. Those temporal variation are obtained including a low-pass filtering operation in both speed and pressure. By inspection of the afore-stated quantities Hirose et al. [13] report that the temporal variation in the pressure terms presented a clear coincidence with the characteristic values of approaching flow. Thus, they argue that the ventilation rates of the buildings in such conditions might temporarily vary due to such variations. This last statements actually reinforces the advantage of open-environment experiments precisely for being able to show variations and discrepancies that happen to be missed in controlled-environment conditions.
As a final comment on the work of Hirose et al. [13], one shall assess the limitations of the afore-presented study. Following the appraisal reported by the authors, further research might be needed to assess the effects of small-scale turbulent flow within the canopy layer. The authors propose more complete air data measurements allowing the gather three-dimensional flow statistics. To that proposal, one can add the implementation of turbulent numerical simulations, such as large-eddy simulation (LES) or even a direct numerical dimulation (DNS), that precisely allow to appraise the smaller flow structures.

Coming back to the work of Zajic et al. [42], early on we introduced the idealised description presented by the authors as a result of the integration of the work of Oke [22]. However, the current study also reports a experimental section complementing the idealised flow behaviour and in particular the analysis of flow regimes. Recall that Zajic et al. [42] focused on the particularities of urban flow in Central Business District (CBD) areas and in particular in the Park Avenue street. From the experiment's perspective, the test area was equipped with three-dimensional ultrasonic anemometers, radiation and infrared temperature sensors as well as thermistors, a soil heat flux

plate, a soil water content sensor and a Doppler lidar. This measurement system was replicated in three different sites. Moreover, an additional measurement system was placed in a semi-rural adjoint site to serve as control experiment, i.e. to help distinguish the effects caused by the urban environment.
Combining both the principal and control experimental system Zajic et al. [42] were able to recover data concerning flow, thermal and soil in the urban environment considered. In this way, their analysis is then focused precisely in those areas, i.e. characterising the flow in terms of both patterns and thermal effects. From the thermal point of view, the characterisation of heat tranfers in the CBD showed, as one could expect, that the heat capacity is higher in the urban environment than it is in the semi-rural environment. This can be explained, as Zajic et al. [42] comment, by the presence of additional heat influxes such as the anthropogenic heat flux. To that matter, other phenomena, such as "radiation trapping" make the temperature significantly higher in the case of urban environments. The afore-statement phenomenon leads to non-development of a stable stratification inside the test area, Zajic et al. [42] explain. To the heat analysis, Zajic et al. [42] added a full appraisal of the flow patterns in the CBD area. Although, results "showed a high sensitivity to large-scale wind direction changes" (Zajic et al. [42]), the authors manage to observe that "close to the canyon edges large vortices form in the horizontal plane and flow tends to channel through an opening on the northern row of buildings" (Zajic et al. [42]). Additionally the authors compared the obtained results with pre-existent literature on the topic, concluding that a sufficient match was found. This last statement shows that the use of a canonical approach, i.e. considering the obstacles as idealised geometries is actually consistent with open-environment results. This matter is particularly interesting for us since numerical simulations precisely rely on the use of idealised geometries and their consistency with the physical phenomena. Furthermore, Zajic et al. [42] finished their exposition with a statistical description of the flow in order to characterise turbulence intensities and how they relate with exogenous factors such as wind direction, time period etc. The results in this past study showed, as one could expect, that higher levels of turbulence were found in the upper parts of the buildings. However, at a pedestrian level, turbulence was influenced by the surroundings buildings, i.e. the spanwise obstacle location, rather than any other factor in the vertical direction.

To close the current section, let us draw some thoughts on the importance of open-environment test and their place in urban flow research. The previous lines were dedicated to the presentation of some of the major works in the discipline. At this stage, it seems clear that open-environment testing in both full-scale and models, brings an unquestionable value to the understanding of the flow in particular applications. In fact, the strength of these methods rely precisely on the characterisation of specific cases. In this way, if one wants to analysis the flow conditions in, for instance, a specific part of city, directly probing the part to be analysed provides significantly valuable information. However, the open-environment resting presents severe flaws in the flow detailed description. Although the majority of open-environment studies recover the available literature on empirical flow descriptions, the detailed behaviour of the flow remains unexplained. From the application's perspective, if one considers design-oriented or purely scientific applications, which tend to have a broader application range, open-environment techniques happen to fall short. This limitations leaves room for the introduction of either additional experimental methodologies, e.g. wind-tunnel testing, or fully numerical approaches.

1.3.1.3 Close-environment testing of urban turbulent flows: Wind-tunnel experiments and alternative techniques

The alternate fundamental approach when dealing with experimentation in fluid mechanics is the close-environment approach, i.e. the testing in indoor controlled conditions. This technique is based on the validity of data extrapolation from the measurements made in the scaled model to the actual system considered. This part will be dedicated to the exposition and appraisal of some of the available literature on the topic. Once again, the same dichotomy appears to form. On the one hand, there are studies that focus on the sole evaluation of the techniques. Those are what we have called *technical or fundamental studies*. Their motivation is almost exclusively to assess and compare the performance of a given experimental techniques in the frame of turbulent urban flows. On the other hand, one might find a kaleidoscope of studies on particular problems where the techniques in question are applied to solve that particular problem. We have denominated those as *applied studies*. Following the afore-stated dichotomy, the revision here reported will be founded dividing the studies in *fundamental* and *applied*. Furthermore, some additional lines will be dedicated to an alternate study that applies enhanced measurement techniques.

Fundamental studies: Appraising the performance of experimental techniques Evaluating a study technique is a fundamental part in the setting of both its adequacy and application range. The following lines will be dedicated to the exposition of the available literature appraising the performance of wind-tunnel experiments within the turbulent urban flows framework.

One of the fundamental studies on the appraisal of both experimental and numerical techniques is the now well-known work presented by Vita et al. [40], where wind-tunnel measurements are compared with full-scale modelling as well as numerical simulations. Note, that in this part only the wind tunnel part of the studies will be addressed, refer to §1.3.1.2 for a full explanation full-scale experiment section of the report. Recall that Vita et al. [40] were studying the performance of various techniques within the framework of pedestrian safety. In this way, their approach consists on reproducing the street-level test in full-scale open-environment over a scaled-model such that it can be tested in the wind-tunnel. The chosen probe system was composed of the combination of three different types of probes. Firstly, Irwin probes were used to determine the mean wind speed. Although Irwin sensors are not the most performant system for such endeavour, their accuracy was estimated sufficient for the task. Their principal advantages lies on their omni-directionality which eases the installation process since no realignment is needed. Secondly, multi-hole probes such as Cobra probes, were used to measure the incoming wind speed on the top of the model. Multi-hole probes are typically used in high-resolution measurements of turbulent flows. However, those probes are limited by their insensitivity in flows slower than two meters per second as well as their directionality. In this way, their exclusive use does not appear to be possible in the application considered by Vita et al. [40]. The third measurement system is hot-wire anemometry, which overcomes the limitations of the two afore-mentioned system. Nevertheless, hot-wire probes present limitations in terms of spacial resolution and sensitivity to wind direction. Recapitulating, Irwin and hot-wire probes were used independently to measure the flow at pedestrian level, i.e. over the model. In addition, a Cobra probe was used to obtain data on the incoming wind speed. As far as the results are concerned, the discussion of the methodologies was previously introduced in §1.3.1.2 and thus we encourage the avid reader to recall the conclusion there drawn.

On the same line, one might cite the work Gadilhe et al. [9], which despite being significantly older than the work of Vita et al. [40], thus more limited in the literature included, provides a clear assessment on the verification of measurements systems and models in urban turbulent flows.

The work reported by Gadilhe et al. [9] consisted on the comparison of numerical prediction of wind flow with the data recovered from a boundary layer wind tunnel experiment. In this way, the approach incorporates the same motivation of the previous mentioned study. However, they differ on the viewpoint as the work of Gadilhe et al. [9] uses the wind-tunnel experiment as verification method rather than as subject of study. From the experimental point of view, a predictive model was developed to be able to compare the results with the measurement obtained in the wind tunnel. Then a scaled model of the testing site was developed to be rested in the wind tunnel under suburban wind conditions. Figure 1.5 provides a schematic view of the testing site, i.e. the model introduced in the wind tunnel, as well as the locations of the probes used in the data acquisition system.

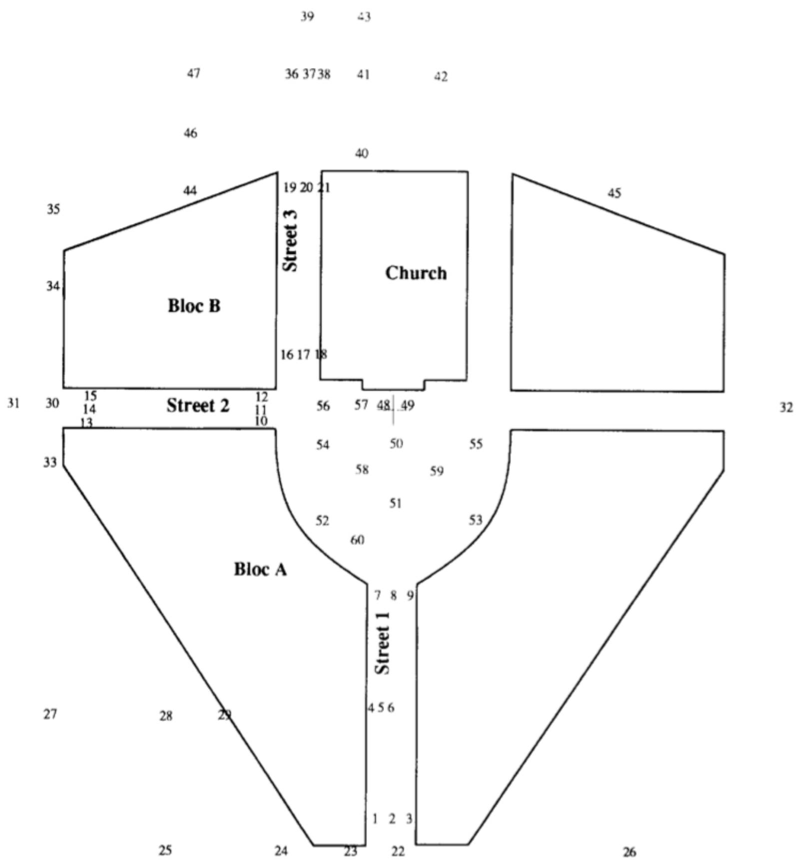


Figure 1.5: Schematic view of the wind-tunnel model prototype and sensor location. Extracted from Gadilhe et al. [9].

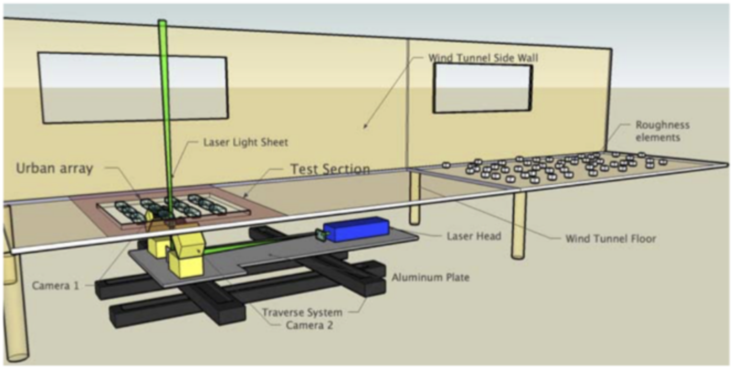
The model was built on a 1/100 scale basis. The measurement focus exclusively on velocity which was acquired in the 60 points shown in Figure 1.5 by means of hot-wire anemometers. The Reynolds number, as described by Gadilhe et al. [9], was defined using the cross-section's hydraulic diameter and was roughly $Re = 10^6$. The analytical strategy applied by Gadilhe et al. [9] consisted on a two axes approach. Firstly, the authors analysed the wind velocity at a constant plane place 1.5 meter above the ground. That part of the study focused on the symmetry of the problem, checking the velocity magnitude and components evolution with the vertical coordinate. In this way, it can be observed that "the transversal and vertical velocities are close to zero both in the wind tunnel experiment nd in the numerical simulation" (Gadilhe et al. [9]). In addition, it's also observed that "in the square, the church [*structure*] and backward flow make the wind velocity decrease (Gadilhe et al. [9]). As far as dispersion is concerned, there is a reasonable agreement between the model and the wind-tunnel experiment in the whole domain with the exception of the inlet and outlet parts. In a second part of the analysis, Gadilhe et al. [9] focus on the wind velocity gradients, taking vertical measurements at some points of the recirculation area, i.e. two on the square and one behind the church. In such areas, the computed vertical velocity component is negative, which is translated in a downward motion while the measured value is positive, i.e. upward motions. On the contrary, "wind intensities are in good agreement" (Gadilhe et al. [9]). The study concludes with a discussion on the methods implemented. Recalling the data comparison between both methods, we saw that the computed data faithfully matched the experimental values in the vast majority of the domain, with exception of the inflow and outflow areas. This discrepancy might have its origin in the computation of the inflow conditions, which includes a significant approximation of the turbulence intensities in the area. In addition, wind tunnel measurements also present their own limitations. Gadilhe et al. [9] explain that at recirculation and wake regions, the measurements might not be reliable. They proposed some solutions, such as the inclusion of laser anemometry. Nevertheless, the overall conclusion reached by the authors suggests that "further comparative studies are required" (Gadilhe et al. [9]).

Applied studies: Wind-tunnel on specific urban environments As stated at the beginning of this section, the studies are mainly divided in *fundamental* or *technical* and *applied*. Fundamental studies were covered in the previous paragraph. Now it is time to focus on the specific application presented in some of the available literature. Note that the studies here assessed will be significantly less generic than the ones belonging to the previous paragraph.

Weerasuriyaa et al. [41] presented a study on the effect of *twisted* winds at a pedestrian level using a scaled model of the Tsuen Wan street in Hong Kong inside a boundary layer wind-tunnel. The setup consisting in a series of wooden vanes twisted with a turning table such that the straight-streamlined flow was curves to obtain *twisted* wind conditions. The twist was set to obtain four distinct cases, at 15 and 30 degrees turning close-wise and counter-clockwise. The measurement system consisted on a five-point probing system to obtain data on mean flow velocity, turbulence intensities and yaw angles. At each point, the afore-stated magnitudes "were measured at 12 discrete heights from 10 mm to 1000 mm for a sampling period of 65 seconds" (Weerasuriyaa et al. [41]).
From the conclusions perspectives, the authors showed how twisted winds have a direct influence in the pedestrian-level wind found in the particular case considered, i.e. Tsuen Wan (Hong Kong). They found how twisted winds can cause more than 35% difference in wind speed at street level among other conclusions. From the appraisal's perspective, the authors explain that in this study the wind profiles were artificially generated and thus result particularly "clean". In fact, in an uncontrolled environment, wind would not fully blow from a single direction and hence the resulting *twisted* wind would very probably be different to the one obtained in the wind-tunnel. In conclusion, this study is an example on how the afore-described techniques are useful to assess the flow condition over a specific area. However, the application range of the observations proposed by Weerasuriyaa et al. [41] is clearly more limited. That is why, our intentions with its inclusion were simply illustrative.

Enhanced measurement techniques: PIV methodology To close the exposition on experimental studies, let us introduce a additional study that introduces an enhanced measurement

technique, the stereoscopic particle image velocimetry (PIV) system. The work reported by Monnier et al. [19], is a research-oriented study that aims to test the applicability of PIV methodologies in the investigation of flows over a urban-like obstacle array. The study incorporates an *a priori* modelling section which purpose is to parametrise the atmospheric boundary layer (ABL). To do so, Monnier et al. [19] run an close-loop wind tunnel experiment, following the work of Najib et al. [20]. This is done using a counter jet consisting of a "60 mm diameter steel tube placed on the floor of the wind tunnel and spans its entire width"(Monnier et al. [19]). The measurement system introduced in the wind-tunnel consists on an array of three hot wires mounted over "a vertical traverse system enabling measurement of the velocity profiles starting from a position close to the floor [...] and extending approximately 400 mm above it" (Monnier et al. [19]).
From the geometry's perspective, the PIV experiment is run over a 120 obstacle array oriented as shown in Figure 1.6.



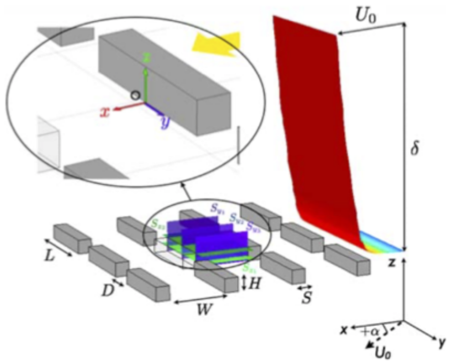


Figure 1.6: PIV setup and obstacle array. Extracted from Monnier et al. [19].

Note that when dealing with PIV system, the geometry is not only limited by any given study-related limit, it is also deeply influenced by the requirements on the PIV resolution. In fact, Monnier et al. [19] explain that the height H in the setup considered in Figure 1.6 was actually influenced by the spacial resolution of the stereoscopic PIV.
The fundamental principle behind the PIV system lies on the introduction of some particles, the seeding, to be measured by means of a sensor. In this way, the seeding is detected by a data acquisition system which then characterises the flow. One of the major inconveniences of the PIV system is the calibration process. In fact, a new calibration is required for each independent data plane, which importantly elongates the testing time. This limitation was solved by Monnier et al. [19] by setting "the whole system up on a single plate sitting on a two-axis traverse system located under the wind tunnel" (Monnier et al. [19]). The full setup is shown in Figure 1.6. As far as seeding is concerned, the system includes atomisers to ensure a sufficiently small particles. Those are injected by three inlets at the floor of the installation.
The analysis presented by the authors focused in three fundamental areas. Firstly, Monnier et al. [19] analysed the effect of the mean flow incidence angle on the streamlines considering the skimming and wake interference flow regimes. In the null incidence case, the flow structures appeared to be identical to the idealisation proposed by Oke [22]. In fact, two recirculation regions are formed, the largest being placed at the bottom of the upper block and the smaller, secondary region formed in the upper part of the bottom block. Nevertheless some differences with the idealised behaviour described by Oke [22] are found. The authors observed that in the isolated roughness regime, "the wake created by the upstream block interacts with the secondary recirculation up-

stream of the downstream block" (Monnier et al. [19]). In addition, the authors also analysed a non-null incidence case, tilting the incoming flow by -4.5 degrees. Under those conditions the skimming flow regime happens to be non-dependent on the incidence of the flow. However, the streamlines are now slightly tilted as well and the recirculation zone is modified, migrating form a purely symmetrical structure to an unsymmetrical one. Turbulent statistics are assessed by the computation of different quantities. Perhaps, one of the most critical ones might be the turbulent kinetic energy (TKE). Monnier et al. [19] present, in the case of wake interference regime, the presence of two large TKE regions close to edges as well as another one located at mid-span. In addition, if a non-null incidence angle is set, "the central high TKE level region disappears and a highly turbulent regions ois created in the region where the stagnation point exists and the secondary recirculation region splits" (Monnier et al. [19]). Furthermore, the study also includes the appraisal of the problem's vorticity as well as the analysis of the velocity gradient tensor.
In conclusion, Monnier et al. [19] verified the correlation between obstacle spacing and flow regimes as it was enunciated by the idealised approach presented by Oke [22]. Under this description, the wake flow in skimming condition happen to be fundamentally two dimensional, for the null incidence angle case. On the contrary, under wake interference conditions, only the primary part of the recirculation region presented a two-dimensionality. The TKE analysis showed that the wake interference regime happens to induce a much more important energy exchange between the flow within the streets and the fluid above. As far as the flow incidence is concerned, it was shown that this angle has a significant effect on the mean streamlines under wake interference regime. The afore-presented work was expanded later on in a study working on the study of the turbulent structure itself. Once again, Monnier et al. [18] used the PIV methodology to characterise the flow in an idealised urban environment, this time focusing on the understating of turbulent structures.

The afore-reported lines have provided a general picture on the available literature on the topic here discussed. Appraising experimental techniques one can see that open-environment approaches appear to provide more reliable results when dealing with specific application studies where one's aim is to characterise the flow over a specific urban area. However, when looking for a wider description of urban flow physics, open-environment strategies appear to be limited. This limitation might be solved using close-environment techniques, which present better results in such endeavours. Nevertheless, we have seen that standard close-environment methods such as wind-tunnel experiments are still limited in the characterisation of the actual physic of the problem. Enhanced techniques, such as the PIV methodology do provide better results but at the expenses of a more complex implementation. Despite the afore-stated improvement, the PIV technique is still limited in the smaller scales of turbulent motion.

1.3.2 Numerical simulations in urban turbulent flows

Numerical techniques also known as computational fluid mechanics start developing around the fifties as an alternative methodology to study the behaviour of flows. With the improvement of computers and numerical methods, the applicability of those methods have completely exploded within our time. Nowadays, the number of available techniques, variants and applications is vast. Nevertheless, one can simply orient the different methodologies with the respect to the amount of modelling that is included. In this way, at one extreme lie actual models, such as the approaches used in applied aerodynamics, e.g. Theodorsen's model, while on the other extreme one would find pure numerical methods, such as direct numerical simulations where the solution is computed at every flow scale. In between those two, one would encounter the rest of methodologies, i.e. Reynolds-averaged Navier-Stokes (RANS), large-eddy simulations (LES) among many others.

This part of the historical perspective will be structured, precisely, following the afore-mention classification. In this way, the exposition will be organised in a three-axis scheme, starting with the literature involving a significant part of modelling, then presenting some of the works on DNS and LES, to finish with a brief exposition on the new techniques currently being developed.

1.3.2.1 Modelling numerical techniques in urban turbulent flows: RANS and others

The following lines will be dedicated to the presentation of some of the available literature involving a significant amount of modelling. Note, that many numerical techniques involve the use of modelling, e.g. LES typically use turbulence models the subscales of the flow. However, during the current presentation, only works explicitly involving an important amount of modelling will be considered.

Lien et al. [16], presented a work focusing on the predictive capabilities of modelling techniques within the frame of urban environments. In particular, their studies present two models applied independently to be compared with one another as well as with an experiment. On the one hand, the authors presented a classic unsteady RANS (URANS), involing the use of a two equation $k - \varepsilon$ model to compute the turbulence. On the other hand, Lien et al. [16] focused on a partially resolved numerical simulation (PRNS). This numerical methodology, can be considered as hybrid approach since it combines a classic RANS with an LES. This concept was introduced by Shih and Liu [32] and was here applied within the context of urban flows. The idea behind the method is to exploit the strengh of both methods, i.e. having a better accuracy and validity in the results than what is obtained in RANS while keeping cost controlled.
As far as application is concerned, the models were tested over two independent cases to be compared with the experimental data available. Firstly a regular obstacle array is considered. The authors integrate the numerical results of Meinders and Hanjalic [17] which basically presented a flow characterisation precisely over the array here considered. Figure 1.7 show the array used in both the experiment presented by Meinders and Hanjalic [17] and the PRNS presented by Lien et al. [16]. Note that the calculations where performed on a mesh of $45 \times 45 \times 45$ in the three Cartesian components.
The results were analysed essentially by comparison of the streamwise velocity profile. Lien et

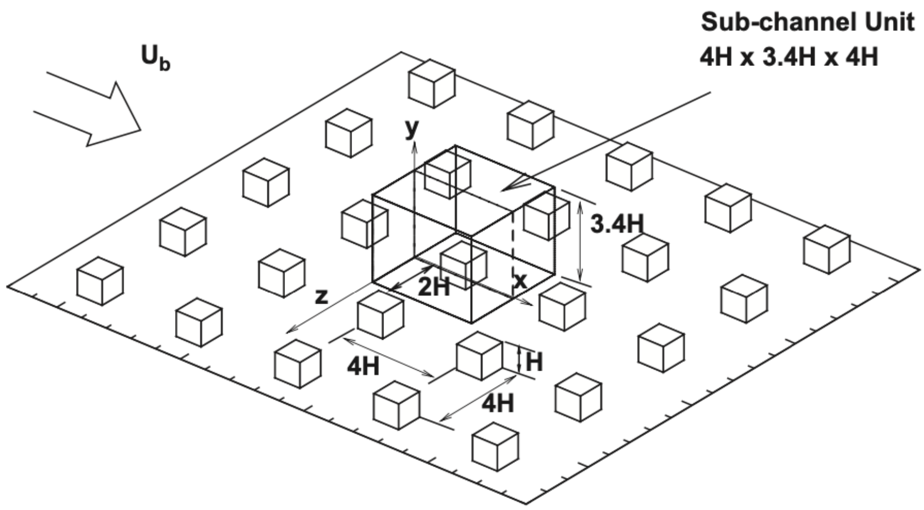


Figure 1.7: Array used in the first case of the PRNS. Extracted from Lien et al. [16].

al. [16] explain that the accordance between PNRS and experimental results was significantly better than in the case of a URANS. In addition, the prediction of the recirculation area was also better in the case of the PNRS comparing with the URANS. This first results confirm the initial guess that the authors had and it is consistent with the conception of the methods which aim lied on the improvement of the URANS approach. Note that Lien et al. [16] don't question the validity nor the limits of the experimental data provided by Meinders and Hanjalic [17]. In this way, one could legitimately question the validity of the experimental data.
The second part of the analysis recovered the Joint Urban 2003 experiment presented by Allwine et al. [1] which aim was to characterise the flow in a central business district (CBD) in Oklahoma city. Lien et al.[16], approached this part of the study analysing the flow field on its own as well as developing a dispersion model from the data obtained with the PRNS. Flow field results were assessed using two metrics, i.e. means speed and wind direction. The flow field was computed using "urbanSTREAM coupled with the steady inflow conditions determined from an interpolation of the flow field" (Lien et al. [16]). The results are then compared with the afore-mentioned experimental data. The quantities were analyse with respect to height. In the case of mean flow speed, the matching between simulation and experimental values is reasonably correct, specially as height grows. On the contrary, Lien et al. [16] show that in terms of wind direction no match is found between both datasets. In addition, the authors run the case using RANS and PRNS methods separately, showing that PRNS values happen to have a better agreement than the RANS "values for $z \leq 200$" (Lien et al. [16]). Once again, the obtained results are consistent with the initial hypothesis formulated by the authors. The dispersion model is developed using exclusively the flow statistics obtained in the PNRS. In this case, "the predictions for mean concentration at or near the mean plume centreline are quite food , with the predicted within a factor of about tow of the observed concentration" (Lien et al. [16]). Despite the good plume's agreement, a deeper analysis reveals a discrepancy between the predicted and the actual values. In fact, the predicted centreline happens to be too far east.
In conclusion, the authors conclude that the considered numerical scheme provides reasonably accurate results, without having to commit to a great computational cost. Using the dispersion model, the authors showed that the interpolated data could reproduce the many features of the flow. However, Lien et al. [16] warn that the afore-exposed results are preliminary and that some further optimisation might be needed to actually reach the complete potential of the technique. Nevertheless, the authors conclude with a encouraging note on the potential of predictive capabilities in the analysis of urban flows.

Some studies combine the use of modelling with higher-accuracy methods such as DNS. The main idea this time is to use some modelling to obtain specific conditions that are then used to run a better simulation scheme. Vinuesa et al. [37] combine lower quality simulation with a DNS in order to improve the efficiency of the simulation. The main procedure consist on running the lower quality simulation to solve the zero-pressure gradient (ZPG) boundary layer in order to incorporate those results in the principal simulation, i.e. the DNS, as a time-dependent inflow condition. The objective of the studies is to show that DNS methodologies can be used in geometries more complex than the classical canonical geometries over which those simulation are historically applied. To do so, they define a complex canonical geometry, i.e. a cylinder over which the simulation will be run. Figure 1.8 shows the domain of both the precursor and the main simulation. It's easy to see that, the authors have optimise the domain by reducing the height of the precursor distribution, since it sole purpose is to simulate the ZPG boundary layer and thus a smaller domain is sufficient.

As far as methodology is concerned, the precursor simulation is run using a Fourier-Chebyshev spectral code, SIMSON. The SIMSON code can simulate simple geometries very efficiently since it

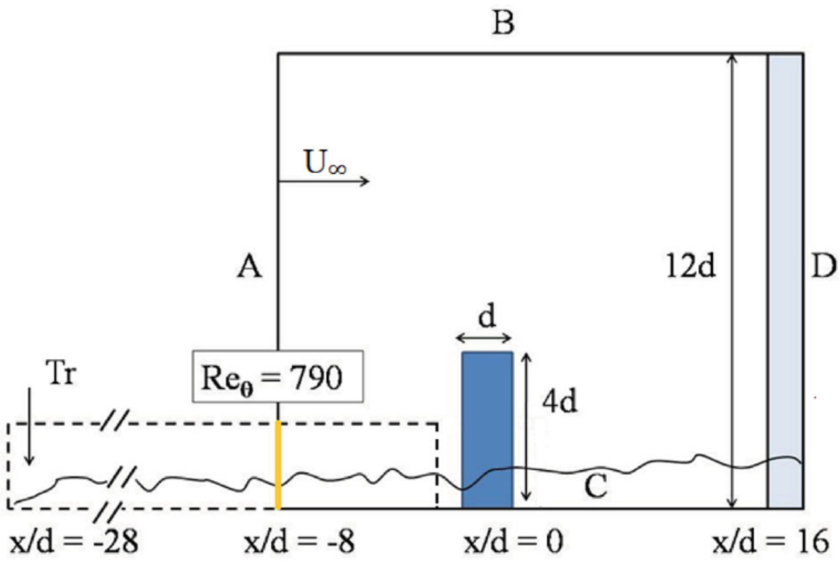


Figure 1.8: Simulation setup showing both the main and precursor simulation's domain. Note that the precursor simulation domain is in dashed line whereas the continuous lines correspond the main simulation's domain. Extracted from Vinuesa et al. [37].

takes advantage of Fourier expansions in both homogenous directions, i.e. spanwise and streamwise. However, SIMSON code is limited in terms of the complexity of the geometry applied. On the contrary, the spectal-based code Nek5000 is applied in the main simulation provides more flexibility but it is less efficient in the computations. In this way, the approach presented by Vinuesa et al. [37], exploits the advantages of both methods, i.e. the efficiency of the SIMSON code is used in the parts where no geometry is found, and those results ar included with in the framework of Nek5000 to be applied in a complex geometry case. The coupling between the methods is done by means of a time-dependent Dirichlet condition that feeds to the main simulation with the solution fields obtained in the precursor computations.

Two test cases are considered in the study presented by Vinuesa et al. [37], one using a laminar [3] inflow and another with a turbulent one. The analysis focus in the comparison between both cases within the frame of instantaneous fields and time-averaged statistics. Note that in this case the results will be described very briefly since our interest actually lies on the afore-described technique rather than in the results. The authors comment that both simulations exhibit the proper behaviour of a turbulent simulation. Thus, the transition to turbulence mechanism appears to function properly. "The flow behind the cylinder is massively separated and exhibits a self-sustained oscillation in both cases, as well as large areas of reversed flow" (Vinuesa et al. [37]). Both simulation capture the near-wall streaks. In the case of the laminar inflow, streaks are only visible after the obstacle, which is consistent with the transition to turbulence. However, comparing the afore-mentioned structures with the turbulent inflow case, one can see that the streaks are less structured in the laminar inflow case. The authors suggest this might be caused precisely by the transition to turbulence. Furthermore, the obstacle's effects appear to be more pronounced in the laminar inflow case. Some additional differences are found by inspection of instantaneous captures of the normalised streamwise velocity fields. In fact, "instantaneously, both wakes have a similar half-width of around $0.8d$ right after the obstacle, up to $x \approx 2.6d$. However, as one moves

[3] Note that the main simulation applies a tripping force strategy to induce the transition to turbulence. In this way, no matter the inflow considered, the main simulation will be turbulent.

downstream the turbulent wake becomes wider than the one in the laminar-inflow simulation, reaching its maximum half-width of around $4d$ at $x \approx 13d$ compared with the half-span of the laminar-inflow case of approximately $3d$" (Vinuesa et al. [37]).
In conclusion, the authors explain that the results have a sufficient quality and the simulation cost is significantly better than the cost that such simulation would have been carrying if fully run using DNS. Nevertheless, Vinuesa et al. [37] also conclude that the spanwise width resulted to be insufficient for the case considered and thus needs to be raised in further applications.
As a final comment on the study here presented, one can easily see that the work of Vinuesa et al. [37], despite not analysis an actual urban canopy, sets the technical basis to use an hybrid methodology that was not applied in such application cases. Furthermore, they also provide an example on how numerical simulations can be used to improve the existent techniques both within the numerical and experimental frameworks.

1.3.2.2 Direct Numerical Simulations and Large-Eddy Simulations studies

The previous paragraphs were dedicated to the introduction of some techniques that involved a significant part of modelling in the solution process. The current paragraph will be dedicated to higher quality solution methods, specially to large-eddy simulations (LES) for being those the object of the work here reported.

Before presenting some of the available literature on urban LES, let us make some comments on the use of DNS in the context of urban flows. As discussed before, DNS have been historically applied to canonical geometries. This is partially due to the high computational cost and the unease to apply it to complex geometries. Nevertheless, studies have attempted to applied this kind of work to more complex geometries and bigger domains. For instance, one can mention the work of Vinuesa et al. [37] that was introduced in §1.3.2.1. However, within the frame of urban environments, this kind of geometries still remain relatively simple and small. Perhaps, the main limitation of DNS in the case of urban environment is precisely the size of the domains and thus the computational cost. In fact, urban canopies tend to have an important size, e.g. the experiment presented by Hirose et al. [13] reported in Figure 1.3. Therefore, improvements need to be made on the numerical schemes such that DNS can be applied in feasible time. That is precisely, what was aimed and exposed in the work of Vinuesa et al. [37]. Nevertheless, LES, if properly set, can provide very similar results to DNS while keeping a significantly lower cost. That is why, LES is the method of choice in the work here presented, as we will see later on. The avid reader might consult §2.2.2 for an early explanation on the general LES methodology.

