Unsteady SpaRTA

Data-driven turbulence modelling for unsteady applications

Master Thesis

Nicolò Miori

Delft University of Technology

Unsteady SpaRTA

Data-driven turbulence modelling for unsteady applications

by

Nicolò Miori

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Thursday December 15, 2022 at 14:00.

Student number:	4839706	
Thesis supervisor:	Dr. Anh Khoa Doan	TU Delft
	Dr. Richard Dwight	TU Delft
Thesis committee:	Dr. Marios Kotsonis	TU Delft, Committee chair
	Dr. Ivan Langella	TU Delft, External member
Project Duration:	Sep, 2021 - Dec, 2022	

Preface

Here comes *The Last Dance*, citing a famous Netflix Series. This work marks the end of my time as a University student. The years here in Delft were the most significant for me. I stayed longer than expected, but it was all worth it. I had yes the opportunity to grow academically and as an engineer, but, most importantly, as a person. It was full of experiences and emotions, shared with amazing people that I will always bring with me.

I would like to thank my parents because without their continuous support and love none of this would have been possible. They are simply the best, and I will never be thankful enough.

I also would like to thank all my friends here in Delft, but also my friends back in Italy for always being there for me. I am not going to name them all, since I would always leave someone behind, but you are all essential to me. Though, a special mention goes to my friend Gigi, since we met on the first day of lectures in the Aerospace Faculty, after which we shared this whole time here in Delft together and now you are like a brother to me.

Last but not least, I am happy to thank my supervisors Anh Khoa and Richard for their continuous feedback during this research project. It was a challenge to me because I had very little background about the topic, but thanks to your guidance I got to know more about it and finally deliver this dissertation which I hope you will find interesting.

Even if sad about ending my student life, I am beyond excited for new challenges and to see what the future holds.

Nicolò Miori
Delft, December 2022

Abstract

Recent years have seen an increase in studies focusing on data-driven techniques to enhance modelling approaches like the two-equation turbulence models of Reynolds-averaged Navier-Stokes (known as RANS). Different methods have been implemented to improve the results from these simulations. In particular, the main focus has been on overcoming the limitations implied by the Boussinesq assumption [8]. This has been approached by using machine learning techniques as a way of discovering new formulations that could overperform when compared to traditional models.

Despite promising results for steady RANS simulations, little has yet been investigated in URANS applications. In this dissertation, this lack of research will be addressed. The main ideas are then, first, to see how the available information in URANS simulations can be used to improve the anisotropic Reynolds stress tensor prediction, and second if and how this can be done by using a sparse regression technique, whose framework is known as SpaRTA [36]. A procedure involving the *triple decomposition* of High-Fidelity velocity fields is applied, aiming at finding a model exclusively for the stochastic component of the anisotropy [21]. The test case which is considered is the flow around a cylinder at $Re = 3900$. The High-Fidelity data was collected by running Large Eddy Simulations in OpenFOAM, after which the velocity was split into its different components through Proper Orthogonal Decomposition.

A priori results have shown good performance of the trained models, outperforming the Boussinesq assumption both in the prediction of turbulence componentiality and also on the values of the single anisotropy components.

Contents

Preface i

Abstract ii

1 Introduction 1

2 Turbulence Modelling 3
- 2.1 Turbulence 3
  - 2.1.1 What is turbulence? 3
  - 2.1.2 Description of turbulent flows 4
  - 2.1.3 Turbulence modeling and RANS. 7
- 2.2 URANS: Unsteady Reynolds Averaged Navier Stokes 8
  - 2.2.1 Governing equations 8
  - 2.2.2 Eddy-viscosity models 10
  - 2.2.3 $k - \omega$ SST Model 10
- 2.3 Bluff-bodies Flow 12
  - 2.3.1 Resolution of turbulence in simulations 14

3 Data-driven Turbulence Modelling 16
- 3.1 Steady flows 16
  - 3.1.1 SpaRTA: Sparse Regression of Turbulent Stress Anisotropy 19
- 3.2 Unsteady flows 22

4 Scientific Gaps and Research Questions 24

5 Simulation of Flow around a Cylinder 26
- 5.1 Computational Domain 26
- 5.2 High-Fidelity Simulations 28
  - 5.2.1 Governing equations 28
  - 5.2.2 Subgrid scale model 29
  - 5.2.3 Boundary conditions 30
  - 5.2.4 Quality of LES simulations 31

6 Data-driven Unsteady Methodology 37
- 6.1 Introduction to the Framework 37
- 6.2 Proper Orthogonal Decomposition. 39
- 6.3 Phase and Ensemble Averaging. 43
  - 6.3.1 Influence of number of vortex shedding periods 44
- 6.4 Unsteady SpaRTA 45
  - 6.4.1 Non-linear eddy-viscosity model for $b_{ij}^{''}$ 45
  - 6.4.2 Library of candidate functions 45
  - 6.4.3 Model selection and inference using sparsity-promoting regression 46
- 6.5 *A priori* study 47

7 Results **49**
7.1 Correlation between shedding and stochastic quantities 49
7.2 Phase-averaged modelling from $\widetilde{U}_i$ to b''_{ij} 52
7.3 Ensemble-averaged modelling from $\overline{U}_i$ to b''_{ij} 53
7.3.1 Comparison of Boussinesq approximation and modelled anisotropy 54
7.3.2 Testing different models at $Re = 6000$ 59
8 Conclusions **60**
9 Future Work **62**
References **65**
A Model Matrix for models from 1 to 30 functions at $Re = 3900$ **66**
B 3-D Tensors and Invariants **67**
C Definition and Performance of $M2$ Model **68**

1
Introduction

As already previewed from the title of the dissertation, this study is focused on turbulence modelling. Let us first start with a simple question: why are we interested in turbulence? Turbulence is a common physical characteristic of fluid flows. In wind turbine design, knowledge of the turbulence in the incoming flow and in the blade boundary layers is important for performance; in internal combustion engines, vigorous turbulence increases fuel/air mixing, improving overall efficiency and reducing emissions; and in airplane design, delaying the occurrence of turbulence in boundary layers over the wing surfaces leads to reduced fuel consumption [10]. It is then of continuous interest to develop techniques to simulate and predict turbulent flows.

The difficulties in the representation of turbulent motions are due to the broad range of active spatial and temporal scales involved and to the strong chaotic nature of the phenomenon. The continuous growth of computer power has enabled direct numerical simulations (DNS) of a number of turbulent flows. DNS exactly resolve the Navier-Stokes Equations, hence providing a fully resolved velocity field, where no approximation is introduced. However, this is possible only for simple flows at low Reynolds numbers. In fact, for example, it would take years to resolve the flow around an airplane by means of DNS simulations.

This is the main reason why, especially for practical applications, simplified engineering approximations remain popular and widely used. Among these, Reynolds-averaged Navier-Stokes (RANS) and large-eddy simulation (LES) approaches are the most common. In this study, particular focus will be on RANS. These models are constructed using a formal averaging procedure applied to the exact governing equations of motion and require closures to represent the turbulent stresses and scalar fluxes emerging from the averaging process. The inherent assumptions in the RANS approach and the process of formulating closure models introduce potential accuracy limitations and, consequently, reduced credibility in the predictive power of the approach. During the last few years, to overcome such limitations, research in the use of data-driven techniques has been performed.

Due to improvements in computational power and the increased availability of large data sets, data science is nowadays implicated in many different disciplines. In addition, significant improvements in data analytics and machine learning (ML) techniques have paved the way for further and more widespread use. Different ML techniques are commonly used today in financial and commercial applications such as stock trading, fraud detection, etc. and scientific applications such as genomics, astrophysics, fluid mechanics, and natural language processing [11].

Depending on the application, the objectives of the tasks can be a combination of automated clustering and classification, feature extraction, predictive modelling and improved decision-making. Specifically, in the area of turbulent flows, previous efforts have used Elastic Net Regression and Gene Expression Programming for the purpose of tensor modelling [36][42], field inversion and neural networks (FIML) to extract spatio-temporal model discrepancies and reconstruct the discrepancies into functional forms that are embedded within physical models [17], neural networks for learning nonlocal constitutive models [48], *etc*.

The thesis work, that I conducted, places itself in the area of tensor modelling, through the application of Elastic Net Regression. Such regression technique has shown the ability to come up with algebraic expressions that can be easily implemented as a substitute for the modelled Reynolds stresses. The novelty of the work here presented comes from the fact that it is not employed for correcting RANS simulations, but for URANS applications, which means that only the stochastic contribution to the Reynolds stresses will be modelled through the ML algorithm mentioned above.

The structure of the dissertation is then organised as follows. In Chapter 2, some theoretical notions are introduced, which are necessary to understand this work and the objectives of this research. Chapter 3 provides a review of past research and works in data-driven turbulence modelling, as well as highlighting the advantages and disadvantages of specific methodologies. This leads to Chapter 4 where the scientific gaps are pointed out, and so the research questions that will be addressed in this Thesis are exposed. In Chapter 5, it is explained the test case and the CFD simulations that were run for collecting the high-fidelity data. Chapter 6 is then the main player of this dissertation since the developed novel methodology is presented in depth. Last but not least, in Chapter 7 the results obtained are presented, followed by final conclusions and suggestions for eventual future work.

2

Turbulence Modelling

2.1. Turbulence

2.1.1. What is turbulence?

In our everyday surroundings, there are a lot of opportunities to observe turbulent flows, whether it be smoke from a chimney, water in a waterfall, or the buffeting of a strong wind. When observing a waterfall, it can immediately be detected that the flow is irregular, unsteady, seemingly random and chaotic, and surely the motion of every eddy or droplet is unpredictable.

An essential feature of turbulent flows is that the fluid velocity varies significantly and irregularly in both space and time. Figure 2.1 shows a typical time history of the velocity of a turbulent flow field at a given spatial location. The dashed horizontal line shows the mean velocity denoted by $\overline{U}$, defined as:

$$\overline{U} = \frac{1}{T} \int_{t_0}^{t_0+T} U(t)\; dt \tag{2.1}$$

It may be observed that the velocity $U(t)$ shows significant fluctuations and the time history features variations on a wide range of timescales [31].

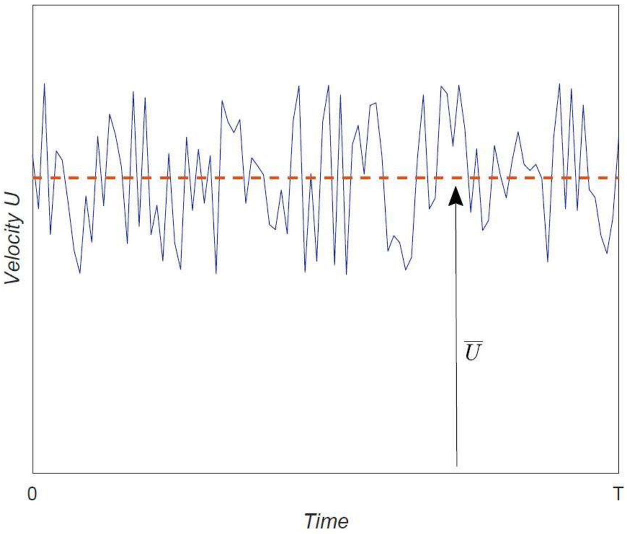


Figure 2.1: Temporal evolution of a turbulent velocity signal.

In engineering applications turbulent flows are prevalent. For example in the processing of liquids/gases with pumps, compressors, pipelines, etc., the flows are generally turbulent. Similarly, the flows around vehicles are mainly turbulent. An important characteristic of turbulent flows is their ability to better transport and mix fluids when compared to laminar flows. This can be an advantage, for example when thinking of a combustion chamber, where we want the mixing to be as more efficient as possible. However, this can also be a disadvantage, because this tendency comes with additional drag, which we usually want to minimize when designing external aerodynamic surfaces, such as on cars or on aircraft wings.

2.1.2. Description of turbulent flows

When considering turbulent flows, the first way of providing a categorization is to distinguish between small-scale turbulence and large-scale motions. The latter ones are strongly influenced by the geometry of the flow, hence the boundary conditions. Their nature then depends on the dimensions of the flow field, its characteristics and/or on external bodies generating flow disturbances. On the contrary, the behaviour of the small-scale motions is determined almost completely by the rate at which they receive energy from the large scales, and by the viscosity. They then have a universal character, since they are somewhat independent of the flow geometry.

When talking about turbulent flows, we have to mention the importance of two quantities of interest: the *turbulent kinetic energy* and the *anisotropy* of the Reynolds stresses. The turbulent kinetic energy is the mean kinetic energy per unit mass associated with eddies in a turbulent flow. Mathematically, it is written as follows:

$$k = \frac{1}{2}\left(\overline{(u')^2} + \overline{(v')^2} + \overline{(w')^2}\right) \tag{2.2}$$

where the turbulent velocity component u' is the difference between the instantaneous and the average velocity. The turbulent kinetic energy is then a quantitative measure that defines the flow behaviour, in particular, it gives an indication about how turbulent the flow is. If the flow were purely laminar along all three orthogonal directions, then the turbulent kinetic energy would be zero.

Anisotropy is also a quantity that tells us about the type of turbulence. In particular, it is a measure of the degree of departure from the ideal isotropic turbulence. The scaled Reynolds stress anisotropy b_{ij} is derived from the Reynolds stress tensor τ_{ij} according to the formula:

$$b_{ij} = \frac{\overline{u'_i u'_j}}{2k} - \frac{\delta_{ij}}{3}, \quad \delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j' \end{cases} \tag{2.3}$$

The *Reynolds stress anisotropy tensor* is the most trustful entity to characterize the turbulence and to ground the hypotheses which are made in classical theoretical development [9]. In fact, its second and third mathematical invariants describe the possible states of realizable turbulence by means of the *anisotropic invariant map* or the Lumley triangle.

Related to the turbulent kinetic energy, is the concept of *energy cascade*. As a first step, the kinetic energy enters the turbulence at the largest scales of motion. By inviscid processes, this energy is then transferred to gradually smaller scales until, at the smallest scales, the energy is dissipated by viscous action. Such procedure is represented in Figure 2.2. The large eddies take energy from the mean flow via the Reynolds stresses, through velocity gradient or shear, and feed it down a cascade. This energy transfer proceeds until the local Reynolds number is sufficiently small that the eddy motion is stable, and molecular viscosity is effective in dissipating the kinetic energy into thermal energy. The *Reynolds number* is a non-dimensional quantity that relates the amount of inertial forces to viscous forces. It is defined as:

$$Re = \frac{uL}{\nu} = \frac{\rho u L}{\mu} \tag{2.4}$$

where

- ρ is the density of the fluid (kg/m^3)

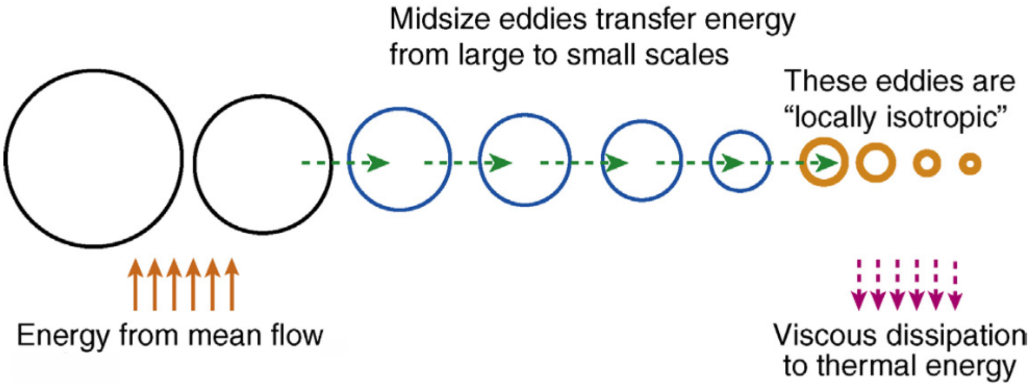

Midsize eddies transfer energy from large to small scales
These eddies are "locally isotropic"
Energy from mean flow
Viscous dissipation to thermal energy

Figure 2.2: Energy transfer from the largest scales to the smallest ones (taken from [39]).

- u is the flow speed (m/s)
- L is the characteristic length (m), which for example for the flow around the cylinder is the diameter D
- μ is the dynamic viscosity of the fluid $(kg/(m \cdot s))$
- ν is the kinematic viscosity of the fluid (m^2/s)

When the Reynolds number is small it means then that the viscous forces are dominating over the inertial forces, and hence the main action is the one from the dissipating molecular viscosity.

The smallest scales were identified by Kolmogorov [20], who developed a theory based on three hypotheses. The assumptions are the following [31]:

- **Kolmogorov's hypothesis of local isotropy.** At sufficiently high Reynolds number, the small-scale turbulent motions ($l \ll l_0$) are statistically isotropic, where l_0 is the scale of the largest eddies, comparable to the flow scale $\mathcal{L}$.
- **Kolmogorov's first similarity hypothesis.** In every turbulent flow at a sufficiently high Reynolds number, the statistics of the small-scale motions ($l < l_{EI}$) have a universal form that is uniquely determined by ν and ϵ, which are respectively the kinematic viscosity and the rate of dissipation of turbulent kinetic energy.
- **Kolmogorov's second similarity hypothesis.** In every turbulent flow at a sufficiently high Reynolds number, the statistics of the motion of scale l in the range $l_0 \gg l \gg \eta$ have a universal form that is uniquely determined by ϵ, independent of ν.
  η is the so-called Kolmogorov length scale and defined as $(\nu^3/\epsilon)^{1/4}$. It is the scale of the finest pulsations, whose energy is directly dispersed into heat energy due to viscosity.

Based on these hypotheses, the following subdomains are defined:

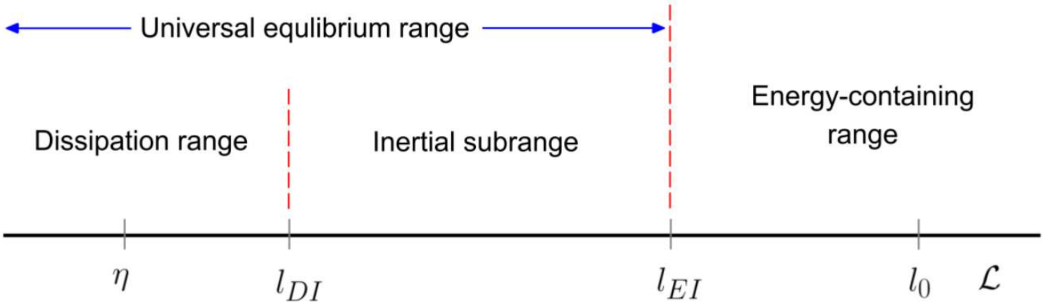


Figure 2.3: Subdivison of lengthscales (taken from [31]).