Once again the application of those methodologies typically aims to have some degree of flow prediction within the context of the case. García-Sánchez et al. [10] presented a study which objective was to study the uncertainty sources that lie beneath the LES method while comparing it with a RANS and an experiment. The study is run within the context of urban flows and in fact the experimental data is recovered from the now well-known Joint Urban 2013 Experiment.
The LES simulation is run using the OpenFOAM coding environment. The considered domain differ in the RANS and LES cases. In fact, the domain was chosen smaller in the LES case with the purpose of constraining computational cost, the authors argue. In addition, the RANS domain allows different inflow-outflow directions, thus reproducing a changing wind direction. On the contrary, in the LES wind direction is fixed. In both cases, the domains cover the zone of interest, i.e. Oklahoma downtown. The mesh was created keeping a cost-resolution balance, i.e. determining refinement zones, four in total, where the element size vary. In this way, mesh resolution progressively increases towards the city's downtown while keeping computational cost controlled.

Figure 1.9 shows a schematic representation of the zones.

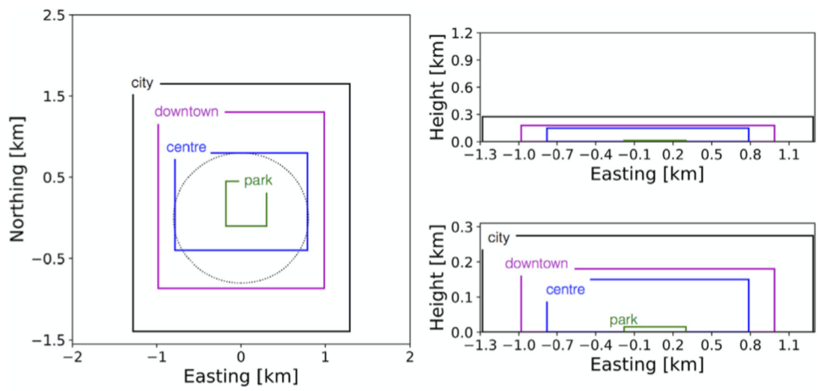


Figure 1.9: Two-dimensional scheme on the refinement areas in the LES domain. Extracted from García-Sánchez et al. [10].

Following the afore-mentioned meshing approach, five levels of refinement are defined starting from a coarser resolution at the city level and progressively increasing resolution, thus reducing the element size, up to the finest area at the park. Refinement is actually one of the most common and effective techniques to ensure a sufficient resolution while keeping cost controlled. In fact, this technique will be used in the application cases here considered.

García-Sánchez et al. [10] assess results by means of flow visualisation, considering both flow fields and time-averaged statistics. Flow fields are visualised at eight meters height and reveal "how the flow impacts the buildings generating recirculation areas with large-scale unsteady structures in the wakes of the geometries" (García-Sánchez et al. [10]). Time-averaged statistics, show that there is a very good agreement between the LES and RANS results in terms of the mean velocity with the exception of some localised areas where LES predicts a stronger acceleration. Furthermore, by analysing the trend of the difference one can see that the upstream zones of the domain present bigger discrepancies in terms of the mean velocity. A deeper analysis of the turbulence reveals bigger discrepancies between the methods, LES outperforms RANS at a rough 80% in the turbulent kinetic energy compared with experimental data. In addition, "the LES produces turbulence spectra that are in good agreement with the measurements regarding the scales and energy content of the turbulence" (García-Sánchez et al. [10]).

The authors conclude highlighting the need of verification in numerical simulations, precisely at the areas where there is better agreement between RANS and LES. In fact, García-Sánchez et al. [10] found in such areas the sensitivity of RANS to inflow is high. For the authors, this fact suggests that "a higher-fidelity turbulence model does not guarantee a better prediction of the flow in areas with large uncertainty related to the inflow boundary condition" (García-Sánchez et al. [10]). Thus leaving room for improvement.

In conclusion, the authors here present a vast evaluation of the uncertainties that underlie numerical simulations and in particular LES. Their final closing sustains that while LES provides very useful information on the flow structures, in some areas the obtained results show insignificant variation from RANS results. While this last statement is certainly true, one must keep in mind that the quality of LES lies very importantly on mesh resolution. In this way, depending on the "standard" applied the quality of the result might vary. For instance, the Linné Flow Centre at KTH Royal Institute of Technology is known for using *well-resolved* LES. Those simulations, as it will be exposed further on, have a quality very close to DNS methods.

1.3.2.3 Enhanced techniques: Application of machine learning methods to the study of urban turbulent flows

DNS or LES despite being complex techniques that are still under development within the frames of urban flows, are well-known techniques that have been providing results in several domains of physical studies. However, regardless of the application considered when dealing with flow simulation it seems that one is always confronted with the cost-quality dilemma. Either one choses to have a very good quality simulation at the expenses of cost or one chooses to have a simulation with less cost and thus with low quality. This is particularly true in the study of urban turbulent flows as domains tend to be large and flows complex. That is why new studies have tried to apply different methodologies to ease the cost of such simulations. Machine learning is one of the trending techniques to deal with vast amounts of data. In this way, part of the recent research is focusing on the application of machine learning (ML) to urban turbulent flows. The main underlying idea is to be able to train neural networks to reproduce and predict flow solutions.

Xiao et al. [6] presented a study of turbulent airflows modelling using a neural network. Their aim was to develop a fast-running non-intrusive reduced order model (NIROM) for predicting turbulent airflows within urban environments. Figure 1.10 shows the fundamental steps in the production of the NIROM.

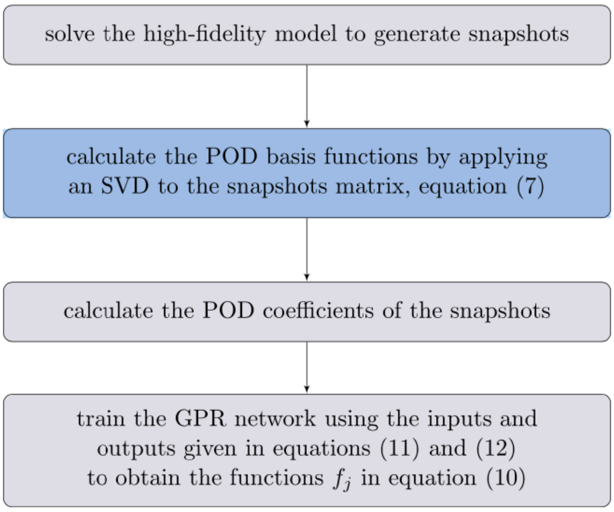


Figure 1.10: NIROM off-line production stages. Extracted from Xiao et al. [6].

In this study the feeding model is assumed to be known. Thus the first actual step in the production of the NIROM is to obtain the proper orthogonal decomposition (POD) functions. This is done using a singular value decomposition (SVD) method which is applied over the snapshot matrix data. Xiao et al. [6] explain that they chose to apply the method simultaneously to all velocity components. This approach aim to capture the correlations that arise between the velocity components naturally.
Once the functions obtained, a Gaussian process regression (GPD) is run to obtain the surface representation here needed. This operation consist on the application of linear combination of Gaussian-shape basis functions. The main advantage of this method, Xiao et al. [6] comment, lies on the small amount of data required to make it function. Other methods are available to obtain the surface functions, e.g. feed-forward neural networks. However, theses typically involve optimi-

sation problems difficult to properly solve.
Now that the POD have been fully characterised, the NIROM is derived. To do so, the authors train a neural network to predict the behaviour of the flow governing equations. The idea is that one trains the network using the high-fidelity model in order to obtain a system that will later on predict the behaviour of those equations, this time without the need of snapshots. Note that the snapshots are projected over a reduced space spanned by the POD base functions obtained previously.
To validate the NIROM, Xiao et al. [6] modelled the airflow around London South Bank University. They demonstrated the ability of the network to "reproduce snapshots and [...] that NIROM is capable of making predictions beyond the range of the snapshots" (Xiao et al. [6]). In fact, the results showed that the method can accurately represent the vast majority of the dynamics showed in the high-fidelity model. They observed that means flow are very well predicted even using a small amount of POD base functions. However, the Reynold stresses required a higher amount of base functions to be properly represented. Thus using a high number of POD base functions, it appears that a vast amount of quantities can be predicted. As far as computational cost is concerned, the authors argue that even with a high number of POD functions, the network "is six orders of magnitude times faster than the high-fidelity model" (Xiao et al. [6]). Nevertheless, the authors also add that further studies research will be needed to be able to expand the capabilities of this methodology to further domains.

1.3.3 Final comments on the historical perspective

Finally to close the historical perspective section let us report some general concepts on the literature. The previous lines were dedicated to briefly explain some of the available literature in experiments and simulations on urban flows. Generally speaking, we saw how experiments tend to provide better results when applying over the specific study of a urban area while simulations were more useful to understand the behaviour of the flow.

Focusing on numerical simulations, it appears that the range of applicability and the quality of the process are somehow confronted. This is that either the method is high-quality but tailored for a given application or the method can be generally applied but quality is reduced. In addition, coming back to the different methodologies it appears that the only method that can provide a deep description of the flow turbulence is LES, besides DNS. In this way, for theoretical applications, i.e. the study of the actual physics of turbulent flows, LES appear to be the right methodology. That is why this project will be focusing in that precise methodology and specifically in creating a systematic procedure to create high-quality LES in urban environments.

2

Theoretical background

Now that a general picture has been given on the past and current situation of the scientific research on urban flows, it is time to get into the fluid's physics, both from the descriptive and formal point of view. In this part, the idea is to provide to the read a sufficient understanding on the behaviour of the flow in urban environments as well as the mathematics that allow to describe it.

2.1 Flow physics : Theory in Fluid Mechanics

First of all, as the method behind any numerical resolution, it is advisable to try to understand the physics of the fluid in an empirical approach before getting into mathematical tools that allow to fully describe the flow's behaviour.

Flows in their general conception can be classified by means of different characteristics. For instance, one could classify the flow by general aspect such as it composition but also with more specific parameters such as the speed or the pressure. Several options are available and their selection strongly depends on the problem that one is solving. However, one of the most common ways to classify a flow, is by its behaviour. In fact, flow behaviour can mainly be constrained into two categories, **laminar** and **turbulent** flow. Their characteristics are significantly distinct to the point of being able to distinguish one another at plain sight. In addition, their solving is also very distinct as it requires the use of different methods and involves distinct assumption. Those among many facts are the reason why the way of classifying the flows is so widely used. Note, that in the work here exposed, only turbulent flows will be described, as all the cases further studied will be dealing with turbulent flows.

2.1.1 Turbulent flow

Let us for the moment blindly assume that the flow in a urban environment is fully turbulent, the complete explanation will be developed in further parts. Turbulent flows are quite common in our everyday life. Whether one is in front of a waterfall, seeing the smoke coming out of chimney or using a household sink, one is exposed to turbulent flow. Empirically, as Pope enunciated in his work *Turbulent flows* [25], can be understood as an unsteady, irregular and seemingly chaotic flow for which the prediction of the motion of every droplet would be unpredictable. From this simple definition, one can already imagine that the problem will be characterized by an important degree of complexity which, as it will be exposed in further parts, will require the use of rather sophisticated numerical tools.

An implicit fact of the description afore-presented is that the turbulent flow is inevitably characterised by a irregular variation in both space and time. This is particularly the case for the velocity. Obviously, depending on the studied case, one might neglect some of the variability but in its most general form, one should consider full variability for this kind of flows. This is one of the sources of complexity in the study of turbulent flows even in the most simple cases. Figure 2.1, presents the time history of the axial velocity in a turbulent jet.

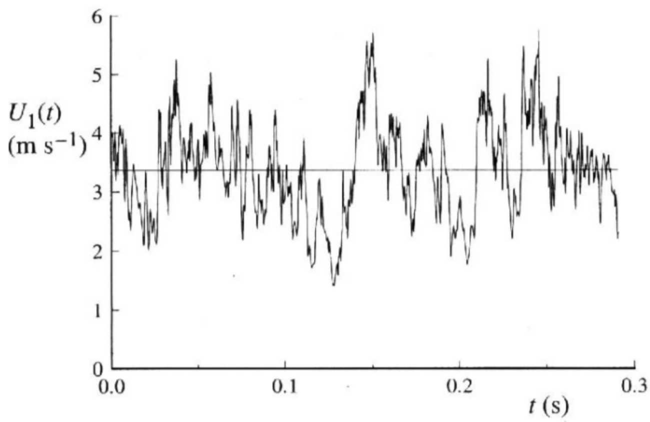


Figure 2.1: Time history of the axial component of the velocity on the centreline of turbulent jet. From Thong and Warhaft [33]. Extracted from Pope [25]

.

One can see that there is not any visible trend in the curve presented in Figure 2.1 and that the the variability within the curve is too important to even try to fit a model. In this way, the reader might now have an idea on the degree of complexity to which we are exposed in this kind of study. A direct consequence of this is the need of statistical tools. Those kind of tools will be presented in further sections.

Moreover, as stated in the first paragraph, our world is full of turbulent flows which makes that engineering application can not ignore them. As far as the engineering toolbox is concerned many approaches are concerned going from the use of empirical models to the full resolution of the fluid mechanics equation with numerical tools. The resolution methodology will be presented in further sections.

Now that a qualitative description of a turbulent flow has been given it is time to formalise a more proper definition. This is where the Reynolds number needs to be introduced. The Reynolds number is a non-dimensional coefficient developed by Oswald Reynolds while he was studying the transition from laminar to turbulent flows in pipes. In his experimental investigation of the motion of water [27], Reynolds observed that there was a relation between the fluid behaviour and relation of the dimensional properties in the body to be studied. The results he presented complemented the insights provided by Professor Stokes which focused on the development of the equation of motion, but, in Reynolds words, “might contain evidence which had been overlooked, of the dependence of the character of motion on a relation between the dimensional properties and the external circumstances of motion”[27]. The result, mainly distilled from his experimental observations, can be synthesised in the following expression,

$$Re = \frac{Ul}{\nu} \tag{2.1}$$

In this way, the Reynolds number can be understood as the ratio between the dynamic and viscous forces.
The Reynolds number, is the main parameter that is used to distinguish laminar and turbulent flows. In the Reynolds experiment if the Reynolds number was below 2300 the flow is laminar,i.e. the flow does not change with time [25]. On the contrary if $Re > 4000$, the flow is fully turbulent.

Now that it is clear what a turbulent flow is and how one can identify it, it is time to get into the tools that allow to describe such a physical problem.

2.1.2 Fluid mechanics equations

The main objective in the subsection is to introduce the mathematical expressions that allow to describe a turbulent flow and that are the ground base of the solution process. Note that for the moment, the resolution schemes will not be tackled as they belong more to the discipline of numerical methods than to the physics of fluids itself. Also note, that it's assumed that the reader is familiarised with the basic equations of motion and physical principles, as those won't be reviewed in the current text.

In the following lines, the full Navier–Stokes equations will be progressively built stating with the description of the physical principles that exude the terms composing the equations up to the full expression of those.

Before getting into the inner details behind the physics of the flow, it is advisable to review some of the basic concepts that constitute the foundation of the derivations in fluid mechanics. Note, that in both this preliminary exposition and the derivation of the Navier–Stokes equations, the work here presented will be following the reasoning once exposed by Stephen B. Pope in his masterpiece *Turbulent flow* [25], please consult the mentioned text for a more complete derivations and comments.

2.1.2.1 Continuum media and general properties of the fluid

When dealing with liquids and gases, two approaches have been historically present in the scientific work. On one side, considering the fluid as a system of study by itself, one can look at a microscopic level how the fluid's particles behave. On the other side, one can consider that the flow is continuous medium and hence analyse it at a macroscopic level, i.e. without taking into account the motion of each of the particles that compose the fluid. This is what is known as the **continuum hypothesis** and it's the foundation of the analysis that is here exposed.

The continuum hypothesis can be justified, as exposed by Pope [25], by comparing the orders of magnitudes of the mean velocities of the molecules and the smaller geometric length scale that in which a flow can be described. The findings presented by Pope show a difference in more than three orders of magnitude between the molecular velocities and the smallest of the length scale. Hence, for the applications of the work here presented, one can safely assume that difference at a molecular level are not significant at the level of the whole fluid, i.e. the continuum hypothesis holds.

Once assuming the continuum hypothesis, the differences in the molecular scales are ignored and one can start invoking the continuum variables that will be, in their most general representation functions of space and time in macroscopic scales. Note that in the whole set of mathematical descriptions and derivations the continuum hypothesis is assumed and therefore all the variables will be continuum variables.

2.1.2.2 Continuity equation

Historically in fluid mechanics, it is common to start by the mass conservation principle and therefore with the continuity equation as a starting point in the derivation of the fluid's equation. Note, that it is assumed that the reader is familiar to the concept of control volume as well as the Lagrangian and Eulerian reference systems.

Consider a general control volume, the mass conservation principle can be written in the following form,

$$\frac{\partial \rho}{\partial t} + \nabla(\rho U) = 0 \tag{2.2}$$

Equation 2.2 is known as the continuity equation. It is the mathematical representation of the mass conservation principle, i.e. that no mass can be created nor destroyed in the process.

As far as the components of Equation 2.2 are concerned, the time derivative term is known as the accumulation term. It represents the quantity of matter that is accumulated in the control volume for a given time duration. The gradient term, describes that amount of matter that that going inside or outside the control volume. Note that Equation 2.2 is the most general expression of the continuity equation, meaning that depending on the case one might simplify it assuming a steady flow which makes the time derivative term vanish or assuming that the flow is incompressible, i.e. ρ is constant.

2.1.2.3 Momentum equation

While the continuity equation already provides a significant amount of information, it is clear that the mass conservation principle is, by itself, insufficient to fully describe the behaviour of the flow, since it does not consider how forces are interacting with the control volume among many other things.

From classical mechanics theory, it is at this time well known that when dealing with the equations of motion of any system, Newton's second law needs to arise. Newton's seconds law basically relates the acceleration of a given particle to the forces that are affecting the system. As far as the momentum equation is concerned, it is simply the application of Newton's Second Law to a control volume, using the terms that allow to describe the momentum in a fluid.

Before getting the full formulation of the momentum equation, following Pope's[25] approach, the different forces have to be formulated. When dealing with flows, two types of forces are typically involved. One the one hand, **surface forces**, which are mathematically described by the symmetric stress tensor, $\tau_{ij}(\mathbf{x}, t)$. On the other hand, **body forces**, which are forces that are applied in the whole control volume. In this case, for the vast majority of applications, it is the gravitational forces that is interesting. Many formulations are available for this term. In the work here presented Pope's[25] formulation will be followed and hence the gravitational force is described in terms of the gravitational potential, i.e.

$$\mathbf{g} = -\nabla(\Psi) \tag{2.3}$$

Now, by direct application of Newton's Second Law,

$$\rho \frac{DU_j}{Dt} = \frac{\partial \tau_{ij}}{\partial x_i} - \rho \frac{\partial \Psi}{\partial x_j} \tag{2.4}$$

Assuming that the fluid is Newtonian, the stress tensor can be replaced by its corresponding expression, and Equation 2.4 can be expressed as:

$$\rho \frac{DU_j}{Dt} = \mu \frac{\partial^2 U_j}{\partial x_i^2} - \frac{\partial P}{\partial x_j} - \rho \frac{\partial \Psi}{\partial x_j} \tag{2.5}$$

Equation 2.5, is known as the Navier–Stokes equations. In this case the equation are presented in tensor notation. Introducing the modified pressure, i.e. replacing P by the following expression:

$$P = p - \rho \Psi \tag{2.6}$$

Equation 2.5 can be written as,

$$\frac{D\mathbf{U}}{Dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{U} \tag{2.7}$$

Now, by looking at different terms in the Navier–Stokes equations (Equation 2.5 and Equation 2.7), the reader can remark the use of the total derivative, which is by itself taking into account both the time variation of the velocity and the variation which respect to the space. Note that in the expressions here presented, the forcing terms considered are pressure and gravitational force. Any additional terms related with forcing, should be added by summation in its correct form depending on whether it's a surface or body force.

2.1.2.4 Passive scalar equation

Up to this point, the continuity and momentum equation have been covered. However, one might very well how to consider additional quantities such as the temperature or any other magnitude related with the flow. In fact, the equations presented up to this point do not consider such additional quantities explicitly. That is why some addition expression needs to be introduced.

Let us consider a passive scalar $\phi(\mathbf{x}, t)$ that is conserved over the flow. The term "passive" denotes a significant assumption that it is introduced when dealing with this kind of magnitudes, it's assumed that the magnitude in question does not have an effect on the material properties of the flow, i.e. on the velocity, pressure etc.

In a constant property flow, the conservation of the passive scalar can be expressed in mathematical terms by means of the following expression,

$$\frac{D}{Dt} \left(\phi(\mathbf{x}, t) \right) = \Gamma \nabla^2 \phi(\mathbf{x}, t) \tag{2.8}$$

Note that the diffusivity here represents the diffusion of the property consider. For instance if one is considering the temperature as the passive scalar expressed in Equation 2.8, the diffusivity Γ, would be the thermal diffusivity.

With Equation 2.8, one can introduce some additional magnitudes of interest in the study of the flow. Please, consult the works by Pope[25] for a complete derivation of the passive scalar conservation equation as well as a deeper study on its implications.

2.1.2.5 Vorticity equation

The afore-stated equations are considered basic in fluid mechanics and thus need to be introduced whenever trying to find the behaviour of any flow, both laminar and turbulent. However, it is well known that turbulent flows are rotational flows, i.e. vorticity needs to be taken into account. That is why, to conclude the exposition on the Fluid's equation an additional equation needs to be introduced.

Before getting into the vorticity equation, let us introduce the vorticity concept. Vorticity is tool that is used to describe the local spinning of the flow. Mathematically, the vorticity is known to be the curl of the velocity, i.e.

$$\omega(\mathbf{x}, t) = \nabla \times \mathbf{U} \tag{2.9}$$

By applying the curl to the Navier–Stokes equations (Equations 2.5 and 2.7, one can obtain the **vorticity equation** (Equation 2.10).

$$\frac{D\omega(\mathbf{x}, t)}{Dt} = \nu\nabla^2\omega(\mathbf{x}, t) + \omega(\mathbf{x}, t)\nabla\mathbf{U} \tag{2.10}$$

2.1.2.6 Synthesis: Fluid Mechanics Equations

In the previous lines, the different equations relevant for the analysis of a turbulent flow. As a conclusion to this section, let us recap most relevant expressions.

$$\begin{aligned}
\text{Continuity equation}: &\ \frac{\partial\rho}{\partial t} + \nabla(\rho U) = 0 \\
\text{Momentum equation}: &\ \frac{D\mathbf{U}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{U} \\
\text{Passive scalar equation}: &\ \frac{D}{Dt}\left(\phi(\mathbf{x}, t)\right) = \Gamma\nabla^2\phi(\mathbf{x}, t) \\
\text{Vorticity equation}: &\ \frac{D\omega(\mathbf{x}, t)}{Dt} = \nu\nabla^2\omega(\mathbf{x}, t) + \omega(\mathbf{x}, t)\nabla\mathbf{U}
\end{aligned} \tag{2.11}$$

With the different expressions synthesised in Equation 2.11, the fluid mechanics theory is considered complete. In the following sections the methodology to solve those equations the cases of interest will be presented among other concepts.

2.2 Computational Theory

In the previous section, the Fluid Mechanics equations were briefly introduced. The idea now is to present some of the methodology that can be used in order to solve the afore-stated equations. Note, that the available literature on this matter is way too vast to be fully considered in the work here presented, hence, only methods suitable to the problem studied in this project will be considered.

Before getting into the details of the methodology used, let us present a general introduction of te workflow in computational fluid mechanics.

2.2.1 Workflow in Computational Fluid Mechanics

Any project in computational fluid dynamics generally follows a scheme consisting in **preprocessing**, **solving** and **postprocessing**.

Preprocessing Preprocessing covers all the necessary steps that precede the solving process of the numerical problem in question. This phase starts by the setting of the physics to be studied. It is clear that one should start by trying to understand the physics behind the problem, since as it will be exposed in further sections, it is essential to be able to validate the results obtained. Once the physics are sufficiently known, the mathematical model needs to be evaluated, i.e. the equations needed to describe the physics are gathered. Finally, it is time to set the solution strategy to used, i.e. the numerical methods that will be applied. Although each method has its own characteristics and limitations, some stages in the solution process are common to the vast majority of them.
The starting point in the resolution scheme typically involves a discretization of several quantities, very often space and time. Conceptually, the idea is to numerically solve the equations in a limited amount of points, hence the continuous domain needs to be discretised in a set of points. Note, that in general terms and within a method, the greater the number of points the better the solution. However, with the increase of points, the computational cost is raised as well, thus, a trade off between accuracy and cost needs to be attained. Further details on this matter will be given on the specific problem that will be presented later.
Another important matter when solving partial differential equation such as the Navier–Stokes equations are boundary conditions. In fact, to be able to find a solution, the values of some quantities, for instance the velocity need to given *a priori*. Once again this depends on the physical understanding that one has on the problem.
When dealing with turbulent flow, there is an additional step to be considered. Typically it is very convenient to have a model for the turbulent quantities of interest. The use of such models, avoids solving the equations in the complete range of points in space and time, saving an important amount of computational cost. Note, that the simulations in which the Navier–Stokes equations are solved in the full range are known as *Direct Numerical Simulations* (DNS). No further details will be given around DNS since those methods are out of the scope of the work here presented. Also note, that once again many different models are available. The next section will be dedicated to the presentation of the methods that are interesting for the problem here solved.

Solving Once the equations are set, and the domain is discretised, an solution scheme is applied. At this stage, the physics of the problem play a secondary role since, from this perspective it is just a numerical problem to be solved. Note that computational fluid dynamics problems tend to involve a very important amounts of data. Hence, an efficient solving scheme is mandatory such that the problems solutions are feasible.

Postprocessing After solving the problem, it is necessary to treat the data such that conclusions can be obtained. The treatment will be very dependent on the problem and its application. Postprocessing could go from simply plotting the computed quantities to the use of those quantities in additional computational process. For instance, when dealing with turbulent flows it is very common to obtain the statistics of several magnitudes such as the velocities.

Now that a brief introduction on the main stages in fluid simulations, let us focus on the methods that are useful to the problem that will be introduced in further chapters.

2.2.2 Turbulent Modelling: large-eddy imulations

Although computational cost has not been directly addressed up to this point, it is one of the most critical concepts along with stability and accuracy when dealing with fluid mechanics simulations. In fact, one needs to ensure that the problem can be solved with the available computational power in feasible time. This is particularly important when dealing with turbulent flow simulations, as it is the case in the work here presented. Focusing only on accuracy, direct numerical simulations (DNS) clearly provide the best results, since those kinds of simulations solve the equations describing the problems, e.g. the Navier–Stokes equations, in every scale of points. However, DNS tend to have a very important simulation cost and usually have problems when the complexity of the geometry is increased. Although, the problems that will be addressed in further sections do not present an important geometrical complexity, DNS will not be considered as a simulation strategy. In fact, for the problems here treated the raise in accuracy obtained by using DNS does not justify its use. Hence, an alternative has to be considered. The natural tendency, as in any other engineering application, is to develop models.

Models allow to obtain solutions without solving the equations in every scale. Since the equations are not solved in every point it is clear that some information is lost and therefore the accuracy of the method decreases. It is part of the preprocessing task to determine which model provides a sufficient accuracy to solve the problem in question. This is usually done by analysing the pre-existent literature. Many different models of turbulence are available, each one having different levels of accuracy. In the work here presented, the Large-Eddy model will be applied.

2.2.2.1 Fundaments of Large-Eddy Simulations

The basic idea behind **large-eddy Simulations** (LES) is that large motions are directly represented, i.e. solved, whereas the smaller motion are modelled. The motivation behind is based on the fact that computational cost is not uniformly distributed over the scale. For instance, when dealing with DNS simulations the vast majority of the computational cost is produced by the smaller scales (Pope 2010 [25]). Hence, by modelling the smaller scales one could significantly improve the computational cost of he method. In addition, as described by Pope (2010 [25]), the energy and anisotropy are mainly contained in the large scales which are affected by the flow geometry and hence are not universal. On the contrary, smaller scales tend to have, to some extend, a universal character. Under this assumption, it seems that the LES approach could be an efficient way to reduce computational cost.

Fundamentally any LES can be stripped into four main steps which are key to the inner workings of the method. Note, that this description follows the approach presented by Pope [25] as well as its notation.

First of all, during the **filtering** process velocity is segregated depending on whether it corresponds to large or small scales. In that way, the velocity term $\mathbf{U}(\mathbf{x}, t)$ is decomposed in the sum of a filtered component (i.e. component corresponding to larger scales) $\overline{\mathbf{U}}(\mathbf{x}, t)$ and a residual component (i.e. component corresponding to smaller scales) $\mathbf{u}'(\mathbf{x}, t)$. Secondly, the Navier–Stokes equations are applied to derive the equations involving the filtered components. Thirdly, the residual-stress tensor is obtained by applying a pre-existent model, for instance in its most simple version an eddy-viscosity model. Finally, the model filtered equations are solved numerically using any of the available tools in numerical methods. This last stage will be further developed for the tool that will be applied in the work here presented.

Now that a general vision of the LES method has been presented it is time to introduce the specific theory behind the simulation. Note that the aim of this part is to provide a brief introduction on the theory over which LES are based. However, if the reader is avid to deepen in such theory, Pope's work *Turbulent Flows* and Sagaut's text *Large-Eddy Simulations for Incompressible Flows* [28] cover a complete explanation on the topic. Let us now introduce the formal expression of the LES filter as exposed on Leonard [15]. Consider a generic function $f(\mathbf{x})$ all the . Using the notation introduced in the previous paragraph, $f(x)$ should be decomposed in a function containing the large scales, i.e. $\bar{f}(\mathbf{x})$ and a function gathering the smaller scales, i.e. $f'(\mathbf{x})$. Note that the apostrophe over f variable does not represent the derivative of f, but rather a function applied over the smaller scales. Now, let us define a filter function G which produced the afore-stated segregation. By integration of the convolution of the function f and the filter function G the large scales quantity is found.

$$\bar{f}(\mathbf{x}) = \int G(\mathbf{x} - \mathbf{x}')f(\mathbf{x}')d\mathbf{x}' \tag{2.12}$$

Applying it now to the velocity term, it is found :

$$\overline{\mathbf{U}}(\mathbf{x}, t) = \int \mathbf{G}(\mathbf{r}, \mathbf{x})\mathbf{U}(\mathbf{x}, t)d\mathbf{r} \tag{2.13}$$

Note, that the filter function $\mathbf{G}$ is actually a transfer function, if ones uses the classical control theory terminology. In that way, this function can take many different forms depending on its shape. For instance one could consider some standard transfer functions such as the box function, the Gaussian function etc. The selection of the filter function will mainly be dependent on the particular application of the method.

As stated it the introduction of this section, the filtered quantity $\overline{\mathbf{U}}(\mathbf{x}, t)$ and the residual quantity $\mathbf{u}'(\mathbf{x}, t)$ are related by superposition, i.e. :

$$\mathbf{U}(\mathbf{x}, t) = \overline{\mathbf{U}}(\mathbf{x}, t) + \mathbf{u}'(\mathbf{x}, t) \tag{2.14}$$

Once the velocity term is filtered, the equations of the filtered components are derived form the Navier–Stokes equations. Smaller scales are obtained by means of modelling while larger scales are obtained from any of the numerical methods available. This last stage will be assessed later on.

Now that some of the basic principles behind the LES method have been presented, it is time to see this methodology in the perspective of the available strategies to solve this kind of problem. The next paragraphs will be dedicated to the evaluation of the strengths and weaknesses that LES have comparing to the other mainstream approaches currently used.

2.2.2.2 LES performance appraisal

The aim in this part is to look at LES with critical eyes by trying to evaluate to strengths and weaknesses of this approach. The assessment in question will be carried following a qualitative approach rooted on the available literature on the topic. Note that a finer appraisal could probably be done following a quantitative approach. However, for the sake of the work here presented a general understanding on the performance of the method should be more than sufficient. Nevertheless, we strongly encourage the avid reader to consult the works of Chapman (1979) [4] as well as the well known book of Pope, *Turbulent Flows*[25].

The appraisal here presented, following Pope's approach, will be supported on five fundamental pillars.

1. **Level of description** : This criteria is related with the physical characteristics of the flow and how those relate to the computational method applied. For instance, the use of statistical quantities such as means can provide a sufficient description for some flows but might not be enough for other specific cases. The selection of the computational method based on this criteria will depend on several aspects such as the inner nature of the flow, the application of the study in question etc. In practice, once again, this evaluation is based on previous experience both generally and precisely.

2. **Completeness** : Completeness is mainly evaluated by inspection of the equations that build the model and how they relate with the flow. A model is considered complete if the equations that are involved do not depend on flow specifications. For example, DNS is complete since the equations are solved in every point and hence no flow specification is required. On the contrary, a mixing-length model introduces a flow specific quantity, the mixing length and hence it is incomplete.

3. **Cost** : Up to this point cost has already been accessed in several occasions. However, the main objective in this block is to provide a more detailed explanations of the factors that influence cost in computational fluid dynamics. It is necessary to distinguish between models and fully computational approaches such as DNS, since they tend to behave very differently as far as cost is concerned.

    (a) Turbulent models : In turbulent models the cost is mainly related with the mathematical objects that are inside the model. For instance if the problem is statically stationary the cost tends to decrease.

    (b) DNS : In direct numerical simulations, there is a strong direct relation between the computational cost and the Reynolds number. As the Reynolds number is increased the cost tends to rapidly raise. This is in fact one of the motivation in developing turbulent models since high Reynolds number are a well-known characteristic of turbulent flows.

    In general, the computational process in any flow simulation can be divided into two main stages. Firstly, the equations of the simulation need to be solved in order to attain certain conditions. For instance, one needs to spend some computational power to achieve turbulent conditions. However, further on we will see that there exist ways to reduce the cost in this area by “smartly” induce the turbulence, for instance with the use of a *tripping* force. This cost will depend on the flow target conditions, the models and the computational tools available. However, it is important to know that this is single time cost, since once the flow conditions achieved, one can save the fields and then perform different simulations with them. Secondly, once the flow conditions are reached, the flow needs to be solved in a specific geometry. This cost is obviously recurrent, since it is the problem itself and hence will be different for every application. Again, this recurrent cost will depend on the flow conditions, model applied, computational tools etc.
    In practice, computational cost is typically assessed in terms of computational time. The idea is whether a given problem can be solved in feasible time. This importantly related with the available computer. The use of supercomputer has allowed up to today to raise the limit possible computations. In this area, quantum computing appears to be a very promising tool. However its application in fluid mechanics simulations remains unavailable for the moment. Also note, that there are computational techniques that allow to decrease computational time, such as the parallel computing protocols, which basically, split the problem to run it simultaneously in many different processors.

4. **Applicability** : Not all the models can be applied to every problem or flow. This fact can be compared with the way in which classical fluid mechanics simplifies the flow equations. In every model there are a set of assumptions which have to be coherent in the specific application considered. For instance, in classical fluid mechanics one can a assume that the flow in a pipe is one dimensional if the pipe is long enough and the its cross-section is constant. However, if the diameter of the pipe is increase over a certain limit, the one dimensional flow assumption can not be applied any more. The same kind of principles applies in fluid numerical simulations, specially when dealing with models. Recalling the example of mixing-length models, those typically include assumptions regarding the flow geometry, hence their applicability range is constrained to a certain set of flow geometries. In the case of DNS, there are not such theoretical limitations. However, due to the relation between cost and Reynolds number, one can state that its applicability is constrained to low or moderate flows.

5. **Accuracy** : Accuracy si one of the most important aspects of a model. In principle, higher accuracy implies a better model. However, maximising the accuracy of a model is not an objective *per se*, since typically accuracy balances with other factors such as cost or applicability range. Accuracy is usually measured by comparing the numerically solved results with experimental results. Note, that this process it done for every numerical simulation, but rather in specific which serve as reference to further studies.

This appraisal system, far from being perfect or unique, aims to provide different angles of evaluation to a given methodology. Note that there is not a "perfect" model when dealing with numerical simulation nor one can state that one of the afore-mentioned pillars is more important than the others. In practice, the discrimination of model over another is typically the results of mixed weighted evaluation of the afore-stated criteria as well as other factors such as the available resources, expertise or application of the specific task.

Now that a brief introduction on the evaluation criteria has been presented, this appraisal scheme will be applied to the LES method which is the one that will be applied further on. The aim in this part is to provide the reader with a general picture of the reasoning behind LES selection as main computational methodology in the work here presented.

Level of description It is well known from flow physics that the energy-containing motions are typically large scale motions. In this way, since LES does solve the proper fields in the large scales, the level of description is usually sufficient for many different kinds of applications. As far as the smaller scales are concerned, the level of description might vary depending on the model applied but usually they do not present significant problem in this area. Further on, it will be exposed how some methodologies allow to increase the level of description and accuracy the smaller scales will keeping the cost restrained thanks to the use of a *well-resolved* LES.

Completeness Recall that the completeness of a simulation is determined by whether the solution depends on the flow. In the particular case of the LES the completeness can not assessed for the methodology in general. In fact, the completeness of an LES mainly depends on the filter width. As Pope's clearly presents in *Turbulent Flows* [25], it is very common in turbulent LES to define the filter width proportional to the local grid size, which is predefined. In such case, if the LES solution is dependent on the grid definition parameters, e.g. grid size, the method is, by definition, incomplete. However, the grid can be defined such that the vast majority of the energy-containing motion is resolved in the domain, i.e. in the larger scales. This time, contrary to the prior case, the LES solution will not have such dependence on the grid parameters and therefore the simulation can be considered complete.

Cost Cost in LES is a deep topic on its own. In fact, the cost of a LES is related with different factors such as the type of flow or turbulence considered. For instance if one consider an homogenous isotropic turbulence, 40^3 modes are sufficient to solve 80% of the energy (Pope [25]). However, the simulation of a wall-bounded flow would have significantly different resolution requirements and hence cost. Note that resolution requirements and their study will be concretely assessed further on, in application cases. One of the most important aspects as far as the cost of LES is concerned, is the fact not being dependent on the Reynolds number. This is particularly interesting for the field here considered, turbulent flow simulation. In addition, this is one of the main factor of distinction between LES and DNS simulations, since DNS cost increases importantly with the Reynolds number. This is one of the many reason why the LES method is preferred over DNS in the kind of simulations here considered. Note that the main reason one could think to justify the preference of a DNS over a LES is precision. However, specially when using well-resolved LES, the small increment of precision is absorbed by the high raise in cost, which makes one lean towards the LES method, at least in the applications here considered. Although, LES is preferable in terms of cost over a DNS, it is not the most cost-efficient method available. In fact, the cost of LES belongs to the high cost section in the general spectrum of the available methodologies. For instance, a Reynolds-Averaged Navier–Stokes (RANS) model would in the vast majority of cases have a lower cost than a LES. However, cost is always analysed as a counter-factor to precision, accuracy and applicability, which are significantly better in a LES. The discrimination of one method over will be fundamentally depending on the application considered.

Applicability The range of applicability of the LES is wide comparing with other methodologies, specially for constant density flows over which this method is generally applicable. However, the cost will be a key factor in the applicability of the method. In practice, the cost and complexity of the problem to solve will determine the applicability of the method. In other words depending on the level of complexity and the cost of the methods considered, one will discriminate a method over another. In addition, the degree of description provided it is also a fundamental factor that determines the applicability of the method. It is clear that more rudimentary methods are typically easier to apply and more cost-efficient. However, the level of description is such methods usually limits their applicability as they, simply can not describe an important set of problems. For instance, the range of applicability of RANS methods is way more constrained than the range of LES method (Pope 2010 [25]) even if LES methods are typically more costly. This is due to the level of description of LES methods which allows to apply it to more complex problems such as aeroacoustic problems, unsteady turbulent problems etc. In addition, combined with statistical modelling in the subgrid scales, LES are applicable even more complex problems such as molecular reaction in flows or molecular mixing.

Accuracy As presented in the begging of §2.2.2.2, accuracy is one of the most important aspects when evaluating a given computational method. Appraising accuracy requires a proper comparison of the method's performance over a wide range of flows. For simple boundary layer flows the available literature appears to present that LES are accurate enough to be applied in such endeavours. However, Pope [25] suggest that the range of tested flows is not large enough to state that for every kind of simple boundary layer flow the method would be accurate enough. Note that even though the accuracy of general LES methods still needs to be properly evaluated, in the following section an enhanced version of LES will be presented and discussed, the *well-resolved* LES. In fact, *well-resolved* LES provide an accuracy close to DNS while keeping the cost contained.

2.2.2.3 Enhanced LES : Well-resolved LES

Well-resolved boundary layer simulations is a term to describe a LES which have a particularly constrained resolution conditions and hence accuracy. In fact, by setting a fine resolution over the scales one can significantly improve the accuracy of the method to achieve a performance close to DNS methods. Schlatter et al. [29] develop a method based on two modifications of the approximate deconvolution model (ADM) approach and they compare it with the results obtained from a fully resolved DNS. The two modification introduced are ADM-2D and the RT-3D, which, as stated are modifications of the standard ADM, known as ADM-3D. The ADM-2D is based on a two dimensional filtering and decomposition over the homogenous wall-parallel directions instead of the three dimensional that is proposed in the standard method. In addition, both variations include a modified version of the non-dimensional Navier–Stokes equations which include a relation term $-\chi(I - Q_N G)\overline{u}_i$. The relaxation parameter χ is defined by Schlatter et al. (2004)[29] as,

$$\chi(t + \Delta t) = \chi(t) \frac{F_2(t + \Delta t)|_{\chi=0} - F_2(t)}{F_2(t + \Delta t)|_{\chi=0} - F_2(t + \Delta t)|_{\chi=\chi(t)}} \tag{2.15}$$

where F_2 is the second-order velocity structure function and $\overline{u}_i$ is the filtered velocity.
The rest of the magnitudes forming the relaxation term are given by the high-pass filtered also presented in the work of Schlatter et al. [29].

$$H_N = I - Q_N G \tag{2.16}$$

where, Q_N is the approximate inverse of the primary low-pass filter, G. Schlatter et al. [29] explain how the deconvolution term is maintained over the whole process in both the transient and turbulent phases. In this way, “the advantages of the ADM techniques are retained, although this model is not as general as the original formulation since it is restricted to filtering in two dimensions only” (Schlatter et al. [29]) . For the RT-3D modification the relaxation term is evaluated in three dimensions. However, the main difference with respect to other methodologies lies in the evaluation of the non-linear terms. In fact, those terms are computed from the filtered quantities. The main difference between the RT approach and the ADM approach, as stated by Schlatter et al. [29] is that the first does not use the deconvolved quantities for the non-linear terms. However, the RT-3D modification has proven to be “as general as the standard ADM procedure” (Schlatter et al. [29]) even if the afore-mentioned deconvolved quantities are not applied over the non-linearities. The study presented by Schlatter et al. [29] concludes that the methods here treated converge in terms of resolution to the results provided by DNS methods. In particular, their study demonstrate that with a proper treatment of the spatial directions, one can “faithfully represent the relevant physical features of the flow” (Schlatter et al. [29]) . This fact supports the simulation choice of the work here presented to select a well-resolved LES over a DNS.

Negi et al. [21] get further in the study of the performance of LES, this time focusing on the afore-presented RT3D variant. In the work of Schlatter et al. [?], it is proved that the RTD3 method is reliable in predicting both the location and flow structures found in transitional flows solved in DNS. In the work here referred, Negi et al. [21] consider both the transition and the fully turbulent phase over an airfoil. However, it is the LES appraisal and comparison that is interesting in the work here presented. In their study, Negi et al. [21] use a much finer resolution than the standard LES resolution which is actually “close to the coarse DNS resolutions used in turbulent flows” (Negi et al. [21]). The results of the study show that a “good agreement with the DNS is found for the mean velocity and all the kinetic energy budget terms (including the total dissipation)” (Negi et al. [21]). Once again the resutls of the study here addressed are consistent with the selection of well-resolved LES method over a DNS method since the require performance is achieved and the costs are constrained, as presented in § 2.2.2.2.

2.2.3 Numerical Method : Nek5000

To close the purely theoretical description of the present work, the numerical tools will be assessed. The main objective in the current section is to describe the foundations of the numerical tools that will be applied further on. Note, that the actual programming process will be addressed later on on this report since the purpose of the current description is to present the theory behind the tool and not the practical functioning.

Recalling §2.1.1, by this point it is know that in addition to the simulation type and the equations that are considered, the selection of the numerical method to run the simulation is a key part of the process to solve any computational fluid dynamics problem. The computational code applied to the application problems here studied, will be Nek5000.

2.2.3.1 General Aspects

Nek5000 is a open-source Navier–Stokes solver that developed by Fisher among many others at the Argonne National Laboratory in the United States of America. Nek5000 was designed to solve incompressible laminar, transient and turbulent problems. In addition, it is also possible heat transfer and species transport problems at low Reynolds number. From the computing point of view, typically, Nek5000 using a parallel computing approach by means of the MPI protocol. Although it is possible to use the tool in a serial mode, i.e. without benefiting from the MPI protocol, during the applications presented further on, it will be used in parallel computing mode such that the problem can be efficiently solved. The source code is coded in f77 such that it can be compiled with any Fortran compiler available. Note, that at this point no more details on the practicalities of the code will be presented since the object of this exposition is to present the theoretical foundations of the method. Please refer to § 3.1.2 for a description on the practical workflow of the tool.

Numerically, the tool is based on a spectral elements method allowing to map complex geometries by means of a high-order spatial discretisation of the domain achieved with the Galerkin approximation. As far as the distribution of points within the elements is concerned, Nek5000 uses the Gauss-Lobatto-Legendre quadrature, also known as the GLL quadrature. In addition, the version that is here considered, is formulated in algebraic type, i.e. $\mathbb{P}_N - \mathbb{P}_{N-2}$. This means that the velocity is expressed with a polynomial of order N whereas the pressure is formulated with a polynomial of order $N - 2$. Further on, the numerical and mathematical concepts here introduced will be properly developed.

2.2.3.2 Navier–Stokes discretisation: Spectral-element method

As introduced in §2.2.1 a key part of the preprocessing stage consist on the discretization of the domain. In the following lines we will focus on describing the **spectral-element method**, since it is the foundation of one of the computational tool applied in this work. Note, that this part will be focused on the spectral discretisation of the Navier–Stokes equations presented in §2.1 following the method presented by Patera in *A spectral element method for fluid dynamics: Laminar flow in a channel expansion* [24]. The method presented consist on a combination of Finite Elements Method (FEM) and Spectral Methods which aims to exploit the advantages that both methods have to offer. On the one hand, FEM as described in Patera [24], presents the generality advantage, allowing their application to many different kinds of problems. In fact, finite-element methods are used in a wide range of engineering disciplines such as structural analysis, heat transfer problems etc. On the other hand, spectral elements have prove to be more accurate. A combination of both, allows to have an accurate tool without excessively narrowing the applicability range of the method.

Although finite-element and spectral-element methods are typically considered as independent, both are based on the same fundamental procedure. As described in Patera [24], the finite-element procedure and therefore the spectral procedure is based on the division of space in small portions, the elements. In each of the elements a series of expansions are applied following a weighted-residual technique. The main advantage of the finite-element methods is found in their generality. In fact, by choosing quadratic elements, one can apply this methodology to all sorts of problems, which is obviously a great advantage. However, this generality is balanced with a significant difficulty to produce an accurate solution, hence the development of spectral methods. Spectral methods, do not provide such generality, but they do have a much better accuracy.

Let the definition of the spaces be the starting point for the development of the spectral elements method (SEM) formulation. The spaces are defined as,

$$L^2(\Omega) = \left\{ f : \Omega \to \mathbb{R} | \sqrt{\int_\Omega |f|^2 d\Omega} < \infty \right\} \tag{2.17}$$

In addition, the following subset is also defined,

$$H_0^1(\Omega) = \left\{ f \in L^2(\Omega) | \frac{\partial f}{\partial \mathbf{x}} \in L^2(\Omega) \right\} \tag{2.18}$$

Moreover, an additional space is introduced for the pressure.

$$L_0^2(\Omega) = \left\{ f \in L^2(\Omega) | \int_\Omega f d\Omega = 0 \right\} \tag{2.19}$$

Time discretisation As stated in the introductory sections of computational theory the numerical methods are based on the discretisation of space and time. In this way, the methodology consists on transforming continuous magnitudes into discrete magnitudes by means of the discretisation operation.

Let us consider a general time-dependent problem in the following form:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathcal{L}[\mathbf{u}]; \qquad \mathbf{u}(t_0, \mathbf{x}) = \mathbf{u_0} \tag{2.20}$$

Note that in Equation 2.20 the space magnitudes and operators are supposed to be gathered in the operator $\mathcal{L}$.

For time variable a backward finite differentiation scheme is applied. Mathematically,

$$\sum_{j=0}^{k} b_j \mathbf{u}^{n+1-j} = \Delta t \mathcal{L}[\mathbf{u}^{n+1}] \tag{2.21}$$

It is easy to see that the backward scheme presented in Equation 2.21, is an implicit scheme. Implicit schemes have proven to be computationally demanding processes (Fisher et al. [8]) and hence it is preferable to find a alternate form such that the implicit form is avoided. By operating,

$$\mathcal{L}[\mathbf{u}^{n+1}] = \sum_{j=0}^{k} a_j \mathcal{L}[\mathbf{u}^{n+1-j}] \tag{2.22}$$

Equation 2.22 is the final expression of the scheme that is introduced in the coding of Nek5000. Further on, the specific routines that assess this expression will be presented.

Space discretisation Now, as far as the space discretisation scheme is concerned, we will focus on the $\mathbb{P}_N - \mathbb{P}_N$ discretisation since this is the general representation of the scheme. Before getting into the details of the method, let us reformulate the Navier–Stokes equations into a more convenient expression. Recall Equation 2.5 as well as the well-known definition of the substantial derivative. In addition, consider the following parameter,

$$\tau = \mu \left[\nabla \left(\mathbf{u} \right) + \nabla \left(\mathbf{u} \right)^T - \frac{2}{3} \nabla \cdot \mathbf{u} \mathbf{I} \right] \tag{2.23}$$

By combining both Equation 2.5 and Equation 2.23, the following expressions for the Navier–Stokes equations are found :

$$\begin{gathered} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \left(\mathbf{u} \right) \right) = -\nabla \left(p \right) + \nabla \cdot \tau + \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{gathered} \tag{2.24}$$

Note, that the Navier–Stokes equations can be considered as a set of non-linear advection-diffusion equations constrained with pressure. This particularity allows to adapt some of the pre-existent literature on the advection-diffusion equation.

Recall Equation 2.20. Applying that expression to the Navier–Stokes equations presented in Equation 2.24, the following expressions are obtained:

$$\begin{aligned} \rho \left(\mathcal{L}[\mathbf{u}] + \mathbf{u} \cdot \nabla \left(\mathbf{u} \right) \right) = -\nabla \left(p \right) + \nabla \cdot \tau + \mathbf{f} \\ \Leftrightarrow \mathcal{L}[\mathbf{u}] = -\mathbf{u} \cdot \nabla \left(\mathbf{u} \right) - \frac{1}{\rho} \nabla \left(p \right) + \nabla \cdot \tau + \mathbf{f} \end{aligned} \tag{2.25}$$

Using the nomenclature presented by Fisher et al. [8], some terms are can be gathered in terms of non-linear terms, $N(\mathbf{u})$ and viscous terms, $L(\mathbf{u})$.

- $N(\mathbf{u}) = \rho \mathbf{u} \cdot \nabla \left(\mathbf{u} \right)$
- $L(\mathbf{u}) = \nabla \cdot \tau = \nabla \cdot \mu \left[\nabla \left(\mathbf{u} \right) + \nabla \left(\mathbf{u} \right)^T - \frac{2}{3} \nabla \cdot \mathbf{u} \mathbf{I} \right]$

Hence, the first expression shown in Equation 2.24, i.e. the momentum equation in differential form can be expressed as,

$$\rho \frac{\partial \mathbf{u}}{\partial t} = -\nabla \left(p \right) + N(\mathbf{u}) + L(\mathbf{u}) + \mathbf{f} \tag{2.26}$$

Fisher et al. [8] remark that for stability reasons, the non-linear terms need to be solved by means of an explicit method whereas the implicit are solved using implicit ones.

Time integration Now that both space and time discretisations have been formulated, the next step is to develop the integration schemes. Once again, the formulation presented by Fisher et al. [8] is followed, since it is the foundational base of Nek5000.
Applying the time discretisation scheme proposed in Equation 2.21 to the terms forming Equation 2.26, it is found:

$$\sum_{j=0}^{k} \frac{b_j}{\Delta t} \mathbf{u}^{n+1-j} = -\nabla \left(p^{n+1} \right) + N(\mathbf{u}^{n+1}) + L(\mathbf{u}^{n+1}) + \mathbf{f}^{n+1} \tag{2.27}$$

Due to the nature of the non-linear terms, this part of the equation needs to be extrapolated. Fisher et al. [8] propose:

$$N(\mathbf{u}^{n+1}) = \sum_{j=1}^{k} a_j N(\mathbf{u}^{n+1-j}) \tag{2.28}$$

Now, coming back to Equation 2.27, it is easy to see that the right-hand terms can be split into:

$$\frac{b_j}{\Delta t}\mathbf{u}^{n+1-j}\bigg|_{j=0} + \sum_{j=1}^{k} \frac{b_j}{\Delta t}\mathbf{u}^{n+1-j} = \frac{b_0}{\Delta t}\mathbf{u}^{n+1} + \sum_{j=1}^{k} \frac{b_j}{\Delta t}\mathbf{u}^{n+1-j} \tag{2.29}$$

Replacing the afore-mentioned terms in Equation 2.27 while sorting explicit and implicit terms, the following expression is found:

$$\frac{b_0}{\Delta t}\mathbf{u}^{n+1} = -\nabla\left(p^{n+1}\right) + L(\mathbf{u}^{n+1}) - \sum_{j=1}^{k} \frac{b_j}{\Delta t}\mathbf{u}^{n+1-j} + \sum_{j=1}^{k} a_j N(\mathbf{u}^{n+1-j}) + \mathbf{f}^{n+1} \tag{2.30}$$

By inspection, one can see that the three last terms in the right-hand side of Equation 2.30 are explicit and depend on the previous velocity step. Note that the forcing term $\mathbf{f}$ also exhibits this dependency although it is not explicitly written. Hence, following Fisher et al. [8] procedure, those terms can be gathered in the unique term $\mathbf{F}(\mathbf{u}^n)$. In this way, Equation 2.30 is rewritten as follows:

$$\frac{b_0}{\Delta t}\mathbf{u}^{n+1} = -\nabla\left(p^{n+1}\right) + L(\mathbf{u}^{n+1}) + \mathbf{F}(\mathbf{u}^n) \tag{2.31}$$

By direct application of the vector identity, the viscous term can be reformulated as follows:

$$L(\mathbf{u}^{n+1}) = \mu\frac{4}{3}\nabla\left(\nabla\cdot\mathbf{u}^{n+1}\right) - \mu\sum_{j=1}^{k} a_j \nabla\times\left(\nabla\times\mathbf{u}^{n+1-j}\right) \tag{2.32}$$

Now, let us define the another term grouping the second term in the left-hand side of the equality presented in Equation 2.32 with the afore-defined $\mathbf{F}(\mathbf{u}^n)$ term.

$$\tilde{\mathbf{F}}(\mathbf{u}^n) = \mathbf{F}(\mathbf{u}^n) + \mu\sum_{j=1}^{k} a_j \nabla\times\left(\nabla\times\mathbf{u}^{n+1-j}\right) \tag{2.33}$$

Hence, Equation 2.30 can be expressed solving the pressure term by means of the term shown in Equation 2.33,

$$\nabla\left(p^{n+1}\right) = -\frac{b_0}{\Delta t}\mathbf{u}^{n+1} - \mu\frac{4}{3}\nabla\left(\nabla\cdot\mathbf{u}^{n+1}\right) + \tilde{\mathbf{F}}(\mathbf{u}^n) \tag{2.34}$$

Equation 2.34 can be recognised to be very close the the pressure Laplace equation. In fact, the Laplace equation is known to be the divergence of the terms presented in Equation 2.34 (Fisher et al. [8]). For a matter of convenience, let us consider the following parameter Q_T such as $Q_T = \nabla(u)$. (Fisher et al. [8]). By taking the divergence of Equation 2.34 one can obtain the pressure Laplace equation in the following form.

$$\nabla\cdot\nabla\left(p^{n+1}\right) = -\frac{b_0}{\Delta t}Q_T^{n+1} - \nabla\cdot\mu\frac{4}{3}\nabla\left(Q^{n+1}\right) + \nabla\cdot\tilde{\mathbf{F}}(\mathbf{u}^n) \tag{2.35}$$

By integration over the space Ω and considering that for incompressible flows $Q_T = 0$ it is found:

$$\oint\nabla\left(p^{n+1}\right)\cdot\mathbf{n}\cdot q dS = \oint\mu\nabla\left(Q_T^{n+1}\right)\cdot\mathbf{n}\cdot q dS + \oint\tilde{\mathbf{F}}(\mathbf{u}^n)\cdot\mathbf{n}\cdot q dS \tag{2.36}$$

where $\mathbf{n}dS = d\Omega$.
At this stage the boundary conditions for the pressure can be obtained by taking the normal the pressure term solved in Equation 2.34. Expressed in terms of $L(\mathbf{u}^{n+1})$ and $F(\mathbf{u}^{n+1})$, this is :

$$\nabla\left(p^{n+1}\right)\cdot\mathbf{n} = -\frac{b_0}{\Delta t}\mathbf{u}^{n+1}\cdot\mathbf{n} + L(\mathbf{u}^{n+1})\cdot\mathbf{n} + \mathbf{F}(\mathbf{u}^{n+1})\cdot\mathbf{n} \tag{2.37}$$

Taking into account the partial separation of the Laplace terms that is in fact incorporated inside $\tilde{\mathbf{F}}(\mathbf{u}^n)$ (Fisher et al. [8]) one can find that,

$$\nabla\left(p^{n+1}\right)\cdot\mathbf{n} = -\frac{b_0}{\Delta t}\mathbf{u}^{n+1} \tag{2.38}$$

Finally, using the expression of the computed pressure and recalling Equation 2.35, the well-know Helmholtz equation which is the one that will be incorporated in the tool to be solved.

$$\frac{b_0}{\Delta t}\mathbf{u}^{n+1} - \mu\Delta\mathbf{u}^{n+1} = -\Delta p^{n+1} + \frac{\mu}{3}\nabla\left(Q^{n+1}\right) - \mathbf{F}(\mathbf{u}^n) \tag{2.39}$$

2.2.3.3 Representation of the magnitudes within the elements

Note that § 2.2.3.2 was dedicated to the understanding of the Navier–Stokes discretisation using the spectral-element method. At that point, the main objective was to divide a continuous domain into a series of elements. Now, within those elements the pressure and velocity are expressed by means of high-order Lagrange interpolants of Legendre polynomials over a specific quadrature of points the Gauss–Lobatto–Legendre quadrature. The following paragraphs will be dedicated to the introduction of those mathematical tools.

Legendre polynomials The Legendre polynomials are used as base functions in order to represent the magnitudes in question. Following the approach exposed by Arfken and Weber [11], the function formulation will be introduced by means of a generating function.
Consider the following generating function formula [11],

$$g(t,x) = \frac{1}{\sqrt{1-2xt+t^2}} = \sum_{n=0}^{\infty} P_n(x)t^n \quad \text{with} \ \ |t| < 1 \tag{2.40}$$

Arfken and Weber [11] prove that:

$$|Pn(\cos{(\theta)}) \leq 1| \tag{2.41}$$

where θ is the angular coordinate in the polar reference system.
From Equation 2.41 it is easy to see deduce that the series expansion presented in Equation 2.40 is convergent $\forall\,|t| < 1$. This statement is in fact very convenient since in Equation 2.40 the generating function was precisely defined for $|t| < 1$, hence the series are convergent for the whole range of t values considered.
From the binomial theorem it is known that the middle term of the equality shown in Equation 2.40 can be expanded by means of the following infinite series,

$$\frac{1}{\sqrt{1-2xt+t^2}} = \sum_{n=0}^{\infty}\frac{(2n)!}{2^{2n}(n!)^2}(2xt-t^2)^n \tag{2.42}$$

In addition, the term $(2xt-t^2)^n$ can be written as a series by means of a double series expansion. In this way, Equation 2.42 can be rewritten as,

$$\frac{1}{\sqrt{1-2xt+t^2}} = \sum_{n=0}^{\infty}\sum_{k=0}^{[n/2]}(-1)^k\frac{(2n-2k)!}{2^{2n-2k}k!(n-k)!(n-2k)!}(2x)^{n-2k}t^n \tag{2.43}$$

Finally by recalling the second equality presented in Equation 2.40, the Legendre polynomials are found,

$$P_n(x) = \sum_{k=0}^{[n/2]} (-1)^k \frac{(2n-k)!}{2^n k!(n-k)!(n-2k)!} x^{n-2k} \tag{2.44}$$

Note, that there exist alternate forms of Equation 2.44 which present different advantages depending on their application. For instance one could consider Rodriguez's formula or the Schlaefli Integral forms (Arfken and Weber [11]). Although for the purpose of the section here presented the considered form should be more than enough for the reader to have an idea on the theoretical foundations of the method, for a matter of convenience, an alternate form will be introduced. The generating function $g(t,x)$ from which the Legendre polynomials were derived can be generalised to obtain the **ultraspherical polynomials**. In this case, the following equality gives the polynomials:

$$\frac{1}{(1-2xt+t^2)^\alpha} = \sum_{n=0}^{\infty} C_n^{(\alpha)}(x) t^n, \tag{2.45}$$

where $C_n^{(\alpha)}(x)$ are the ultrapherical polynomials. Note, that those polynomials will be useful during the derivation of the Gauss-Lobatto-Legendre quadrature expressions

Gauss-Lobatto-Legendre quadrature In the previous section we introduced the Legendre polynomials which, as stated, serve as base functions in the representation of magnitudes within the elements. From the grid-point's perspective, a distribution scheme needs to be developed. Nek5000 utilises the Gauss-Lobatto-Legendre quadrature in order to distribute the points within the elements.

The Gauss-Lobatto-Legendre (GLL) quadrature could be understood as some kind of evolution from the classical Gauss quadrature. In fact , the main distinction between the GLL quadrature and the Gaussian one is that the former considers the extreme points in the integration interval whereas the later does not[26]).

Now, following the procedure presented by Brix et al. [3], consider two parameters k and N such that $0 \leq k \leq N$. Consider also a given GLL node defined as ε_k^N of order N. Then, by definition the GLL nodes of order N are the $N+1$ zeros of the following polynomial expression:

$$(1-x^2)L'_N(x) \tag{2.46}$$

where $L'_n(x)$ is the derivative of the N^{th} Legendre polynomial as formulated in Equation 2.44.

By differentiation over the ultraspherical polynomials (Equation 2.45), the derivative of the Legendre polynomials can be obtained:

$$L'_N(x) = \frac{1}{2} P_{N-1}^{(\frac{3}{2})}(x) \tag{2.47}$$

The GLL nodes are distributed in ascending order, i.e. $\varepsilon_k^N < \varepsilon_{k+1}^N$ forming a grid of order N. In addition, the nodes are symmetrically distributed with respect to the origin, i.e. $\varepsilon_k^N = -\varepsilon_k^N$ (Brix et al. [3]).

3

Problem setup and Implementation

This chapter will be focusing on the practicalities of the flow simulations. In fact, this chapter will be complementing the comments gathered in the previous chapter, now focusing on the implementation of the set of tools applied in this work. Note that in this case the theoretical derivations and explanations will be kept minimal since the main objective of this part is to provide a full guide on the process of designing and running a high-order spectral simulation using the *well-resolved* LES methodology implemented in Nek5000.

The structure of the exposition here presented will be based on the well-known scheme §2.2.1 which is actually the general approach to any physical simulations. More precisely, the description will be start with the routines in both Nek5000 and MATLAB necessary to setup the case, then the characteristics of the solving process and finally, the postprocessing tools needed during the different stages of the analysis.

3.1 Preprocessing tools and routines

In § 2.2.1 we described how any flow simulation starts through a series of implementations that built the different inputs of the solution tool. In addition to the solver's inputs, some specific tools need to be developed in order to produce an *a priori* analysis to determine certain parameters that are key to the well-solving of the problems. Such tools and analysis will also covered as a part of the preprocessing stages.

The exposition here presented will be built over three fundamental axes, which describe the thinking and design processes followed in the application problems. First of all, the mesh has to be generated. Secondly, the mesh files, among others are input into the Nek5000 solver, such that the resolved fields and statistics are obtained. Finally, the output files from Nek5000 are combined with a set of tools that produces the quantities of interest. The afore-stated structure is the fundamental procedure, in general terms, that will be applied in each of the cases presented further on. The next lines will be dedicated to provide the reader with a good understanding on how those processes are designed and carried out.

3.1.1 Mesh generation

The generation of the mesh is actually a topic on its own, for which several tools and studies are available. However, the purpose of the work here presented is not the study of meshes but rather the study of flows in urban environments. Hence, the mesh generation processes is relies on a platform of cloud parallel computing which allows, among many other things, to generate the mesh in Nek5000 format. The platform was developed in collaboration with Parallel Works Inc. [23]. It is based on the Swift parallel computing scripting language as well as the Parsl parallel scripting library. However, the user's experience is a semi-programming experience. The routines to compile and run the case are built-in inside the system such that the user changes a series of parameters which define the setup of the workflow. Nevertheless, the platform still allows the user to see the files and check the programming errors. In this way, the system is oriented to both users and developers. The advantage of this platform lies on the fact that it allows to automate complex procedures such that given a series of parameters the system directly runs the routines and outputs the results.

3.1.1.1 Workflow setup and processes

The main workflow that is used in the present study is a urban environment which generates the mesh, among many other input files, of a domain with a given number of building-like obstacles. The parameters that are input in the workflow can be gathered mainly in two groups, geometrical and mesh parameters. To those two groups, the simulation parameters are added. Those parameters are actually influencing the simulation and with exception of the boundary conditions will not be affecting the mesh and will be redefined in the Nek5000 setup run locally. In fact, the Parallel Works platform is prepared to run some small cases. However, due to the size and nature of the problems here considered, the simulations are in a specific Nek5000 setup outside the platform. Hence, Parallel Works is only applied in the mesh-generation stage.

Geometrical parameters This set of parameters defines both the type and dimensions of the domain. In fact, the geometry is generated from the centre of the first obstacle. The origin of the coordinate system is centred in the intersection of the first obstacle's centreline with bottom horizontal plane. Then the different obstacles are located with respect to the origin following the structure shown in Figure 3.1.

Note that the dimensions shown in Figure 3.1 are scaled as a function of the height of the building-like obstacle h. This allows a fast rescaling of the cases by changing a single parameter. Moreover, number of obstacles and their disposition is determined following a matrix approach by means of the parameters I and J. The I quantity determines the number of rows while J determines the number of columns. For instance for $\{I, J\} = \{2, 3\}$ the resulting geometry is a matrix composed of six obstacles disposed as shown in Figure 3.1. As far as the workflow's limitations are concerned, theoretically the workflow is prepare to handle all kinds of cases. However, limitations might appear in the computational resources. In the work here presented, the platform is used a "mesh generator", hence the limitations of the computational resources do not represent a significant limitation since the mesh generation process does not result in an extraordinary computational cost.