The lengthscale l_{EI} is then the demarcation between the anisotropic large eddies and the isotropic small eddies. As the energy goes through the cascade, the directional information and the information about the geometry of the large eddies are lost. That is why we say that the statistics of the small-scale motions are universal, similar in every high Reynolds number turbulent flow.
An additional separation was introduced with the second hypothesis. A length scale where viscous effects are negligible and in Figure 2.3 titled as inertial subrange. As a consequence, the eddies in this range are independent of ν, and only motions in the dissipation range experience viscous effects and hence they are responsible for all the dissipation.

In this dissertation, incompressible flows are considered. Such type of flows are governed by a set of differential equations, whose solution defines the variation of each field variable with respect of space and time. The main equation we need to consider is the *momentum* equation. Based on Newton's second law, it relates the fluid particle acceleration $\mathrm{D}\boldsymbol{U}/\mathrm{D}t$ to the surface forces and body forces experienced by the fluid. In general, surface forces are described by the stress tensor τ_{ij}, which is symmetric ($\tau_{ij} = \tau_{ji}$). Concerning the body forces, the one of interest is gravity. Defining Ψ as the *gravitational potential*, the body force per unit mass is

$$\boldsymbol{g} = -\nabla\boldsymbol{\Psi} \tag{2.5}$$

The surface and body forces together cause the fluid to accelerate according to the momentum equation

$$\rho\frac{\mathrm{D}U_j}{\mathrm{D}t} = \frac{\partial\tau_{ij}}{\partial x_i} - \rho\frac{\partial\boldsymbol{\Psi}}{\partial x_j}. \tag{2.6}$$

If we restrict the study to flows of *constant-property Newtonian fluids*, we have that the stress tensor is given by the expression

$$\tau_{ij} = -P\delta_{ij} + \mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) \tag{2.7}$$

where P is the pressure and μ is the dynamic viscosity coefficient. By substituting this expression into the momentum equation (2.6), and reminding that ρ and μ are uniform and that $\nabla\cdot\boldsymbol{U} = 0$, we obtain the so-called *Navier-Stokes equations*

$$\rho\frac{\mathrm{D}U_j}{\mathrm{D}t} = \mu\frac{\partial^2 U_j}{\partial x_i\partial x_i} - \frac{\partial P}{\partial x_j} - \rho\frac{\partial\Psi}{\partial x_j} \tag{2.8}$$

In addition, by defining the *modified pressure p* as

$$p = P + \rho\Psi$$

the equation simplifies to

$$\frac{\mathrm{D}\boldsymbol{U}}{\mathrm{D}t} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{U} \tag{2.9}$$

where $\nu \equiv \mu/\rho$ is the kinematic viscosity. In conclusion, the behaviour of constant property fluids, both laminar and turbulent flows, is described by the Navier-Stokes (N-S) equations together with the solenoidal condition $\nabla\cdot\boldsymbol{U} = 0$ derived from mass conservation.

It has to be remarked that the N-S equations are valid when the *continuum hypothesis* is valid, which means that the mean free path of the molecules is very small relative to the smallest physical length scale [39]. This validity is controlled by the *Knudsen* number $Kn = \lambda/l$ where λ is the mean free path, while l is the smallest geometric length scale. When $Kn \ll 1$, and this is the case since even the smallest turbulent lengthscales exceed the molecular scales, the flow can be considered as a continuous matter, which obeys the laws of conservation presented above.

All the equations governing turbulent flows were presented, specifically the Navier-Stokes equations. When they are directly solved, without any modelling introduced, it is said to be running Direct Numerical Simulations (DNS). However, this is possible only for simple flows. In fact, as the number of cells and the Reynolds number increase, it becomes computationally too expensive to resolve them explicitly. In order to simulate flows for more complex applications, for example, flows around cars, planes, etc., some simplifications have to be introduced, which we will see in the next section.

2.1.3. Turbulence modeling and RANS

The simplest way of resolving/modeling turbulence is by introducing the *Reynolds decomposition* [32]. Such procedure leads to the split of the velocity $\boldsymbol{U}(\boldsymbol{x},t)$ into its mean $\overline{\boldsymbol{U}(\boldsymbol{x},t)}$ and the fluctuation $\boldsymbol{u}^{'}(\boldsymbol{x},t)$, such that we have

$$\boldsymbol{U}(\boldsymbol{x},t) = \overline{\boldsymbol{U}(\boldsymbol{x},t)} + \boldsymbol{u}'(\boldsymbol{x},t) \tag{2.10}$$

Let us see now how we can obtain the most basic equations, which govern the mean velocity field $\overline{\boldsymbol{U}(\boldsymbol{x},t)}$. We can start by taking the mean of the momentum equation, which is not straight-forward, because of the nonlinear convective term. Applying the averaging operator to the substantial derivative in conservative form, we obtain:

$$\frac{\overline{\mathsf{D}U_j}}{\mathsf{D}t} = \frac{\partial \overline{U_j}}{\partial t} + \frac{\partial}{\partial x_i}\overline{U_i U_j} \tag{2.11}$$

At this point, we then substitute the Reynolds decomposition (equation (2.10)) for U_i and U_j,

$$\begin{aligned}\overline{U_i U_j} &= \overline{\left(\overline{U_i} + u'_i\right)\left(\overline{U_j} + u'_j\right)}\\ &= \overline{\overline{U_i}\,\overline{U_j} + u'_i\overline{U_j} + u'_j\overline{U_i} + u'_i u'_j}\\ &= \overline{U_i}\,\overline{U_j} + \overline{u'_i u'_j}.\end{aligned} \tag{2.12}$$

The velocity covariances $\overline{u'_i u'_j}$ are known as the *Reynolds stresses*. By injecting equation (2.12) into equation (2.11), we come to the following:

$$\begin{aligned}\frac{\overline{\mathsf{D}U_j}}{\mathsf{D}t} &= \frac{\partial \overline{U_j}}{\partial t} + \frac{\partial}{\partial x_i}\left(\overline{U_i}\,\overline{U_j} + \overline{u'_i u'_j}\right)\\ &= \frac{\partial \overline{U_j}}{\partial t} + \overline{U_i}\frac{\partial \overline{U_j}}{\partial x_i} + \frac{\partial}{\partial x_i}\overline{u'_i u'_j}\end{aligned} \tag{2.13}$$

where we get to the second step by using the continuity equation $\nabla \cdot \overline{\boldsymbol{U}} = 0$ and hence $\partial \overline{U_i}/\partial x_i = 0$.

It is possible to re-express the obtained result by introducing the *mean substantial derivative*

$$\frac{\overline{\mathsf{D}}}{\overline{\mathsf{D}}t} \equiv \frac{\partial}{\partial t} + \overline{\boldsymbol{U}} \cdot \nabla \tag{2.14}$$

Generally speaking, for any quantity $Q(\boldsymbol{x},t)$, $\overline{\mathsf{D}}Q/\overline{\mathsf{D}}t$ represents its rate of change following a point moving with the local mean velocity $\overline{\boldsymbol{U}(\boldsymbol{x},t)}$. By using the derivative from equation (2.14), equation (2.13) can be contracted and rewritten as

$$\frac{\overline{\mathsf{D}U_j}}{\mathsf{D}t} = \frac{\overline{\mathsf{D}}}{\overline{\mathsf{D}}t}\overline{U_j} + \frac{\partial}{\partial x_i}\overline{u'_i u'_j} \tag{2.15}$$

It can be readily seen that the mean of the substantial derivative $\overline{\mathsf{D}U_j/\mathsf{D}t}$ does not match the mean substantial derivative of the mean $\overline{\mathsf{D}}\,\overline{U_j}/\overline{\mathsf{D}}t$. In fact a new term appeared, which is the derivative of the covariances of the velocity fluctuations. Since the other terms in the momentum equation are linear in $\boldsymbol{U}$ and p, it is now easy to take the mean of the momentum equation. The result is the well known *Reynolds-averaged Navier Stokes* (RANS) equations

$$\frac{\overline{\mathsf{D}}\,\overline{U_j}}{\overline{\mathsf{D}}t} = \nu\nabla^2\overline{U_j} - \frac{\partial \overline{u'_i u'_j}}{\partial x_i} - \frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j} \tag{2.16}$$

Equations (2.9) and (2.16) look very similar, with the crucial difference that the RANS equations have the term involving the Reynolds stresses. This term is important, since it is the reason why the RANS equations *cannot* be solved directly.

It is recognized as the *closure problem*, since such set of equations is said to be *unclosed*. The reason is that there are more unknowns than equations. The RANS equations can in fact be written as three momentum equations for the three velocity components, and one mean continuity equation, so four equations in total. These equations need to be solved for $\overline{\boldsymbol{U}}$ and $\overline{p}$, but also for the Reynolds stresses $\overline{u_i'u_j'}$.
When discussing turbulence modelling, much of it revolves around the modelling of these Reynolds stresses. In the next chapter, further details about various turbulence models are given.

2.2. URANS: Unsteady Reynolds Averaged Navier Stokes

2.2.1. Governing equations

In the previous section, the RANS equations have been introduced (equation (2.16)). The peculiarity comes from the fact that they are solved for the mean velocity field $\overline{\boldsymbol{U}}$. In order to treat the unsteady version of the Reynolds Averaged Navier Stokes equations, we recall the *triple decomposition* of the velocity [4], which performs an additional split on the fluctuating quantity $\boldsymbol{u}^{'}(\boldsymbol{x},t)$ (equation 2.10):

$$\boldsymbol{U}(\boldsymbol{x},t) = \underbrace{\overline{\boldsymbol{U}}(\boldsymbol{x})}_{\text{mean component}} + \underbrace{\widetilde{\boldsymbol{u}}(\boldsymbol{x},\phi(t))}_{\text{deterministic component}} + \underbrace{\boldsymbol{u}''(\boldsymbol{x},t)}_{\text{stochastic component}} \tag{2.17}$$

where the mean component is independent of time and the deterministic part is phase-dependent. If we merge these two quantities together we can rewrite the triple decomposition as follows:

$$\boldsymbol{U}(\boldsymbol{x},t) = \underbrace{\widetilde{\boldsymbol{U}}(\boldsymbol{x},t)}_{\text{resolved fluctuation}} + \underbrace{\boldsymbol{u}''(\boldsymbol{x},t)}_{\text{modelled turbulent fluctuation}} \tag{2.18}$$

The same decomposition can also be applied to the values of pressure, and hence: $p(\boldsymbol{x},t) = \widetilde{P}(\boldsymbol{x},t) + p^{''}(\boldsymbol{x},t)$.
For simplicity we now refer to $\widetilde{U}_i(\boldsymbol{x},t)$ and $u_i''(\boldsymbol{x},t)$ as $\widetilde{U}_i$ and u_i'' respectively, and similarly for the pressure terms.

In order to obtain the Unsteady Reynolds Averaged Navier Stokes equations, we substitute the decomposition proposed above into the incompressible Navier Stokes equations, which, as a reminder, are the following:

$$\frac{\partial U_i}{\partial x_i} = 0 \tag{2.19}$$

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} \tag{2.20}$$

The first equation represents the continuity equation, while the second one, the momentum equation.

Now, by substituting the decomposition of the velocity into the Navier Stokes equations reported above, we retrieve:

$$\frac{\partial(\widetilde{U}_i + u_i^{''})}{\partial x_i} = 0 \tag{2.21}$$

$$\frac{\partial(\widetilde{U}_i + u_i^{''})}{\partial t} + (\widetilde{U}_j + u_j^{''})\frac{\partial(\widetilde{U}_i + u_i^{''})}{\partial x_j} = -\frac{1}{\rho}\frac{\partial(\widetilde{P} + p^{''})}{\partial x_i} + \nu \frac{\partial^2(\widetilde{U}_i + u_i^{''})}{\partial x_j \partial x_j} \tag{2.22}$$

Making use of the product rule we can rewrite the second term on the left handside as follows:

$$(\widetilde{U}_j + u_j^{''})\frac{\partial(\widetilde{U}_i + u_i^{''})}{\partial x_j} = \frac{\partial(\widetilde{U}_j + u_j^{''})(\widetilde{U}_i + u_i^{''})}{\partial x_j} - (\widetilde{U}_i + u_i^{''})\underbrace{\frac{\partial(\widetilde{U}_j + u_j^{''})}{\partial x_j}}_{=0} \tag{2.23}$$

From the continuity equation, we have that the last term cancels out, and hence:

$$\frac{\partial(\widetilde{U}_i + u_i^{''})}{\partial t} + \frac{\partial(\widetilde{U}_j + u_j^{''})(\widetilde{U}_i + u_i^{''})}{\partial x_j} = -\frac{1}{\rho}\frac{\partial(\widetilde{P} + p^{''})}{\partial x_i} + \nu \frac{\partial^2(\widetilde{U}_i + u_i^{''})}{\partial x_j \partial x_j} \tag{2.24}$$

If we now split the terms of the above equation and we apply an averaging operation, we obtain:

$$\frac{\partial \overline{\widetilde{U}_i}}{\partial t} + \underbrace{\frac{\partial \overline{u_i''}}{\partial t}}_{= 0} + \frac{\partial \overline{\widetilde{U}_i \widetilde{U}_j}}{\partial x_j} + \underbrace{\frac{\partial \overline{\widetilde{U}_i u_j''}}{\partial x_j}}_{= 0} + \underbrace{\frac{\partial \overline{u_i'' \widetilde{U}_j}}{\partial x_j}}_{= 0} + \frac{\partial \overline{u_i'' u_j''}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{\widetilde{P}}}{\partial x_i} - \frac{1}{\rho}\underbrace{\frac{\partial \overline{p''}}{\partial x_i}}_{= 0} + \nu \frac{\partial^2 \overline{\widetilde{U}_i}}{\partial x_j \partial x_j} + \nu \underbrace{\frac{\partial^2 \overline{u_j''}}{\partial x_j \partial x_j}}_{= 0} \tag{2.25}$$

where the time derivative of the fluctuating term cancels out since its average is zero, while the terms multiplying a deterministic component by a stochastic one are cancelled because of their uncorrelation. Finally, rearranging some of the terms and making use of the averaging properties, we obtain the URANS equations:

$$\rho \left[\frac{\partial \widetilde{U}_i}{\partial t} + \widetilde{U}_j \frac{\partial \widetilde{U}_i}{\partial x_j} \right] = -\frac{\partial \widetilde{P}}{\partial x_i} + \frac{\partial}{x_j} \left(\mu \frac{\partial \widetilde{U}_i}{\partial x_j} - \rho \overline{u_i'' u_j''} \right) \tag{2.26}$$

If we compare them to the RANS equations (2.16), we can see that they are essentially the same equations, with the difference that the resolved velocity is now time-dependent, so also some fluctuations are being resolved, and the modelled Reynolds Stresses are based only on the turbulent, stochastic fluctuations.

In the figure below, the different combinations of modelled and resolved turbulence are represented for RANS, URANS, LES and DNS simulations in the energy cascade.

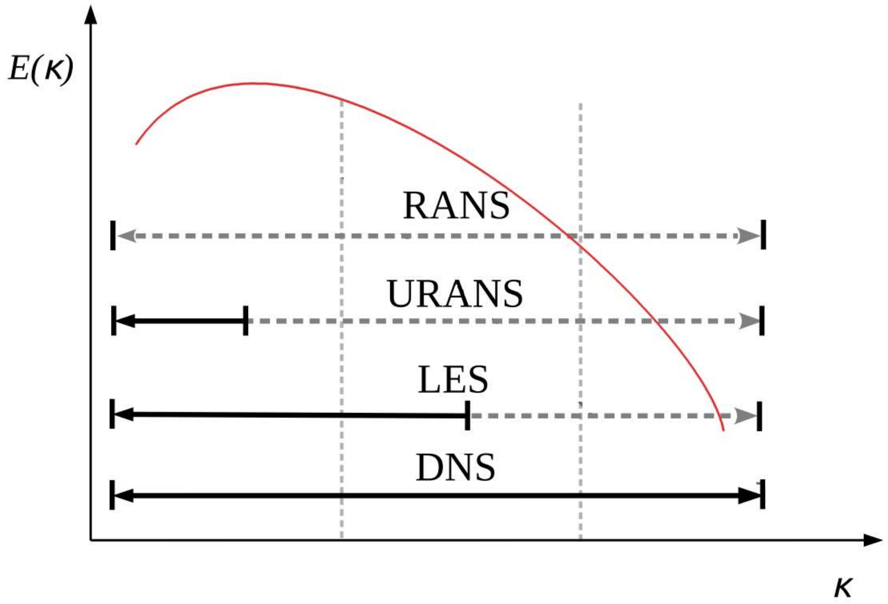


Figure 2.4: Spectrum for turbulent kinetic energy overlaid by turbulence modelling techniques with ranges of resolved and modelled turbulence. Horizontal dashed grey lines: modelled turbulence; solid black lines: resolved turbulence (taken from [3] and modified).

Both RANS and URANS do not resolve turbulence. In fact, it is from the first dashed line that turbulence is being resolved, in the range of -5/3 decay [31]. LES simulations do this in part. Unsteady motions are, however, resolved by URANS simulations, but not by RANS.

2.2.2. Eddy-viscosity models

In Section 2.1 we introduced the problem of *turbulence closure*. As a reminder, when we apply the averaging operation to the Navier Stokes equations, a new term appears, which is the Reynolds Stresses. At this point, there are fewer equations available than the number of variables, so some sort of modelling has to be implemented in order to resolve the RANS/URANS equations.

With regard to closing the RANS equations, it is possible to apply the same idea behind the Newtonian closure for the viscous stress tensor. In that case, the viscous stress tensor is linked to the fluid motion using a flow property called molecular viscosity μ.