Mesh parameters and generation process Along with the geometrical and dimensional parameters the second fundamental part of this process involve the definition of the sizes of the elements that form the mesh. Recall that the main idea of "elements" methods such as FEM or SEM is to discretise the space in a certain number of elements. The computational object that gathers those elements is the mesh. Generally, for non-uniform elements three size parameters are enough to determine the size of the element, one for each direction. However, the mesh here considered follows a refinement scheme such that the element size is reduced as one approaches the walls of the obstacles. Refinement is a very common approach in those kind of simulations as it allows to have a greater resolution at the key areas without having to reduce the size of the elements in the whole domain. Note, that reducing the size of the elements increases resolution but

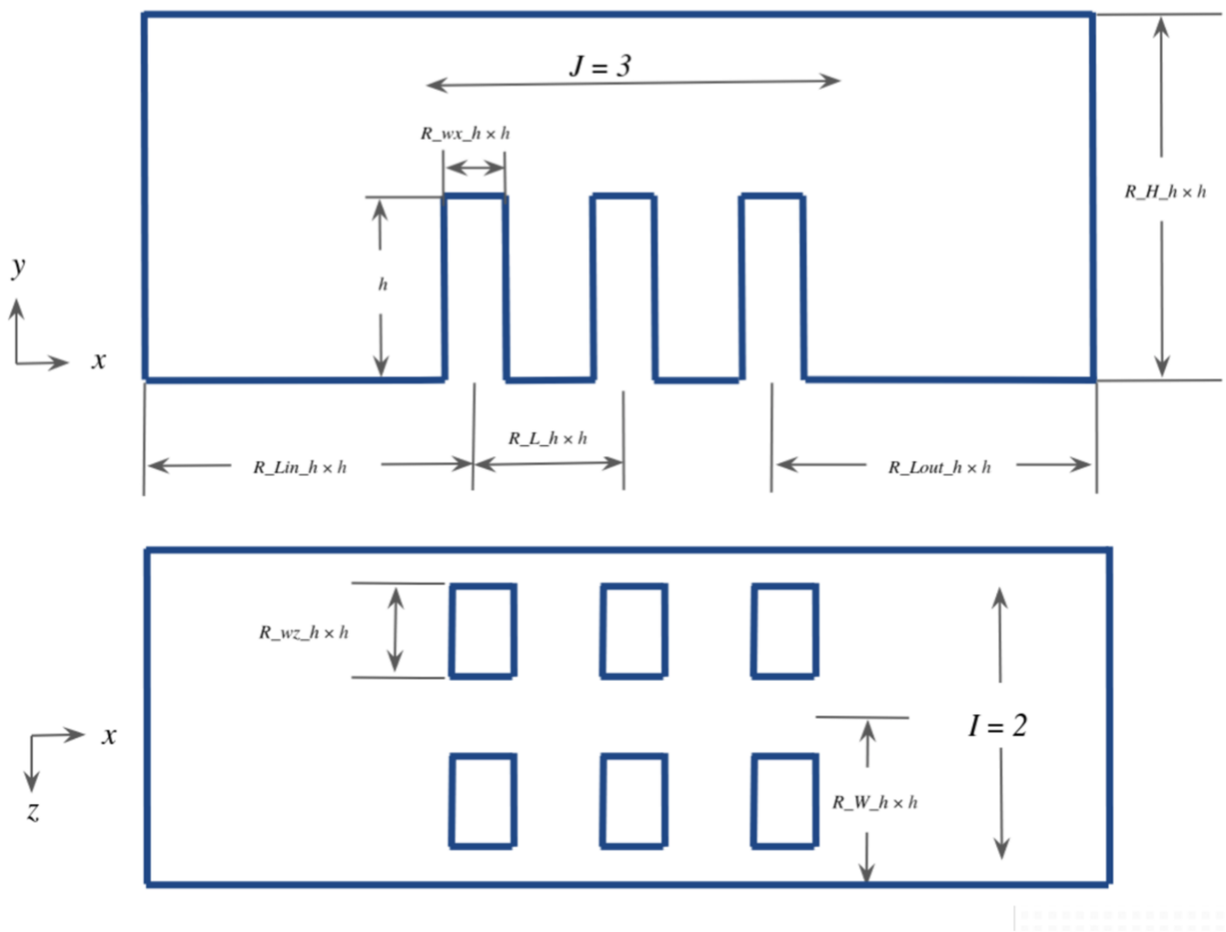


Figure 3.1: Geometrical parameters defining the mesh

also cost, as for a given set of dimensions the number of elements will increase. In the workflow here considered the mesh is progressive in the wall-normal direction to the obstacles. In fact, in the regions near the obstacles the size element will expand in the normal direction from the minimal value at the wall $dmin$ to the maximal value defined in near-obstacle zone, $domax$ at a defined expansion rate r. Note, that this is only applied in the near-obstacle areas, the rest of the domain has a fixed element size that is determined by three parameters $\{dxmax, dymax, dzmax\}$. Figure 3.2 shows a two dimensional cut of the z-plane for an example mesh. In the figure one can see how the mesh is progressive in the direction normal to the obstacle.
Note, that the mesh is not only refined in the near-obstacle areas but also in the bottom y-plane. In fact, it is important to have a fine mesh in that area specially at the surroundings of the inflow since the boundary layer is solved in that area. Later on, we will see how solving the boundary layer properly is a critical stage in the well functioning of a turbulent flow simulation.

The parameters defining the mesh in the platform are gathered in Table 3.1.

From the processing point of view, the platform starts by generating the mesh using the *gmsh*tool, then the *gmsh* data is translated to the Nek5000 format. *gmsh* is a "three dimensional finite element mesh generator with build-in CAD engine and post-processor"(Geuzaine and Remacle [12]). This tool provides a systematic and simple way to generate three dimensional meshes. Although the tool provides a full development environment, in this case the generation tool is integrated with the meshing platform. The main advantage of *gmsh* over other mesh gen-

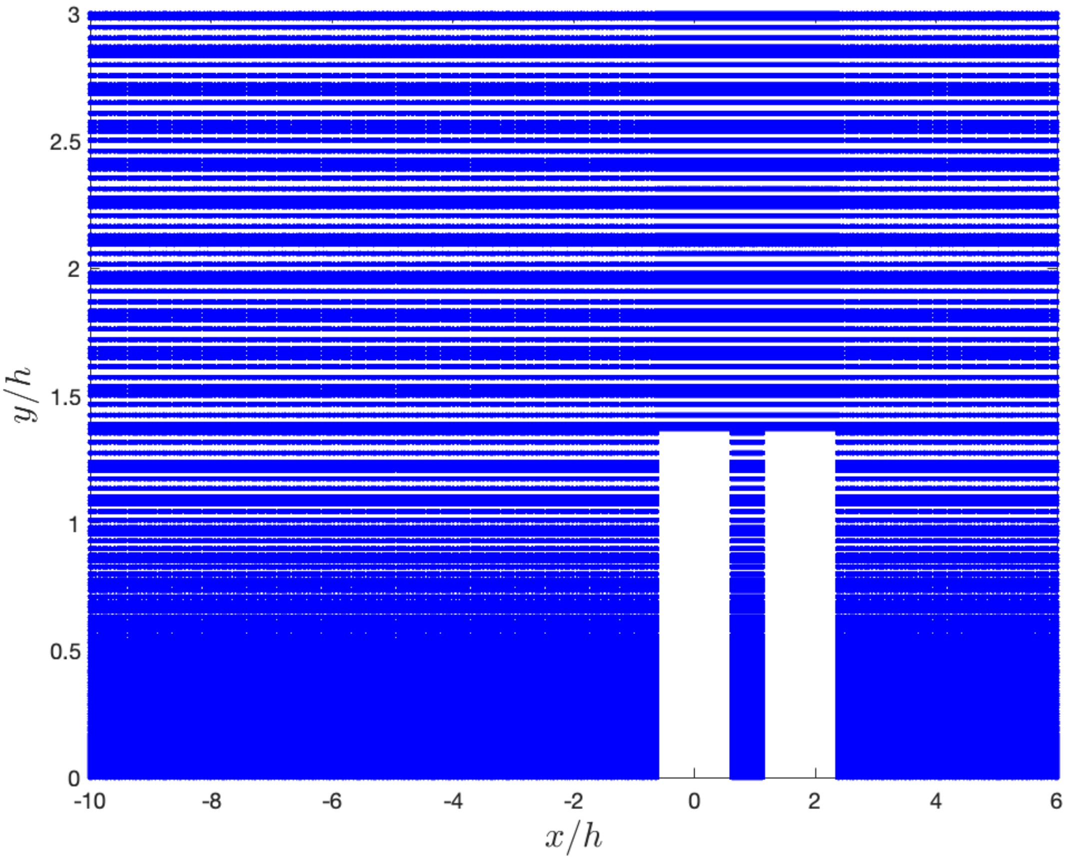


Figure 3.2: wo-dimensional cut at plane $z/h = -0.5$ for the final simulation mesh

Parameter	**Description**
dxmax	Maximum element size in the x - direction
dymax	Maximum element size in the y - direction
dzmax	Maximum element size in the z - direction
dmin	Minimum element size in the wall-normal directions
domax	Maximum element size in the near-obstacle region
r	Expansion rate of the elements in the wall-normal directions

Table 3.1: Mesh parameters

erators that directly implement Nek5000 format is the versatility of the coding process which is significantly more flexible in the *gmsh* tool. In fact, an alternative meshing tool was actually tried outside the platform to check whether a Nek5000 tool was more convenient than using the meshing platform. The Nek5000 built-in *genbox* tool was tried. This tools allows to mesh the domain by means of different "boxes" containing the discretised elements. Although this tool could be directly implemented in the Nek5000 workflow without having to use alternative systems, the design and processing of the mesh is actually more complex having to deal with non-compatibilities and boundary condition mismatches quite frequently. That is why the *gmsh* tool within the meshing platform was selected. Once the *gmsh* file (*.geo*) is generated, a Nek5000 routine is implemented

such that Nek5000 files are produced. The function that enables to transform the *.geo* file into a Nek5000 format file (*.re*2) is the *gmsh2re2* routine.
Now, for a simple case, the afore-stated routines and files are sufficient to define the mesh. However, in bigger cases, such as the ones that will be considered further on, it might be convenient to run the simulation using a parallel computing tool such as MPI. In such cases, an additional file is needed to set how the elements will be splitting over the different processors. The *genmap* file runs a recursive spectral bisection to determine the elements' split. This tool is also implemented in the meshing platform although in the bigger cases it appear to have memory problems. Hence in the application cases, the *genmap* function is actually run locally. The function in question produced a *.ma*2 file which is actually the mesh file required in addition to the *.re*2 file when running Nek5000 in parallel mode. Figure 3.3 serves as a visual synthesis of the afore-explained processes.

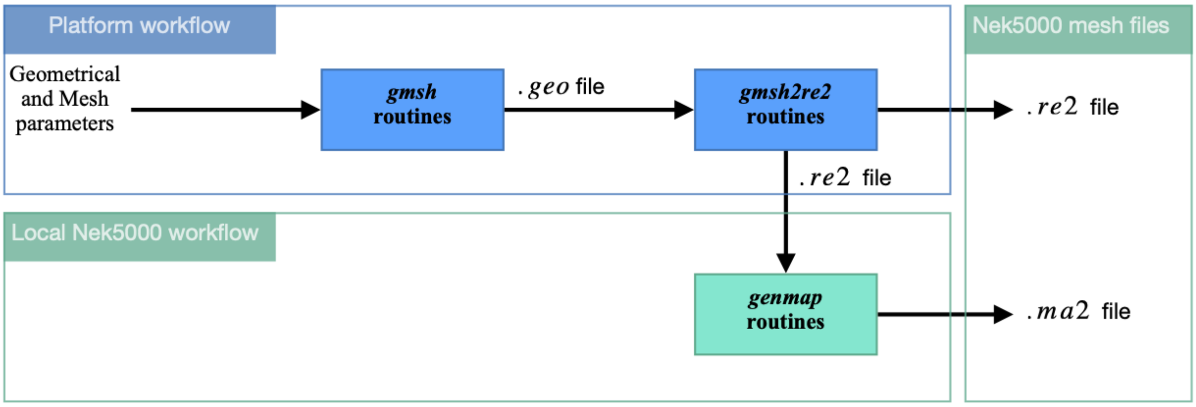


Figure 3.3: Meshing workflow using both the platform and the Nek5000 tools

3.1.1.2 Computational resources

To conclude with the presentation on mesh generation, the last topic to cover is related with the computational resources that are involved in the mesh generation. Before getting to the insights of the computational resources available in the meshing platform let us make a note on the concept of parallel computing. Both the meshing platform and Nek5000 run following a parallel computing approach.

Parallel Computing Parallel computing, is type of computation strategy in which several processes are executed simultaneously. Traditionally, all computers worked in serial processing. Under this strategy, the Central Processing Unit or CPU processes a single computation or instruction at a time. Although, from Moore's law it's known that the computing power of humankind doubles every few years, there exist a vast domain of problems big enough to be able to solve it using conventional serial methods. That is why more efficient ways to solve big computation were developed. The tools here performed run under a particular type of parallel computing which involves the use of a multi-core processor. The main idea is to split the problem into smaller problems that are solved simultaneously using the multiple processors available. Figure 3.4 shows graphically the fundamental differences between the serial and parallel approach. Consider a given computational problem composed of I_N operations. Consider also the time variable t assumed to start at t_1. It is easy to see that for a given time unit the parallel approach can process more operations than the serial approach, hence solving the problem faster.

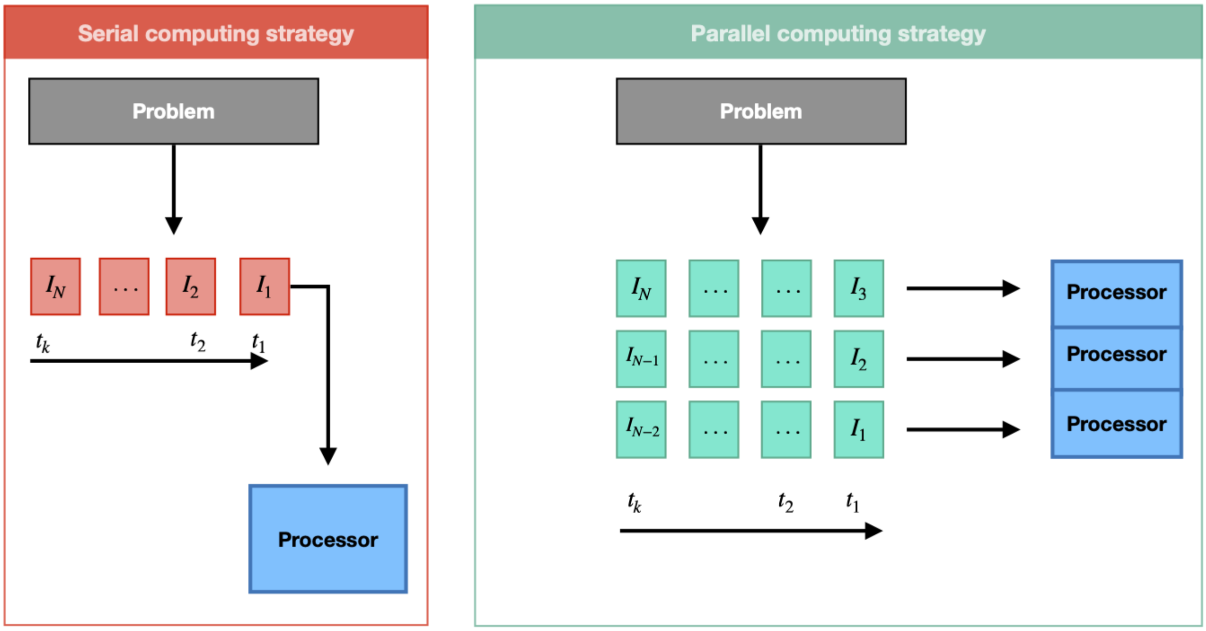


Figure 3.4: Serial computing vs Parallel computing

As far as the particular protocols are concerned, several choices are possible. In the case of both Nek5000 and the computational resources used in the Parallel Works platform the Message Passing Interface (MPI) is applied. MPI is a standard library to "pass information", i.e. distributing tasks and operations throughout the cores. The MPI protocol is compatible with a wide range of programming languages such as C and Fortran. To explain the inner working of MPI one should recall the way common computers treat information. A process inside a computer is typically a program counter and an address space. MPI ensures the right communication between processes by means of synchronisation and data transfer from a given address to another one. This interface works in a cooperation scheme, i.e. any data transfer involves two distinct processes, one sending the information and another one receiving it. Hence the synchronisation and transmission process run simultaneously. Note, that there exist another version of the interface, MPI-2 which is used in one-sided operations, i.e. involving just one process.

3.1.2 Nek5000 setup

This part will be focusing on the Nek5000 setup environment and in particular on the basic files that are required before launching any simulation. Nek5000 is a computational tool that was developed by Fisher et al. [8] at Argonne National Laboratory. The whole system is built on a combination of Fortran and C files and it can be directly run in the terminal only using a classic Fortran and C compiler. Hence, the tool is not particularly user-friendly and requires some expertise before working properly. However, it is precisely the absence of superfluous rendering that makes the tool so versatile and efficient that it is still preferred over commercial software in complex flow simulation.

3.1.2.1 Basic structure and files

The most basic structures to have into consideration are files, folders and paths. From the operability point of view, the are a series of rules that is convenient to follow.

- The fundamental routines that form the basic container of the Nek5000 solver are located in an independent container which is never edited.
- Each case should have a folder which locates the files that define the simulation and geometry of the specific case. The general folder is then divided into the *compile* and the *run* folders, as those operation are operated sequentially and require different files.

The afore-stated rules are simple guidelines to the well organisation of the simulations files. However, those are not requirements for the functioning of the tool. In addition, sometimes a specific toolbox needs to be applied and hence change the structure of the operation scheme.

Every Nek5000 simulation is defined by three fundamental files,

1. *.usr* file
2. *.rea* or *.par* file when using the newer versions of Nek
3. $SIZE$ file

In addition, if the system is working in parallel run a mapping file is needed such that the element's distribution throughout the cores is set. Please refer to §3.1.1.1 for a complete explanation on how the map generation routine works.
Moreover, some additional files such as the $SESSION.NAME$ contain key information, e.g. paths, key files' name etc. In following lines, the basic functionalities of the main files will be exposed. Note that the particular content of the routines found will vary with the simulation. Hence, in the work here presented only the fundamental characteristics, common to all simulations, will be covered. In addition, some specific routines that are specially interesting for the application cases will be also introduced.
The routines included in those files are coded in Fortran 77 programming language. Typically, in F77 programes the different routines are introduced in the same file one after another. Starting with the *subroutine* command and closing with *end*, the different routines can be programmed one after another in the same file. Then the command *include* allows to read data from external files to be used in the routines.

.usr **file** Table 3.2 gathers the principal routines, common to the vast majority of simulation, that can be found in the *.usr* file. The *uservp* routine is actually imputing the main parameters

Routine	**Call schedule**	**Description**
uservp	Once per processor & once per point in every processor	Contains variable properties & gov. eq.
userdat	Once per processor for all points	Allows element & the boundary conditions defined in *.rea* file modification
userdat2	After GLL points distribution	Allows affine transformation
userf	Once per processor & once per point	Allows to define forcing terms
userbc	Once per processor	Boundary conditions (BC) definition
useric	Once per processor	Initial conditions (IC) definition
userchk	Once per processor after each time step	Solution checking & modification

Table 3.2: *.usr* file main routines

of the governing equations. In fact, those parameters are all input by means of three variables:

udiff, *utrans* and *ifield*. In this way, the variable represent different parameters depending on the equation over which they are applied. Table 3.3presents the definition of those variables as a function of the equation to which they are applied. The *userdat* routine is called after the distribution

Equation	**Parameter in the equation**		
	udiff	*utrans*	*ifield*
Momentum	ρ	μ	1
Energy	ρc_p	k	2
Passive Scalar	$(\rho c_p)_i$	k_i	i-1

Table 3.3: Parameters assigned in the *uservp* routine. Extracted from Fisher et al. [8].

of the GLL points and hence allows to change the elements if the topology remains constant. In addition, it allows to change the BC defined in the *.rea* file if there are any. On the contrary the *userdat2* file is called after the GLL distribution and therefore the elements can not be modified. However, the routine allows to change the mesh coordinates and boundary conditions. In general, it only allows affine transformations on the geometry. The *userf* routine allows to define a forcing term to be applied. In fact, it is sometimes interesting to be able to include an external force in the computations. For instance, imagine a case in which a flow is moving around a paddle which is moving in the counter direction powered by an electric motor. In such a case, that external force should be consider on in the *userf* routine. However, late on we will see how introducing a forcing term can be used as a computational tool to induce the turbulence, even no external force is applied in the physical system.
From mathematical theory it is well known that time-dependent partial differential equations systems require the use of both boundary and initial conditions. Boundary conditions are specified in the *userbc* routine in the *.usr* file whereas Initial conditions are introduced in the *useric* routine. Typically BCs are assigned to the correspondent elements' faces by means of the code specified in Table 3.4. In this way, the user only has to input the code and the implementation of the BC will be carried out by the pre-existent Nek5000 routines. However, if one wants to impose a particular BC different from the common types then it will have to be implemented manually in the *.usr* file. For instance, one could think about the Dong condition which later on will be introduced.

Nek code	**Description**	**Mathematical formulation**
O	Open or outflow condition	Non-stress formulation: $[-p\mathbf{I} + \nu(\nabla(\mathbf{u}))] \cdot \mathbf{n} = 0$ Stress formulation[1]: $[-p\mathbf{I} + \nu(\nabla(\mathbf{u}) + \nabla(\mathbf{u})^T)] \cdot \mathbf{n} = 0]$
SYM	Symmetric condition[2]	$\mathbf{u} \cdot \mathbf{n} = 0$ $(\nabla(\mathbf{u} \cdot \mathbf{t})) \cdot \mathbf{n} = 0$
P	Periodic condition	Assigns last point to the first point via $\mathbf{u}(\mathbf{x}) = \mathbf{u}(\mathbf{x} + L)$
W	Wall condition	$\mathbf{u} = 0$
v	Dirichlet condition	$\mathbf{u} = f(\mathbf{x})$

Table 3.4: Nek5000 pre-exist boundary conditions. Extracted from Fisher et al. [8].

As far as the initial conditions are concerned, Nek5000 allows different specifications:

- **Null initial conditions**: The default option.

- **Fortran function**: This option allows the user to specify a function as initial condition. Note, that this function can only be space-dependent.
- **Presolv**: This option is characteristic of temperature problems, in which the temperature conduction problem is solved and used as initial condition.

Note that initial conditions might be incompatible depending on the setup, geometry, etc. A careful study of both boundary and initial conditions is required in terms of the computational methods and the physical problem. In fact, one needs to ensure that the boundary conditions are consistent with the physical problem that one tries to simulate but one should ensure that the boundary conditions will be adequate from the computational point of view.

.rea **file** This file, in contrast with the *.usr*4 file does not contain a series of Fortran routines but rather a list of parameters that control the simulation. The *.rea* file is composed of three fundamental sections that gather different type of information.

Parameters The first section of the *.rea* file is dedicated to the simulation parameters that control the run. Those can be both physical and computational. For instance, viscosity, conductivity, number of timesteps etc. In addition, Nek5000 also allows a certain number of uncategorised parameters which can be used to introduce some parameters in the *.usr* file directly from the *.rea* file. This can be useful in order to deal with parameters that are frequently changing in some given set of simulations. In addition the passive scalar parameters presented in Table 3.3 can be specified in this file instead of introducing them in the *uservp* routine of the *.usr* file. Once again, choosing to introduce a parameter in the *.rea* file over the *.usr* file will depend on how frequently it is changed. Later on we will see how the simulation process takes place and how it's faster to change parameters in the *.rea* file, *ceteris paribus*, since only the run has to be repeated. Finally, the parameters' section in the *.rea* file also contains some logical controls that allow to enable and disable some properties of the simulation, e.g. unsteady or steady simulation.

Mesh and BC The *.rea* file also contains information on the geometry and curvature which is found a list of the locations in the cartesian reference system of the eight vertices defining the element for every element constituting the domain. Also this section list the curves that might be present when dealing with complex geometries. Curves are classified using a single letter code: C for circle, s for sphere and m for midside-node positions associated with quadratic edge displacement. In the the application cases presented further on, this section will remain empty as the urban environment is modelled only with straight lines. Note, that the curves are limited by the order of the spectral method. The curved surface can be "as high order as the polynomial used in the spectral method" (Fisher et al. [8]).
Boundary conditions are also specified in the *.rea* file. BCs are listed for each face of the elements forming the domain.

Output information The last section of *.rea* file contains the output information parameters. Those can be gathered into three groups,

1. **Restart conditions**: If the simulation is not starting a null time, it is in this section where the field that serves as starting point is specified. Note, that it is possible to use different

[1] Depending on the formulation two situations are consider. In the non-stress formulation the boundary is actually open whereas on the stress formulation it's a free traction boundary condition

[2] In the SYM condition, if the normal vector **n** and the tangent vector **t** are not aligned then the stress formulation has to be used.

source files for the different variables, e.g. velocity and temperature fields could be introduced from two different files (Fisher et al. [8]).

2. **Output specifications**

It is important to note that typically Nek operates with *.rea* and *.re2* files. The main distinction between those is that the first is in ASCII format while the second is in binary format. In fact, the afore-stated parameters which memory requirement is a function of the number of elements (e.g. geometry parameters) are typically in the *.re2* file. Recall that the *.re2* file was introduced in § 3.1.1.1 since it is generated by the Parallel Works platform. Note that the data transfer from *.rea* to *.re2* can be disabled by changing the negative sign in front of the number of elements parameter in the *.rea* file. During the application presented further on the simulations will be carried using both *.rea* and *.re2* files.

***SIZE* file** The *SIZE* file determines the size of the problem to solve. It gathers the parameters that define the sizes of the geometrical objects of the simulation. For instance, it includes the number of spatial points within the elements, the number of elements per core etc. Note, that this files is critical in the well functioning of the simulation as it controls the memory usage of the vast majority data structures. Also note that any change in this file requires the revaluation of the whole simulation process from compiling to the simulation's run. Table 3.5 gathers the principal parameters that can be found *SIZE* file.

Parameter	**Description**
$ldim$	Determines the dimensions considered in the simulation, i.e. $ldim = 2$ for a two-dimensional simulation and $ldim = 3$ for a three-dimensional one.
$lx1$	Polynomial order of the approximation
lxd	Polynomial order of the integration for convective terms
$lx2$	Determines the formulation of the Navier-Stokes equations solver. $lx2 = lx1$ implies the use of $\mathbb{P}_N - \mathbb{P}_N$ formulation whereas $lx2 = lx1 - 2$ implies the use of the $\mathbb{P}_N - \mathbb{P}_{N-2}$ formulation. [3]
$lelt$	Maximum number of elements per processor
lp	Maximum number of processors
$lelg$	Maximum number of elements to be used in the simulation

Table 3.5: *SIZE* file main parameters

The *SIZE* file, as previously stated, controls the vast majority of the memory allocation since it determines the size of the problem. A careful planning on memory allocation is advisable since the computational resources needed will be strongly dependent on such allocation.
"Per-processor memory requirements for Nek5000 scale roughly as 400 8-byte words per allocated gridpoint" (Fisher et al. [8]). The number of allocated grid points per core can be computed from the $lx1$ parameter using the formula proposed by Fisher et al. [8].

$$n_{max} = lx1 * ly1 * lz1 * lelt \tag{3.1}$$

where $lx1 = ly1 = lz1$ for the three-dimensional case and $lx1 = ly1$, $lz1 = 1$ for the two-dimensional case.

[3] Recall the formulation of the Navier-Stokes solver in §2.2.3.2.

3.1.3 *A priori* analysis

The last stage in the preprocessing of any computational fluid mechanics case requires an *a priori* approach. The principle behind the design of Fluid Mechanic simulation is that one designs a small baseline case with the sole motivation of extracting a series of qualitative and quantitative information to be able to design a proper final case. In this way, the baseline case results form the combination of the available literature, one's experience and intuition on the problem to solve.

The next lines will be dedicated to expose the qualitative and quantitative tools used in this project to design the cases and verify their validity. Note, that the vast majority of these tools require a baseline solved case. Later on we will see how the baseline case, although it does not carry a deep *a priori* analysis nor its derived results are valid, is extremely useful to produce a valid second case.

3.1.3.1 Data processing : MATLAB routines

The first step in the analysis here presented requires processing the data to obtain a more convenient format. Nek5000 is a very powerful system to deal with the simulation processes. However, as far as data processing is concerned it is more convenient to use high-level [4] programming languages which typically provide better data visualisation tools, a faster debugging process etc. During the analysis here presented the programming language will be MATLAB. In this way the first step is to be able to read the data produce by Nek5000 using the MATLAB programming language. Up to this point the actual Nek5000 solving process hasn't been introduced yet. For the moment let's assume the solving process is a **black box** that produces a series of files the nature of which is irrelevant at the moment.

The data processing routines are based on a MATLAB script developed by the Linné Flow Centre which allows to transform the data in a Nek5000 solution field into a MATLAB array. The MATLAB function *readnek.m* takes as sole input the Nek5000 solution file and extracts among many other parameters a MATLAB array containing the solution data. Note the data format in the output array might vary depending on whether one is running the 3D or 2D version of Nek5000. In this case only the 3D version will be considered as in the work here presented all simulations are run in the 3D version. the extracted data is allocated in multidimensional array, i.e. an array with three dimensions. In this way, the two first dimensions of the array correspond to the elements of the problem while the third correspond to the variables that are stored. Hence, regardless of the problem the third dimension of the array will always be eight as it's the number of variables stored in the Nek5000 file with the current setup[5].

Although the use of multidimensional arrays is efficient, it is not convenient from the user's perspective since for large data set, MATLAB is unable to show the data in the variable window. Hence the next step consist in reallocating the more convenient formats. This is done in a separate function *readnekel.m* which contains the *readnek.m* function and it is actually the one that it used in the analysis programs that will be introduced further on. The data extracted from the Nek5000 field files is transfer into a *struct* as follows.

[4]The term 'high-level" is here related with the level of abstraction a programming language has with regard to the fundamental working of a computer. Higher level programming languages present stronger level of abstraction from the working of the computer than lower level ones. E.g. HTML programming language used in the website development has a higher level than C programming language.

[5] Note that it is possible to have different variables stored in the Nek5000 field file by changing the setup. E.g. by selecting a 2D simulation.

```
size_data_in = size(data_in) ;
for i=1:size_data_in(1)
    el(i).GLL(:,1) = data_in(i,:,1) ; % Mesh points  in x
    el(i).GLL(:,2) = data_in(i,:,2) ; % Mesh points  in y
    el(i).GLL(:,3) = data_in(i,:,3) ; % Mesh points  in z
    el(i).Vel(:,1) = data_in(i,:,4) ; % Velocity field in x
    el(i).Vel(:,2) = data_in(i,:,5) ; % Velocity field in y
    el(i).Vel(:,3) = data_in(i,:,6) ; % Velocity field in z
    el(i).Pre(:,1) = data_in(i,:,7) ; % Pressure
    el(i).Tem(:,1) = data_in(i,:,8) ; % Temperature
end
```

In this way, the data is organised in terms of both the element and the variable. The advantages of this data structure is that it allows an easier comprehension on both the data itself and the inner workings of the functions. However, in order to apply data to further computations, the most convenient form is the standard two-dimensional array. This operation is also computed in the *readnekel.m* routine using the following instructions.

```
j_data = zeros(size_data_in(1)*size_data_in(2),6);
k = 1 ;
j = 1 ;
for i1 = 1:size_data_in(1)
    for i = 1:size_data_in(2)
        j_data(k,1:3) = el(i1).GLL(i,:) ;
        j_data(k,4:6) = el(i1).Vel(i,:) ;
        k = k+1 ;
    end
end
```

Note the operation shown in the first line of the code section above. This operation is known as preallocation and it is an optimisation yet simple techniques to save running time. In fact, it is more time efficient to change an array component rather than the size of an array. In this way, if the final size of the array is known by firstly producing a null array of such size a significant amount of computational time. It is important to keep in mind that even in the smaller cases the number of components in the arrays is important, e.g. in the baseline small case the *readnek.m* output array size is $5880 * 512 * 8$, i.e. the total number of points is roughly 24 million. Hence, it is mandatory to keep in mind the processing time and optimise as much as possible.

3.1.3.2 Boundary-layer analysis

The fundaments of the *a priori* analysis here presented is centred on two basic domains. On one hand the mesh resolution. On the other hand, the boundary layer analysis. In both cases the analysis requires a previous simulation, which as stated is the results of either previous studies or a guessing strategy.

Generally, the boundary layer of a fluid is defined as the layer of fluid in the near-wall region over which the velocity of the fluid changes from the null value at the wall to the free-stream value. Being able to solve the boundary layer properly, i.e. having enough resolution to represent the solution is a critical part in any flow simulation. However, in the simulations here considered, the turbulence is induced by means of a tripping force mechanism. Therefore, it is specially crucial to check that the turbulent regime and boundary layer are well established before the "test zone".

The considered appraisal methodology consist on computing a series of parameters either for their sole evaluation or for their comparison with the ZPG correlations. Therefore, the presentation here reported will be founded on those two axes.

Boundary-layer parameters A wide range of parameters both physical and artificial are available to describe a turbulent boundary layer. Here, only the parameters applied during the analysis will be considered. Note that, the parameters are here considered *at wall*, i.e. typically the first set of elements such that $y \neq 0$ is taken into account. Table 3.6 gathers the parameters that will be include in the analysis.

Parameter	**Description**
θ	Momentum thickness
τ_w	Local wall shear stress
δ_{99}	Boundary-layer thickness considered at the point where the velocity U is 99% of the free-stream velocity U_∞
C_f	Skin-friction coefficient
u_τ	Friction velocity evaluated at the wall[6]
Re_θ	Reynolds number computed as a function of the momentum thickness
Re_τ	Reynolds number computed as a function of the friction velocity

Table 3.6: Boundary-layer analysis parameters

In the following lines the different parameters will be discussed and formulated.

Boundary-layer thickness : δ_{99} This parameter is defined as the boundary layer thickness evaluated at the point where the velocity is the 99% of the free-stream velocity. In this way, there is no mathematical expression to be presented here. However, this parameter requires a careful selection of the point where the 99% of the velocity is considered. MATLAB provides a series of tools that allow to apply the point's selection using a couple of lines.

```
ind_d99 = min(find(abs(U - 0.99*Uinf)<tol99));
delta99 = y(ind_d99)                          ;
```

Note, that usually this quantity is defined for a given z-plane as a function of the x-coordinate.

Momentum parameters One of the physical quantities that can provide an important amount of information is the momentum. One can define the momentum thickness as,

$$\theta = \int_0^{+\infty} \frac{U(y)}{U_\infty} \left(1 - \frac{U(y)}{U_\infty}\right) dy \tag{3.2}$$

where $U(y)$ is the horizontal velocity at a given y and U_∞ is the free stream velocity. In practice the momentum thickness is computed numerically by means of any integration numerical method. In this case, the integral will be evaluated using the MATLAB built-in trapezoidal integration function *trapz*.

Using the expression presented in Equation 3.2 one can define the Reynolds number as a function of the momentum's thickness, i.e.

$$Re_\theta = \frac{U_\infty \Theta}{\nu} \tag{3.3}$$

Note that the Reynolds number generally relates the kinematic forces with the viscous forces. In this case the kinematics are evaluated using the momentum thickness of the boundary layer as characteristic length. The main advantage of the Reynolds number and in fact any other non-dimensional parameter is that one can very easily compare it with other cases without taking into consideration the dimensional differences between those cases.
Typically, both the momentum thickness and Re_θ are computed for a given z-plane as a function of the x-coordinate. In this way, one can obtain a graphical representation of the quantities evolution.

Friction parameters Following the same approach applied in the momentum parameters, one can define the shear stress at the wall as,

$$\tau_w = \mu \left(\frac{dU}{dy} \right)_w \tag{3.4}$$

where the velocity derivative can be directly from the simulation or can be computed using the different available methods such as finite differences.
Using the shear stress at the wall the friction velocity can be defined as,

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{3.5}$$

The Reynolds number can therefore be computed using the friction velocity and the boundary layer thickness δ_{99}.

$$Re_\tau = \frac{u_\tau \delta_{99}}{\nu} \tag{3.6}$$

Finally, the friction coefficient can be defined by the combination of the shear wall stress and the well-known dynamic pressure,

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho U_\infty^2} \tag{3.7}$$

ZPG correlations When dealing with turbulent boundary layers (TBL), typically one distinguishes adverse-pressure gradient (APG) from zero-pressure gradient (ZPG) turbulent boundary layers. The main distinction between the APG and ZPG boundary layer can be very easily understood by considering the pressure term in the momentum equation. Recall Equation 2.11 and consider the pressure gradient term axially, i.e. $\frac{dp}{dx}$ being x the axial coordinate. In the APG boundary-layer the static pressure grows with the flow, i.e. $\frac{dp}{dx} > 0$. On the contrary when dealing with ZPG boundary layer the pressure remains constant, i.e. $\frac{dp}{dx} = 0$. Vinuesa et al. [39] presented a study on the conditions that make a APG turbulent boundary layer *well-behaved*, i.e. independent from the inflow and exempt of numerical artefacts. In addition, their studies propose a series of empirical correlations that predict the Re_Θ evolution of the friction coefficient C_f in both APG and ZPG TBL. In the analysis here presented only the ZPG TBL is considered since it is the one to be applied during the applications.

[6]The expression "at the wall" is here used only as matter of emphasis since the magnitude in question can only be defined in the vicinity of the wall.

The empirical correlations here considered is formulated as the friction coefficient in function of the momentum Reynolds number Re_Θ. In particular two correlations are considered, a first approach considers exclusively the logarithmic variation of Re_Θ and hence at some point “misses” the flow effect at low Reynolds numbers. A second approach includes higher-order terms outside the logarithm expression considered in the first approach. The use of higher-order terms allows to retain the effects of the flow at low Reynolds number. Following the formulation presented by Vinuesa et al. [39], Equations 3.8 and 3.9 gather the correlation expression for the two afore-stated approach, respectively.

$$C_f = 2\left[\frac{1}{\kappa}\ln\left(Re_\Theta\right) + C\right]^{-2} \tag{3.8}$$

$$C_f = 2\left[\frac{1}{\kappa}\ln\left(Re_\Theta\right) + C + \frac{D_0\ln\left(Re_\Theta\right)}{Re_\Theta} + \frac{D_1}{Re_\Theta}\right]^{-2} \tag{3.9}$$

As in any empirical correlation the value of the coefficients are determined by the combination of the available data sets and empirical experience. In this case, the parameters presented by Vinuesa et al. [39] are gathered in Table 3.7. The afore-presented correlations are implemented as

Case	κ	C	D_0	D_1
ZPG TBL	0.384	4.127	220	-1945
APG TBL with constant $\beta = 1$	0.361	5.300	250	-2100
APG TBL with constant $\beta = 2$	0.349	6.886	260	-2500

Table 3.7: Empirical simulation coefficients. Extracted fromVinuesa et al. [39] .

a MATLAB function such that they can be compared with the case C_f. The comparison is done plotting the friction coefficient's evolution as a function of the Reynolds number Re_Θ.

3.1.3.3 Resolution analysis

The concept of resolution is the result of combining the domain dimensions with the element dimensions. It appears to be logical that for a given domain if one decreases the element size, *ceteris paribus*, the number of elements needed to cover the domain will increase. Previously, it was discussed how increasing the number of elements tends to improve the precision of the simulation although it also increases the cost. Those ideas are all gathered in the resolution analysis. Once again, the objective is to have a sufficient resolution in the domain's critical zones while ensuring that the simulation remains cost-efficient. Please recall §2.2.2.2 for the full discussion on cost in LES.

In the work here presented resolution is appraised using two methodologies. Firstly by the inspection of grid spacing evolution in each direction. Secondly, comparing grid spacing with the Kolmogorov scales.

Normalised grid spacing analysis The analysis here introduced consists on producing a graphical representation of grid spacing evolution as a function of space coordinates. Once the plot is produce the evolution of such spacing is confronted to the available literature and one's own experience. Typically, those magnitudes are evaluated in the domain's key areas, i.e. in the zones of interest either from the study's or simulation's perspective. In the case of urban environments, interesting areas tend to be at the surrounding of building-like obstacles. However, in turbulent simulations it's commonly interesting to check the resolution in areas where the turbulent boundary

layer is computed and developed. In the work here presented, normalised grid spacing analysis will be applied in every spacial direction over the zone enclosed by the inflow and the first obstacle, since it's precisely there where the turbulent boundary layer is solved.
The normalised quantities are given by the following expressions.

$$\Delta x^+ = \frac{u_\tau \Delta x}{\nu} \quad ; \quad \Delta y^+ = \frac{u_\tau \Delta y}{\nu} \quad ; \quad \Delta z^+ = \frac{u_\tau \Delta z}{\nu} \tag{3.10}$$

where $\{\Delta x, \Delta y, \Delta z\}$ are the grid spacing in the x, y, z directions, respectively and u_τ is the friction velocity. Note, that the expression presented in Equation 3.10 might be applied over different points depending on the considered direction. Also note that the grid spacing is evaluated in vector form, i.e. the local spacing for every pair of adjoining points is computed such that for N points in a given direction the grid spacing vector will have $N-1$ components.

Kolmogorov-scale analysis The second analysis applied on the simulation's resolution is based on the comparison of mesh scales with the Kolmogorov scales, by means of a ratio between the mesh elements volume and the Kolmogorov length scales. In fact, urban environment flow simulations typically involve complex flow which happens to be far from the wall. In such flow condition the normalised grid spacing analysis does not hold, hence this alternative method is applied. The idea behind this analytical tool is to analyse the afore-mentioned ratio in given sections of the domain. Once again, the actual interpretation of the quantity is based on previous experience and the available literature. Later on, the *statistic toolbox* will be introduced as the main tool to compute several of the terms here applied. Again, in this case a baseline simulation is required in this part of the analysis.

Energy cascades and Kolmogorov hypothesis Before getting to define the Kolmogorov scales, one should refer to the concept of *energy cascade*. The concept was introduced by Richardson in the early twenties to describe the flow structures' behaviour in the turbulence. He argued that the turbulence is made by a series of eddies that of different sizes which have their own characteristic length, time and velocity. From fluid theory it is known that kinetic energy is introduced in the turbulence by the larger scales which is then transfer to the smaller scales and finally is dissipated by viscous mechanisms. In this way, larger eddies will have a greater characteristic velocity and hence the viscous effects will have lower importance. As one decreases in the eddies' size, the kinetic energy decreases and hence the viscous effect become more and more important, up to the point, in the smallest scales where sole viscous dissipation mechanism are present. The concept of energy cascade precisely refers to that energy transformation throughout the scales. The length scales in the larger eddies have been found to be very close to the flow scales (Pope [25]). However, Richardson did not assessed the size of the smaller eddies, responsible for the viscous dissipation. Kolmogorov advanced Richardson's theory following a series of hypothesis sustained on the fact that both velocity and time-scales decrease with the length scale. In this way, the first hypothesis consist on assuming that, contrary to large-scale motions, small-scale motions are statistically isotropic (Pope [25]). In fact, Kolmogorov argued that as energy flows down the cascade all in the information concerning the shape of the eddies is lost. Under this assumption, the geometry of the small-scale eddied is independent from the mean flow field, boundary conditions etc. In this way, "the statistics of the small-scale motions are in a sense universal" (Pope [25]). From the modelling perspective, the two fundamental mechanism in the energy cascade are the energy-transfer rate from larger scales to smaller scales and the viscous dissipation rate, concerning the smaller-scales eddies. Those processes are evaluated with the Kolmogorov similarity hypothesis which states that the statistics of the smaller-scale motions in fully turbulent flows are universal and can be exclusively described with the kinematic viscosity ν and the viscous dissipation ε. From the combination of the isotropy and similarity hypothesis the Kolmogorov scales are formulated.

Kolmogorov scales formulation The length, velocity, and time Kolmogorov scales, respectively $\{\eta, u_\eta, \tau_\eta\}$, are expressed exclusively with ν and ε as follows.

$$\eta = \left(\frac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}} \quad ; \quad u_\eta = (\varepsilon\nu)^{\frac{1}{4}} \quad ; \quad \tau_\eta = \left(\frac{\nu}{\varepsilon}\right)^{\frac{1}{2}} \tag{3.11}$$

Although Equation 3.11 presents the three Kolmogorov scales, i nthe analyssi here presented the only scale considered is the length scale η.

Assessing resolution with the Kolmogorov scale Now, that the scales have been introduced it's time to introduce the evaluation procedure. The volume of a given element can be computed as the product of the element size in each direction. From the volume one can define the parameter h as the cubic root of the volume of a given element.

$$h = \sqrt[3]{\Delta x \cdot \Delta y \cdot \Delta z} \tag{3.12}$$

Note, that h as well as the Kolmogorov length scale is evaluated for each element. Hence, two vectors which can be related with the spacial location of the element. This is particularly useful to obtain contour plots over the domain. Later on, we will see how the Kolmogorov scales are computed in the workflows here considered and how they introduce the use of a interpolation mesh which typically is smaller than the solution mesh. Hence, very often one has to interpolate the Kolmogorov scale to fit the solution mesh.
Finally, combining Equation 3.11 and Equation 3.12 one obtains the evaluation parameter h_η,

$$h_\eta = \frac{h}{\eta} = \frac{\sqrt[3]{\Delta x \cdot \Delta y \cdot \Delta z}}{\sqrt[4]{(\nu^3)/\varepsilon}} \tag{3.13}$$

h_η is computed for each of the elements in the solution mesh and then is plot using a contour plotting tool. In this way, the resolution can be assessed in every part of the considered domain.

3.2 Solution process

Once the preprocessing phase if completed and the setup is fixed, the next phase consists on running the simulation, i.e. solving the equations to obtain the solution fields among many other files. Note, that the solution process will be only presented in general terms as it doesn't carry an important amount of implementation in our case.

3.2.1 Fundamental stages in the solving process

Figure 3.5 combines the files presented in the preprocessing stage with the processes that take place in the solving stages.

Solving process The two principal processes that form the solving block are compiling and running, which are executed sequentially. As it is shown in Figure 3.5 part of the preprocessing files are compiled using the *makenek* routine which produces the *nek5000* executable file. The file in question carries the "skeleton" of the simulation, i.e. the equations to solve, the *SIZE* parameters

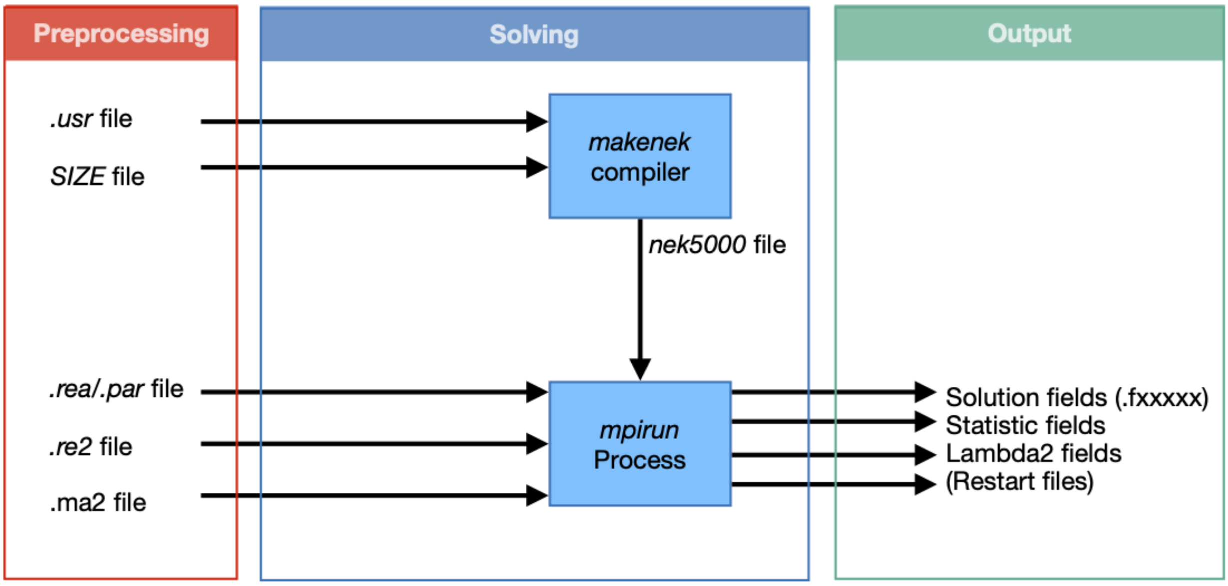


Figure 3.5: Solving process stages

etc. Then, this executable file is run using the MPI running platform, *mpirun*. During the running process some additional files coming from the *preprocessing* stage have to be included. On the one hand, the geometry files need to be included. Recall §3.1.1.1 for the full description on how those files are generated and what properties characterise them. The run process, i.e. the simulation, is controlled with the *.rea* or *.par* file [7] which contains not only the parameters defining the equations but also the control parameters, e.g. the number of time-steps to be simulated, how often are the field files written etc. In addition, depending on the workflow, the simulation might save additional files such as restart files or statistic files. Later on, the *statistic toolbox* will be introduced and we will see how the *.rea* or *.par* file control the toolbox's output. In practice, one might save some field files at the beginning of the simulation to check the correct implementation of the conditions and then wait till the end to save the fields of interest. In fact, in three dimensional simulation field files tend to occupy an important amount of disk space, hence one should only save the fields that are useful for the study in question. Note, that one needs to simulate a sufficient amount of time to have a convergent solution, i.e. the solution does not depend on any simulation or numerical parameter. As far as the file format is concerned, field files are saved following a five digit numerical system (*.fxxxxx*), e.g. the fist file will be save as *.f00001*, the tenth file as *.f00010*, etc.
Note that while field files and restart files are a standard output of Nek's solution process, λ_2 criterion and statistic files are actually part of a setup developed by the Linné Flow Centre at KTH Royal Insitute of Technology to provide a series of quantities that are useful in the analysis here performed.

Solution process appraisal The are several advantage in having a two stage process in the solving scheme. Firstly, having the compiling and running processes segregated allows to very easily change the simulation parameters without having to repeat the whole process. This fact

[7] Originally Nek5000 simulation could only run using the *.rea* file which mainly consist on a list of parameters. Newer Nek versions allow the use of *.par* file instead. This last is built following a "description approach", i.e. the parameters are assigned to specific variables instead of just being listed. The *.par* file provides a more user-friendly approach however it tends to be more limited when dealing with specific workflow in which non-standard parameters are required. Once the selection of one over the other will depend on the specific workflow considered.

combined with the restart capabilities allow to incorporate pieces of a previous simulation into a given simulation and change a particular simulation parameter [8]. Secondly, the compiling process allow to detect some errors that might be present in the setup without having to do the much longer running process. This is a clear advantage for debugging as the simulation process takes an important amount of time. However, the compile process focus on the element constitution, i.e. integration order, formulation scheme, etc. Hence, important sources of mistakes related with the geometry are not appraised during the compile process. The *nek5000* file run using *mpirun* contains checking routine that allow to check the fundamental properties of the simulation such as boundary conditions, mesh etc. Those routines are executed at the begging of the simulation such that it can be interrupted if any of the checks in questions fails.

3.2.2 Special routines in the solving process: Tripping

§3.2.1 was dedicated to the presentation of the simulation process in general. The current section will be rather focusing on some specific routines that are not standard in the platform but are fundamental in the workflows applied later on. The idea now is not to cover the inner workings of the Nek processes but rather to present some tools that are applied during the solution process but are not included in a Nek5000 standard workflow. Concretely we will focus on the tripping routine since it's a fundamental part of the simulation that does not produce any output.

The tripping routine is a numerical tool applied in turbulent simulations to induce turbulence. The main idea behind is to be able to achieve a fully turbulent boundary layer rapidly from a non-turbulent inflow. Without the use of a tripping force method, one needs to importantly raise the Reynolds number and simulate till the flow becomes turbulent, thus increasing cost. The tripping methodology here presented was designed by Schlatter and Örlü precisely to study the transition to turbulence in DNS.

Physical similarity The numerical tripping methodology is inspired in an actual physical method implement in wind tunnel experiments. Wind tunnels tend to be limited in terms of size, power and thus experimental capabilities. However, the tested prototypes correspond to flying objects which, at some point, will face turbulent boundary layers, for which classical wind tunnels can not very usually provide the testing conditions. In order to solve this problem, the transition is induced by means of adhesive bands over which are distributed some kinds of "perturbators", e.g. a series of dots, such that the flow is disturbed and the turbulence is induced. The numerical approach here presented aims to reproduce this kind of behaviour in the simulation. The main idea is to introduce a weak random volume force in the forcing terms of the Navier–Stokes equation acting in the wall-normal direction such that those disturbances are created in the simulation flow. Once the disturbance are set, those will grow in the simulation thus creating turbulent flow.

Tripping formulation Schlatter and Örlü [30] present the mathematical formulation of the tripping force here reported. The actual forcing term introduced in the Navier-Stokes equations is given by Equation 3.14.

$$F_2 = \exp\left(\left[\frac{x - x_0}{l_x}\right]^2 - \left[\frac{y}{l_y}\right]^2\right) g(z,t) \tag{3.14}$$

[8] Note that this capability has some restrictions. One cannot make significant modifications in the geometry or element's constitution and use a restart field that is significantly different. In fact, changing the $SIZE$ file very usually requires the clean and compilation of the whole dataset.

where $\{l_x, l_y\}$ represent the Gaussian discretisation of the forcing region and $g(z,t)$ is the forcing function. Note, that the force is applied in the lower y-plane of the domains around the x_0 position. The forcing function is the element that actually introduces the randomness in the force varying both in time and space.

Tripping implementation Now that the mathematical formulation has been introduced it's time to deal with the implementation of the tripping force within the Nek5000 framework. Recalling Table 3.2, we saw that forcing terms are usually implemented within the *userf* routine. However, the complexity of the tripping force requires the call of additional function as well as separate file where the forcing function is implemented.
In the setup here considered the tripping force function is called in the *userf* routine in order to allocate the force terms, which are outputs of the tripping force function, in the Nek5000 standardise format. The function in question is actually in an independent set of files which are all part of the tripping toolbox. One of the fundamental files of the toolbox is the *TRIPD* file which controls the different subroutines that allow to compute the tripping force terms. Note that for a matter of convenience, the complete tripping toolbox is not found in Appendix. In addition to the afore-stated terms, the function requires an initialisation as well as periodic checks. Those are implemented in the *.usr* file more precisely in the *userchk* routine. In addition, the input parameters that define the tripping force are introduced in the simulation in a specific file, the*forparam.i* file.

Tripping force solution Summarising the previous points, the routine here presented basically applies a random force on a specific part of the domain to induce the transition to turbulence. The user defines amplitude, span-wise length and temporal frequency as well as the location of the tripping line. Then the routine generates a series of random forces which produce very particular flow structures that are responsible for inducing the turbulence.

Figure 3.6 shows an example domain developed to test the tripping routine. In this case, the location of the tripping line happens to be at the inflow. Once the origin fixed, the tripping forces vary in every spacial direction due to the randomness of the process. Note that the scale of the structures can be controlled using the input parameters files. In fact, one needs to properly adjust the terms such that the simulation properly works, e.g. an oversize tripping force could cause the simulation to explode. One needs to keep in mind that the tripping terms are a numerical object which sole purpose is to induce the turbulence. Hence, one should try to minimise the collateral effects since this force is not part of the physical description of the problem.

3.3 Postprocessing

The last stage in the implementation of a flow simulation consists on treating the obtained numerical results such that a proper analysis can be carried out. Once again, this process will be strongly dependent on the application of the simulation as well as the tools involved. In our case postprocessing will be founded on two pillars. On one hand, a qualitative approach, in which both mesh and solutions are inspected. On the other hand, a quantitative approach obtaining and assessing flow statistics. Note, that processing considers the treatment of the obtained data and not the verifications that might be needed to ensure the quality of the simulation. Recall that the resolution analysis presented in §3.1.3 serves both to design the final simulation and to check it once the simulation is finished. In this way, no verification will be presented in the current section.

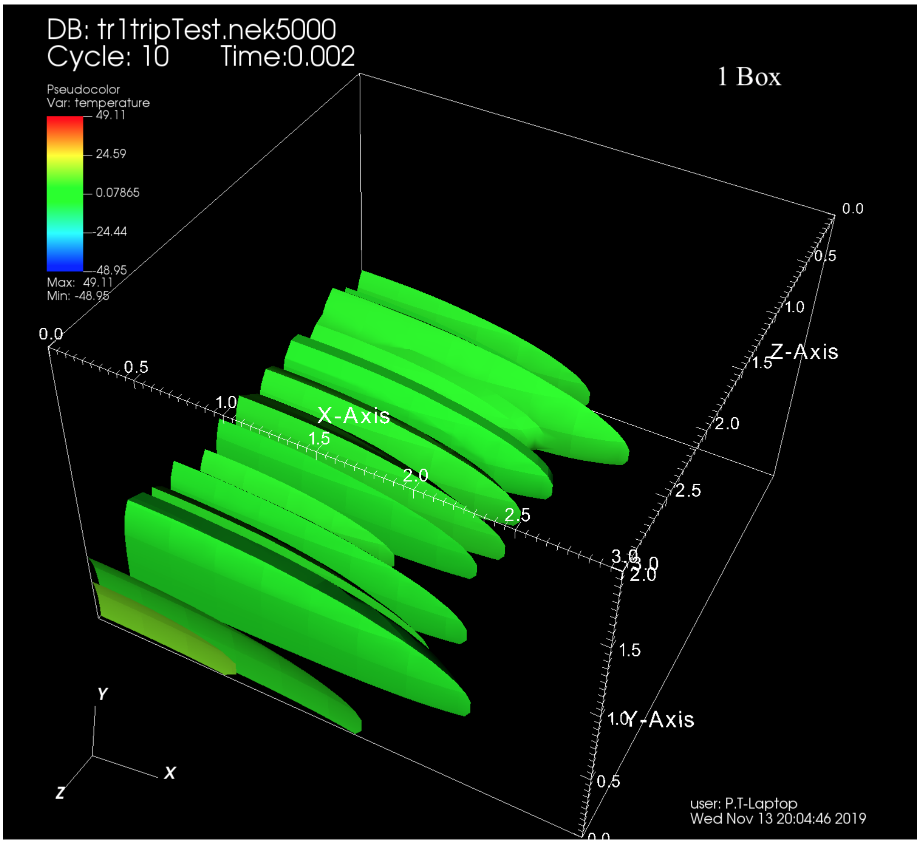


Figure 3.6: Isosurface level curves of the tripping force in a single box domain

3.3.1 Qualitative postprocessing: Results visualisation

In § refsec:3.2 the solution process was presented concluding in the simulation's output, the solution fields. The simplest postprocessing procedure consists on the solution's visualisation. Several tools are available for the Nek5000 format. In fact, Nek5000 provides a built-in solution viewer *Postnek* which allows to directly see the solution in Nek's platform. However, *Postnek* is a quite rudimentary tool thus limited to more complex analytical needs. That's why typically Nek user's rely on external tools for visualising the solution. In our case *VisIt* will be used in all fields visualisation.

VisIt is a open-source graphical visualisation tool developed by the Advanced Simulation and Computing Initiative of the United States of America Department of Energy to serve as platform to post-process results obtained from various computational domains. The tools allow the visualisation of scalar and vector fields in both two and three dimensions. In addition, the integration with the Nek format is fast and simple. The sole integration process required is the generation of a $.nek5000$ file [12] which purpose is to gather the format information such that VisIt can find the fields to be visualised. In fact, the file in question only contains the field name and the starting and ending fields number, allowing to visualise several fields one after another, i.e. showing the simulation's evolution over several moments. In fact, VisIt tool can be used for any kind of field, e.g. statistic fields, restart fields etc. However, in our case this kind of visualisation will be mainly

[12]It's important to distinguish the *nek5000* executable file presented in the solving process (§3.2) from the $.nek5000$ file here introduced. The fore is an executable file that carries essential simulation information whereas the last only serves a link between Nek and VisIt.

considered for solution fields.

3.3.2 Quantitative post-processing: Statistics toolbox

The statistic toolbox is a set of Nek5000 and MATLAB routines developed with the purpose of computing turbulence statistics in Nek5000. The technicalities of the toolbox can be found in the work of Vinuesa et al. [36]. In addition, further studies show how the toolbox is applied and how the obtained data can be used in the characterisation of turbulent flows. In this area, one might highlight the works of Vinuesa et al. [35], [38], for showing precisely that application. The toolbox is implemented transversally throughout the Nek5000 workflow, i.e. statistic routines are run in very stage of the workflow from compiling to postprocessing. However, in the lines here presented we will focus mainly on the postprocessing part for being the most interesting one.

3.3.2.1 Statistic toolbox workflow

Firstly let us comment the general working of toolbox such that the reader might have an idea on how this is implemented. As stated in the previous paragraph, the statistic toolbox incorporate files in every stage of the typically Nek5000 solution process. However, the toolbox also requires some additional processes that happen outside Nek.

Compiling and running the statistic toolbox Firstly we will assess the required setup within the classical Nek5000 workflow, i.e. in the compile and run phases. Recall the Nek5000 solution process described in §3.2. The compile stage's output is the *nek5000* which contains the fundamental information to be simulated in the running phase. Thus, it's seems consistent to introduce the statistic's computations framework at this stage. Concretely, the statistic routines are introduced in the compiling folder which are then called in the well-known *.usr* file. Then, after compiling the *nek5000* file is produce and then has to be simulated in the running folder. During the running process statistic computations and writing protocols are controlled within the *.rea* file environment just as the rest of simulation. More precisely four parameters are actually added in the *.rea* file such that one can control the statistic's simulation. Vinuesa et al. [36] present a list of the control parameters that control the simulation. Find those gathered in Table 3.8.

Name	Number	Description
STAT_COMP	p068	Determines the number of time steps between the samples used to compute the time-averaging the statistic quantities.
STAT_OUTP	p069	Determines the time-step interval between the writing of statistic field files. E.g. a STAT_OUTP of ten implies that the files will be written every 10 time-steps.
CHKPTSTEP	p070	Determines the time-step interval between the writing of restart files
IFCHKPTRST	p071	Determines whether the simulation starts from a null time, i.e. from initial conditions, or it is started using restart files

Table 3.8: Statistic toolbox control parameters

From the routine's perspective, the process start with the initialisation of the variables in the subroutine *stat_avg* the first time that the statistics toolbox is called in the simulation. Then, if "the current step is a multiple of STAT_COMP then the subroutine *stat_compute* is called to perform the statistic computation" (Vinuesa et al. [36]). In addition, if the step happens to be a multiple of STAT_OUTP then the computed statistics are written in a field field. The output of this toolbox, in the 3D version, consist on 44 time-averaged variables[13] stored in 11 files which are all name using the prefix sXX where XX denotes the number of the statistic file, starting from $s01$ and closing with $s11$. In addition, the Nek fields numbering system is also applied here, i.e. the suffix $.fXXXXX$ is used the eleven files. Hence, for each time-step over which the statistics are saved, i.e. multiple of STAT_OUTP, eleven files are saved starting at $.f00001$. From now on, we will refer to the eleven output files as s-files. Note, that typically one saves statistic at the middle and end of a given simulation, thus in this case the disk occupancy of those files, despite being important, is not a limiting factor.

General postprocessing workflow Once the s-files are obtained, there are several routines that need to be implemented in order to post-process the statistic quantities. Firstly the time average of the run if computed. This operation consist on taking the statistic files obtained during the simulation and taking the time average. There is a particular routine in the postprocessing folder inside the toolbox. This operation produces 22 files following the same naming system with the exception of the prefix which this time is aXX. We will call those the a-files. The a-files contain the three-dimensional turbulence statistics and several derivatives. Now, from the format's point of view those files are identical to the solution field files and thus can be visualised with VisIt or any other graphical tool compatible with the Nek format. However, a further postprocessing is also possible by interpolating the results in a mesh and transferring the interpolated files to MATLAB. Later on, this part will be specifically assessed.

3.3.2.2 Statistic's postprocessing: MATLAB routines

Now that the general workflow in the statistic toolbox has been introduce, we will focus on the last part of the postprocessing stages, i.e. the interpolation of the a-files as well as further steps. In fact, it's precisely this stage that it's the most interesting since it's almost custom-made for the cases.

Interpolation mesh First of all the interpolation mesh has to be created. This time the mesh is generated using a MATLAB script which creates the grid-points that are afterwards transformed into a suitable Nek format. This mesh is not held to the same quality standard as the solution mesh, i.e. this mesh will be significantly simpler in terms of resolution and methodology. In fact, no spectral methods are here applied only a uniform or progressive point distribution. As far as resolution is concerned, the number of points will be significantly lower having thus a much coarser resolution. Note that mesh resolution is not a critical factor this time since the interpolation mesh is exclusively used to represent the statistic results and thus no computations are carried over it.
The interpolation mesh is built by removal, i.e. the domain is filled with points and then some of those points are removed in order to produce the obstacle's geometry.

Interpolation mesh generation As stated in the previous lines, the domain is firstly filled with points and then the obstacle are "carved" over it. The points are distributed over the domain by means of several stencils which are then reproduced in order to fill the domain. In fact, a series of x-coordinate are set over which y and z stencils are distributed. As far as the

[13] The 44 time-averaged variables include both fundamental variables such as the velocities as well derivative and tensor variables. Find the complete list of variables in the work of Vinuesa et al. [36]

stencil are concerned, in the whole range of application cases the meshes are uniform in x, z and double-progressive [14] in y. The stencils are introduced using MATLAB arrays as follows,

```
k       = 1:nmax                                                  ;
x       = linspace(xmin,xmax,npt_x)                               ;
yn      = [ymin 1-cos((2*(1:nmax)-1)/(2*nmax)*pi/2)...
           2-cos((2*(1:nmax)-1)/(2*nmax)*pi/2) ymax]              ;
zn      = linspace(zmin,zmax,npt_z)                               ;
```

Note that in this case the progression over the y-coordinate using a cosine function. This formulation produces a concentration of points in the lower y-coordinates of the domain's half-plane which is precisely the desired effect.

Once the stencils are set, the data is reproduced such that the full three dimensional space is covered and the data is split in coordinate form. The obtained format consists on three coordinate vectors $\{x, y, z\}$ which contain sequentially the coordinates of every grid-point. This operation is obtained using a series of array operations,

```
np = length(zn)*length(yn)*length(x);
xx = zeros(length(x),length(yn));
yy = zeros(length(x),length(yn));

 for i=1:length(x)
     xx(i,:) = x(i)    ;
     yy(i,:) = yn    ;
 end
x_pts = xx(:);
y_pts = yy(:);

%Create coordinates
x = zeros(np,1);
y = zeros(np,1);
z = zeros(np,1);

for i = 1:length(zn)
    f                 = length(x_pts)  ;
    x(1+(i-1)*f:i*f)  = x_pts          ;
    y(1+(i-1)*f:i*f)  = y_pts          ;
    z(1+(i-1)*f:i*f)  = zn(i)          ;
end
```

The next step consists on selecting the coordinates that correspond to the obstacle's location and remove such points. This can very easily achieved using the *find* MATLAB built-in function. Note that this operation is the fundamental removal operation and thus has to be repeated for every obstacle in the domain. The following lines of code allow to find and remove the coordinates such that the obstacle shape is obtained.

[14] The term double-progressive here signifies that domain's length in the y direction is here split in two in order to produce two progressive stencils. This is done in order to concentrate grid-points not only at the domain's bottom but also in the y mid-plane.

```
r_z = find(abs(z)< obs_w);

k = 1;
for i=1:length(r_z)
    yzi = y(r_z(i));
    xzi = x(r_z(i));
    if (abs(yzi)<obs_h)
        if (abs(xzi)==min(abs(x)))
            r_xyz(k) = r_z(i) ;
            k        = k +1   ;
        end
    end
end

x(r_xyz) = NaN ;
y(r_xyz) = NaN ;
z(r_xyz) = NaN ;
```

Note that the number of grid points must be sufficient to properly represent the obstacles, i.e. if the space between points is to wide compared with the obstacle's directions, the resulting geometry might be badly represented.

Finally, a series of standardised commands allow to transform the MATLAB distribution of points into a Nek-compatible format. The output files consist on three $.fort$ files, one per direction, that contain the grid-point's coordinates.

Interpolation process Once the mesh created, an alternative *.usr* file is run in a distinct folder in order to interpolate the statistic files over the interpolation mesh. This new *.usr* file is used with the sole purpose of doing the interpolation thus all the non-related parameters are put to zero. In addition, the interpolation routine is introduced in the *userchk* Nek function. The function in question loads the interpolation mesh (in $.fort$ format) and interpolates the statistic files over it. The result is the *int_fld* file which contains the interpolated results. Those results are read in MATLAB to be further post-processed.