A similar parallel is referred to as the *Boussinesq assumption* [8]. Such approximation tries to relate the Reynolds stresses to the fluid motion by introducing a new term called ”eddy” or ”turbulent” viscosity. According to this,

$$\tau_{ij} = \overline{u_i' u_j'} = -\nu_t \left(\frac{\partial \overline{U_i}}{\partial x_j} + \frac{\partial \overline{U_j}}{\partial x_i} \right) + \frac{2}{3}\delta_{ij} k \tag{2.27}$$

where τ_{ij} is known as the Reynolds Stress Tensor. Given the turbulent viscosity field $\nu_t(\boldsymbol{x}, t)$, the equation above yields a closure for the RANS equation, which can now be resolved in terms of $\overline{\boldsymbol{U}}$ and p. The fluctuations are hence being represented by an averaged quantity coupled to the mean flow.

The problem with using such an approximation is that for many flows the accuracy of the hypothesis is poor. That is why RANS equations are generally low in terms of precision. However, for fast calculations and preliminary assessments at an industrial level, RANS simulations are the most widely used.

If the turbulent viscosity hypothesis is accepted as an adequate approximation, what remains is to come up with an appropriate specification of the turbulent viscosity $\nu_t(\boldsymbol{x}, t)$. Such term is treated as a scalar quantity which can be written as the product of a turbulent velocity scale $u^*(\boldsymbol{x}, t)$ and a length scale $l^*(\boldsymbol{x}, t)$:

$$\nu_t = u^* l^*$$

There are different ways of prescribing u^* and l^* and hence of closing the eddy viscosity. In algebraic or zero equation eddy viscosity models (EVM), l^* is specified on the basis of the geometry of the flow. In one equation EVM's one transport equation for a turbulent quantity is solved, while the other turbulent quantity is obtained from an algebraic expression. In two equation EVM's, two turbulent quantities are solved from two transport equations. In the next section, we will explore the details of a specific two-equation EVM, known as $k-\omega$ *SST model*, which is the most widely used as it has shown over the years better general prediction capabilities.

2.2.3. $k - \omega$ SST Model

Before talking about the $k-\omega$ SST model, a brief introduction to other two two-equation models must be done. These models are the $k-\epsilon$ and $k-\omega$ models. The quantities involved are k, the turbulent kinetic energy, ϵ, the rate of dissipation of turbulent kinetic energy, and ω, which is the specific rate of dissipation of turbulent kinetic energy into thermal energy. Both these models come with advantages and limitations.

For example, the $k-\omega$ model is preferred in the logarithmic part of the boundary layer, since it has been shown to be superior in equilibrium adverse pressure gradient flows and in compressible flows [31]. However, in the wake region of the boundary layer, the $k-\epsilon$ model shows better behaviour, mainly because of its lower sensitivity to the arbitrary freestream values ω_f specified for ω outside the boundary layer.

The idea behind the $k-\omega$ SST model was then to switch between the two and use the best characteristics from each model. It was introduced by Menter [29] and the combined implementation of the two models is done by introducing a blending function F_1. In the following steps, we will see how the new model is derived, and how the standard $k-\epsilon$ equations will be transformed to a $k-\omega$ formulation.

The original $k-\omega$ model is:

$$\frac{\mathsf{D}\rho k}{\mathsf{D}t}=\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta^*\rho\omega k+\frac{\partial}{\partial x_j}\left[\left(\mu+\sigma_{k1}\mu_t\right)\frac{\partial k}{\partial x_j}\right] \tag{2.28}$$

$$\frac{\mathsf{D}\rho\omega}{\mathsf{D}t}=\frac{\gamma_1}{\nu_t}\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta_1\rho\omega^2+\frac{\partial}{\partial x_j}\left[\left(\mu+\sigma_{\omega 1}\mu_t\right)\frac{\partial \omega}{\partial x_j}\right] \tag{2.29}$$

while the transformed $k-\epsilon$ model:

$$\frac{\mathsf{D}\rho k}{\mathsf{D}t}=\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta^*\rho\omega k+\frac{\partial}{\partial x_j}\left[\left(\mu+\sigma_{k2}\mu_t\right)\frac{\partial k}{\partial x_j}\right] \tag{2.30}$$

$$\frac{\mathsf{D}\rho\omega}{\mathsf{D}t}=\frac{\gamma_2}{\nu_t}\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta_2\rho\omega^2+\frac{\partial}{\partial x_j}\left[\left(\mu+\sigma_{\omega 2}\mu_t\right)\frac{\partial \omega}{\partial x_j}\right]+2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \tag{2.31}$$

As we can see, an additional cross-diffusion term appears in the ω equation and the modelling constants are different.

Now we multiply equations (2.28) and (2.29) by F_1, and equations (2.30) and (2.31) by (1-F_1), after which the corresponding equations of each set are added together to lead to the new model:

$$\frac{\mathsf{D}\rho k}{\mathsf{D}t}=\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta^*\rho\omega k+\frac{\partial}{\partial x_j}\left[\left(\mu+\sigma_{k}\mu_t\right)\frac{\partial k}{\partial x_j}\right] \tag{2.32}$$

$$\frac{\mathsf{D}\rho\omega}{\mathsf{D}t}=\frac{\gamma}{\nu_t}\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta\rho\omega^2+\frac{\partial}{\partial x_j}\left[\left(\mu+\sigma_{\omega}\mu_t\right)\frac{\partial \omega}{\partial x_j}\right]+2\rho\left(1-F_1\right)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \tag{2.33}$$

The function F_1 is then designed to be equal to 1 in the sublayer and logarithmic region of the boundary layer and to gradually switch to 0 in the wake region. The new $k-\omega$ SST model is then based on a $k-\omega$ formulation, with the original Wilcox model [44] activated in the near wall region and the standard $k-\epsilon$ model activated in the outer wake region and in free shear layers.

The blending of two different models is not the only novelty introduced by the $k-\omega$ SST formulation. In fact, the acronym SST stands for *Shear-Stress Transport*. While Reynolds-stress models feature a transport equation for the turbulent shear stress τ, based on Bradshaw's assumption that the shear stress in a boundary layer is proportional to the turbulent kinetic energy k:

$$\tau=\rho a_1 k \tag{2.34}$$

in two-equation eddy viscosity models, the shear stress is computed as:

$$\tau=\mu_t\Omega \tag{2.35}$$

with $\Omega=(\frac{\partial u}{\partial y})$. For two-equation models, the relation above can be rewritten as

$$\tau=\rho\sqrt{\frac{\mathsf{Production}_k}{\mathsf{Dissipation}_k}}a_1 k \tag{2.36}$$

as derived in [28]. However, the problem here is that when we are dealing with adverse pressure gradient flows the ratio of production to dissipation can be a lot larger than one, which would lead to an overprediction of τ when using equation (2.36). In order to satisfy equation (2.34) within the framework of an eddy-viscosity model, the eddy viscosity is redefined as:

$$\nu_t=\frac{a_1 k}{\max\left(a_1\omega;\Omega F_2\right)} \tag{2.37}$$

where F_2 is a new blending function, equal to one for boundary layer flows and to zero for free shear layers. For example, in an adverse pressure gradient boundary layer, the production of k is larger than its dissipation ($\Omega>a_1\omega$), so equation (2.37) guarantees that equation (2.34) is satisfied; on the contrary, for the rest of the flow (when $\Omega<a_1\omega$) the original formulation $\nu_t=k/\omega$ is used.

If we denote any constant in the original model as ϕ_1 and any constant in the transformed $k-\epsilon$ model as ϕ_2, the relation to find the corresponding constant ϕ of the new model is:

$$\phi=F_1\phi_1+\left(1-F_1\right)\phi_2 \tag{2.38}$$

2.3. Bluff-bodies Flow

In this chapter, we are going to introduce a particular class of flows, which is the wakes behind bluff bodies. In aerodynamics, a ”bluff body” is characterised by a large region of separated flow, a high value of the coefficient of drag, and last but not least, the phenomenon of vortex shedding.

Bluff body wakes are complex, they involve the interactions of three shear layers in the same problem, i.e. a boundary layer, a separating free shear layer, and a wake [45]. The basic problem is still the one that was recognized about two centuries ago in the D’Alembert paradox, which is to ”predict” the drag of a bluff body. In particular, probably the most widely known bluff-body is the circular cylinder. Such a body has been used extensively for research purposes, both due to its engineering significance and the simplicity of setting up such an arrangement in an experimental or computational laboratory. In fact, in a laboratory setting, it is readily available as piping or tubing, structurally convenient for mounting and for pressure tapping, and it can be rotated around its axis [33].

The engineering significance comes from the fact that the alternate shedding of vortices in the near wake induces large fluctuating pressure forces in a direction transverse to the flow and these may cause structural vibrations, acoustic noise, or resonance, which in some cases can lead to failure.

Another body to study could have been the sphere, instead of the cylinder. However, although its similar appeal, it would be less convenient to investigate in a laboratory environment. Also, from such bluff bodies, it is particularly interesting to see the ”vortex shedding” which is more prominent in two-dimensional flow, hence it would be ruined by three-dimensional effects when using a sphere.

The behaviour of the flow in the wake of the cylinder can be described by four main states, as proposed by Zdravkovich [46], which are represented in Figure 2.5. It shows the whole wake development from fully laminar to fully turbulent. The first transition flow, Figure 2.5(a), is when 3D distortions happen in the wake of the cylinder, which makes the laminar vortices in the wake become turbulent. In the subsequent stage, the free shear layers also get turbulent. We can see there is an initial laminar region (L) where two boundary layers are attached to the surface of the cylinder, and then transitioning to turbulent (Tr). The third transition state (c) is characterised by a complex interaction between the separation and transition before the boundary layers become fully turbulent along the separation line. In the last transition state, the shear layers upstream of the separation point get turbulent. As the Reynolds number increases, the shear layers get turbulent even up to the stagnation point. After this last transition, we have a wide downstream region of separated flow, called the wake.

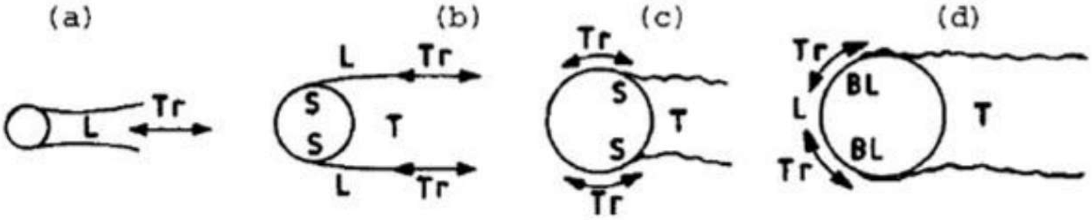


Figure 2.5: History of transition to turbulent state, as Reynolds number increases (taken from [46]).

Another way of categorizing the different flow regimes is through the base suction coefficient ($-C_{pb}$), as proposed by Williamson [45]. Such coefficient is defined by using the pressure at a point 180 degrees from the stagnation point. The plot in Figure 2.6 shows the coefficient as the Reynolds number varies. The different flow regimes can be easily distinguished, and hence summarized as follows:

- up to point A, the wake is made of a steady and symmetric flow with two stationary vortices behind the cylinder, called *laminar steady regime*
- region from A to B is where the vortex shedding starts appearing, with periodic oscillations, and is known as *laminar vortex shedding*
- from B to C is the *3D wake transition regime*

- from regime C to D there is an increasing disorder in the fine-scale three-dimensionalities, which causes a reduction in the two-dimensional Reynolds stresses, a consistent reduction in base suction and an increasing length of the formation region
- region D to E is denominated as Shear-Layer transition regime, and it goes from Re = 1000 to 200000. Due to the developing instability of the separating shear layers from the sides of the body, the 2-D Reynolds stress level increases
- from regime E to G is the *asymmetric reattachment regime* (or critical transition)
- regime G to H is known as *symmetric reattachment regime*, while H to I is called *boundary-layer transition regime*

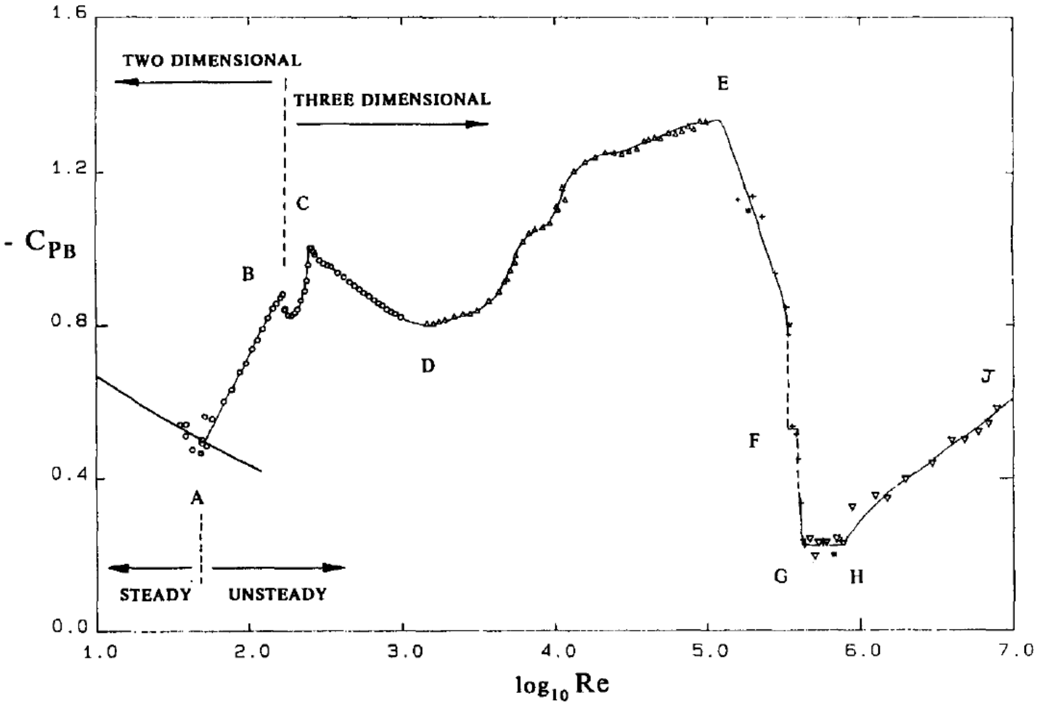


Figure 2.6: Suction coefficient evolution in function of Reynolds number (taken from [45]).

Regardless of the different methods for classifying the different regimes, we can summarize them as follows: the shedding generally starts at a Reynolds number of about 35-49, the wake is 2-dimensional up to Reynolds number equal to 200, while after Reynolds of 1000 the wake becomes turbulent and complex.

The regime that I will investigate in this research is the one associated with the range of Reynolds number from 10^3 to 10^5. The rise in base suction and drag (from D to E in Figure 2.6) was associated with forward movement of the "turbulent-transition point" in the separating free shear layers [33]. Responsible for the turnaround in base suction is the development of Kelvin-Helmholtz instability in the shear layers. Such phenomenon is a fluid instability that happens when there is a velocity shear in a single continuous fluid or a velocity difference across the interface between two fluids. K-H instabilities can be nicely observed in the atmospheres of planets and moons, as also in cloud formations on Earth (see Figure 2.7).

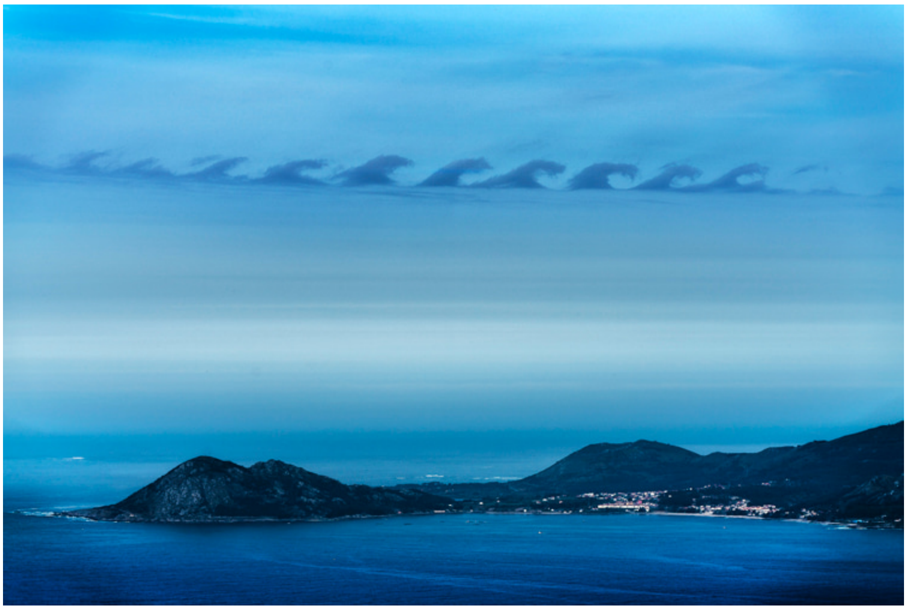


Figure 2.7: Kelvin-Helmholtz instabilities in clouds. Retrieved from: ”*https://www.surfertoday.com/environment/kelvin-helmholtz-the-clouds-that-look-like-breaking-waves*”

2.3.1. Resolution of turbulence in simulations

As was already explained in Section 2.1, DNS simulations are the most accurate simulations since they solve all length and time scales of the flow. However, they are excessively computationally expensive. And that is why turbulence modelling techniques have been developed for faster simulations. It is observable in Figure 2.8 that in DNS every eddy in the wake is resolved, from the large vortex shedding to the tiniest dissipative eddies.

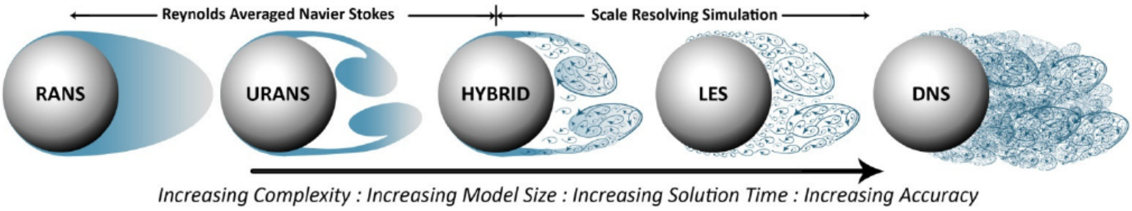

Reynolds Averaged Navier Stokes
Scale Resolving Simulation
RANS
URANS
HYBRID
LES
DNS
Increasing Complexity : Increasing Model Size : Increasing Solution Time : Increasing Accuracy

Figure 2.8: Representation of different scales of resolution, depending on the type of simulation (taken from [15]).

Going from right to left, the accuracy of resolution is decreasing, since more scales are modelled and fewer are resolved. Large Eddy Simulations (LES) are the second most accurate type of simulation. They are here classified as Scale Resolving Simulations, which means that the cutoff between resolved and modelled scales, depends on the size of the eddies. They allow the extraction of additional information that cannot be obtained when applying time-filtering/averaging. In particular, LES are obtained by spatially filtering the Navier-Stokes equations. The large energetic scales are resolved, while the influence of the small unresolved scales is modelled using a subgrid-scale (SGS) model and tuned for the generally universal character of these scales.

Between Scale Resolving and RANS simulations, there is the hybrid RANS-LES formulation. The computational cost of LES is basically independent of the Reynolds number for free-shear flows, weakly dependent on the outer portion of the turbulent boundary layer, but it gets strongly dependent on the innermost layer. Because of this, in most hybrid RANS-LES methods, RANS is applied for the inner portion of the boundary layer while large eddies are resolved by an LES. One of the most known hybrid RANS-LES models is the Detached Eddy Simulation (DES), developed originally by Philippe Spalart.

The name DES is based on the method of covering the boundary layer by RANS model and switching to LES mode in detached regions.

Continuing to the left side of the figure below, we find the URANS simulations. Their mathematical formulation was presented in Section 2.2.1, and it was explained that the unsteady term $\mathrm{D}\boldsymbol{U}/\mathrm{D}t$ is kept in the equations. This allows to capture and resolve the mean velocity, plus the velocity corresponding to the vortex shedding. However, now the smaller and turbulent scales are modeled, and in fact we see in the figure only the two large vortices being resolved.

Last on the left we see the RANS resolved velocity field, which is given only by the mean velocity $\overline{\boldsymbol{U}}$. No vortex shedding is here captured.

3 Data-driven Turbulence Modelling

As it was anticipated in the Introduction of this dissertation, the possibility of leveraging large amounts of data has spurred interest in turbulence models that rely more on the available data than on traditional physical approaches. Many new methodologies have been developed in recent years to overcome the limitations of using the Boussinesq assumption (equation (2.27)) as a turbulence closure for RANS solutions. In particular, regression techniques involving the use of *evolutionary algorithms* and *elastic-net*, but also approaches based on *neural networks* and *random forests*. A brief overview of the here mentioned algorithms is given in the following sections.

3.1. Steady flows

First, some data-driven approaches for steady applications will be presented. The term *steady* means that the corrections are applied to RANS solvers, where all turbulence is modelled and we only solve for the steady $\overline{\boldsymbol{U}}$ mean velocity field. The methods applied until now can be split into two main groups. One group known as tensor modelling, which aims at obtaining an algebraic expression for the Reynolds Stress Tensor, and a second group which just implements a correction to the turbulence model, based on the discrepancies of some quantity of interest.

Let us start with the first type of approach, hence the one involving tensor modelling. In RANS simulations, the turbulent or Reynolds stress τ_{ij} (presented in Section 2.2.2) is an artefact of the Reynolds-averaging procedure, and it must be modelled. The use of time-averaged solutions hence reduces the complexity of the problem, but at the same time, it introduces uncertainties and errors. The modelling is generally based on the Boussinesq assumption, which implicates that the anisotropic part of the Reynolds stresses is linearly dependent on the mean strain rate tensor S_{ij}, scaled by a constant known as eddy-viscosity ν_t. However, this is known to be valid only for a limited amount of flows.

To overcome this limitation, the idea behind the data-driven approach of tensor modelling is to obtain a non-linear relationship between the anisotropy and the mean strain rate tensor $S_{ij} = \frac{1}{2\omega}\left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i}\right)$, but also making it dependent on the mean rotation rate tensor $\Omega_{ij} = \frac{1}{2\omega}\left(\frac{\partial \bar{U}_i}{\partial x_j} - \frac{\partial \bar{U}_j}{\partial x_i}\right)$, as proposed by Pope [30]. He suggested an extension to the assumption that the Reynolds stresses are uniquely related to the rates of strain and local scalar quantities. In particular, the non-dimensionalized anisotropic stress tensor b_{ij}, expressed as

$$b_{ij} = \frac{\overline{u_i' u_j'}}{2k} - \frac{1}{3}\delta_{ij} \tag{3.1}$$

can be substituted by a combination of ten independent tensors $T_{ij}^{(n)}$ and five scalar invariants I_m.

In the general form, we will then have the anisotropy expressed as

$$b_{ij}\left(S_{ij},\Omega_{ij}\right)=\sum_{n=1}^{10} G_n\left(I_1,\ldots,I_5\right)T_{ij}^{(n)} \tag{3.2}$$

where the base tensors and invariants are defined below. For statistically two-dimensional flows, the first three base tensors form a linearly independent basis and only the first two invariants are nonzero. So the set of tensors and invariants reduces as follows:

$$\begin{aligned}
T_{ij}^{(1)} &= S_{ij},\;\; T_{ij}^{(2)} = S_{ik}\Omega_{kj}-\Omega_{ik}S_{kj}\\
T_{ij}^{(3)} &= S_{ik}S_{kj}-\frac{1}{3}\delta_{ij}S_{mn}S_{nm}\\
I_1 &= S_{mn}S_{nm},\;\; I_2=\Omega_{mn}\Omega_{nm}.
\end{aligned}$$

Deriving Explicit Algebraic Reynolds Stress Models (EARSM) then consists in finding a set of scalar functions G_n. This can be done from physical considerations [13][30], but as already anticipated, in the last years machine learning techniques have been applied to find such unknown terms.

Some of the first to address this problem were Weatheritt and Sandberg, who used ***Gene Expression Programming*** for tensor modelling, on different test cases. They first applied it to a backward-facing step [42], then for flows in ducts [43], but also for heat flux closures for a fundamental trailing edge slot [35]. Together with Akolekar et al. [1] they also used GEP for enhanced prediction of wake mixing in low-pressure turbines. While the cases presented above were all frozen approaches, in a later stage they upgraded their framework to a CFD-driven methodology [2].

Generally speaking, GEP is an evolutionary algorithm, that mimics nature's survival-of-the-fittest by iterating non-deterministically to optimise a problem. A population of chromosomes (each representing an algebraic expression) are randomly generated and then evolved according to survival of the fittest. During each successive generation, expressions that better approximate the target values are more likely to mate and provide genetic material to the next generation. Once a stopping criterion has been met, which is usually a finite number of generations, the best expression from the population is taken as the output. The fitness, or cost, of an expression, is calculated as the distance from the training data,

$$Fit\left(\boldsymbol{b}_{ij}^{GEP}\right)=\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{3}\sum_{j\leq i}\left(\left|\boldsymbol{b}_{ij}^{Hi-Fi}-\boldsymbol{b}_{ij}^{GEP}\right|\right)^2 \tag{3.3}$$

$|\cdot|$ is the absolute value and the summation is over the N training data points.

In the same year as GEP was first used for turbulence modelling, also deep neural networks were tried for regressing the anisotropy tensor. Ling, Kurzawski and Templeton [24] presented a novel ***neural network*** architecture, which uses a multiplicative layer with an invariant tensor basis to embed Galilean invariance into the predicted anisotropy tensor. Galilean invariance is an essential property for our model discovery so that the obtained model is independent of the reference system. It is worth noting that if the neural network is not trained on rotationally augmented data, one may observe a deteriorated flow field compared to the baseline RANS solution and/or even encounter converging issues to the specified problem. Galilean invariance can be obtained in two different ways [23]:

- either by using a basis of invariant inputs, thereby embedding the invariance into the model;
- or by training our algorithm on multiple transformations of inputs and targets, which though enlarges significantly our dataset and hence memory and time requirements, and also it achieves only approximate invariance.

Their work was applied in improving Reynolds stress predictions for duct flow and flow over a wavy wall. They managed to predict corner vortices for the duct case and flow separation in the wavy wall case; two key flow features that a linear eddy viscosity model could not predict.

Another machine learning approach is the one based on ***random forests***. Wang, Wu and Xiao [41] trained the Reynolds Stress discrepancies by using DNS databases. The method was then evaluated by two classes of flows: fully developed turbulent flows in a square duct at various Reynolds numbers and flows with massive separations. A drawback of such a method is that the mapping offered by random forests regression is determined based on a number of decision trees. Contrary to the techniques above, it then does not allow obtaining an algebraic expression for the anisotropy.

Another option for targeting the anisotropy tensor b_{ij} is by implementing an ***elastic-net*** regression, as proposed by Schmelzer, Dwight and Cinnella [36] with their SpaRTA framework (Sparse Regression of Turbulent Stress Anisotropy). Their method was tested on four different cases of separating flows: periodic hills, converging-diverging channel, curved backward-facing step and for three-dimensional flows around bluff bodies [18]. An advantage of SpaRTA compared to GEP is that it is a deterministic symbolic identification algorithm, so at each run, when using the same training data we will obtain the same ”best” model. As it happened for the GEP methodology, SpaRTA was developed as a frozen approach at first, but it was later developed as a CFD-driven framework [34].

The so-called ”***frozen***” approach is based on keeping the training quantities fixed. This method is fast since it does not require any additional CFD computation, but it has the main disadvantage that we need the full field data from the high-fidelity source (LES or DNS). Also, due to the offline training, the learned model may cause numerical stiffness once coupled with a CFD code and may not guarantee exact conservation of energy.
The procedure can be generalized as follows:

- calculate the tensor of interest b_{ij} from the training data through equation (3.1);
- apply a regression technique to create algebraic forms for this tensor, based on scalar invariants I_m and tensors $T_{ij}^{(n)}$, in *a priori* setting;
- insert the new algebraic form into a baseline model;
- make observations about the model by testing its predictive capability *a posteriori*

Both the inputs $T_{ij}^{(n)}$, I_m and the targets b_{ij} are built from the high fidelity data $\overline{U}_i$, turbulent kinetic energy k and Reynolds stresses τ_{ij}. To non-dimensionalize the mean strain and mean rotation rate tensors, a high fidelity value for ω is also needed. For this purpose, a so-called k-corrective-frozen RANS algorithm is used, which consists in injecting $\overline{U}_i$, k and τ_{ij} in equation (2.28) and iteratively computing ω from it. That is why we address this methodology as the ”frozen” approach, because the training data is all provided *a priori* and there is no CFD calculation that updates this data.

The ***CFD-driven*** alternative is a method to diversify the way that Explicit Algebraic Reynolds Stress Models are created. The models obtained from GEP/SpaRTA are explicit and can be instantly and automatically implemented into RANS solvers during the model development process. This way, the performance of candidate models is already evaluated during the training loop, hence there is no need to test the models individually *a posteriori*. The resulting ”CFD-driven” models are then ready to be implemented into industrial design tools. Another benefit of ”CFD-driven” training is that, in contrast to the ”frozen” technique, the definition of the cost functions is more flexible [47]. In fact, they can be tailored to capture any important flow feature of interest, without being restricted only to quantities that

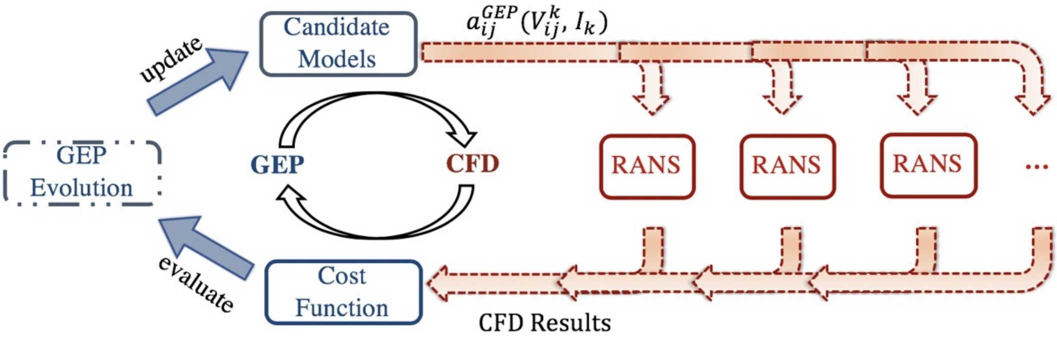


Figure 3.1: Representation of the CFD-driven framework for GEP (taken from [47]).

are part of the closure terms such as anisotropy tensors. This comes very handy when for example only a few profiles from high-fidelity or experimental data are available.

In Figure 3.1, the CFD-driven framework is shown for simplicity. Here b_{ij} is referred to as a_{ij}, which has the same definition as equation (3.1). When using CFD-driven SpaRTA, the methodology is the same, with the only difference being that the regression is through elastic-net regression. It is possible to observe that once a candidate model is obtained, it is immediately run through a RANS calculation, which ensures that the final models will already be numerically stable and realizable. It also guarantees the coupling of the mean field and turbulent transport equations at any stage of the training process, thus respecting the conservation of mean kinetic energy. It was noted that the CFD-driven model exhibits a much simpler form compared to the frozen model, as the coefficient functions contain fewer high-order invariants, even though the training of the frozen model is based on the same data. This is because the higher-order terms in the frozen model usually result in very high-amplitude values on some points in RANS calculations, reducing numerical stability. Therefore, such terms fail to survive in the CFD-driven evolution.

It was earlier anticipated that there are two different options for improved turbulence modelling. The methodologies presented above based on tensor modelling, and the methods based on correcting the turbulence models themselves. The first ones to do so were Holland, Baeder and Duraisamy [16] with their ***field inversion*** technique. Their objective to reduce model form errors in a RANS setting is targeted through a correction applied to the production term. The learning step is here integrated into the field inversion process. This integrated approach makes sure that the process generates learnable model discrepancy, and thus results in a consistent machine-learned model that can be embedded in a predictive setting. This methodology can be summarized in the following points:

1. A cost function (J) is formulated which represents the misfit of a RANS quantity of interest with the data, such for example lift coefficient, skin friction, surface pressure distribution, *etc*.
2. A model discrepancy correction function $\beta(x)$ is applied to the turbulence model as a field variable (which is defined at every point in the computational domain). In this case, it was implemented for improving the Spalart-Allmaras (SA) turbulence model [38].
3. The best correction field for the turbulence model is discovered using a gradient-based optimization approach, such that the correction can be reproduced from an appropriate selection of variables from the model.

It was demonstrated that the proposed procedure was able to generate learnable model discrepancy functions and improve the prediction of RANS solvers.

In the following subsection, a more detailed explanation of the SpaRTA framework will be provided, since it will be implemented as part of our methodology.

3.1.1. SpaRTA: Sparse Regression of Turbulent Stress Anisotropy

The methodology developed by Schmelzer, Dwight and Cinnella [36] aims at inferring algebraic stress models for the closure of RANS equations, derived directly from LES or DNS data. The term SpaRTA stands for *Sparse Regression of Turbulent Stress Anisotropy*, which highlights that sparsity-promoting regression techniques will be used. This is preferable since it was observed that models using only a few nonlinear terms show low training and prediction error, as well as high numerical robustness for industrially relevant flow cases [42].

As anticipated in the previous section, the main deficiencies of Linear Eddy Viscosity Models come from the fact that they are based on the Boussinesq approximation (equation (2.27)). The RANS model chosen for modification is a slight variant of the SST. The SST is known to over-predict the size of recirculation regions making it an ideal target for improvement. One way to improve the results is then by adding a correction built from high-fidelity data. After having quantified the model-form error, the method proposed in SpaRTA consists of the general steps presented above as the ”frozen” procedure. The novelty of this work is that not only a correction of the stress-strain relation is identified (here referred to as b_{ij}^{Δ}), but also one for the turbulent transport equations (R). We recall that the Reynolds stresses, which are the subject of modelling, are defined as

$$\tau_{ij} = 2k\left(b_{ij} + \frac{1}{3}\delta_{ij}\right) \tag{3.4}$$

with the baseline model $b_{ij}^{o} = -\frac{\nu_t}{k}S_{ij}$, as proposed by Boussinesq. By injecting the high-fidelity data $\overline{\boldsymbol{U}}^*$, k^*, τ_{ij}^* and ν_t^*, into (3.4), it is possible to obtain the residuals of the baseline turbulence model as an additive term b_{ij}^{Δ}:

$$b_{ij}^{*} = -\frac{\nu_t^*}{k^*}S_{ij}^* + b_{ij}^{\Delta} \tag{3.5}$$

As we can see from the relation above, we also must prescribe a high fidelity value ν_t^*, and that is obtained by passively solving the ω transport equation substituting $\overline{\boldsymbol{U}}^*$, k^* and τ_{ij}^*, recalling that $\nu_t^* = \frac{k^*}{\omega^*}$. Since the addition of b_{ij}^{Δ} also alters the production of turbulent kinetic energy P_k, a correction is introduced to account also for the residuals of the k equation. The residual is equivalent to an additive correction term, defined as R, which leads to an augmented $k-\omega$ SST model

$$\frac{\mathsf{D}k}{\mathsf{D}t} = P_k + R - \beta^*\omega k + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_{k1}\nu_t)\frac{\partial k}{\partial x_j}\right] \tag{3.6}$$

$$\frac{\mathsf{D}\omega}{\mathsf{D}t} = \frac{\gamma_1}{\nu_t}(P_k + R) - \beta_1\omega^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_{\omega 1}\nu_t)\frac{\partial \omega}{\partial x_j}\right] \tag{3.7}$$

where the turbulent kinetic energy production P_k is augmented by b_{ij}^{Δ} and bounded following Menter's limiter

$$P_k = \min\left(-2k\left(b_{ij}^{o} + b_{ij}^{\Delta}\right)\frac{\partial U_i}{\partial x_j}, 10\beta^*\omega k\right) \tag{3.8}$$

Now that the corrections b_{ij}^{Δ} and R are identified, the next step is to build a functional model targeting such quantities. The approach used is the one presented by equation (3.2), so that the anisotropy correction can be expressed as a combination of base tensors and invariants.

We introduced earlier the term R, which provides local information to correct the transport equations. Depending on its sign, it can either increase or decrease the net production P_k locally. It then acts as an additional production or dissipation term, which can overcome the error in k. It is modelled here in a similar way as the turbulent production

$$R = 2kb_{ij}^{R}\frac{\partial U_i}{\partial x_j} \tag{3.9}$$

which has the advantage that it can be represented by using the framework of nonlinear eddy viscosity models. Two different models will then be obtained, one for modelling b_{ij}^{Δ} and one for b_{ij}^{R}.