Interpolated fields postprocessing The final stage in this process consists on loading the interpolated fields to MATLAB such that one can apply further computations using this data. This procedure starts by reading the interpolated data in MATLAB. Then the data is separated using in terms of fields using a MATLAB cell data structure. This applied using the following iteration loop,

```
for field_number=1:22
    %U
    fname=[strcat(pathName,'/U'),num2str(field_number,'%2.2d')];

    [fid,message] = fopen(fname,'r','ieee-le');
    hdr           = fread(fid,1,'int32')        ;
    F             = fread(fid,hdr,'*char')'     ;
    dum5          = fread(fid,1,'*float64')     ;
    time          = fread(fid,1,'*float64')     ;
    sfn           = fread(fid,1,'int32')        ;
    dum6          = fread(fid,1,'*float64')     ;
```

```
    aa=fread(fid,Inf,'*float64');

    %Assign points taking into account gaps among processors
    for i=1:ncores
        bb(1+sum(history2(1:i-1,2)):sum(history2(1:i,2)),1)=...
        aa(1+sum(history2(1:i-1,2))+(i-1):sum(history2(1:i,2))+(i-1));
    end

    UD{field_number}=bb;

    fclose(fid);
end
```

Note that this loop operation is applied over the velocity in each direction and their correspondent derivatives as well as over some additional quantities, all in separated cell data structures. Once the data sorted, each of the quantities is manually assigned to a specific variables and finally saved in a MATLAB *.mat* file for further use.

Further postprocessing To close the statistic toolbox description, one shall make some comments on the available steps after the obtaining afore-mentioned quantities. From this particular point and further on, the analysis applied is fully dependent on the nature of the study being carried out. In this way, the current paragraph is more of a example than it's a prescription. In our case the statistics are partially used not only to analyse the case from which those have been extracted but also to generate better cases. Recall the *a priori* analysis introduced in §3.1.3 and how the quantities used in the boundary layer analysis depended on both velocities and their derivatives. Thus to be able to compute those quantities one has to use the afore-mentioned quantities. From the workflow's perspective, one launches a preliminary simulation to compute the quantities needed to obtain the *a priori* analysis such that the actual simulation can be designed. In addition, this approach allows to debugging of the tools applied.
Moreover, in the case of a final simulation the afore-presented quantities are actually interesting on their since they are the statistic quantities of the flow. By having those quantities in MATLAB format one can easily analyse them using any method of preference.

4

Simulation and Results

4.1 Introduction

The previous chapters and sections were dedicated to the introduction of the theoretical and technical concepts that sustain the processes applied in the present study. We saw the theoretical description of turbulent flows as well as the numerical methods that are the foundations of the simulations. From the technical perspective, the programs and methodologies were also described. The current chapter will be focusing on the final simulation that is the object of the present study. The exposition will follow the same structure introduced before, i.e. we will be starting with the simulation setup, then the solution process and finally the postprocessing.

Recall that in §3.1 we saw how it is necessary to produce a preliminary simulation to be able to properly define the final one. This was the case in the present study were a reduced-domain simulation was implemented in order to determine the resolution needed to properly represent the flow. The preliminary simulation consisted a two-obstacle geometry evaluated in a $10h \times 2h \times 2h$ domain. The element resolution considered was rather coarse and the simulation was run using a $N = 5$ GLL distribution within the elements. It is clear that the resolution of such case is insufficient to be able to fully describe the turbulent quantities of the flow. However, the interest of the preliminary simulation is not the solution *per se* but rather the information extracted in terms of resolution. Furthermore, the preliminary simulation is also useful to check the well functioning of the routines since the running time is significantly smaller. Using the afore-described simulation we were able to determine the resolution needed and thus the setup parameters that are presented in the current chapter. In further section the specificities of the case will be introduced.

As far as the objective is concerned, the present simulation aims to produce a canonical case that serves as foundations of the workflow integrated in the present study. In fact, this final case is the result of the tuning of the tools that were put together in the development of the workflow. In this way, the objective is double. On the one hand, the present case allows to show the well functioning of the workflow, which was one of the aims of the present study. On the other hand, this application case, despite being rather theoretical, provides a better understanding of the flow and it's representation using the current methodology. Nevertheless, recalling the objective of the present study, the focus was driven towards the development of the methodology and thus the canonical case here considered should be more than sufficient for its appraisal.

4.2 Preprocessing and setup

Recall the concepts introduced in §3.1 and in particular the importance of preprocessing in the proper development of the simulation. In fact, the preprocessing here considered in the result of an iterative process that includes the results of the preliminary simulation as well as some additional tests required to properly tune the different routines that form the workflow. The current section will be dedicated to the discussing of the process that are more significant in the case design. The exposition will be founded over three axes. Firstly, the geometrical design, i.e. the selection of the parameters that define the shape and dimensions of the considered geometry. Secondly, the mesh design will be introduced. This time the focus is not put in the domain itself but rather in the parameters that form the mesh and in particular their relation with resolution requirements. Finally, the boundary conditions will also be described.

4.2.1 Geometrical design

In §3.1.1.1 we introduced the processes used in the meshing platform to define the geometry and generate the mesh. Recall that the domain is built using a series of parameters which are gathered in Figure 3.1 and it's normalised using the obstacle's height. As stated, the idea with this case was to build a case sufficiently large to be able to show proper results but keeping complexity constrained such that the workflow can be appraised.

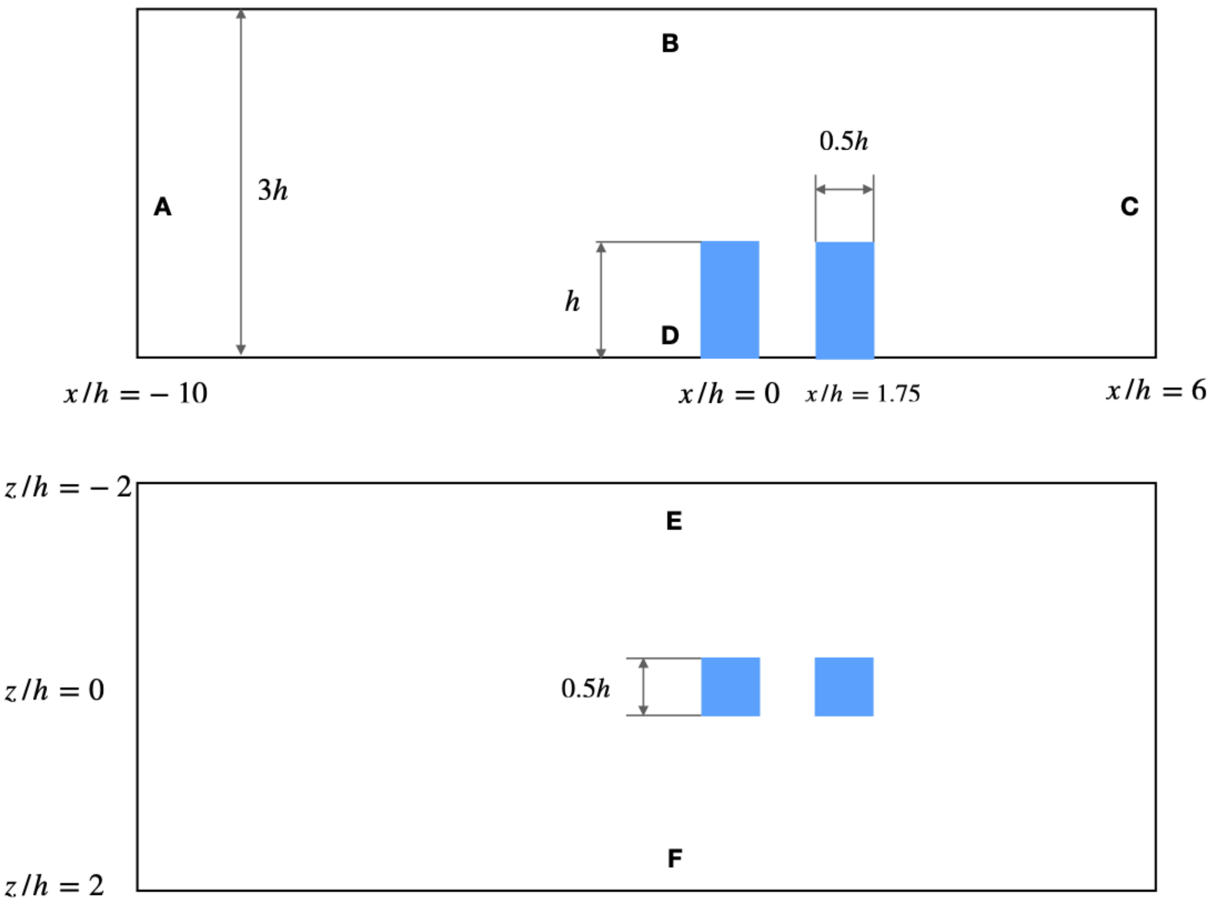


Figure 4.1: Geometrical scheme of the domain

Figure 4.1 shows a schematic representation of the xy and xz planes of the domain. As it can be seen in the figure, the origin of the reference frame is centred with the first obstacle such that the array of obstacle develops in the positive side of the domain. Note that the letters **A** to **F** name the different faces of the domain. Those will be recalled later on during the boundary conditions definition. The dimensions shown in Figure 4.1 can be expressed following the meshing platform nomenclature, i.e. using the quantities shown in Figure 3.1. Table 4.1 gathers the parameters introduced in the meshing platform following the nomenclature defined in Figure 3.1.

Parameter	**Value**
$R\_Lin\_h \times h$	10
$R\_L\_h \times h$	1.75
$R\_Lout\_h \times h$	4.25
h	1
$R\_H\_h \times h$	3
$R\_wx\_h \times h$	0.5
$R\_wz\_h \times h$	0.5
$R\_W\_h \times h$	4
I	1
J	2

Table 4.1: Geometrical setup parameters in the meshing platform nomenclature

From the design perspective some guidelines need to be considered to ensure the well functioning of the simulation.
Firstly, the distance from the inflow to the the first obstacle must be large enough to ensure that the flow is fully turbulent before interacting with the obstacle. Recall that turbulence is induced with a tripping force at the surroundings of the inflow. In this way, the inflow Reynolds number is low since the transition to turbulence is ensured by the tripping force. Hence, a certain distance is needed such that the flow fully transition to turbulence. From the combination of the preliminary simulation and previous experience, the inflow-obstacle distance was set to ten. Later on the characteristics of the turbulent boundary-layer (TBL) are assessed. A similar reasoning can be applied to the outflow of the domain. In fact, the outflow must be separated from the obstacle area such that the boundary condition does not affect the flow in the area of interest, i.e. the obstacle region.
Secondly, the vertical upper boundary of the domain, i.e. face B in Figure 4.1, needs to have a sufficient separation such that the flow structures are not affect by the boundary. It is easy to see that separation between boundary and region of interest is critical in any of the directions considered. Nevertheless, in the case of the upper vertical boundary it is particularly important since many of the structures of interest occur in the upper section of the obstacle. In the preliminary simulation, the vertical dimension of the domain was set to $2h$ which resulted insufficient to ensure that the flow would not be affected by the location of the upper boundary. That is the reason why it was raised to $3h$ in the present simulation.
Furthermore, the obstacle separation is also a critical parameter both in the meshing and simulation process. Recall that the mesh in the near-obstacle region is built following a progressive scheme in which the element size is expanded in the wall normal direction from the smaller size at the wall to the bigger size. Hence, ones needs to ensure that the distance between obstacles is sufficient to allow the mesh to expand. Failing the fulfilment of this conditions interrupts the meshing process. In addition, from Oke [22] we know that the separation of the obstacle, *ceteris paribus*, determines the flow regime experienced. In our case flow regime is not a major issue since the objective of

the present work is the development of the technique rather than the actual study of urban flow regimes. Nevertheless, the parameters need to be considered within the frame of flow behaviour.

In conclusion, combining the experience extracted from the preliminary simulation with previous knowledge, the dimensions of the case were set to fulfil as much as possible the well-functioning criteria of the simulation. Note that despite the preliminary simulation one can only check the structures once the simulated has started, thus the verification of the afore-stated criteria will be introduced in the postprocessing section.

4.2.2 Mesh design

The second part in the creation of the meshed domain consists on defining the meshing parameters. In §3.1.1.1 we introduced the meshing process applied in the present work and in particular how it was designed using a set of parameters. Mesh design was actually one of the mos time consuming parts in the preprocessing operation. In fact, those parameters determine the resolution of the mesh and thus have a direct impact on the solution's representation. In this way, the design stage consist on several design-test loops in which the mesh is generated with a certain resolution and than run during a small period of time. Then, mesh resolution is analysed and corrected. The difficulty lies in the optimal selection of the resolution since it has to be high enough to properly represent the quantities while keeping computational cost constrained. In our case the design process took several iterations loops balancing the required resolution with computational cost. Note that others factors such as the order of the GLL scheme have an influence on resolution and computational cost. Nevertheless, at this stage we only focus on the element scale. Recall the parameters presented in Table 3.1. The resulting input mesh parameters are synthesised in Table 4.2 for both the preliminary and final simulation.

Parameter	Preliminary simulation	Final simulation
$dxmax$	0.25	0.25
$dymax$	0.4	0.15
$dzmax$	0.2	0.15
$dmin$	0.02	0.006
$domax$	0.2	0.05
r	1.5	1.13

Table 4.2: Meshing parameters for the preliminary and final simulations

By inspection one can see that the final mesh has been refined in every direction, with the exception of the streamwise direction. In particular one of the key areas is the refinement zone in the near-obstacle region. This particular area had to be expanded and refined from the preliminary simulation to obtain a good representation of the turbulent structures. This expansion and refinement was achieved by means of the expansion rate and element sizes in the near-obstacle region. Focusing on the spanwise direction one can see that the same element size is maintained. This fact empathises the utility of the design-test approach and in particular the importance of starting with a coarser mesh. By carefully analysing resolution in every direction one can see in which directions the element size is adequate. In this way, computational cost is reduced by approximation the optimal resolution in each direction. Furthermore, the refinement within the mesh, i.e. the progressive element size in the near-obstacle region, is also a cost-efficiency strategy which allows to have a coarser mesh in the areas outside the region of study.

Figure 3.2 shows a two-dimensional cut in the z-direction of the mesh defined in the final simulation. Figures 4.2 and 4.3 show two-dimensional cuts in the streamwise and heightwise directions.

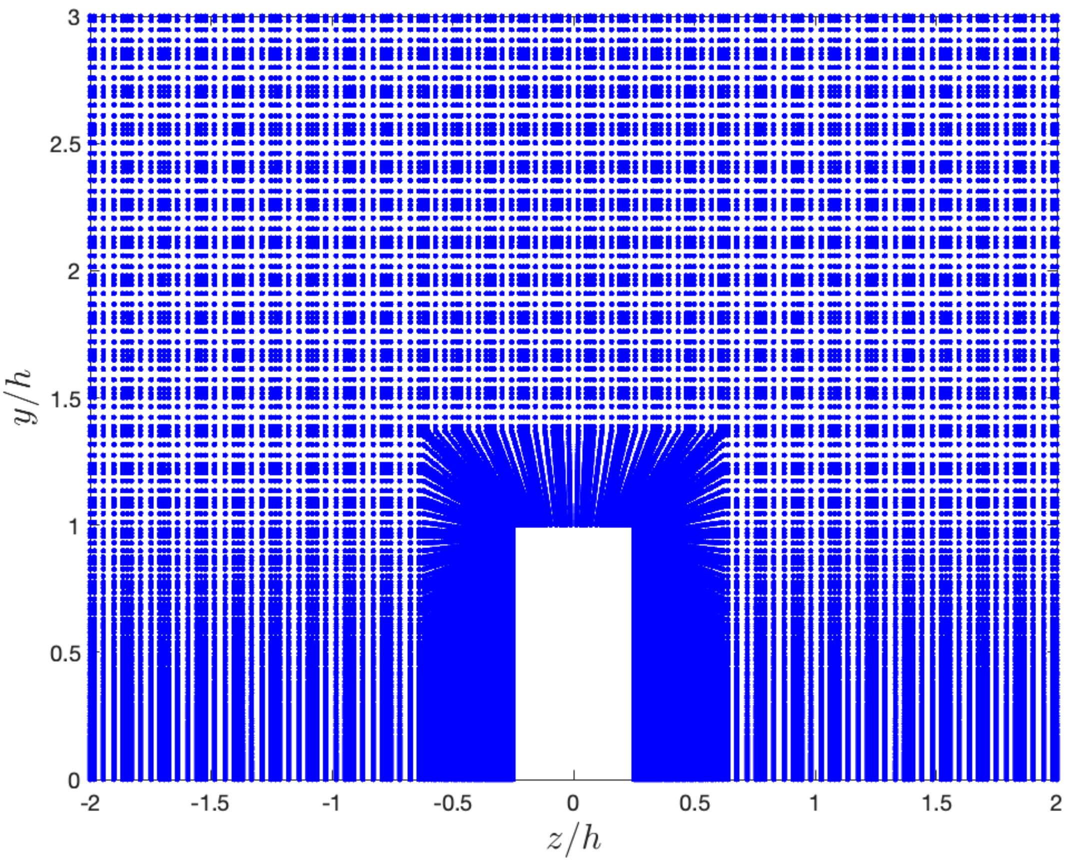


Figure 4.2: Two-dimensional cut at plane $x/h = 1.75$ for the final simulation mesh

By inspection of Figure 4.2 one can observe how the refinement process works in the near-obstacle region. In fact, the refinement is applied in the wall-normal direction in every side of the obstacle, creating thus a refine volume around the obstacle. It is particularly important to have a fine mesh at the edges and upper horizontal face of the obstacles since it is precisely in those areas where flow structures need to be carefully represented. As far as resolution is concerned, a proper resolution study will be presented later on. Nevertheless, by simple inspection one can see that grid-point [1] concentration is significantly higher in the near-obstacle region so it is resolution. Furthermore, the mesh is generally progressive over the lower vertical wall face, i.e. in the $y/h \approx 0$ zone. This resolution requirement is set to ensure that the turbulent boundary layer (TBL) is properly solved in the pre-obstacle region. During the resolution analysis it is precisely this region that is analysed.

[1]Note that it is important to distinguish between element and grid-points. Recall that spectral-element method are based on a series of elements that contain a distribution of points within. This is particularly critical in the meshing and resolution analysis processes. In fact, in the meshing process only element resolution is defined. However, the elements contain a GLL distribution of seven points, thus having a finer resolution. Both the graphical representation of the mesh and the resolution analysis grid-points are considered, i.e. the GLL points in each of the elements.

Figure 4.3 shows the two-dimensional cut at the lower horizontal plane. Again, the mesh refinement is here visible in the wall-normal direction.

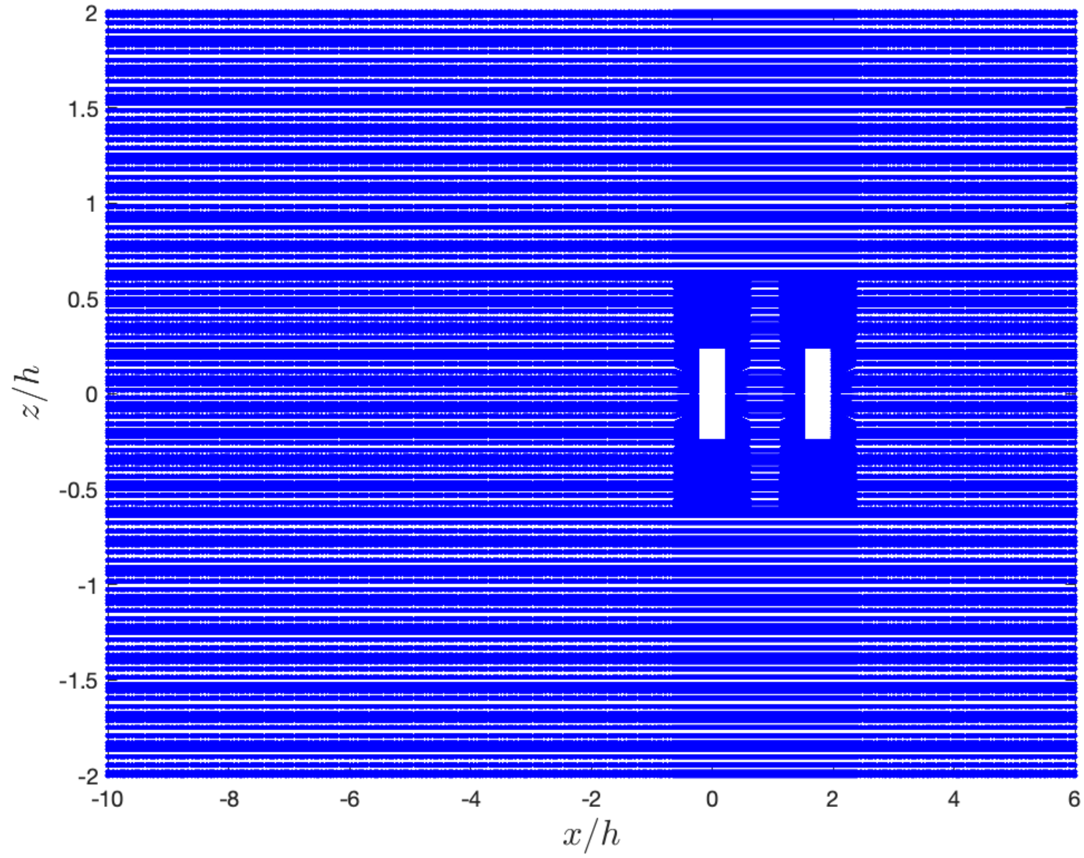


Figure 4.3: Two-dimensional cut at plane $y/h = 0$ for the final simulation mesh

4.2.3 Boundary conditions

To close the setup presentation the last main topic to assess is the definition of boundary conditions (BC). Boundary conditions are a mathematical requirement of any PDE such that the solution scheme can be run. In §3.1.2 we explained how BC are implemented in the Nek5000 setup and which are the particular routines that make use of them. Recall that boundary conditions are set by assigning a letter code to the element faces. This code was presented in Table 3.4.

Both the preliminary and final case were design with using the x-direction as the streamwise direction, having thus the inflow at $x/h = -10$ and the outflow at $x/h = 6$. In Figure 4.1 we named the different faces in the domain using letters A to F. This notation will be applied in the following lines to link boundary conditions with the domain faces. Table 4.3 gathers the BC applied in the final simulation at the boundaries of the domain. In addition to the BC listed in the table one needs to add the faces of the obstacle which are obviously set as walls.

Face	BC	Description
A	v	This BC is applied at the inflow by means of the Dirichlet condition. In particular, a Blasius profile is applied.
B	ON	This BC is a combination of an outflow and Dirichlet condition. In fact, a zero stress condition (O) is applied in the normal direction to the boundary whereas a Dirichlet condition is applied in the other two.
C	O	The stabilised Dong's [5] outflow condition is applied.
D	W	Wall condition is considered at the bottom horizontal plane.
E and F	P	Periodic conditions are applied in the spanwise boundaries.

Table 4.3: Boundary conditions applied in the final simulation

Unlike mesh design, BCs were not significantly changed during the final design process, only testing and debugging was done on the less standard conditions, i.e. Dong's BC. In fact BCs are constrained by the physics of the problem and the stability of the numerical method. From the physical perspective, the idea is to simulate an open urban environment. In this way, the ON condition was set at face B in order to simulate an open boundary. Periodicity is set in the spanwise direction to encure than the spanwise boundaries don't affect te flow. Finally, Dong's condition is applied precisely to set an outflow without adverse upstream effects. This last condition is precisely important from the stability's perspective.

The outflow condition here considered provides a stabilised outflow. Dong et al. [5] presented a method which allows to maximise the truncation at the outflow boundary without inducing adverse effects in the flow. In fact, the method allows "the influx of kinetic energy into the domain through the outflow boundaries" (Dong et al. [5]) while preventing the uncontrolled growth in the energy that is introduced in the domain. In this way, the method ensures the stability of the simulation during the afore-mentioned operation. As far as the algorithm is concerned, it "is developed on top of a rotational velocity-correction type strategy to de-couple the pressure and velocity computation" (Dong et al. [5]). In conclusion, Dong's outflow condition provides a better performance than a standard outflow condition, allowing a reduction in the outflow distance and thus reducing computational cost.

4.3 Simulation run

In §3.2.1 we saw how the solving process works. In particular one might recall Figure 3.5 which gathers the different stages in the solution process as well a the files that characterise each of those stages. The main objective of the present section is not to describe the solution process extensively considered in §3.2.1, but rather give some information about the particularities of the final simulation run.

The present simulation is run using a $\mathbb{P}_N - \mathbb{P}_{N-2}$ formulation. The element discretisation consists on 205,605 elements which contain a eight GLL points quadrature within the elements. The total number of points ascends to roughly 105 million grid-points. The inflow Reynolds number defined with a unitary characteristic length is $Re_{\delta^*} = 450$, which is given by the laminar Blasius solution. Note that the inflow Reynolds number is low for a turbulent simulation. However, recall that turbulence is induced using a tripping force and thus the inflow Reynolds number can be lower.

The simulation is carried in the Cray XC40 system "Beskow" located in the PDC Centre for

High Performance Computing at KTH Royal Institute of Technology. In total, the system has 67 456 cores and a 156.4 TB primary memory. In the present study, simulations were run using up to 512 cores.

4.4 Postprocessing

As described in §3.3 postprocessing is the main procedure to apply once the simulation starts outputting data. The object in the postprocessing here considered is double. On the one hand, our aim is to check the well functioning of the simulation process at different levels, i.e. resolution, statistic etc. On the other hand some quantities are obtained to better characterise the flow, in particular we will focus on the boundary layer analysis. As far as the implementation of postprocessing techniques is concerned, one might recall §3.3.2 where the different operation were described.

The following lines will focus on presenting the particularities and results obtained in the postprocessing of the final simulation. Firstly, we will discuss the setup of the interpolation mesh which is the foundation over which the statistic and boundary layer quantities are obtained. Secondly, the resolution analysis[2]will be presented. Finally the boundary layer will be analysed using the interpolated results.

4.4.1 Interpolation mesh design

The interpolation mesh is the base over which the statistic solution is obtained. In §3.3.2.2 we saw how the afore-stated mesh is obtained and it is integrated in the postprocessing procedure. The current interpolation mesh was designed to cover only the pre-obstacle region since the analysis will be focusing on that part of the domain.

The interpolation mesh considered in the final simulation consist 700,000 points in total. The streamwise and spanwise directions consider a linear distribution whereas in the heightwise direction the grid-points are distributed progressively. In fact, in the spanwise direction a logarithmic function is applied to concentrate the grid-points in the lower part of the domain. In §3.1.3.2 we saw how the boundary quantities are computed at the wall, thus a higher concentration of points is needed in that area. In Figure 4.4 one can appreciate the grid-points distribution in the streamwise and heightwise direction. It is easy to see to see that the concentration of points is significantly more important in the lower region of the domain, precisely where the boundary layer parameters will be computed. The streamwise coordinate is truncated at $x/h = 0$ since only the pre-obstacle region is consider. However, due to the nature of distribution there is a set of points at $x/h = 0$ which coincides with the first obstacle's centreline. Nevertheless, the afore-stated fact is not problematic as the evaluation range of the quantities can be reduced later on. The same way, Figure 4.5

[2]Note the resolution analysis here presented was introduced in §3.1.3 under the name of "*a priori* analysis". In fact, this analysis applied *a priori* using the preliminary simulation solution but also after the final simulation is run. As it was discussed in §3.1.3, the analysis using preliminary solution helps in the design of the final simulation. Nevertheless, the analysis here presented is based on the final simulation results to serve as validation of the design introduced in §4.1

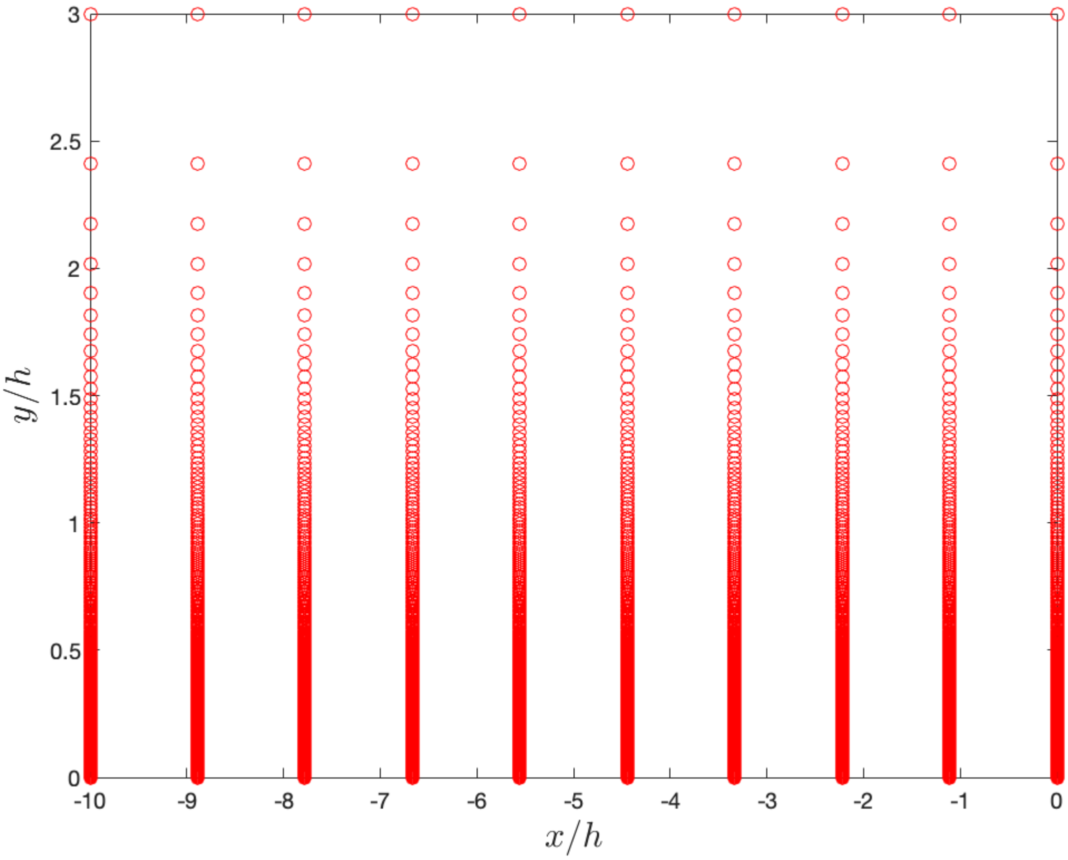


Figure 4.4: Two-dimensional cut of the interpolation mesh at $z/h = 2$

shows the distribution in the spanwise direction. By inspection, it can be seen that the number of points in the spanwise direction is significantly higher than in the streamwise direction. This can be explained by the analytical method applied. In fact, later on we will see that both the resolution and boundary layer metrics are compute in two-dimensional slices at the spanwise coordinate of the mesh. Once all the slices have been computed, the quantities are averaged over the spanwise coordinate. By considering a sufficient amount of points in the spanwise direction one ensures that a significant amount of the data is taken into account during the averaging process.

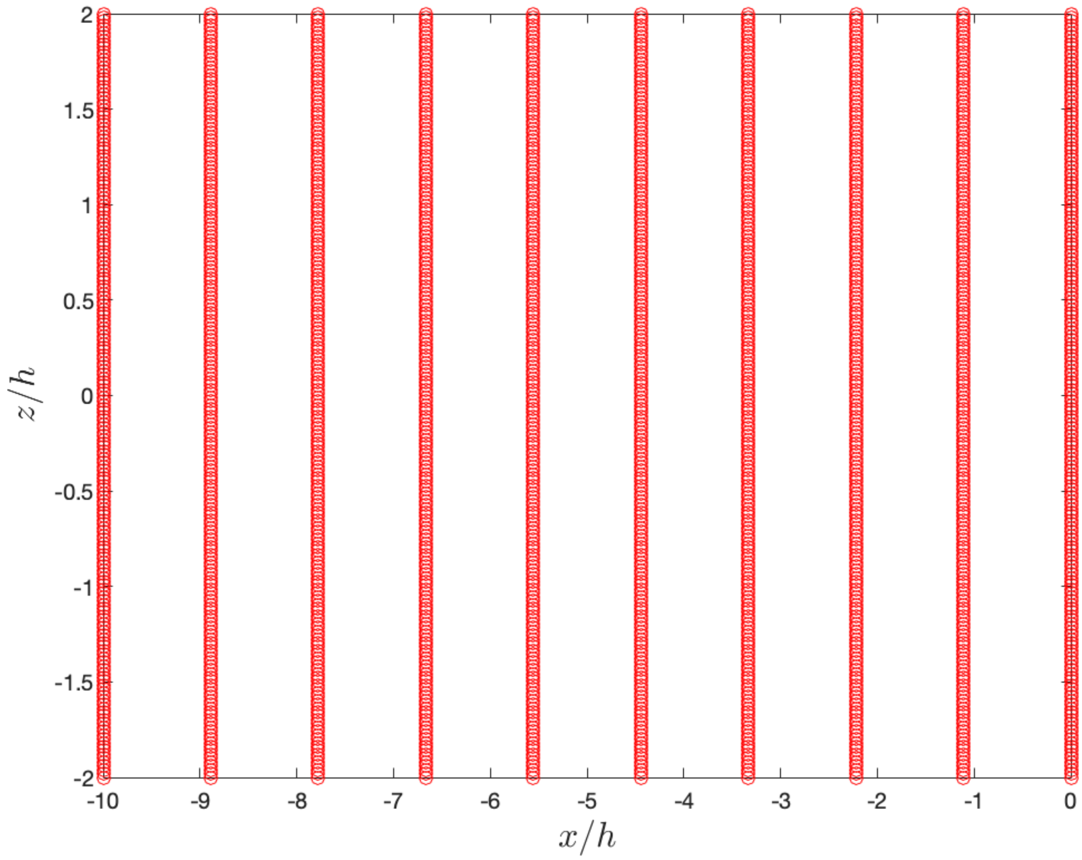


Figure 4.5: Two-dimensional cut of the interpolation mesh at $y/h = 1$

4.4.2 Resolution analysis

Resolution analysis was introduced in §3.1.3.3 in terms of normalised grid spacing and Kolmogorov scales. As it was previously discussed the resolution analysis was applied in both the preliminary and the final simulations. The present section will be dedicated to the presentation and discussion of resolution for the final mesh. In particular, we will focus on the pre-obstacle region using normalised grid spacing quantities.The procedure is applied in every plane in the spanwise direction such the averaged quantities can be obtained. Note that the process for the final case differs from both the preliminary simulation and the procedure previously introduced. Due to the vast amount of data to be processed by the MATLAB routines, the mesh applied during the resolution analysis was considered using a $N = 5$ GLL points distribution instead of an $N = 7$, which is the mesh used in the solution process. In this way the amount of data input in the MATLAB routines is significantly lower. Then, an additional MATLAB routine is implemented in order to change the points within the elements with the GLL quadrature mapped over the element. Using the method described above one can obtain the actual mesh resolution without having to directly process the vast amount of data that form the complete solution mesh.