As can be seen from Figure 3.2, the first step of SpaRTA is to build a library of candidate functions. Starting from the invariants, a vector based on their combinations is generated:

$$\begin{aligned}\mathcal{B} =& \big[1, I_1, I_2, I_1^2, I_2^2, I_1^2 I_2^3, I_1^4 I_2^2, I_1 I_2^2, I_1 I_2^3\\ & I_1 I_2^4, I_1^3 I_2, I_1^2 I_2^4, I_1^2 I_2, I_1 I_2, I_1^3 I_2^2, I_1^2 I_2^2\big]^T\end{aligned} \tag{3.10}$$

In order to regress models for B_{ij}^{Δ} each function of $\mathcal{B}$ is multiplied with each base tensor $T_{ij}^{(n)}$, leading to the library of tensorial candidate functions

$$\mathcal{C}_{b_{ij}} = \left[T_{ij}^{(1)}, T_{ij}^{(2)}, \ldots, I_1^2 I_2^2 T_{ij}^{(3)}\right]^T \tag{3.11}$$

In a similar way, the library $\mathcal{C}_R$ is obtained by multiplying each function in $\mathcal{C}_{b_{ij}}$ with the mean velocity gradient tensor $\frac{\partial U_i}{\partial x_j}$:

$$\mathcal{C}_R = \left[T_{ij}^{(1)}\partial_j U_i, \ldots, I_1^2 I_2^2 T_{ij}^{(3)}\partial_j U_i\right]^T \tag{3.12}$$

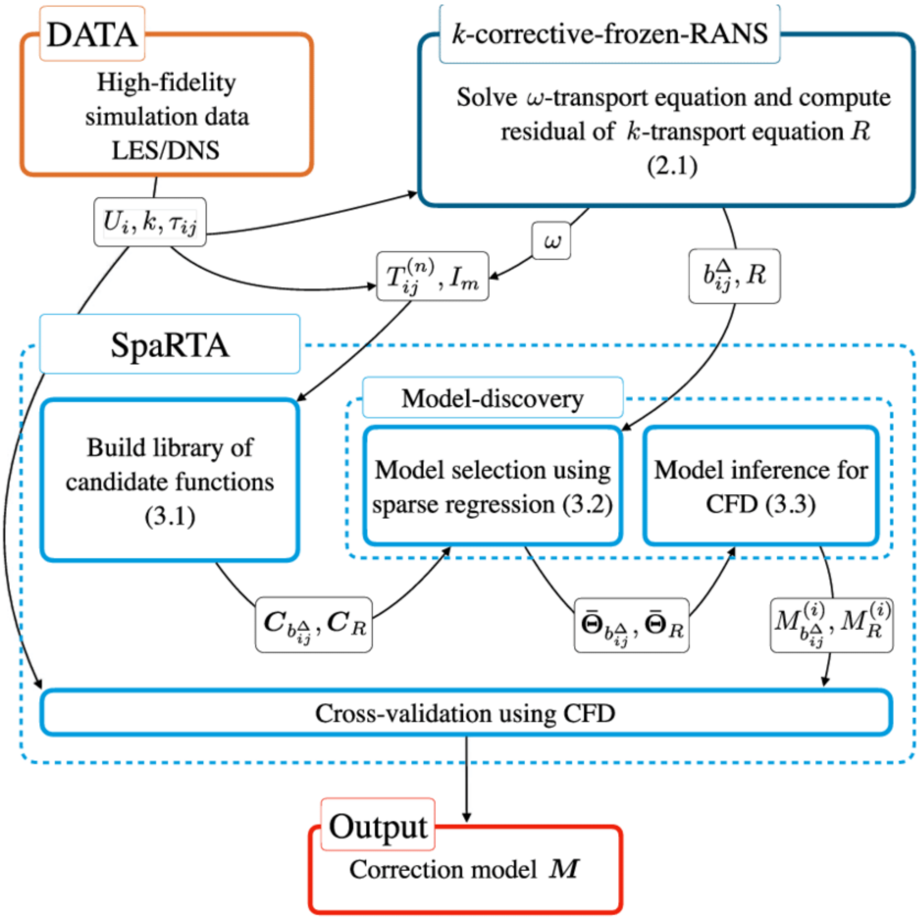

DATA
High-fidelity simulation data LES/DNS
k-corrective-frozen-RANS
Solve ω-transport equation and compute residual of k-transport equation R (2.1)
U_i, k, τ_{ij}
ω
$T_{ij}^{(n)}, I_m$
b_{ij}^Δ, R
SpaRTA
Model-discovery
Build library of candidate functions (3.1)
Model selection using sparse regression (3.2)
Model inference for CFD (3.3)
$C_{b_{ij}^\Delta}, C_R$
$\bar{\Theta}_{b_{ij}^\Delta}, \bar{\Theta}_R$
$M_{b_{ij}^\Delta}^{(i)}, M_R^{(i)}$
Cross-validation using CFD
Output
Correction model M

Figure 3.2: Schematic of SpaRTA framework (taken from [36]).

Step 2 is known as the *model selection*, where, given the defined libraries, the target data $\Delta = b^\Delta$ or R is linearly regressed by finding the coefficient vector Θ

$$\Delta = \mathcal{C}_\Delta \Theta \tag{3.13}$$

which represents a large, overdetermined system of equations. The idea proposed by [36] is then to use sparsity-promoting regularisation. In particular, the following elastic net formulation is used

$$\boldsymbol{\Theta} = \arg\min_{\hat{\boldsymbol{\Theta}}} \left\| \boldsymbol{C}_\Delta \hat{\boldsymbol{\Theta}} - \boldsymbol{\Delta} \right\|_2^2 + \lambda\rho\|\hat{\boldsymbol{\Theta}}\|_1 + 0.5\lambda(1-\rho)\|\hat{\boldsymbol{\Theta}}\|_2^2 \tag{3.14}$$

which blends the l_1- and the l_2- norm regularisation, through the mixing parameter $\rho \in$ [0,1]. The first one, known as Lasso-regression, promotes sparsity by permitting only a few nonzero coefficients and shrinking the rest to zero, while the l_2 is called Ridge-regression and imposes relatively small coefficients without setting them equal to zero. The optimisation is done on different values of ρ and λ, and as the value of λ decreases, the amount of nonzero coefficients increases, making the discovery of sparse models possible. The selection step of SpaRTA ends with obtaining the set of D unique abstract model forms $\mathcal{D}_\Delta = \left\{ \overline{\boldsymbol{\Theta}}_\Delta^d \mid d = 1, \ldots, D \right\}$.

The third step is the *model inference*. While the abstract models $\mathcal{D}_\Delta$ were found using standardised candidates, an additional regression is now performed using the unstandardised candidate functions for each subset determined by the abstract model forms in $\mathcal{D}_\Delta$. This is done with the scope of defining a model with the correct units. In this case, the regression implemented is a Ridge regression so that the magnitude of the nonzero coefficients is shrunk, but they are not set to zero.

3.2. Unsteady flows

In Section 3.1, some data-driven methods for correcting RANS simulations are presented. Concerning unsteady RANS improvements, little literature is available, mainly due to the additional complexities that still have not been addressed and investigated. However, in this section, a new framework for developing closures for URANS calculations is briefly explained [21] [22]. In unsteady RANS simulations, the deterministic scales associated with vortex shedding are resolved, while the stochastic scales of pure turbulence are modelled. The methodology here proposes then to separate the stochastic from the deterministic length scales and then develop a turbulence closure only for the stochastic scales.

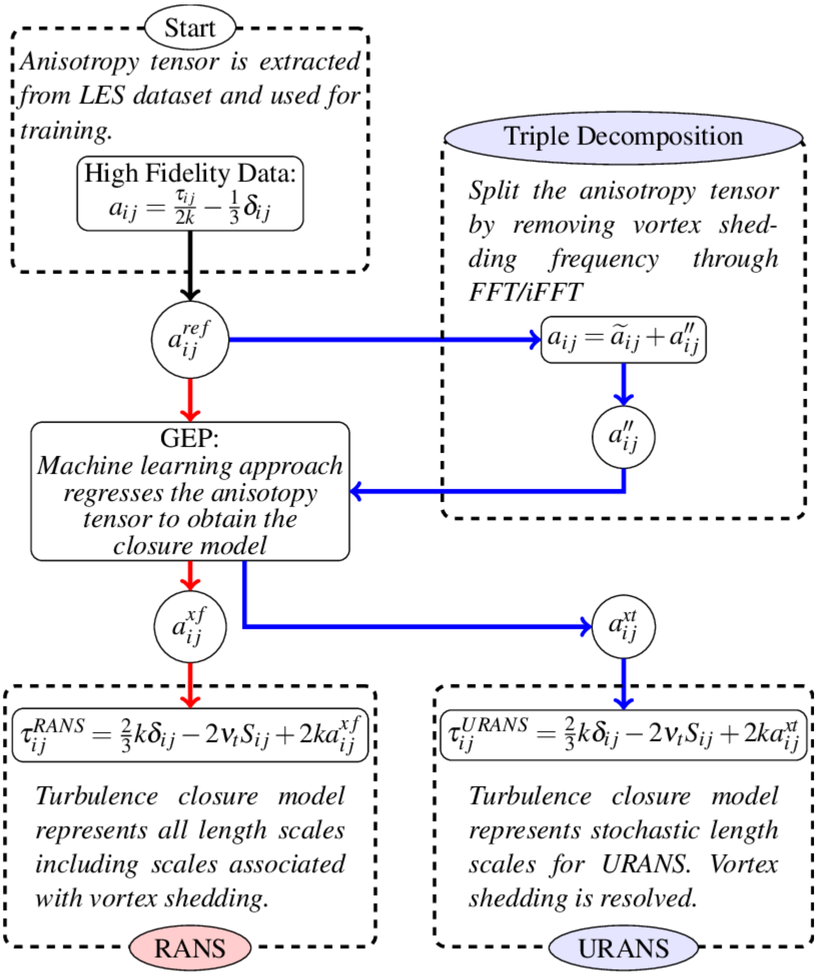

Start
Anisotropy tensor is extracted from LES dataset and used for training.
High Fidelity Data: $a_{ij} = \frac{\tau_{ij}}{2k} - \frac{1}{3}\delta_{ij}$
a_{ij}^{ref}
Triple Decomposition
Split the anisotropy tensor by removing vortex shedding frequency through FFT/iFFT
$a_{ij} = \tilde{a}_{ij} + a''_{ij}$
a''_{ij}
GEP:
Machine learning approach regresses the anisotopy tensor to obtain the closure model
a_{ij}^{xf}
a_{ij}^{xt}
$\tau_{ij}^{RANS} = \frac{2}{3}k\delta_{ij} - 2\nu_t S_{ij} + 2ka_{ij}^{xf}$
Turbulence closure model represents all length scales including scales associated with vortex shedding.
RANS
$\tau_{ij}^{URANS} = \frac{2}{3}k\delta_{ij} - 2\nu_t S_{ij} + 2ka_{ij}^{xt}$
Turbulence closure model represents stochastic length scales for URANS. Vortex shedding is resolved.
URANS

Figure 3.3: Representation of Sandberg framework for unsteady applications (taken from [21]).

As was previously introduced in tensor modelling approaches, the target of modelling is the anisotropy, here referred to as a_{ij}, while the b_{ij} notation is used in [36]. From Figure 3.3, it is shown how the data is first collected from high fidelity datasets and then split into a deterministic $\widetilde{a}_{ij}$ and a stochastic a''_{ij} component as

$$a_{ij} = \widetilde{a}_{ij} + a''_{ij} \tag{3.15}$$

which are defined after the triple decomposition in equation (2.17). For instance, the stochastic component a''_{ij} is built from the stochastic velocity contribution u'' as

$$a''_{ij} = \frac{\overline{u''_i u''_j}}{2k''} - \frac{1}{3}\delta_{ij} \tag{3.16}$$

where $k'' = 0.5 \cdot \left(\overline{(u'')^2} + \overline{(v'')^2} + \overline{(w'')^2}\right)$.

Following this novel unsteady framework, as can be observed in Figure 3.3, in the machine learning regression step, only the stochastic part a''_{ij} is the one which is targeted for improvement. The resulting modelled anisotropy a^{xt}_{ij} is then added as a correction term to the modified turbulence model

$$\tau_{ij}^{URANS} = \frac{2}{3}k\delta_{ij} - 2\nu_t S_{ij} + 2ka^{xt}_{ij} \tag{3.17}$$

The entire process is essentially the same as that of the GEP frozen approach. The sole distinction is that only the stochastic portion of the anisotropy is the target here.

4 Scientific Gaps and Research Questions

After a careful literature review, the following limitations from current works were identified:

- Random forests and field inversion approaches do not produce a tangible algebraic expression, so making it harder to implement in a solver.
- Gene Expression Programming is non-deterministic, which means that at each run a different expression is obtained, and not necessarily the ”best” model.
- CFD-driven techniques require a longer training time, due to the coupled RANS simulations that need to be run.

At the same time, some positive considerations and takeaways can be highlighted:

- The general extension of the anisotropy proposed by Pope allows the introduction of new terms and hence allows not to be limited by the Boussinesq assumption.
- SpaRTA has shown good performance in regressing models for the anisotropy tensor, obtaining algebraic, interpretable expressions [36].
- Little work has yet been done on addressing unsteady simulations.

As a consequence, it would be of interest to investigate the performance of SpaRTA frozen methodology, when applied to improve unsteady simulations. The main research question comes then from this motivation and can be formulated as:

”How can we use SpaRTA to improve turbulence modelling in an unsteady RANS framework?” (Q1)

As already anticipated in Section 2.2, when running URANS simulations they are solved both for the mean velocity ($\overline{U}$, as in RANS), but also for a coherent unsteady component ($\widetilde{u}$), which are merged as: $\widetilde{U} = \overline{U} + \widetilde{u}$. Since additional resolved data is available, a first step can be to study how it can potentially be used to improve URANS modelling. In particular, whether it is possible to relate such component $\widetilde{U}$ to the stochastic part of the anisotropy b''_{ij}, following a similar procedure to the one presented in Section 3.2. The main research question can then initially be approached by answering the following sub-question:

”How can we leverage the resolved coherent velocity field to improve the anisotropy tensor prediction? (Q1.1)

Again, this can be split into two other sub-questions. The first one which will give us an initial indication of the correlation between the resolved field and the stochastic quantities:

”Is there any correlation between $\widetilde{U}$ and b''_{ij}?”
(Q1.1a)

If such a correlation exists, then SpaRTA can be used for regressing a model between these quantities. So the following sub-question that could be addressed is:

”How can SpaRTA find such a model between $\widetilde{U}$ and b''_{ij} and how accurate is this model?”
(Q1.1b)

Such an attempt is not trivial given the uncertainty related to modelling of b''_{ij} based on $\widetilde{U}$.

Therefore, if the modelling from $\widetilde{U}$ to b''_{ij} shows limitations, then an analogous approach to the one proposed by Weatheritt and Sandberg [21] will be studied, but using SpaRTA rather then Gene Expression Programming. So the successive sub-question to the main research question (Q1) comes naturally as:

”How can we use SpaRTA to model the relation between the mean flow quantities (for example $\overline{U}$) and the anisotropic Reynolds stress tensor b''_{ij}?”
(Q1.2)

In addition, it must be assessed whether the model discovery has positive outcomes, and the following subquestion answered:

”How accurate is the SpaRTA-based model?”
(Q1.2a)

It was explained earlier in Chapter 2 that the main limitations of turbulence modelling come from the non-general validity of the Boussinesq assumption. It is clear that our model, in order to be satisfactory, must improve the anisotropy prediction when compared to the Boussinesq assumption. So the performance of the SpaRTA-based model will be assessed by comparing it to the Boussinesq-based model.

5

Simulation of Flow around a Cylinder

5.1. Computational Domain

The flow that was investigated, and at which the method developed will be applied is the flow around a circular cylinder. This type of flow has been of academic importance over the years, employed as a prototype to study bluff body flows. Because of the different regimes at which it undergoes (presented in Section 2.3), it shows diverse physics at every distinct Reynolds number range. Particularly interesting is the *vortex shedding*, which is an oscillating process that takes place when a fluid such as air or water flows past a bluff (as opposed to streamlined) body at certain velocities, depending on the size and shape of the body. This phenomenon starts appearing when the Reynolds number equals 40. From this point on, vortices are created at the back of the body and detach periodically from either side of the body forming a Von Kármán vortex street.

The computational domain is a rectangular box whose dimensions are reported in the figure below.

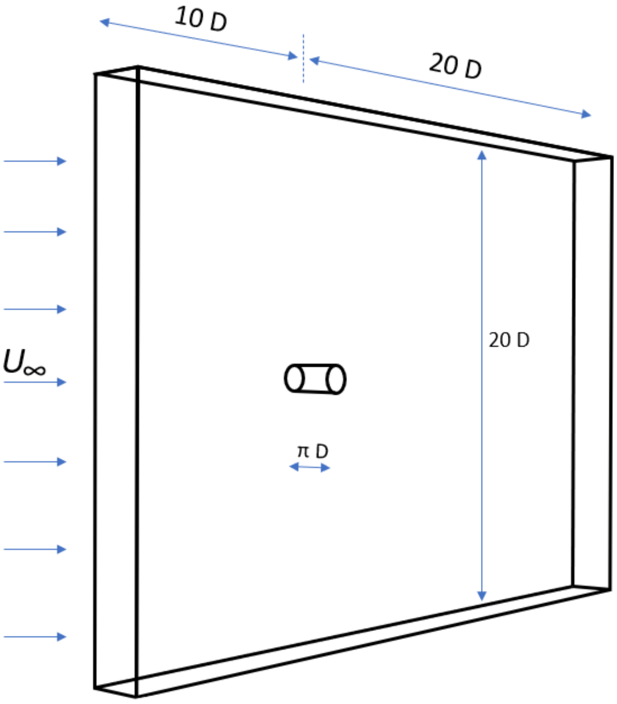


Figure 5.1: Dimensions and boundaries of study domain.

The dimensions were taken from literature [12]. The domain shown in Figure 5.1 was then filled with hexahedral cells, in a non-orthogonal block structure. The cells in the spanwise (z) direction were distributed uniformly, while in the x and y directions they were highly refined around the cylinder and in the wake, then gradually increasing size towards the boundaries of the domain. In the spanwise direction, simulations were run with 40 grid points, which led to a total number of cells equal to about 2 million. In Figure 5.2 it is reported a picture showing the refinement around the cylinder. What it was aimed at, was obtaining a y^+ value smaller than 1 and a very small expansion ratio between successive cells.

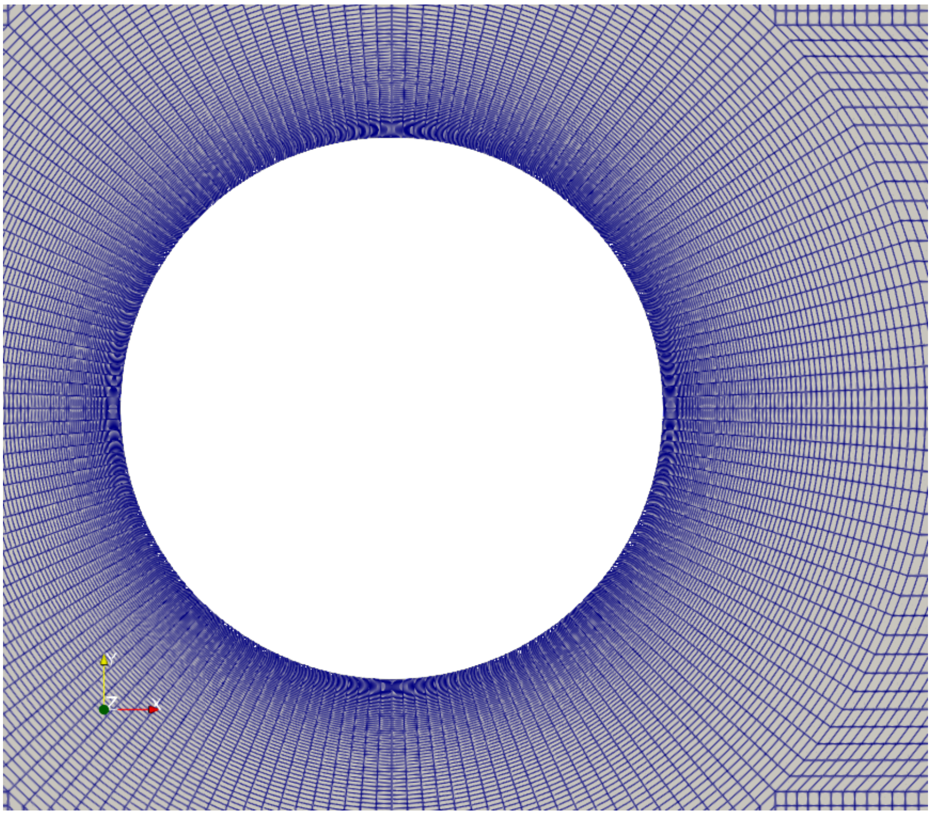


Figure 5.2: Refinement of mesh around the cylinder.

The y^+ value introduced just above is a non-dimensional measurement of the distance from a wall. It is defined as follows:

$$y^+ = \frac{y\, u_T}{\nu} \tag{5.1}$$

where y is the absolute distance from the wall, ν is the kinematic viscosity and u_T is the so-called friction velocity, which is based on the wall shear stress τ_w and prescribed as $\sqrt{\tau_w/\rho}$.
The y^+ is used in CFD simulations to describe the height of the first grid element next to a wall. This non-dimensional parameter was introduced because it was observed from experiments that flows of all scales tend to exhibit very similar flow patterns as the flow approaches the wall, in that thin region known as *boundary layer*. Turbulent flows are significantly affected by the presence of walls, hence an accurate representation of the near-wall region determines a successful prediction of wall-bounded turbulent flows. It is then of essential importance to know the value of y^+ in our simulations. In fact, there are two different ways for dealing with near wall region, and hence obtain reliable results:

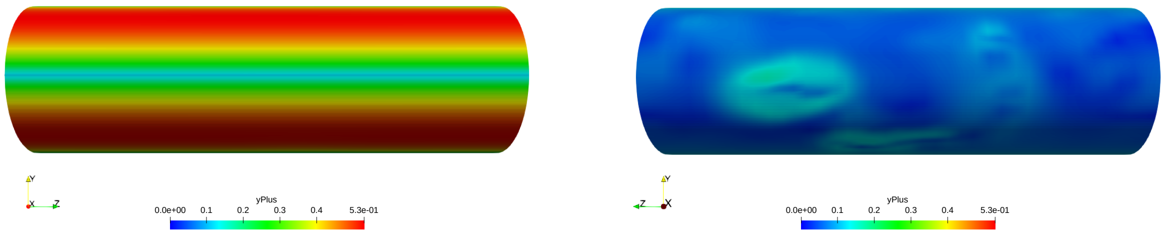
1. either by integrating turbulence to the wall. In this way, also the viscosity-affected region is resolved with all the mesh down to the wall, including the viscous sublayer. For solving the near-wall region, the first cell must be placed in the viscous sublayer, hence y^+ must be equal or smaller

than 1. This approach is usually followed when we are interested on the forces acting on the wall (for example around airfoils);

2. or by using *wall functions*, which model the near wall region. In this case though, the first cell needs to be placed in the log-law region to ensure the accuracy of the results. This region is characterised by a y^+ value comprised between 30 and 300, otherwise the model is invalid. The use of wall functions obviously leads to a reduction in cells and so in computational time. However, these are usually implemented when we are more interested in mixing rather than the forces on the wall.

Whatever approach is implemented, it is clear the importance of knowing the value of y^+ in order to apply the proper modelling.

In our case, since we are also interested about the forces acting on the cylinder, a y^+ smaller than 1 was targeted. As we can observe from the figures below, the values of y^+ are overall smaller than 1, with a maximum value of 0.51 (red areas).



(a) y^+ on the front view of the cylinder.

(b) y^+ on the back view of the cylinder.

Figure 5.3: y^+ values on the cylinder surface

As it can be observed, the higher values of y^+ are where the velocity magnitude is higher, hence at the front of the cylinder, when the flow accelerates due to the curvature of the cylinder.

5.2. High-Fidelity Simulations

As I anticipated in Chapter 3, data-driven tensor modelling is built from high-fidelity data. In this case, the data will be from Large Eddy Simulations (LES). Both RANS and LES solve the averaged Navier-Stokes equations, in a very different way though. In fact, while in RANS the average is a statistical/temporal average, in LES it is a spatial average over a small volume. The non-linearity of the Navier-Stokes equations causes their averaged forms to contain additional terms for which models are required. I talked earlier about turbulence models for RANS, while for LES the modelling is referred to as subgrid or subfilter scale models.

5.2.1. Governing equations

The Navier-Stokes equations are able to describe even the most complex turbulent flow, and hence no turbulence model would be needed. However, DNS is impracticable for high Reynolds number flow problems. Reynolds averaged equations were applied for reducing computational time. They showed though limited success, in particular for highly complex turbulent flows, such as those in bluff-body aerodynamics. This is because these methods have to describe the whole spectrum of turbulent eddies. The concept of LES was developed for overcoming this drawback. In fact, large eddy simulations are so called because only the small eddies are modelled by a subgrid-scale model, whereas the large energy-carrying eddies are directly computed. The development of appropriate models is simplified since small eddies are more universal, random, homogeneous and isotropic.

For obtaining this separation between large and small-scale motions, the three-dimensional, time-dependent Navier-Stokes equations are filtered.

$$u = \bar{u} + u' \tag{5.2}$$

The governing equations are then given by:

$$\begin{aligned}
&\frac{\partial \bar{u}_i}{\partial x_i} = 0, \\
&\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\bar{u}_i \bar{u}_j\right) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\frac{1}{Re}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)\right] - \frac{\partial \tau_{ij}}{\partial x_j}, \\
&\tau_{ij} = \overline{u_i u_j} - \bar{u}_i u_j,
\end{aligned} \tag{5.3}$$

where $\bar{u}_i$ is the velocity component of the resolved scales, $\bar{p}$ is the corresponding pressure and Re is the Reynolds number. The filtering procedure leads to the governing equations for the resolvable scales of the flow field, analogously to the averaging procedure which was leading to URANS (equations 2.26). Due to the filtering of the non-linear convective fluxes, an additional term for the non-resolvable subgrid-scale stresses appears: τ_{ij}. Such quantity describes the influence of the small-scale structures on the larger eddies. Subgrid scale models are then aimed at modelling this quantity. In the following Subsection, the specific model used in these simulations will be discussed.

5.2.2. Subgrid scale model

There is a need to use models for approximating the stress tensor that comes out of the filtering operation. Many different models have been developed over the years. In particular, the most widely known is the Smagorinsky model which takes its name from its inventor. It is the simplest model since it is a 0-equations model, and hence also the most extensively used, because of its simplicity and low computational cost. Though it is not a particularly good choice for wall-bounded flows, it can be quite adequate for flows far from solid boundaries. Similarly to the Boussinesq assumption, the sub-grid scale stress tensor τ^{SGS} is modelled as a linear relationship with the resolved strain-rate tensor $\overline{S}$ as follows [37]:

$$\tau_{ij}^{SGS} = -2\nu_{SGS}\overline{S} + \frac{1}{3}\delta_{ij}\tau_{kk}^{SGS} \tag{5.4}$$

where ν_{SGS} is the sub-grid scale eddy viscosity, and is computed as

$$\nu_{SGS} = (C_S\Delta)^2|\overline{S}| \tag{5.5}$$

where Δ is the filter width, C_S is the Smagorinsky constant and $|\overline{S}|$ is the magnitude of the strain-rate tensor. The constant C_S requires adjustment for different flows.

Despite this short introduction to the Smagorinsky model, the model I used for my simulations to collect the high-fidelity set of data is the ***Dynamic Kinetic Energy Transport*** model [14]. It was chosen because it showed to overcome some of the limitations of the other available models (Smagorinsky, Smagorinsky-Lilly, Dynamic Smagorinsky, etc...). This though comes with extra computational cost, in fact an additional transport equation for the subgrid scale kinetic energy must be solved. Such model is implemented in *OpenFOAM* and named as *dynamicKEqn*. The novelty comes then from the fact that the following transport equation for the subgrid-scale kinetic energy k_{SGS} has to be solved:

$$\frac{\partial k_{SGS}}{\partial t} + \bar{u}_i\frac{\partial k_{SGS}}{\partial x_i} = -\tau_{ij}^{SGS}\frac{\partial \bar{u}_i}{\partial x_j} - \varepsilon + \frac{\partial}{\partial x_i}\left(\nu_{SGS}\frac{\partial k_{SGS}}{\partial x_i}\right) \tag{5.6}$$

In general, dynamic models are strongly recommended because the coefficients are dynamically computed in function of space and time, contrary to older approaches where such coefficients were taken as constants. In addition, solving a supplementary equation for the turbulent kinetic energy improves the prediction and also gives information about the modeled TKE. Having time and space varying coefficients allows to get around one of the major drawbacks of earlier models, which was their inability to model correctly the unresolved subgrid stresses in diverse types of turbulent flow fields with a unique universal constant. Secondly, thanks to the dynamic evaluation, the model coefficient can turn negative in certain regions of the flow field and thus, suggests that it can mimic backscatter of energy from the subgrid-scales to the resolved scales.

The Dynamic Kinetic Energy transport model not only addresses the problem of the conventional fixed-coefficient model but also another weakness of the Smagorinsky model. The Smagorinsky's time-independent, algebraic eddy viscosity model used in Germano *et al.*'s dynamic model is derived by presuming local equilibrium between the SGS energy production and dissipation rate. As such, non-local and history effects of the turbulence evolution are completely ignored. Thanks to the coupling of equation (5.6), the direct computation of the SGS kinetic energy is expected to account for the local details of the flow structure and the turbulence development history.

The three terms on the right-hand-side of equation (5.6) describe, respectively, the production rate, the dissipation rate and the transport rate of k_{SGS}. The sub-grid stresses τ_{ij}^{SGS} are modeled in terms of the SGS eddy viscosity ν_{SGS} as:

$$\tau_{ij}^{SGS} = -2\nu_{SGS}\overline{S} + \frac{2}{3}\delta_{ij}k_{SGS} \tag{5.7}$$

where

$$\nu_{SGS} = c_\nu\sqrt{k_{SGS}}\overline{\Delta} \tag{5.8}$$

Equation (5.6) is closed by implementing a model for the dissipation rate ε. By simple scaling arguments, ε is usually modeled as

$$\varepsilon = c_\varepsilon\frac{k^{1.5}}{\overline{\Delta}} \tag{5.9}$$

The dynamic modelling method then aims at obtaining appropriate values of the coefficients c_ν and c_ε.

5.2.3. Boundary conditions

Table 5.1: Boundary conditions for LES simulation.

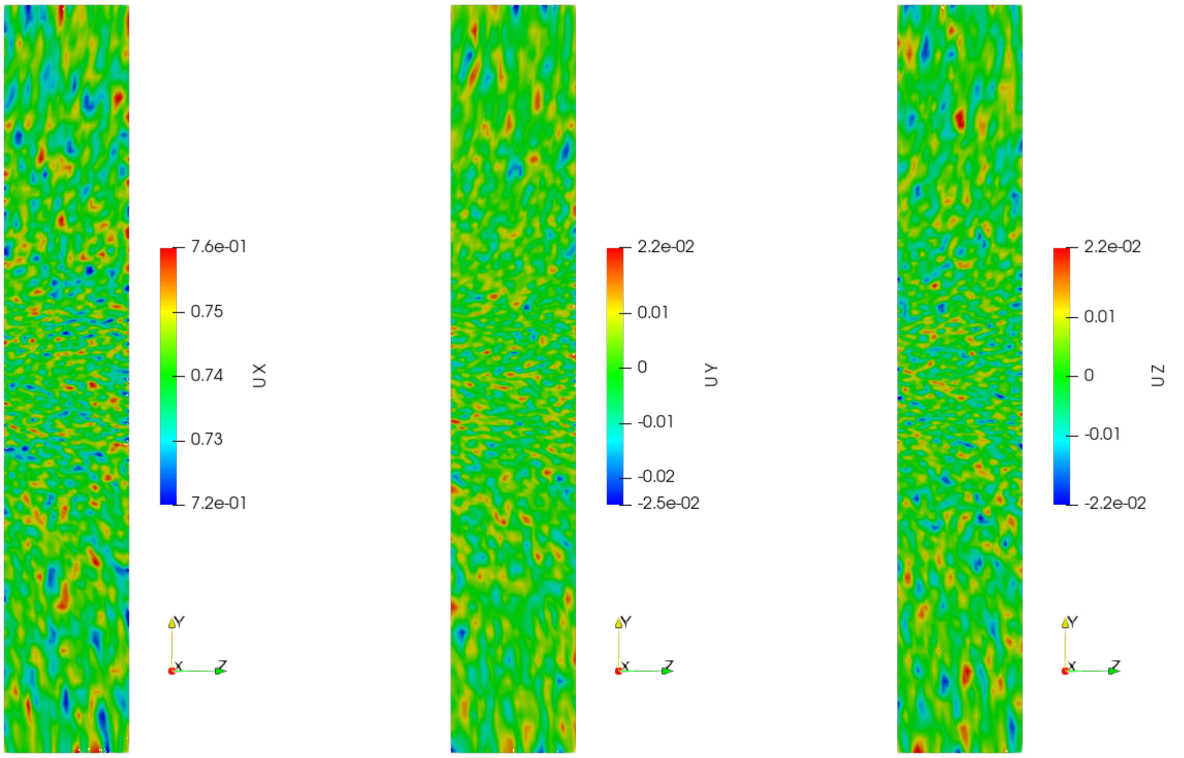
Boundary	u	p	k	ν_t
Inlet flow	0.48m/s	zeroGradient	2e-05	zeroGradient
Cylinder	noSlip	zeroGradient	0	zeroGradient
Outlet	zeroGradient	fixedValue 0	inletOutlet 0	zeroGradient
Front and Back	cyclic	cyclic	cyclic	cyclic
Top and Bottom	symmetryPlane	symmetryPlane	symmetryPlane	symmetryPlane

The domain considered in our simulations was represented in Figure 5.1. The left boundary, where the U_∞ is prescribed is called the *Inlet*, and the opposite is the *Outlet*. The lateral sides were then defined as *Front* and *Back*, while the upper and lower walls as *Top* and *Bottom*.

Large Eddy Simulations are unsteady simulations so it is necessary to introduce turbulent velocity time histories at the inlet boundary conditions. The most accurate way of producing inlet conditions that satisfy the Navier-Stokes equations is the use of precursor simulations. Here, the turbulent upstream flow is simulated, and the time-varying data produced are subsequently introduced to the simulation's inlet. However, this approach is computationally very expensive and requires the storage of large data amounts.

To reduce the cost of the simulation of spatially evolving flows, the inflow boundary is placed close upstream of the flow position of interest and some approximated turbulent velocity inflow boundary conditions are assigned to the inlet. The inflow data are collected using techniques that create artificial turbulence fields from specified characteristics of the turbulent approaching flow, such as for example the magnitude of the applied fluctuations. In fact, the simplest method is to superpose random fluctuations (or noise) on the mean velocity profile. The only information needed to implement this method is the mean velocity and the amount of turbulence at the entrance. This though has a big downside, that the fluctuations tend to die out quickly downstream of the inlet. However, in our case, it was investigated that turbulence was still present at the location where the flow was hitting the cylinder. Hence, it was decided that for our application this boundary condition was sufficiently satisfying for obtaining a turbulent velocity field.

In *OpenFOAM* this is implemented through a patch inlet condition known as *turbulentInlet*. Very simply, what has to be specified is the *referenceField* which is basically the mean velocity profile (u, v, w) and the *fluctuationScale* which is the magnitudes of the superposed velocity fluctuations (p, q, r). The output field will then be the sum of the velocity field plus the random fluctuations: ($u + p$, $v + q$, $w + r$). In our case, it was decided to add $p = q = r = 0.02$. The resulting velocity profiles at the inlet are shown in figure 5.4. In Table 5.1, the boundary conditions of the other quantities are reported.



(a) U_x **(b)** U_y **(c)** U_z

Figure 5.4: Velocity profiles at the inlet.