Figure 4.6 shows the spatial evolution of the grid-spacing in the streamwise direction using the normalised quantities introduced in §3.1.3.3. By inspection, one can see that the curve shown

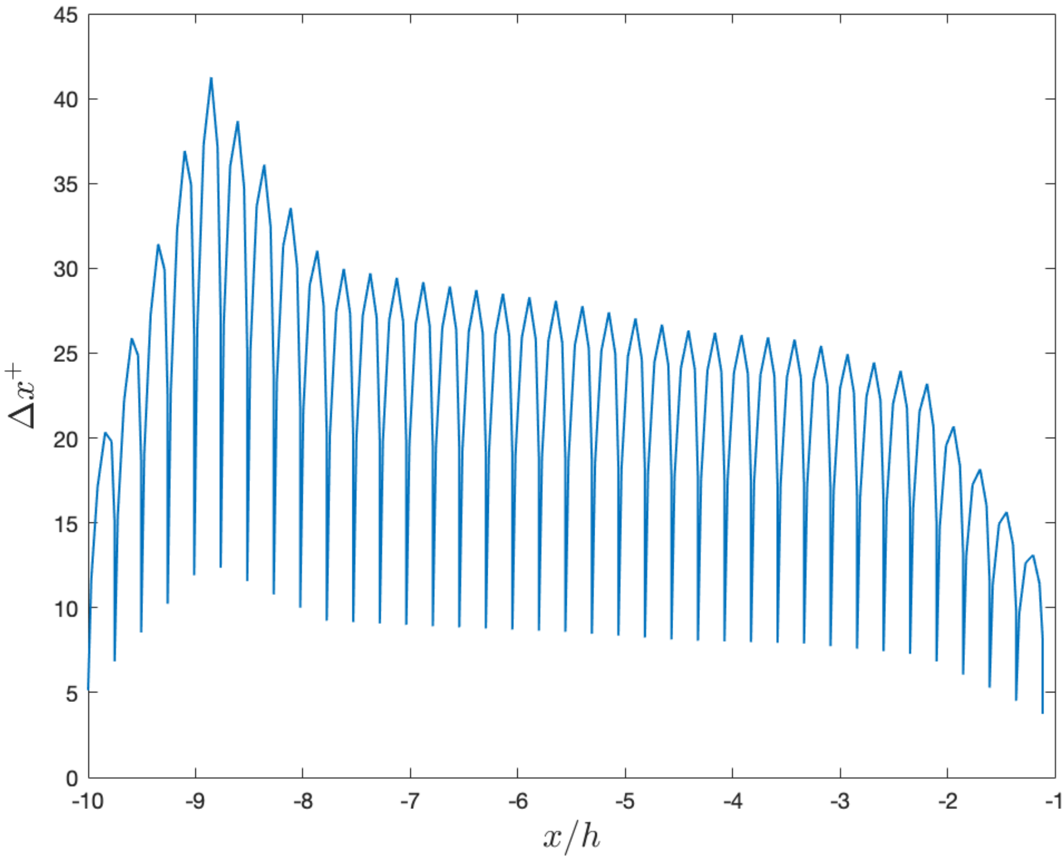


Figure 4.6: z–averaged normalised x–spacing as a function of x/h

in the figure presents some oscillations over the streamwise direction. This behaviour is due to the GLL distribution in the x–direction and thus the results are consistent with our expectations. Note that normalised spacing is only considered for $-10 \leq x/h \leq -1$ since we are mainly interested in the pre-obstacle region.

Resolution criteria for a *well-resolved* LES is significantly higher than in a classic LES. Negi et al. [21] state that resolution in the streamwise direction should be $\Delta x^+ < 19$ such that the mesh has the resolution of a *well-resolved* LES. By inspection of Figure 4.6 one can see that the average grid spacing of the elements fulfils this criterion for approximately $x/h > -7$. For $x/h < -7$ one might thinks that the criterion is not satisfy. However, in this region the tripping force is applied. Thus, the flow is affected by the local tripping and it undergoes transition to turbulence, therefore this criterion does not apply to that part of the flow. In fact, at the immediate surroundings of the obstacle, the peak resolution is roughly $\Delta x^+ = 15$ which is finer than what was proposed by Negi et al. [21]. In conclusion, the mesh resolution in the streamwise direction fulfils the *well-resolved* LES criteria and thus it is a satisfactory mesh in the considered dimension.

The same procedure is applied in the heightwise direction. However, this time the spacing is constant through the streamwise direction. The point considered corresponds to the first non-null coordinate in the y–direction, i.e. the space between the first GLL point and the wall. Figure 4.7

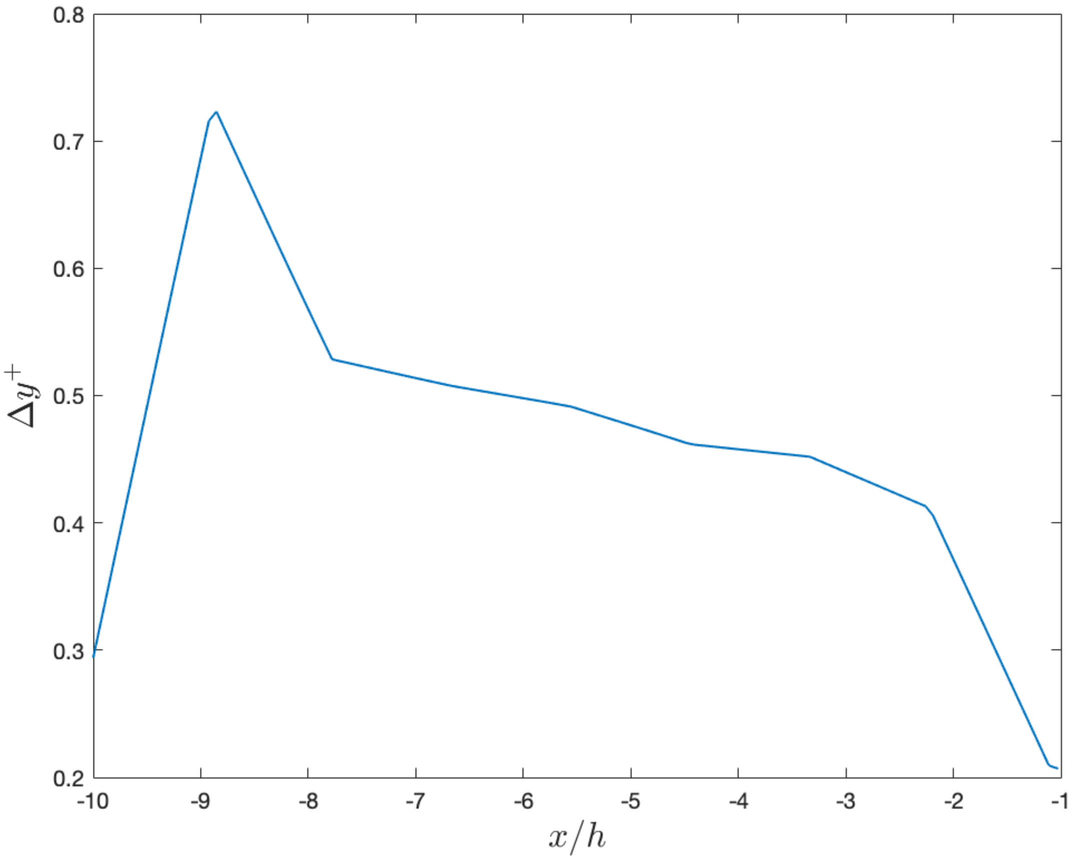


Figure 4.7: z–averaged normalised y–spacing as a function of x/h

shows the spatial evolution of the normalised grid-spacing in the heightwise direction averaged in the spanwise coordinate.

Once again, Negi et al. [21] state that $\Delta y^+ < 0.65$ such that resolution is adequate. By inspection of Figure 4.7, one can see that for $x/h > -7$, the heightwise spacing is below $\Delta y^+ = 0.75$ thus the afore-stated condition is perfectly fulfilled. In addition, approaching the near-obstacle region resolution is significantly improved, reaching $\Delta y^+ \approx 0.2$ at $x/h = -1$. Appraising the obtained resolution, one can see that the grid-spacing in the wall normal direction is significantly lower than the minimum threshold described by Negi et al. [21]. In conclusion, the resolution in the heightwise direction is adequate and thus one can expect to have a very good representation of the quantities.

The resolution is also analysed in the spanwise direction. This time no averaging is applied and the maximum and minimum grid-spacing are considered. Once again, grid-space is constant along the streamwise direction. This time the criterion is $\Delta z^+ < 9$. However, the GLL distribution in the spanwise direction is not uniform. Hence, both the minimum and maximum grid-space have to be computed.

By inspection of Figure 4.8 one can see for the vast majority of the domain, i.e. approximately $x/h > -7$, $\Delta z^+ < 9$. Hence the criterion is once again verified.

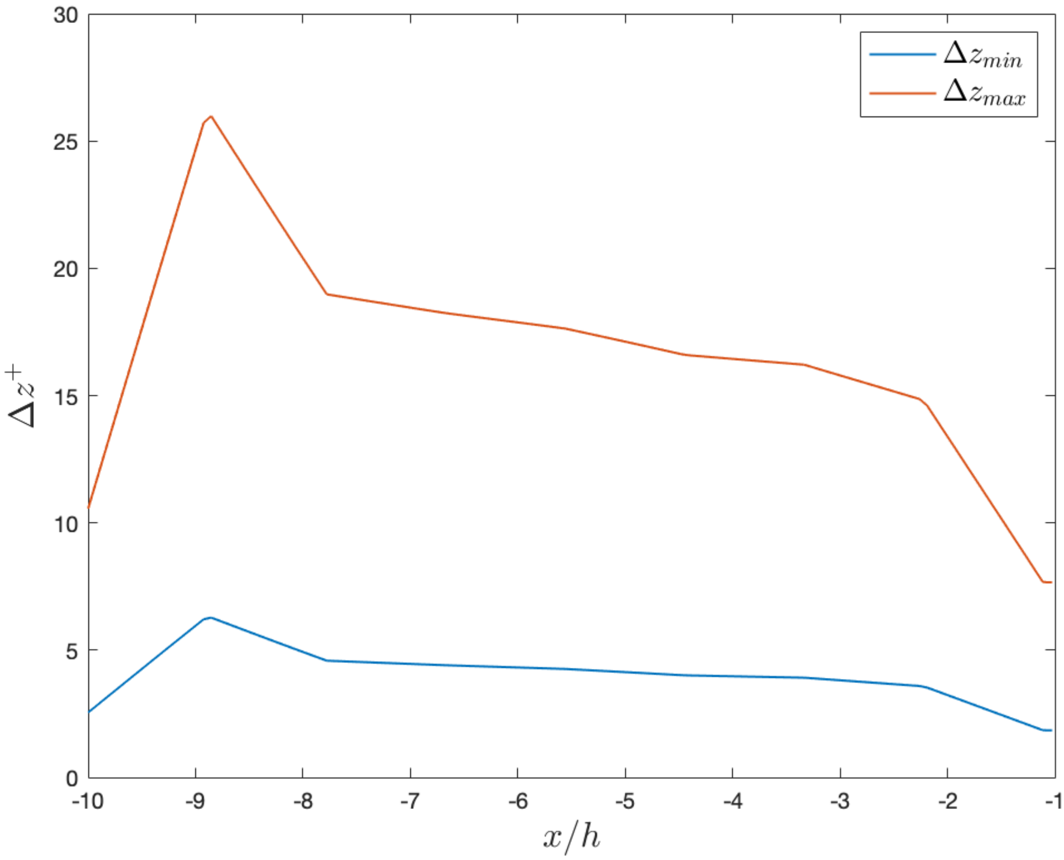


Figure 4.8: Normalised z–spacing as a function of x/h for both Δz_{min} and Δz_{max}

The mesh resolution analysis using normalised coordinates shows that in principle, the mesh should be able to represent all the turbulent quantities with a very good resolution. In addition, following the criteria presented by Negi et al. [21], we can state that the simulation has the resolution of a well-resolved LES. Furthermore, we also compared the grid resolution with the local Kolmogorov scale in the region between the obstacles, and we observed that the h_η parameter defined in Equation 3.13 was always below 9. This further justifies the adequacy of the considered resolution, also around the obstacles.

Now that the normalised resolution analysis has been presented, we will assess resolution focusing on the statistical quantities representation. In particular, the following lines will be dedicated to the analysis of the mean velocity and Reynolds-stress evolution with respect to the heightwise coordinate. Those quantities are obtained from the statistics toolbox introduced in §3.3.2 and are then compared with the ZPG DNS data presented by Schlatter and Örlü [31]. All the quantities will be evaluated in normalised coordinate using the same characteristic quantities than during the resolution analysis. Note that only the quantities are only evaluated in $-8 < x/h < -2$ since it is in that range where a turbulent boundary-layer is (TBL) is found.

Moving closer to the inflow, the mean velocity profiles are either laminar or transitional. Thus

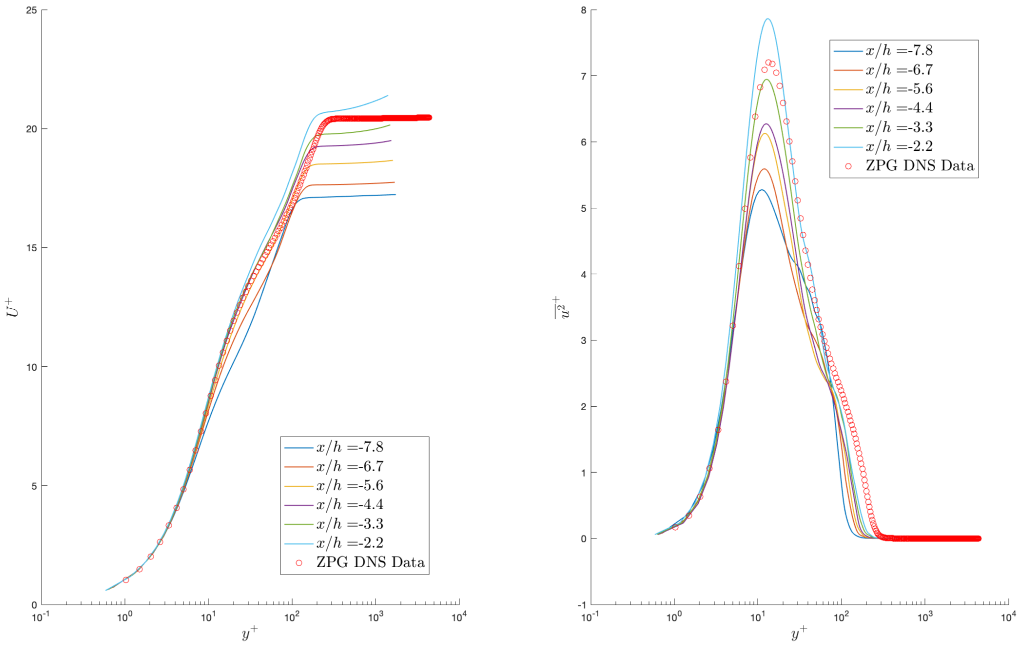


Figure 4.9: Comparison of ZPG DNS by Schlatter and Örlü [31] with the LES resolution of the final simulation. (Left) Normalised mean velocity and first component of the Reynolds-stress tensor (right).

the present analysis is not valid in that region. In fact, theses observations are consistent with the methodology applied to induce the transition to turbulence, i.e. the tripping method, since the force is applied after those points. On the contrary, for $x/h > -2$ the flow starts being influenced by the obstacle and thus the TBL is disturbed. By inspection of Figure 4.9, one can see that in the vast majority of scales, both the mean velocity and the first component of the Reynolds-stress tensor match the data obtained with the DNS. This indicates that the TBL in $-8 < x/h < -2$ is a canonical TBL and thus that the resolution in the region is sufficient for the correct solving and representation of the turbulence.

In conclusion, resolution was appraised using two distinct methodologies. Firstly, we saw how the normalised grid-spacing varied with respect to the streamwise direction and we evaluated whether it fulfilled the *well-resolved* LES criteria. Secondly, the evolution of the mean velocity and the first component of the Reynolds-stress tensor was presented. In this case, the obtained results were compared with the data extracted from a DNS study. In both cases, it was concluded that the resolution was more than sufficient to represent the quantities that characterise the flow. Hence, with the afore-presented discussion the mesh is verified.

4.4.3 Boundary-layer analysis

The last part of the postprocessing will focus on the appraisal of the incoming TBL. The following lines will present the results related to the boundary-layer quantities obtained during the final simulation. As far as theory and implementation are concerned the present analysis is founded on the processes described in §3.1.3.2. The appraisal here reported will be structured over two fundamental axes. First of all, the flow conditions in the pre-obstacle region will be evaluated by means of the Reynolds number. Then, we will focus on the development of the TBL assessing several metrics that describe the boundary-layer. The quantities are computed for each over each spanwise coordinate of the interpolation mesh such as averaging can be applied.

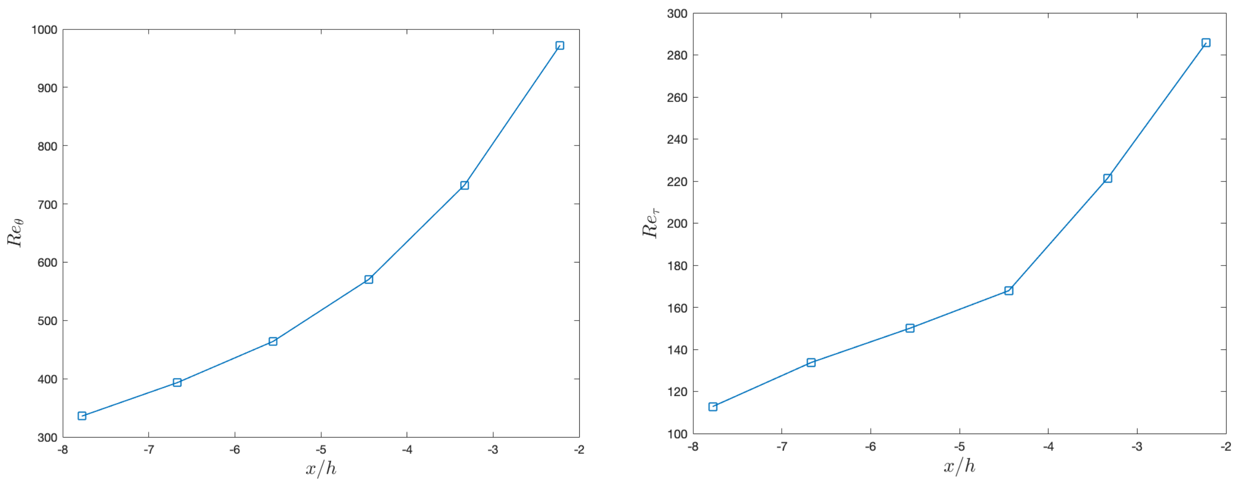


Figure 4.10: z-averaged streamwise evolution of (left) the Reynolds number based on the momentum thickness Re_θ and friction Reynolds number Re_τ(right).

Figure 4.10 shows the streamwise evolution of Reynolds number based on the momentum thickness Re_θ and the friction Reynolds number Re_τ. By inspection, it is easy to see that both quantities grow over the streamwise coordinate. In both cases, the maximum is found at $x/h = -2$. This shows that the TBL thickness grows with the streamwise coordinate up to the surrounding of the obstacle. In fact, the figures are considered up to $x/h \approx 2$ since for greater values, the TBL is disturbed by the obstacle. In addition, the curves' trendline shown in Figure 4.10 are consistent with the friction reduction with the streamline. As friction is reduced it is consistent to find a growing tendency in the TBL. Regarding Re_τ, an abrupt change increase is found at $x/h \approx -3$. The afore-stated increase can be explained by the tendency of the boundary layer thickness δ_{99} which happens to raise its rate of increase precisely at $x/h \approx -3$.

Figure 4.11 presents the streamwise evolution of the boundary-layer thickness computed at the 99% of the free-stream velocity and the friction coefficient. The interpretation of those figures is directly related with the results reported in Figure 4.10. The boundary-layer thickness increases with the free-stream. The curve presents a raise of the growing rate at $x/h \approx -3$, which is consistent with the streamwise evolution of the Reynolds numbers. In fact, when approaching the obstacle, i.e. $x/h \to 0$, the adverse-pressure gradient (APG) induced by the presence of the obstacle tends to make the boundary-layer thickness increase. Those effects can be partially seen in Figure 4.11 at $x/h \approx -2$. Nevertheless, if one keeps approaching the obstacle, the TBL will detach precisely due to the presence of the obstacle. As far as the friction coefficient is concerned, there is an almost

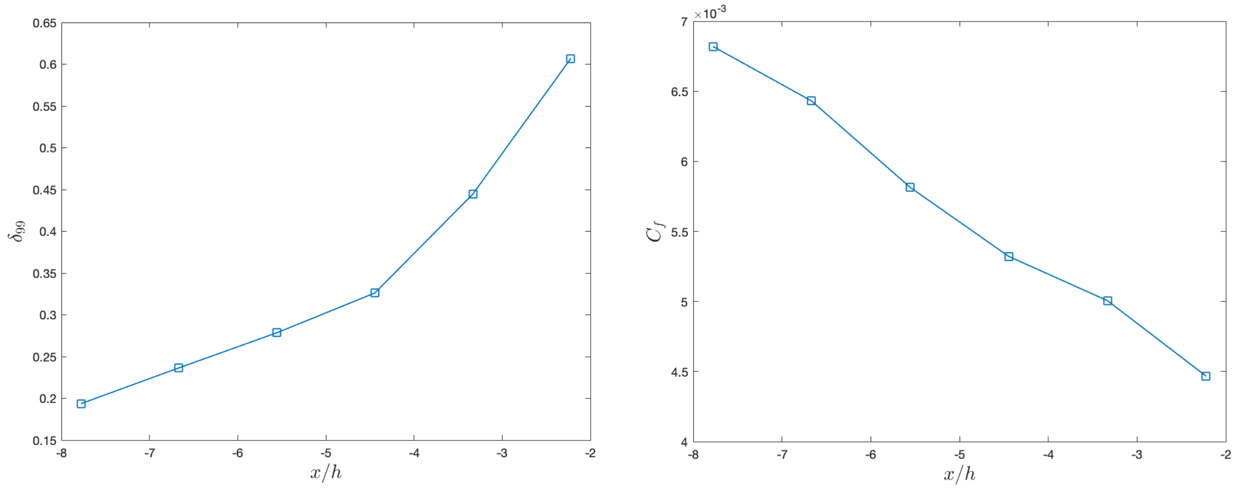


Figure 4.11: z-averaged streamwise evolution of (left) the boundary layer thickness evaluated at 99% of the free-stream velocity δ_{99} and the friction coefficient C_f(right).

linear decreasing tendency in friction over the streamwise direction. Again, this is consistent with the behaviour of the Reynold curves shown in Figure 4.10. In fact, the Reynolds number can be understood as a metric measuring the inertial to viscous forces ration. Friction is directly linked to viscosity, thus an increase in the Reynolds number implies that viscous terms are losing influence, fact that it is supported by the friction coefficient evolution presented in Figure 4.11.

Figure 4.12 shows the evolution of the friction coefficient with respect to the Reynolds number based on the momentum thickness. In this case the results from the LES are compared with the ZPG correlation introduced in §3.1.3.2. In particular the correlation considered is the one presented in Equation 3.9 characterise by the coefficient gathered in Table 3.7. As far as the evolution of the friction coefficient is concerned, the obtained results are consistent with the conclusions stated in the previous lines, i.e. the friction coefficient and the Reynolds number are inversely related. Comparing the results with the correlation data, it can be seen that both curves have a very similar decreasing trendline. However, some discrepancy is found between correlation and simulation. This mismatch can be explained by the fact that the LES is implemented with a low Reynolds number. In fact, as reported by Vinuesa et al. [39], the ZPG correlation was built using cases with Reynolds numbers in the range of $1000 < Re_\theta < 4060$. From Figure 4.10 we saw that in our simulation $Re_\theta < 1000$ in the whole region considered. Hence, such a small discrepancy implies that the condition in the pre-obstacle areas are excellent. In fact, the discrepancy between both datasets is small.

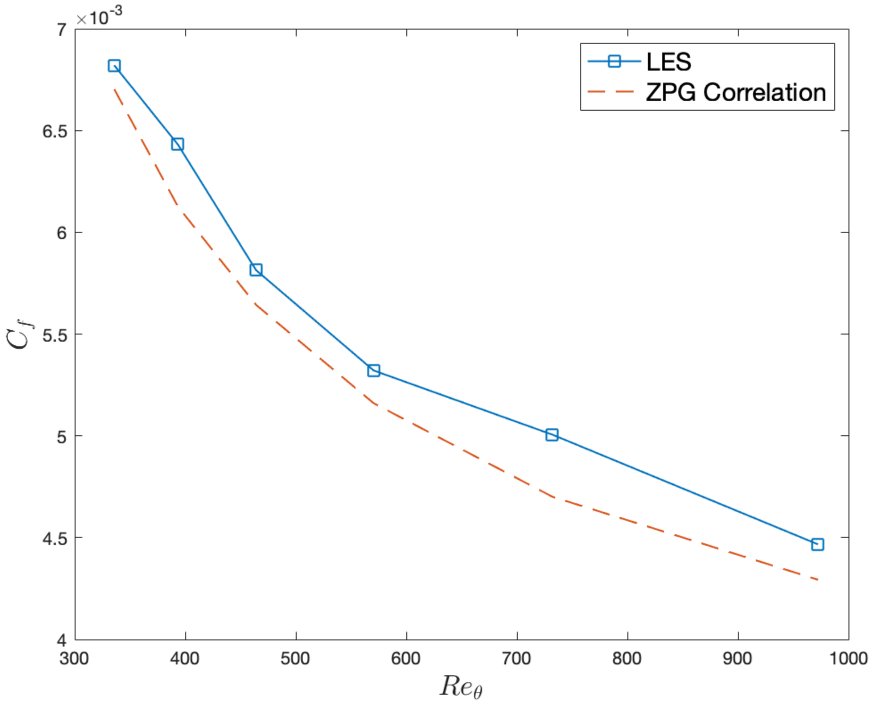


Figure 4.12: z-averaged evolution of the friction coefficient C_f with the Reynolds number based on the momentum thickness Re_θ.

4.5 Results and discussion

The present section will be dedicated to the analysis of the results obtained during the simulation. Contrary to prior analysis, the full domain is now considered and the attention will be driven towards the obstacles instead of the region that precedes them. As far as quantities are concern, flow statistics are now considered. Note that the statistical quantities here presented are not fully converged. Nevertheless, they provide an approximation to the statistical characterisation of urban turbulent flows. The statistical quantities considered are the result of the averaging process described in §3.3.2 applied from $t = 14.2$ to 29.4. In addition, some instantaneous quantities will also be reported.

The analysis will be here divided in four axes. Firstly, we will briefly describe the flow structures visible instantaneous captures of the simulation. Secondly, we will discuss the mean flow velocity fields treated in two-dimensional slices at constant planes as well as the mean pressure field. , To conclude Reynolds-tensor components will also be assessed.

Before analysing flow statistics, it is interesting to have a general idea of the flow structures that are formed over the obstacles. Figure 4.13 shows vortical structures over the first obstacle using the λ_2 criterion for vortex identification presented by Jeong and Hussain [14] . By inspection one can see that the structures from the incoming TBL interact with the first obstacle forming a fluid layer attached to the edges of the obstacle. This layer is very fast detached forming vortical motion that are then carried to the second obstacle.

This behaviour has many implications on the study of urban flows as the fluid structures arriving to

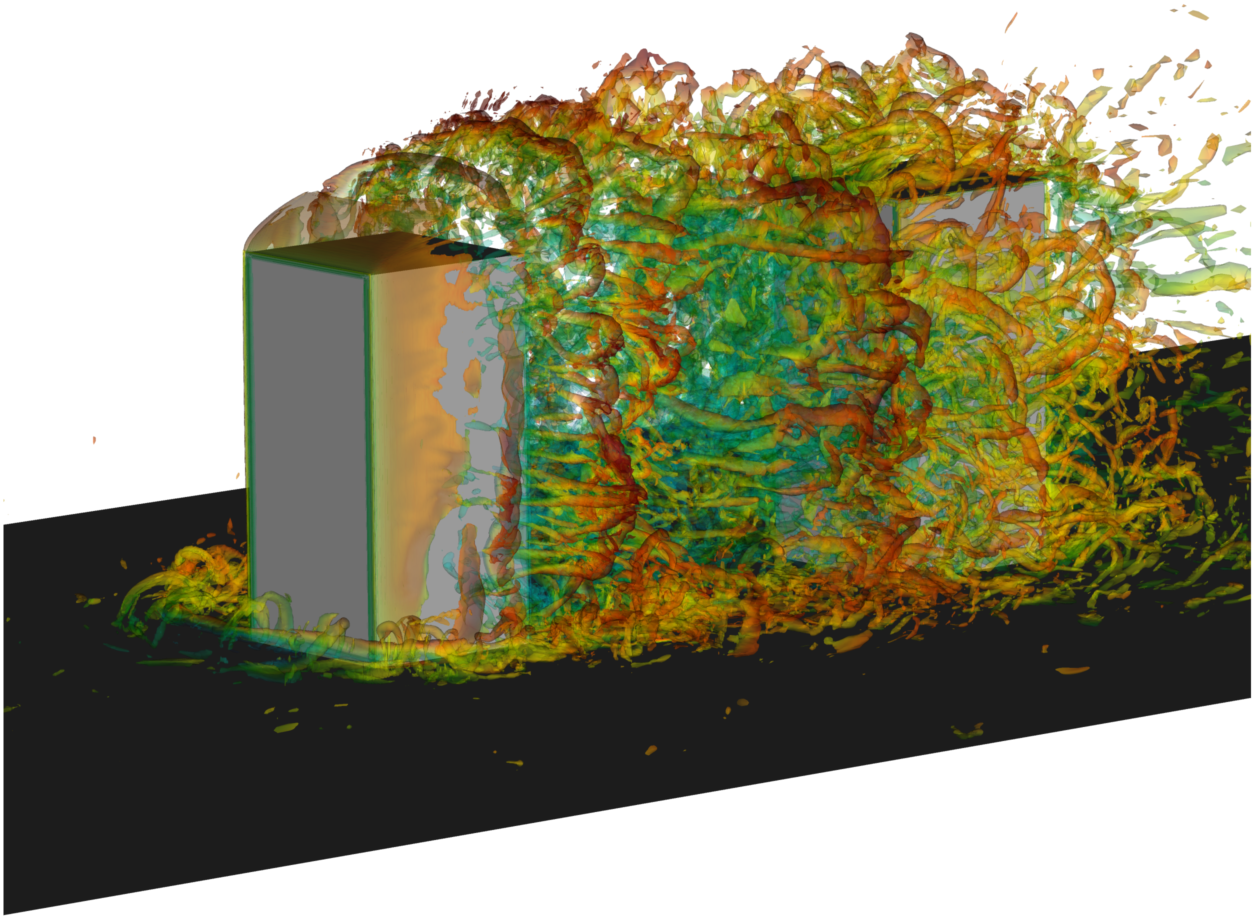


Figure 4.13: Vortical structures identified with the λ_2 method [14] represented using an isosurface at -80 and it is colored by streamwise velocity, where dark blue and red represent low and high velocity, respectively. The isosurface is scaled with both the free-stream velocity U_∞ and the height of the obstacles h, and it is colored by streamwise velocity, where dark blue and red represent low and high velocity, respectively.

the second obstacle and significantly more disturbed than the ones at the first obstacle. Depending on the application considered the afore-described phenomenon might be critical.

4.5.1 Time-averaged velocity fields and mean pressure

Now that the flow behaviour has been introduced, we will focus on the analysis of the time-averaged quantities. The resulting files are then visualised using the VisIt software. Figure 4.14 shows the time-averaged streamwise velocity fields planes $z = 0$ and $y = 0.1$.