5.2.4. Quality of LES simulations

When running an LES simulation, there are different criteria that can be evaluated in order to assess the quality of the solution. In Section 5.1, I already discussed the importance of having a y^+ value smaller than 1. However, there are other parameters that can be computed which give an idea about the goodness of the simulation. As a benchmark, I will then investigate the following values:

- Courant–Friedrichs–Lewy (CFL) Condition
- Q-Criterion
- Fluctuating pressure
- Pope Criterion
- Strohual Number

CFL number

The CFL condition is a convergence criterion which is necessary for convergence when solving certain partial differential equations numerically. It is defined as follows

$$C = \frac{u\Delta t}{\Delta x} \tag{5.10}$$

where

- u is the magnitude of the velocity
- Δt is the timestep
- Δx is the length interval

The condition is then that the Courant number must be smaller than a maximum value, which we refer as C_{max}. Depending on the method used to solve the discretised equation, the value of C_{max} changes. If an explicit (time-marching) solver is used then typically C_{max} = 1. In my case then, the mesh size and the timestep were chosen to aim at obtaining a CFL number smaller than 1. It was obtained a maximum value of 0.7, hence fully satisfying the condition.

Q-Criterion

Another quantity that can be studied for evaluating our simulations is the *Q-criterion*. In particular, it is used for visualizing the vortical structures. It is defined as

$$Q = \frac{1}{2}\left[(\text{tr}(\nabla\mathbf{u}))^2 - \text{tr}(\nabla\mathbf{u}\cdot\nabla\mathbf{u})\right] \tag{5.11}$$

and by plotting the iso-surfaces of such quantity, it is possible to observe the vortices shedding in the wake of the cylinder.

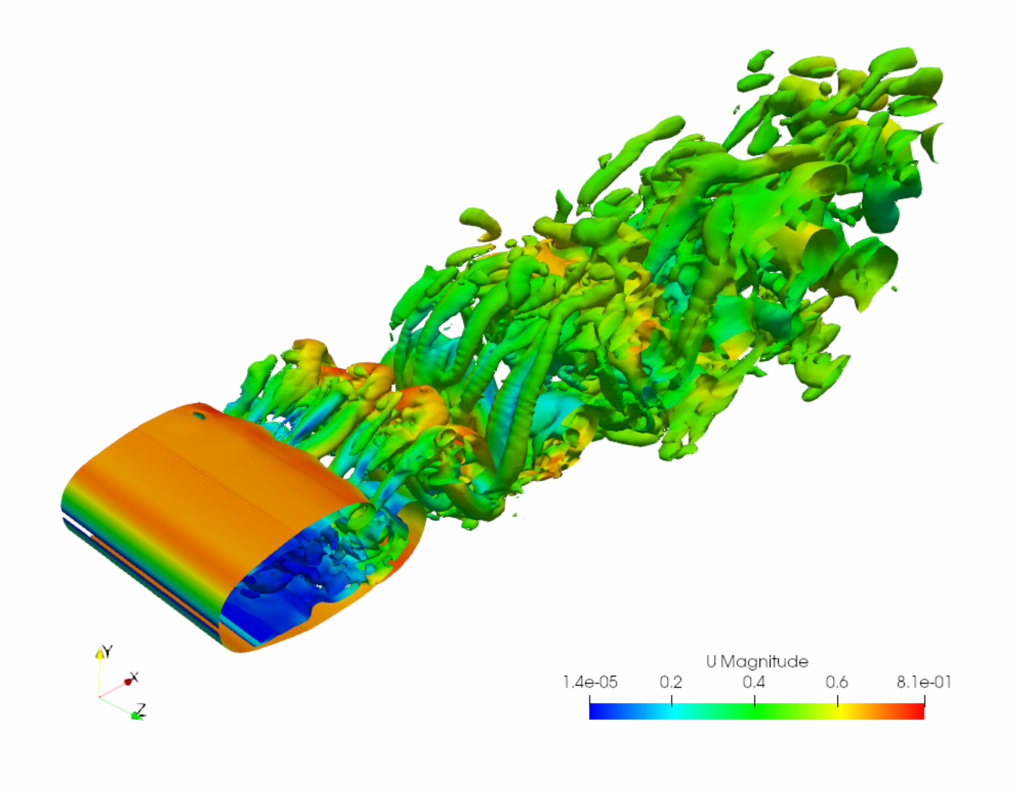


Figure 5.5: Q-criterion.

From the representations in Figure 5.5 and 5.6 it can be visualized how the area around the cylinder is highly laminar and quite extended. This was also observed in [27] where 2 different subgrid-scale LES models were compared: the conventional Smagorinsky (SMAG) and the dynamic k-equation (TKE) model. The simulations showed that the TKE solution was characterized by long, mainly two-dimensional, free-shear layers with a stream-wise length of approximately $2D$. The 2-D to 3-D transition happens after this recirculation region, in the wake area. The same phenomenon is here documented.

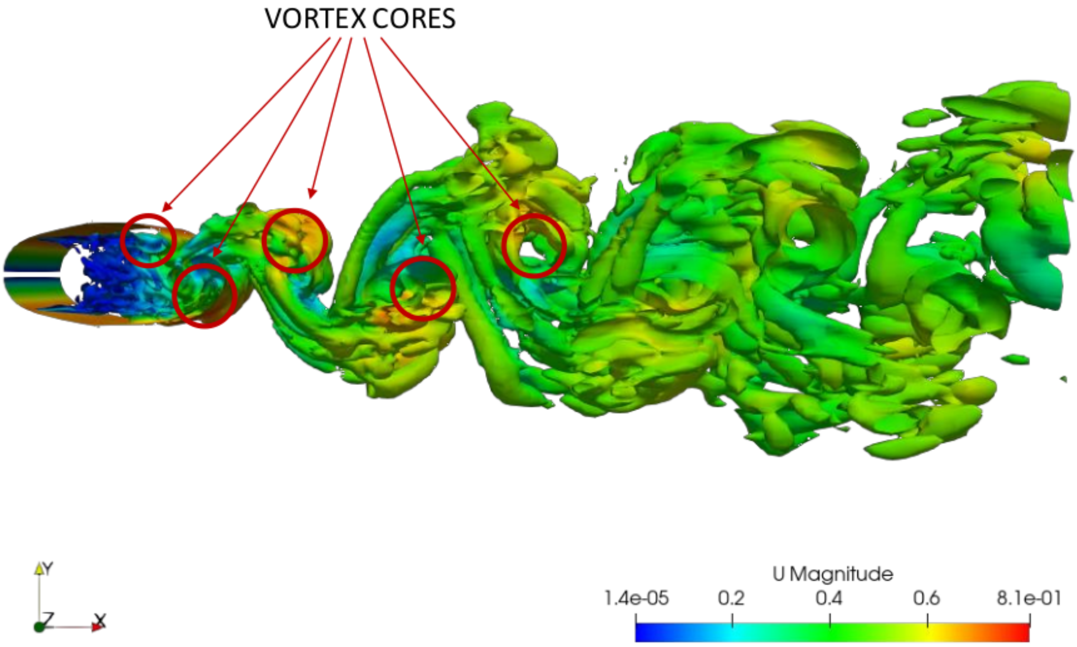


Figure 5.6: Q-criterion in $x - y$ plane.

Fluctuating pressure

One of the assumptions that were initially made about the flow field studied, is that the flow is statistically 2-dimensional. In fact, it was introduced in Section 3.1 that the set of tensors and invariants could be limited to three base tensors and two invariants.

One way to determine whether this assumption is here valid, is by plotting the iso-surface of the fluctuating pressure [27], defined as:

$$p' = 2\frac{(p - \overline{p})}{\rho_\infty U_\infty^2} \tag{5.12}$$

The iso-surface establishes the vortex cores that originate from the roll-up of the shear layers from the upper and lower parts of the cylinder surfaces. As can be observed in Figure 5.7 these vortices are parallel to the cylinder axis and they are convected further downstream by the mean flow. The fact that they keep their cylindrical shape without showing excessive vortex stretching, leads to the conclusion that the initial assumption of two-dimensional flow it can be confirmed.

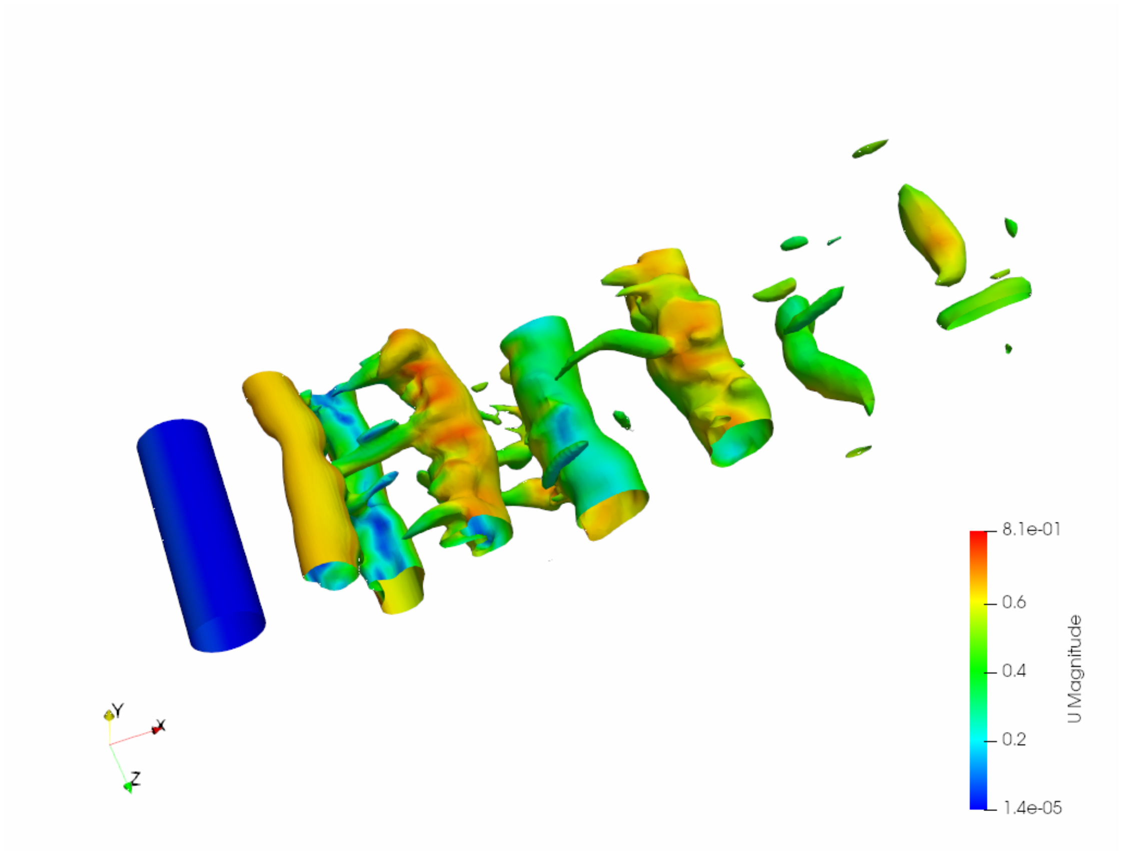


Figure 5.7: Iso-surface of the fluctuating pressure at $p' = -0.1$

Pope Criterion

Another way for measuring the quality of an LES simulation is by using the Pope criterion [31]. Simply, it is a measure of the fraction of turbulent kinetic energy in the resolved motions. It is expressed as

$$M(\mathbf{x}, t) = \frac{k_{SGS}(\mathbf{x}, t)}{k_{SGS}(\mathbf{x}, t) + k_{RES}(\mathbf{x}, t)} \tag{5.13}$$

where k_{SGS} is the turbulent kinetic energy of the SGS eddies and k_{RES} is the kinetic energy of the resolved eddies and defined as

$$k_{RES} = 0.5\left(u'^2 + v'^2 + w'^2\right) \tag{5.14}$$

with the $'$ values are the fluctuating components of the resolved velocity field.

In general, a good LES simulation aims at resolving 80% of the turbulent energy spectrum, which then corresponds to a maximum M value of around 0.2. In fact, a value of M = 0 corresponds to a DNS simulation, and a value of M = 1 corresponds to a RANS simulation. In Figures 5.8 and 5.9 the values of Pope criterion are represented in the $x - z$ and on the $x - y$ plane. In the red region, there is no turbulence, hence the LES resolved field coincides with the RANS velocity field since no turbulence has to be modelled. Focusing on the blue area, where M is smaller than 1, it can be stated that M is in general smaller than 0.2. There are some red spots, corresponding to the smallest eddies, but they are very restricted zones compared to the overall well-resolved velocity field, close to DNS solution. If M values close to 1 want to be avoided, then it would be needed to further refine the mesh. However, for the purposes of this work, these levels of resolution are very satisfactory.

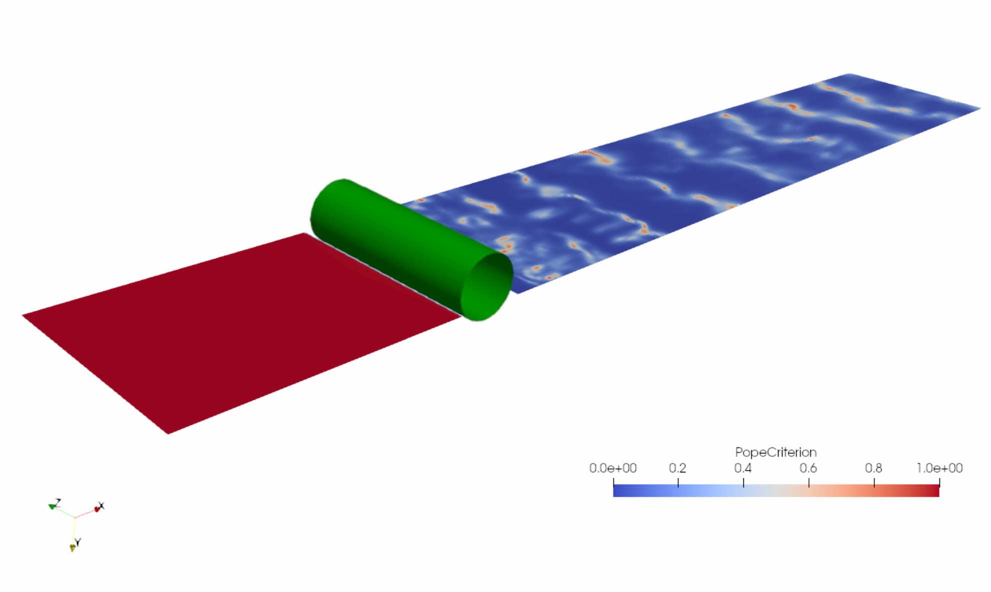


Figure 5.8: Pope Criterion at $y = 0$ in $x - z$ plane.

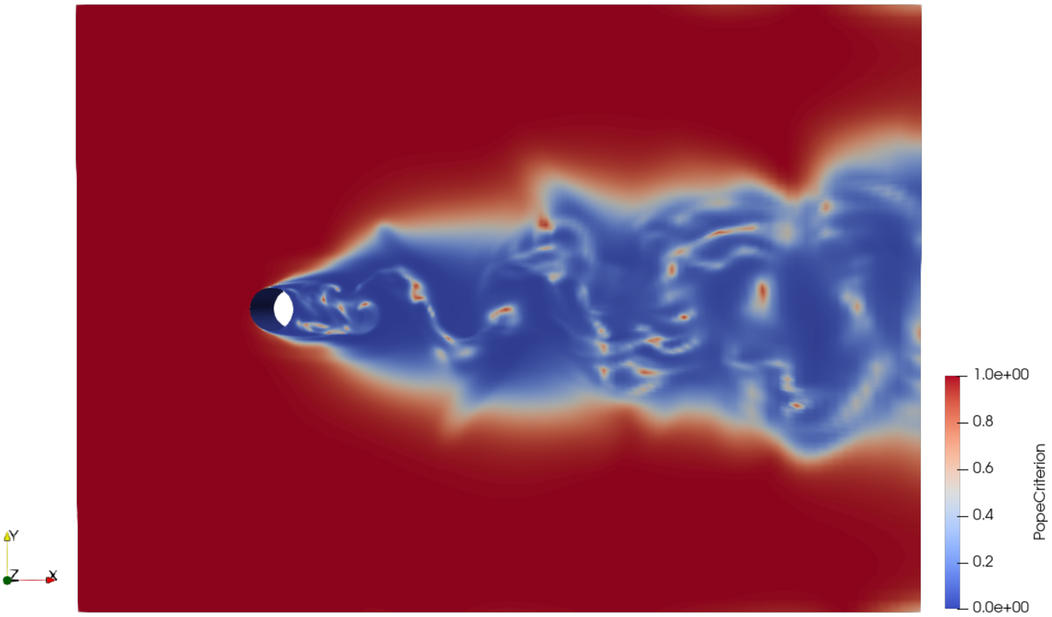


Figure 5.9: Pope Criterion in $x - y$ plane.

Strouhal Number

It was explained that the main feature of the investigated flow is the characteristic of vortex shedding. This happens in a periodic way, and the frequency of this event is almost constant at a given Reynolds number. This is described by the so-called Strouhal number, which is defined as

$$St = \frac{f_{vs}\, D}{U} \tag{5.15}$$

where f_{vs} is the vortex shedding frequency of a body at rest, D is the diameter of the cylinder and U is the free-stream velocity [19]. The wake dynamics is then characterized by two dimensionless parameters: the Reynolds number (equation (2.4)) and the here presented Strouhal number. The dependence of St on Re is shown in the graph in Figure 5.10.

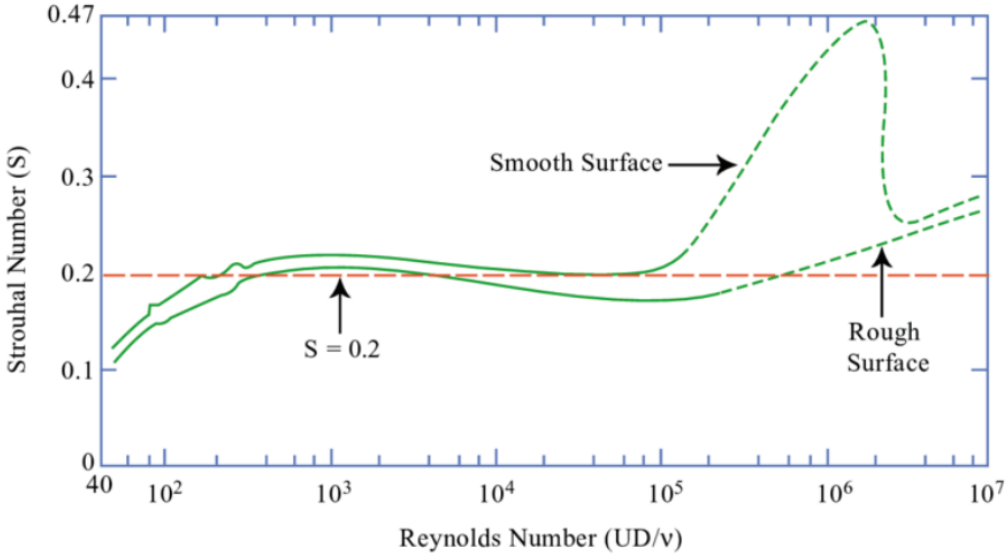


Figure 5.10: Relation between St (S) and Re (taken from [7]).