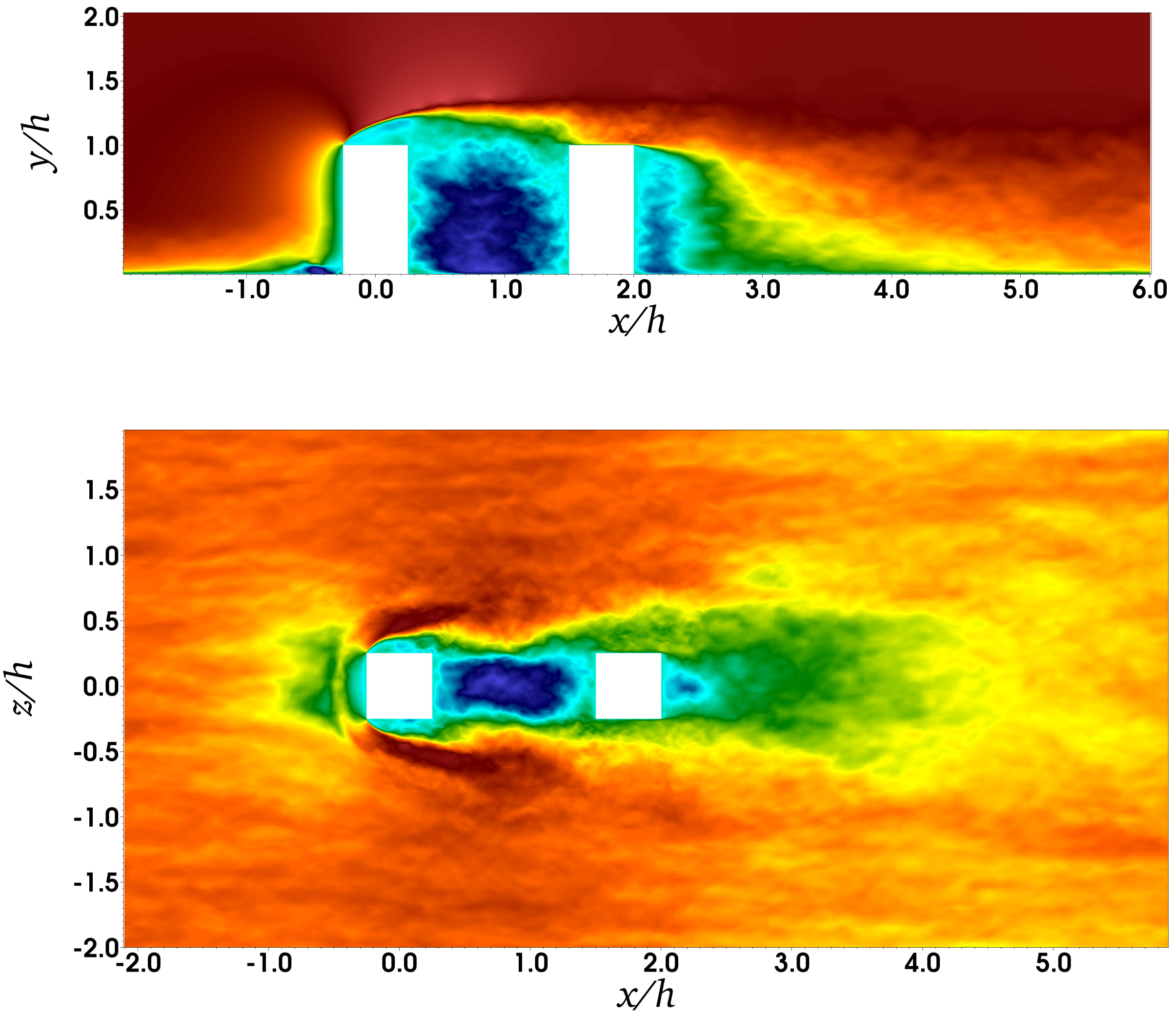
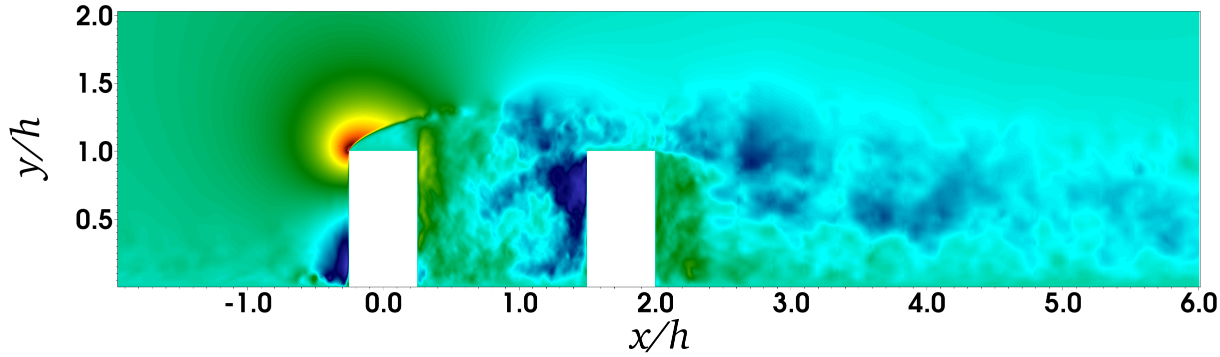


Figure 4.14: Time-averaged streamwise velocity fields (top) U at $z/h = 0$ and $y/h = 0.1$ (bottom). Colour scale starts with blue with $U = -0.54$ and ends with the free-stream velocity in red $U = 1.2$.

By a general inspection that the flow regime, following Oke [22] criterion is skimming flow, the fluid almost doesn't penetrate the space between the obstacles, forming a layer of fluid over them. Focusing on the space between obstacles, one can see that the flow circulates inside the cavity with a slow negative velocity. In fact, due to the flow at the top of the obstacles and the cavity, the flow trapped in the cavity slightly spins remaining inside. This is a typical behaviour in the skimming flow regime. In the bottom part of Figure 4.14 stagnation zones are visible in green. As expected, there is a stagnation region at the surface of the first obstacle front face. In addition the wake appears to maintain it shape over the rest of the domain. This observation can be explained by the flow surrounding the obstacles from the sides. This part of the fluid, contains the wake avoiding its propagation over the sides. In addition, a very small circulation zone is visible in blue after the second obstacle. However, the afore-mentioned region is significantly smaller than the region observed after the first obstacle. This is due both to the absence of an obstacle after the region and by the surrounding flow, which is significantly slower than the flow enveloping the space between the obstacles. Going back the top representation in Figure 4.14, we see that the disturbance does not occupy a vast region in the heightwise direction. Note that the representation was truncated at $y/h = 2$ since the free-stream velocity is attained at $y/h = 1.5$. Nevertheless, the actual domain ends at $y/h = 3$. Once again, this behaviour coincides with the skimming flow regime described by Oke [22]. Furthermore, the flow at the outflow maintains its behaviour thus one can deduced that Dong's outflow conditon is functioning properly.



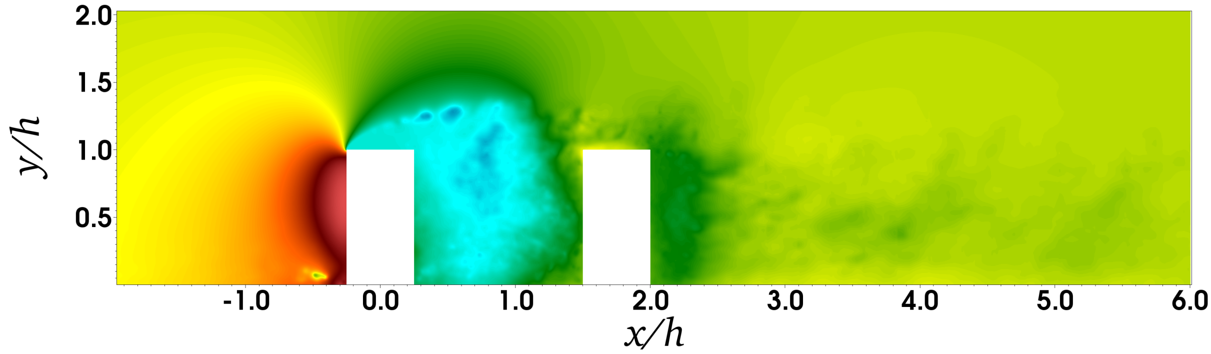


Figure 4.15: Time-averaged streamwise velocity field (top) V and mean pressure P (bottom) at $z/h = 0$. The colour scale range for the velocity (top) starts with $V = -0.5$ in blue and ends with $V = 1$ in red. For the pressure (bottom) the scale starts with $P = -0.6$ in blue and ends with $P = 0.5$ in red.

Figure 4.15 shows the time-averaged heightwise velocity and mean pressure at $z/h = 0$. As far as velocity is concerned, contrary to the streamwise velocity, the overall vertical disturbance in the domain is small. A high velocity small region is visible at the top edge of the first obstacle. This pocket of vertical velocity is the result of the impact of the flow over the edge. Heightwise velocity in the area between obstacles presents lower variation than streamwise velocity. Thus time no circulation area is visible and the area does not behave as a cavity. The heightwise velocity at the wake is negative, thus the fluids is directed towards the bottom of the domain. This might be explained by the overall behaviour of the flow. The heightwise velocity profile coming from the inflow is close to nullity, i.e. the flow is almost fully horizontal. Then, after the encounter with the first obstacle, the flow is abruptly deviated to the upper parts of the domain. However, as it is visible in the bottom graph of Figure 4.15, the gap between obstacles is a low pressure area, thus deviating flow towards the lower part of the domains. In this way, the flow at the gap is mainly

directed towards the down side of the domain, generating a re-circulation zone that joins the wake after the second obstacle. This last observation might be one of the causes why the wake appears to have a greater heightwise velocity.
The mean pressure distribution, is consistent with the behaviour exhibited by the velocity components. A high pressure area is found at the vertical fore-wall of the first obstacle. This area coincides with the stagnation region described before. The area between the obstacle is, as expected, a low-pressure region, which is consistent with the cavity-like behaviour that was described above. In the rest of the domain the pressure remains roughly constant. Again this behaviour matches our expectations as the domain reproduces an open environment.

4.5.2 Time-averaged Reynolds-tensor stresses

The second part of the result analysis will focus on Reynolds stresses. Figure 4.16 gathers the time-averaged normal Reynolds stresses. Focusing on the streamwise direction, one can see the Reynolds stress fields is very small in the whole domain even in area between obstacle. This indicates that the transport is efficient in the streamwise direction in the overall domain. However, locally, the greatest value is found in the gap between the obstacles. This observation is consistent with the conclusion extracted from the velocity analysis, as the region between obstacles contains a re-circulation zone thus reducing the transport in the streamwise direction. The wake exhibits a slight increase in the values however it is still very small, thus the transport is not significantly affected in that area.

In the heightwise normal direction, the Reynolds stress is significantly higher in the area between obstacles, specially at the surface of the second obstacle. This indicates that there is a loss of the transport efficiency in the zone. Interestingly, the area coincides with the negative velocity region shown in Figure 4.15. This overlapping is consistent as the flow is being redirected towards the bottom of the domain, thus generating a larger amount of stress. The wake also exhibits greater values of Reynolds normal stress compared with the streamwise case. Once again this is consistent with the observations of the previous sections.

The spanwise components presents the highest values of the stress tensor. Following a very similar distribution, the higher values concentrate in the area between obstacles. This is not directly aligned with our previous observations. Nevertheless, this behaviour might be explained by the flow surrounding the obstacles. In fact, as it's visible in the bottom representation of Figure 4.16, there are flow structures that surround the obstacles and "close" the space in spanwise direction. Those fluid structures might be raising the quantity. As far as the wake is concerned, interestingly, the values are also significantly higher in this region. This behaviour might be caused by the turbulent structures of the wake. This effect was not visible through the evaluation of the other quantities. Thus, an additional study might be required to draw further conclusions.

Furthermore, the Reynolds shear stress $\overline{uv}$ was also appraised. In Figure 4.17 it can be seen that the greatest value of the Reynolds shear stress are found in the area between obstacles. For the rest of the domain the field values do not exhibit abrupt changes. Nevertheless, a particularly lower value area is found at the top-left corner of the second obstacle. This seems to be the central area of a vortical structures, thus having a lower momentum transport. Furthermore, the Reynolds shear stress is found greater at the wake. This is consistent with our expectations as the wake area exhibits a greater turbulence and thus a greater momentum transport.

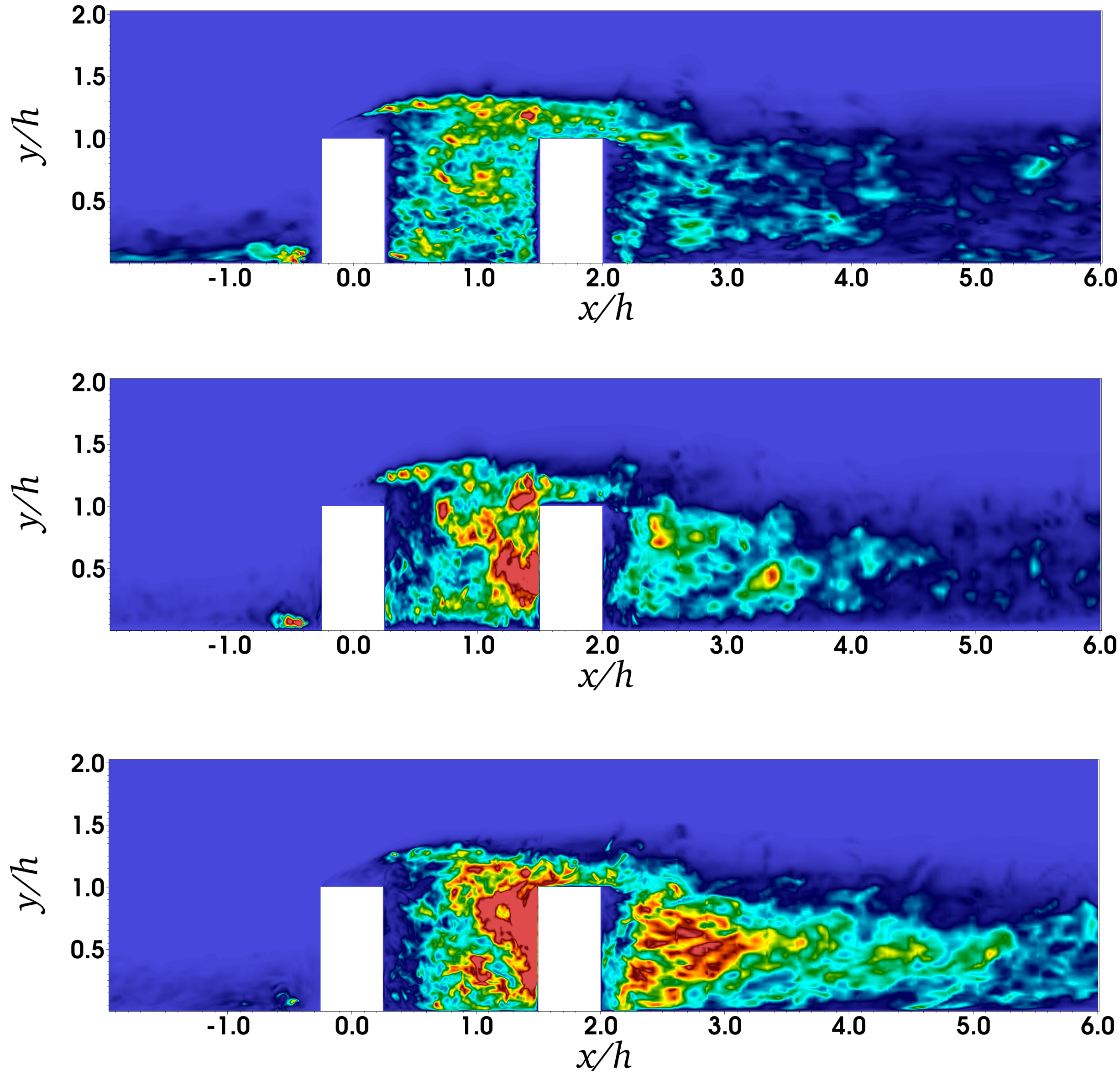


Figure 4.16: Time-averaged Reynolds normal stresses in the (top) streamwise $\overline{u^2}$, (middle) heightwise $\overline{v^2}$ and spanwise (bottom) $\overline{w^2}$ directions. The colour scales starts with a value of 0 in blue and ends in red with a value of 0.1 for the top, middle and bottom graphical representations.

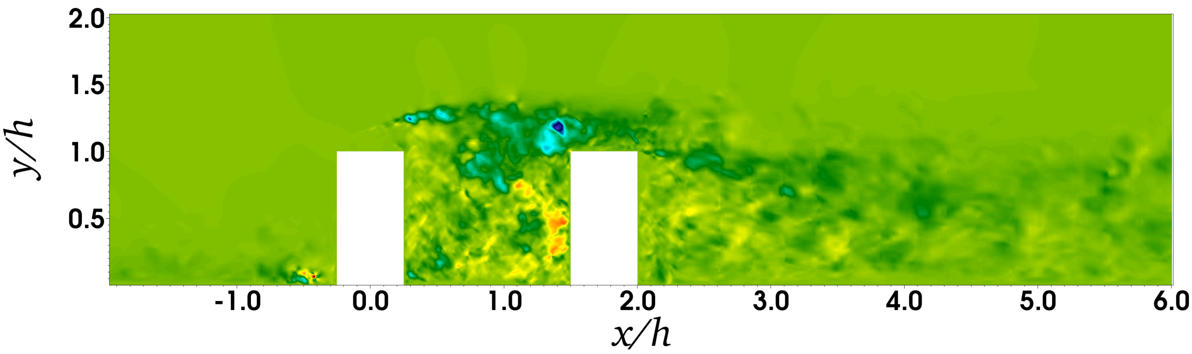


Figure 4.17: Time-averaged Reynolds shear stress $\overline{uv}$. The colour scale starts with $\overline{uv} = -0.1$ and ends in red with $\overline{uv} = 0.1$

4.6 Conclusions

The present section aims to reflect on the objectives enunciated in §1.2. The idea is to re-evaluate the objectives and appraise their fulfilment. The main objective of this project was to gather and develop the tools needed to produce a LES of turbulent flows in a urban environment. At this point we can state that the principal objective of the project has been covered.

Globally, we started by appraising the current state of experimental and computational study in urban turbulent flows. Then, we presented the theoretical concepts that are needed for a correct study of urban flows. During the third chapter, the development of the solution workflow as well as the analytical tools was introduced and discussed. In that part, we saw how the geometry and the mesh are generated as well as the different processes that take part during the simulation. In addition, we also introduced the postprocessing and verification methodologies applied during the final simulation. The fourth chapter presented the design and results of the final simulation. In that part we saw how the design process took place as well as how the results can be appraised. We concluded that result were satisfactory within the limitations of the available resources. In this way, we saw that the tools and process here introduced work properly and can provide accurate results. As far as the flow study is concerned, the simulation was able to characterise the flow as well as the turbulent structures.

An extension of the present work would start by running further the final simulation such that the statistic quantities are fully converged. This additional run would allow to have a deeper understating of the physical implications of the case. Furthermore, the study could be expanded by increasing the complexity of the cases. For instance, the meshing process can easily be modified to allow the variation of the obstacle's height. In this way, a more realistic geometry would be considered. In fact, the afore-stated study is currently under development. Moreover, more complex application could also be considered, e.g. modifying the workflow to take into account the presence of particle's that might be used to simulate the propagation of a contaminants.

References

[1] Allwine, K., Leach, M., Stockham, L., Shinn, J., Hosker, R., Bowers, J., and Pace, J. Overview of joint urban 2003—an atmospheric dispersion study in oklahoma city. in: Symposium on planning, nowcasting, and forecasting in the urban zone. *American Meteorological Society* (2004).

[2] Britter, R. E., and Hanna, S. R. Flow and dispersion in urban areas. *Annual Review of Fluid Mechanics 35* (2003), 469–496. DOI:10.1146/annurev.fluid.35.101101.161147.

[3] Brix, K., Canuto, C., and Dahmen, W. Legendre-gauss-lobatto grids and associated nested dyadic grids. *RWTH Aachen University, Aachen Institute for Advanced Study in Computational Engineering Science* (2013).

[4] Chapman, D. Computational aerodynamics development and outlook. *AIAA Journal 17 (12)*, 79-0129R (1979), 1293–1313. DOI:10.2514/3.61311.

[5] Dong, S., Karniadakis, G., and Chryssostomidis, C. A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains. *Journal of Computational Physics 261* (2014), 83–105. DOI:10.1016/j.jcp.2013.12.042.

[6] D.Xiao, Heaney, C. E., Mottet, L., Fang, F., W.Lin, Navon, I. M., Y.Guo, Matar, O. K., Robins, A. G., and Pain, C. A reduced order model for turbulent flows in the urban environment using machine learning. *Building and Environment 148* (2019), 323–327. DOI:10.1016/j.buildenv.2018.10.035.

[7] Fernando, H. J. S., Zajic, D., S. Di Sabatino, R. D., Hedquist, B., and Dallman, A. Flow, turbulence, and pollutant dispersion in urban atmospheres. *Physics of Fluids 22* (2010). DOI:10.1063/1.3407662.

[8] Fischer, P. F., Lottes, J., and Kerkemeier, S. G. NEK5000: Open source spectral element CFD solver, 2008.

[9] Gadilhe, A., Janvier, L., and Barnaud, G. Numerical and experimental modelling of the three-dimensional turbulent wind flow through an urban square. *Journal of Wind Engineering and Industrial Aerodynamics 46& 47* (1993), 755–766.

[10] García-Sánchez, C., van Beeck, J., and Gorlé, C. Predictive large eddy simulations for urban flows: Challenges and opportunities. *Building and Environment 139* (2004), 146–156. DOI:10.1016/j.buildenv.2018.05.007.

[11] George B. Arfken, H. J. W. *Mathematical Methods for Physicists, sixth edition*. Elsevier Academic Press, Oxford, OH, 2005.

[12] Geuzaine, C., and Remacle, J.-F. *Gmsh Reference Manual*. 2020.

[13] Hirose, C., Ikegaya, N., Hagishima, A., and Tanimoto, J. Outdoor measurement of wall pressure on cubical scale model affected by atmospheric turbulent flow. *Building and Environment 160* (2019). DOI:10.1016/j.buildenv.2019.106170.

[14] Jeong, J., and Hussain, F. On the identification of a vortex. *Journal of Fluid Mechanics 285* (1995), 69–94. DOI:10.1017/S0022112095000462.

[15] Leonard, A. Energy cascade in large eddy simulations of turbulent fluid flow. *Advances in Geophysics 18* (1974), 237–248. DOI:10.1016/S0065-2687(08)60464-1.

[16] Lien, F.-S., Yee, E., Ji, H., and Hsieh, K.-J. Partially resolved numerical simulation and rans modeling of flow and passive scalar transport in an urban environment. *Journal of Wind Engineering and Industrial Aerodynamics 96* (2008), 1832–1842.

[17] Meinders, E., and Hanjalic, K. Vortex structure and heat transfer in turbulent flow over a wall-mounted matrix of cubes. *International Journal of Heat Fluid Flow 20* (1999), 255–267.

[18] Monnier, B., Goudarzi, S. A., Vinuesa, R., and Wark, C. Turbulent structure of a simplified urban fluid flow studied through stereoscopic particle image velocimetry. *Boundary-Layer Meteorology 166(2)* (2018), 239–268. DOI:10.1007/s10546-017-0303-9.

[19] Monnier, B., Neiswander, B., and Wark, C. Stereoscopic particle image velocimetry measurements in an urban-type boundary layer: Insight into flow regimes and incidence angle effect. *Boundary-Layer Meteorology 135* (2010), 243–268. DOI:10.1007/s10546-010-9470-7.

[20] Nagib, H., Morkovin, M., Yung, J., and Tan-atichat, J. On modeling of atmospheric surface layers by the counter-jet technique. *AIAA Journal 14(2)* (1974), 185–190.

[21] Negi, P. S., Vinuesa, R., Hanifi, A., Schlatter, P., and Henningson, D. Unsteady aerodynamic effects in small-amplitude pitch oscillations of an airfoil. *International Journal of Heat and Fluid Flow 71*, 0142-727X (2018), 378–391.

[22] Oke, T. R. Street design and urban canopy layer climate. *Energy and Buildings 11* (1988).

[23] Parallel Work, I. About Parallel Works.

[24] Patera, A. A spectral element method for fluid dynamics : laminar flow in a channel expansion. *Journal of Computational Physics 54* (1984), 468–488. DOI:10.1016/0021-9991(84)90128-1.

[25] Pope, S. *Turbulent flows*. Cambridge University Press, Ithaca, N.Y., 2000.

[26] Quarteroni, A., Sacco, R., and Saleri, F. *Numerical Mathematics*. Springer-Verlag, New York, NY, 2000. ISBN: 0-387-98959-5.

[27] Reynolds, O. An experimental investigation of the circumstances that determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Philosophical Transactions of the Royal Society 74* (1883), 935–982. DOI:10.1098/rstl.1883.0029.

[28] Sagaut, P. *Large Eddy Simulation for Incompressible Flows : An Introduction*. Springer, Paris, France, 1998.

[29] Schlatter, P., Stolz, S., and Kleiser, L. Les of transitional flows using the approximate deconvolution model. *International Journal of Heat and Fluid Flow 25* (2004), 549–558. DOI:10.1016/j.ijheatfluidflow.2004.02.020.

[30] Schlatter, P., and Örlü, R. Turbulent boundary layer at moderate reynolds numbers: inflow length and tripping effects. *Journal of Fluid Mechanics 710* (2012), 5–34. DOI:10.1017/jfm.2012.324.

[31] Schlatter, P., and Örlü, R. Comparison of experiments and simulations for zero pressure gradient turbulent boundary layers at moderate reynolds numbers. *Experiments in Fluids 54(6)* (2013). DOI:10.1007/s00348-013-1547-x.

[32] Shih, T., and Liu, N. Partially resolved numerical simulation from rans towards les for engine turbulent flows. *AIAA Journal 160* (2008).

[33] Tong, C., and Warhaft, Z. Passive scalar dispersion and mixing in a turbulent jet. *Journal of Fluid Mechanics 292* (1995).

[34] Vinuesa, R., Azizpour, H., Leite, I., Balaam, M., Dignum, V., Domisch, S., Felländer, A., Langhans, S. D., Tegmark, M., and Nerini, F. F. The role of artificial intelligence in achieving the Sustainable Development Goals. *Nature Communications 11,233* (2020). DOI:10.1038/s41467-019-14108-y.

[35] Vinuesa, R., Negi, P., Atzori, M., Hanifi, A., Henningson, D., and Schlatter, P. Turbulent boundary layers around wing sections up to $Re_c = 1,000,000$. *International Journal of Heat and Fluid Flow 72*, 86–99.

[36] Vinuesa, R., Peplinski, A., Atzori, M., Fick, L., Marin, O., Merzari, E., Negi, P., Tanarro, A., and Schlatter, P. Turbulence statistics in a spectral-element code: a toolbox for high-fidelity simulations. *KTH Publications* (2018).

[37] Vinuesa, R., Schlatter, P., Malm, J., Mavriplisb, C., and Henningsona, D. S. Direct numerical simulation of the flow around a wall-mounted square cylinder under various inflow conditions. *Journal of Turbulence 16*, 6 (2015), 555–587. DOI:10.1080/14685248.2014.989232.

[38] Vinuesa, R., Schlatter, P., and Nagib, H. M. Secondary flow in turbulent ducts with increasing aspect ratio. *Physical Review Fluids 3* (2018). DOI:10.1103/PhysRevFluids.3.054606.

[39] Vinuesa, R., Örlü, R., Vila, C. S., Ianiro, A., Discetti, S., and Schlatter, P. Revisiting history effects in adverse-pressure-gradient turbulent boundary layers. *Flow Turbulence Combust 99* (2017), 565–587. DOI:10.1007/s10494-017-9845-7.

[40] Vita, G., Shu, Z., Jesson, M., Quinn, A., Hemida, H., Sterling, M., and Baker, C. On the assessment of pedestrian distress in urban winds. *Journal of Wind Engineering & Industrial Aerodynamics 203* (2020), 1–18. DOI:10.1016/j.jweia.2020.104200.

[41] Weerasuriyaa, A., Tsea, K., Zhanga, X., and Lib, S. A wind tunnel study of effects of twisted wind flows on the pedestrian-level wind field in an urban environment. *Building and Environment 128* (2018), 225–235. DOI:10.1016/j.buildenv.2017.11.041.

[42] Zajic, D., Fernando, H. J. S., Calhoun, R., Princevac, M., Brown, M. J., and Pardyjak, E. R. Flow and turbulence in an urban canyon. *Journal of Applied Meteorology and Climatology 50* (2011), 203–222. DOI:10.1175/2010JAMC2525.1.

Part II

Blueprints, Solicitation document and Budget

5

Plans and blueprints

Due to the nature of the present project, no plans, drawings or blueprints are applicable. Thus, this page is intentionally left blank.

6

Solicitation Document

The present chapter aims to report the technical and legal implications of the present study. In particular we will focus on the work and safety conditions that need to be ensure during the development of a project of this nature. The whole range of conditions and specification ar understood within the frame of a computing-oriented project. The solicitation document follows the specifications established by the Spanish Law. However, due to the international nature of the work here presented, some of technical or legal specificities of third countries might no be reported.

6.1 Functions of the involved parties

This section will focus on the generalities related with the working conditions and the task corresponding to each of the parties involved in the project. Note that the present section will treat legal and advisory measures indistinctly. The exposition will be structured following the title of the parties involved, i.e. engineer or student, supervisor and advisor.

6.1.1 Functions of the student

The engineer or students is the central party in the well development of the project. Its fundamental task consist on the implementation, within its limitations, of the simulations and analysis needed during the project. In addition, he is also responsible for the research project, i.e. with the help of the tutor, gathering, reading and analysing of the technical and scientific literature to support the work done in the project. As far as strategy is concerned, the engineer is required to discuss and plan the strategy to be followed during the project and to ensure, as much as possible, the well-execution of the plans. Furthermore, he is in charge of the writing of the final report and presentation as well as the discussion of the documents with the evaluation committee established by the university board. Finally, from the ethical perspective, the students is required to ensure the quality and trustworthiness of the resources used during all the stages of the project as well as the respect of non-plagiarism policy established by the University board.

6.1.2 Functions of the director

The director has the main task of ensuring the well-functioning of the project in general, planning and supervising the different studies with the student. From the analysis perspective, the director must also supervise the results to validate the obtained data and conclusion. In addition, he also provides help with major issues regarding any of the stages of the project. Finally, it is also his responsibility to provide the necessary equipment, outside the personal sphere, to fulfil the project's objectives, e.g. compute software, hardware etc. Furthermore due to the nature of the computational resources required in the present project, i.e. the use of a supercomputer, the director will also assess and execute the operations that need to be covered in the supercomputer.

6.1.3 Functions of the advisor

The advisor is in charge of supporting the work of the student and director as much as needed. His particular role will be strongly dependent on the nature of its capabilities and the project's needs. In the case of the present work, the advisor also helps in orienting the student in the particular paperwork required by the University where the project is presented.

6.2 Working environment conditions

This section will focus on the particular conditions relative to the working place. Those are regulated by the Spanish legal edict, *Real Decreto* 488/1997 which covers the minimal security and health conditions that need to be ensured by any worker exposed to computing devices, paperwork, confined environment working place etc. The section will be covered following the classification contemplate by the Spanish regulatory institutions.

General working environment conditions The following points gather some of the fundamental instructions on the cleanness and order of the working-place.

- The working-place must be kept clean and ordered within the standards of health saferty, i.e. avoiding dust and dirt accumulation in any of the equipments that used during working sessions.
- Kept within the safety standards storage rooms and storing devices both for personal and general use.
- Keep the safety infrastructure free from any obstruction that might difficult its used.
- Ensure that the personal waste produced during the working section is properly dispose in the correspondent containers.
- Temperature, humidity and ventilation must ensure not to put in danger the worker nor be a source of discomfort. The standard in this area dictates that the temperature should be kept between 20°C and 24°C during the winter and between $23°C$ and $26°C$ during the summer. In addition, relative humidity should be kept always within the rang of 45% and 65%.

Lighting and noise Once again the following lines will cover the minimal requirements in terms of lighting and noise established by the Spanish regulation institutions.

- Lighting conditions will be strongly dependent on the particularities of the task considered. Nevertheless, the Spanish regulation agency contemplates some minimal requirements described in the *Real Decreto* 486/1997
  - Within the limitations encountered, natural light will always be preferred over any alternate source of light as long as the environmental conditions allow it.
  - In addition to natural lighting, artificial lighting might be available to complement natural sources. Artificial sources must be adaptable such that the work can properly set the the lightning level depending on the moment.

- Under no circumstance, lights that might put in danger the vision or in general the health of the workers will be used.
- Illumination systems must be properly distributed such that the lighting level is uniformly distributed over the working place. Glares will be avoided as much as possible.

- Noise level is regulated by means of the *Real Decreto* 1316/989 covering the workplace protection.
  - The law establishes that noise exposure should be minimised as much as possible considering the technicalities of the working place.
  - The maximum noise level defined using the LAeq index is $50dB$. The work equipments and installations must ensure that the afore-stared limit is not surpassed. Nevertheless if this limits had to be exceeded, the person responsible of the institution must provide the equipment and instruction needed to ensure the safety of the workers implicated.

Protection and emergency conditions In addition, to the conditions specified in the previous paragraphs, the company or University department must ensure that the emergency measure, i.e. exits, equipment, instructions etc., function properly. In addition, it is the responsibility of the institution to correctly inform the workers on the specificities of the place regarding risk and emergency measures. Furthermore, the installations must fulfil the fire and electrical standards established by the regulatory agencies.

Working site conditions Due to the nature of the project, it is expected that the engineer will spend large periods of time at the work stations, thus it is advisable to cover some ergonomic guidelines to ensure that worker's long-term health is not risked. The nature of computational projects dictates that the worker will spending large period of times in a static position in front of a screen. In this way, it is advisable to have a proper table setup in order to avoid unnatural postures that might induce health problems. In this way, the computational resources must be adjusted to ensure the afore-stated guidelines. It is also advisable to have an office chair that ensure a good postural stance and it is equipped with wheels to allow mobility. Furthermore, the worker needs to have enough space in the workstation to move without important mobility restrictions.

7

Budget

In this final chapter, the main objective is to estimate the monetary cost of the project here reported. The vast majority of the cost is concentrated in the work of the parties involved in the project. Note that very expensive tools are utilised in the present work, e.g. a supercomputer. However, due to the public nature of the resource and its limited access, no realistic estimation of the redeemed quantity can be computed. In this way, we will assume that the cost of the supercomputer is redeem such that this variable can be excluded. The cost of the Cray XC40 supercomputer is estimated at 156,000,000 \$. However, the redeemed quantity remains unknown.

Monetary quantities will be considered in Swedish krona since the vast majority of the project was developed in Sweden. Nevertheless, final quantities will be converted to Euros. The cost will be split in two categories. Firstly, we will consider the cost of work, i.e. the salary that would have been perceived by the parties. Table 7.1 gathers the estimated cost before takes. Note that salaries for engineer and thesis director are estimated from the average salary in Sweden for a starting engineer (less than one year of experience) and an assistant professor. Indirect cost are assumed to be a 30% of the other cost combined. Finally, taxation is estimated through the SAT which is a 25% in Sweden.

Type	**Concept**	**Usage time** (h)	**Cost**/h (SEK)	**Total cost** (SEK)
Salary	Engineer's pay	700	130	91,000
	Director's pay	90	250	22,500
	Adviser pay	10	200	2000
Fixed cost	External computer	–	–	12,509.64
	Expendable goods	–	–	500
Indirect costs	–	–	–	38,552.89
Total cost before taxes (SEK)				167,062.53
Total cost before taxes (€)				16,025.65
Total cost after taxes (SEK)				208,828.16
Total cost after taxes (€)				20,032.06

Table 7.1: Cost before taxes

In conclusion, the project cost is estimated at 20,032.06 €.