Performing the inverse Fourier transform of the velocity signal associated to the vortex shedding (as explained later in Section 6.2), the Strouhal frequency was found to be equal to $\sim$ 1.1, displayed in Figure 6.3. Substituting this value in equation (5.15) leads to $St \sim 0.22$. Reminding that the chosen Reynolds is 3900, the Strouhal value obtained is in line with the findings of Figure 5.10. It is then an additional assessment of the quality of the High-Fidelity LES data that will be used for the training of the models.

6 Data-driven Unsteady Methodology

6.1. Introduction to the Framework

In this chapter, all the steps necessary for the analysis conducted in this thesis, are described and explained. In Figure 6.1, the whole framework is represented. As this work relies on data-driven techniques, the first thing needed is data (box 1 in Figure 6.1). In this case, the required data is the velocity field obtained from an LES or DNS simulation. In Chapter 5, it was shown how the simulations were implemented in terms of domain size and boundary conditions. Some quantities were post-processed for assessing the quality of the solution (Subsection 5.2.4).

After having the full dataset of velocities in the whole domain, only the values from a limited region of the flowfield are extracted. This helps in terms of time and memory requirements. Such procedure is explained visually in Figure 6.2. This picture shows the velocity field at an instant in time t, while the smaller figures display the mean stochastic turbulent kinetic energy for different threshold values in the region bounded by the rectangle. We only consider this subregion for our data-driven modelling research.

This approach is justified as using the data from the full domain would not provide any additional information. In fact, the representations of TKE show that turbulence, hence the modelling objective, is only present in the coloured region in the wake of the cylinder.

After selecting this region, we will use POD (box 2 from Figure 6.1) to get the coherent velocity field. POD modes will later be explained in Section 6.2. This will allow us to perform the triple decomposition of the velocity, as introduced in Section 2.2.

When having the different components, they will be either phase or ensemble-averaged (box 3), and also they will be averaged over the spanwise (z) direction. In fact, the LES simulations were run in a 3D domain, but this modelling is aimed at being implemented in a 2D URANS simulation.

Next, the input and target quantities for our data-driven modelling are computed (box 4). The definitions of τ''_{ij} and k'' are provided in Section 2.1, while tensors $T_{ij}^{(n)}$ and invariants I_m will be defined in Section 6.4.

At this point, the unsteady SpaRTA methodology (box 5 in Figure 6.1) can be applied through the steps explained in Section 6.4 (boxes 5a, 5b and 5c). This will provide us with the correction model M, as a substitute for the Boussinesq assumption.

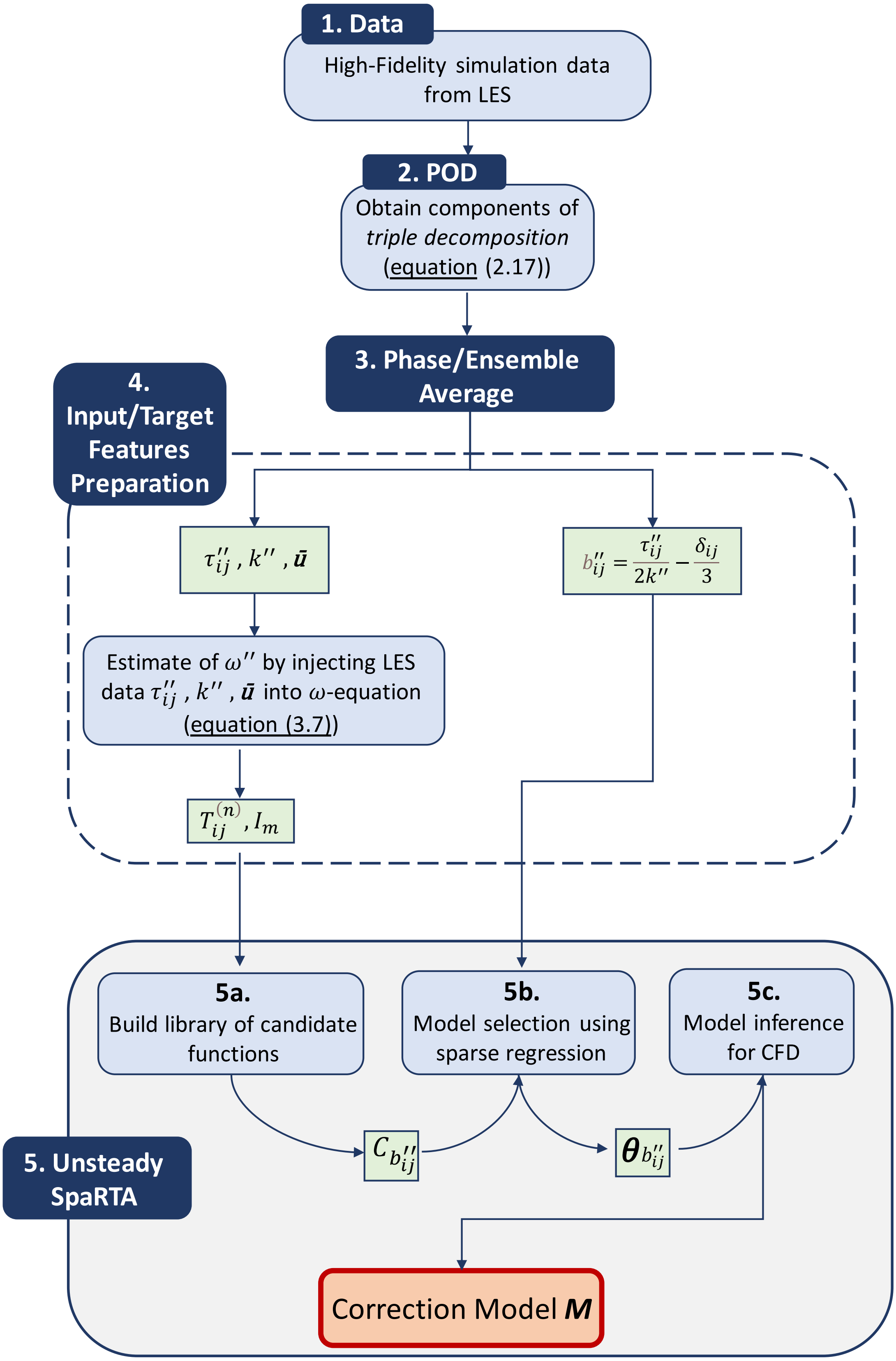
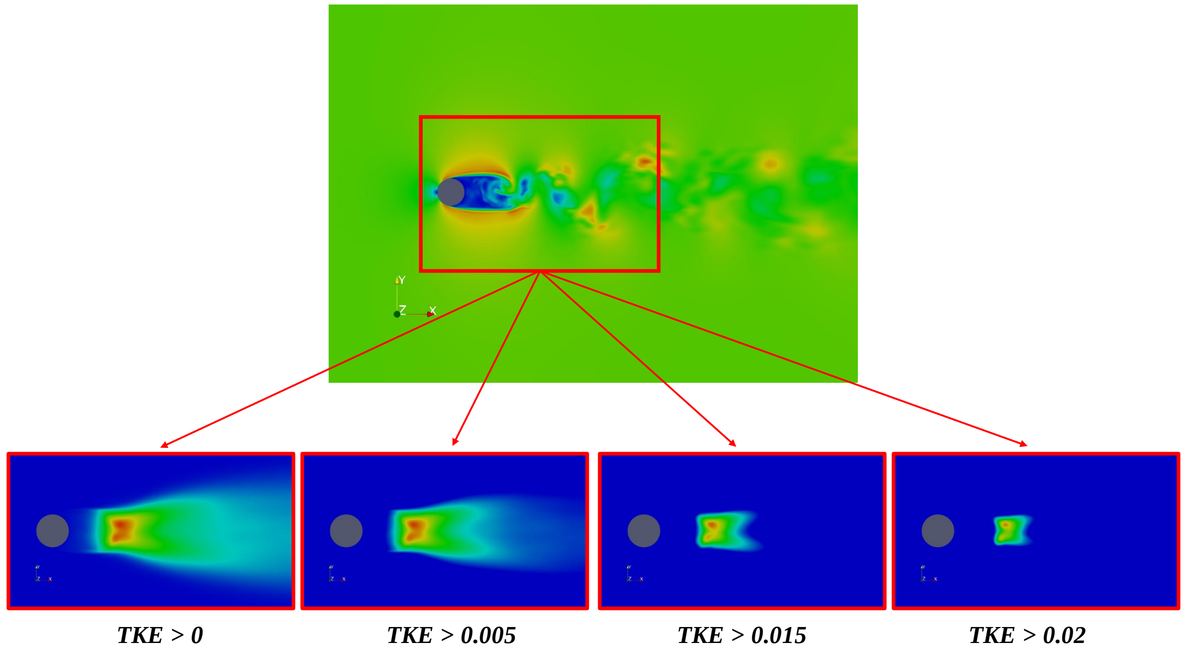

1. Data
High-Fidelity simulation data from LES
2. POD
Obtain components of *triple decomposition* (equation (2.17))
3. Phase/Ensemble Average
4. Input/Target Features Preparation
τ''_{ij} , k'' , $\bar{\boldsymbol{u}}$
$b''_{ij} = \frac{\tau''_{ij}}{2k''} - \frac{\delta_{ij}}{3}$
Estimate of ω'' by injecting LES data τ''_{ij} , k'' , $\bar{\boldsymbol{u}}$ into ω-equation (equation (3.7))
$T^{(n)}_{ij}$, I_m
5a. Build library of candidate functions
5b. Model selection using sparse regression
5c. Model inference for CFD
$C_{b''_{ij}}$
$\boldsymbol{\theta}_{b''_{ij}}$
5. Unsteady SpaRTA
Correction Model $\boldsymbol{M}$

Figure 6.1: Unsteady SpaRTA framework.



TKE > *0* *TKE* > *0.005* *TKE* > *0.015* *TKE* > *0.02*

Figure 6.2: Representation of the extracted region. In the big picture, the velocity field at one phase step is shown. From this flowfield, only a portion of the whole domain is extracted, and the mean stochastic turbulent kinetic energy k'' is represented in the small pictures (at different TKE thresholds).

6.2. Proper Orthogonal Decomposition

As was explained, turbulent flows are of very complex nature, made of a combination of different time and length scales. In order to lower the complexity, dimensionality reduction techniques are used, which lead to a hierarchical decomposition of flow data.

The technique that I will be using is POD, which stands for *Proper Orthogonal Decomposition*. This method provides a rational way for the extraction of coherent structures in turbulent flows. Such structures are organized spatial features which repeatedly occur and undergo a characteristic temporal life cycle. POD has three main advantages [6]:

1. it is statistically based;
2. its analytical foundations supply a clear understanding of its capabilities and limitations;
3. it permits the extraction, from a turbulent field, of spatial and temporal structures and it offers a precise mathematical framework for their description.

It is then a tool not only for the analysis and synthesis of data from experiments or simulations but also for the construction of low-dimensional dynamical models for the interaction of these fundamental structures. More in general, it is a procedure for deriving a basis for a modal decomposition from an ensemble of signals. In the context of turbulence, it was introduced by Lumley. However, in other disciplines, the same methodology is known with different names, for example, Karhunen-Loève decomposition or principal component analysis.

Let us now focus on the application that will be done in the thesis here. POD will be used for extracting the three different components of velocity, the ones split from the triple decomposition (equation (2.17)), so the average, deterministic and stochastic components. When analyzing a time series of data, containing for example velocity fields at a grid of spatial locations, as is the case here, it is beneficial to use snapshot-based methods.

For example, if we have a two-dimensional vector field at a time t_k, it is flattened into a single tall column vector:

$$\begin{aligned}\boldsymbol{x}\left(\boldsymbol{r}, t_k\right) &= \begin{bmatrix} x\left(r_{1,1}, t_k\right) & x\left(r_{1,2}, t_k\right) & \cdots & x\left(r_{1,p}, t_k\right) \\ x\left(r_{2,1}, t_k\right) & x\left(r_{2,2}, t_k\right) & \cdots & x\left(r_{2,p}, t_k\right) \\ \vdots & \vdots & \ddots & \vdots \\ x\left(r_{q,1}, t_k\right) & x\left(r_{q,2}, t_k\right) & \cdots & x\left(r_{q,p}, t_k\right) \end{bmatrix} \\ \Rightarrow \quad \boldsymbol{x}_k &= \begin{bmatrix} x\left(r_{1,1}, t_k\right) \\ x\left(r_{1,2}, t_k\right) \\ \vdots \\ x\left(r_{2,1}, t_k\right) \\ \vdots \\ x\left(r_{q,p}, t_k\right) \end{bmatrix}\end{aligned} \tag{6.1}$$

Here, x represents a flow variable, for example the velocity in our case, $\boldsymbol{r}$ denotes the spatial coordinate, while the index k stands for the kth time step. The vector $\boldsymbol{x}_k$ is called a *snapshot* of data. All these snapshots together can be collected into a single data matrix $\boldsymbol{X}$:

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_m \\ | & | & & | \end{bmatrix} \tag{6.2}$$

In this application, $\boldsymbol{X}$ will be a tall matrix, since the number of snapshots m is smaller than the state dimension n. Now that the matrix is filled with the state variables at each snapshot in time, it is possible to apply POD for extracting its different components.

POD is often formulated by taking the SVD (Singular Value Decomposition) of the above data matrix $\boldsymbol{X}$, so that it is possible to rewrite it as:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* \tag{6.3}$$

We see that the matrix was factorized into 3 different matrices $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, $\boldsymbol{V*}$.

- $\boldsymbol{U}$ contains the spatial correlation of the data
- $\boldsymbol{\Sigma}$ is a diagonal matrix whose entries are the singular values of $\boldsymbol{X}$
- $\boldsymbol{V}$ contains all the temporal information

The matrix $\boldsymbol{U}$ contains then the POD modes, while the matrix $\boldsymbol{\Sigma}$ contains the singular values, ordered by their magnitude. The POD modes are also in order of contribution to the total amount of energy captured by each of them. This means that often it is enough to use the first few modes to describe 90% or more of the actual energy.

In our specific case, the matrix $\boldsymbol{X}$ will then contain the velocity vector column-wise at each snapshot, :

$$\boldsymbol{X} = \begin{bmatrix} u_{1,1} & u_{1,2} & & u_{1,m} \\ u_{2,1} & u_{2,2} & \ldots & u_{2,m} \\ | & | & & | \\ v_{1,1} & v_{1,2} & \ldots & v_{1,m} \\ | & | & & | \\ w_{n,1} & w_{n,2} & & w_{n,m} \end{bmatrix} \tag{6.4}$$

where $u_{i,j}$ is the $x-$component of the velocity at the i^{th} cell at timestep j.

Once we obtain the 3 matrices, we can then reconstruct the values corresponding to the POD mode in which we are interested. If we then compute the frequency of the signal of the modes we can assign certain modes to the related quantities. For example, we have that the first mode has frequency zero, the second and third modes have the frequency corresponding to the vortex shedding, while the other

modes can all be grouped together to form the stochastic part of the velocity $\boldsymbol{u}_{ij}^{''}$.
Using the Python indexing notation we can write the reconstruction of the stochastic velocity as:

$$\boldsymbol{u}^{''} = \boldsymbol{U}[:, 2:] * \boldsymbol{\Sigma}[2:, 2:] * \boldsymbol{V}^{*}[2:, :] \tag{6.5}$$

which leads to a matrix of dimension $n \times m$ where n is the number of cells and m is the number of snapshots.

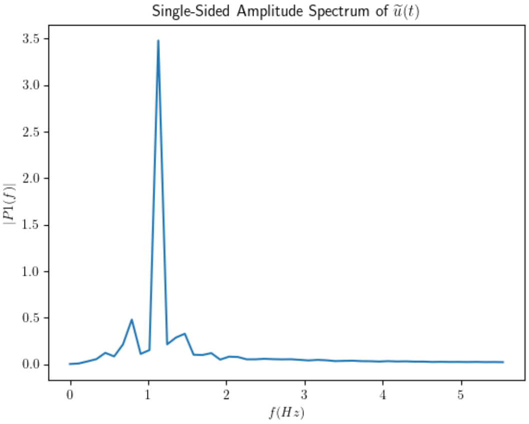


Figure 6.3: Main frequency of the shedding component.

After having obtained the three velocity components, $\overline{U}$, $\tilde{u}$ and u'', through the use of inverse Fourier transform (*ifft*) applied to the time evolution of each, it was possible to obtain the corresponding frequencies of the three different signals. For the mean contribution, the frequency is equal to 0, so it is here not reported.

In figure 6.3 the result of applying *ifft* to the shedding component is shown. One main frequency was found, which corresponds to the frequency of vortex shedding. In particular, it is equal to 1.1, which is close to values found in literature [12][27].
The same procedure was implemented and applied to the stochastic velocity signal and the corresponding main frequencies are reported in figure 6.4. In this case, more dominant frequencies are visible, confirming the chaotic nature of this contribution.

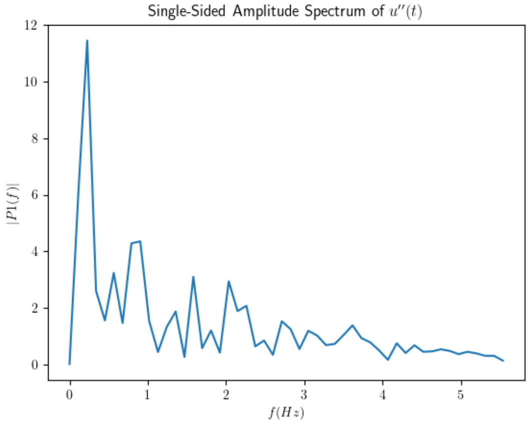
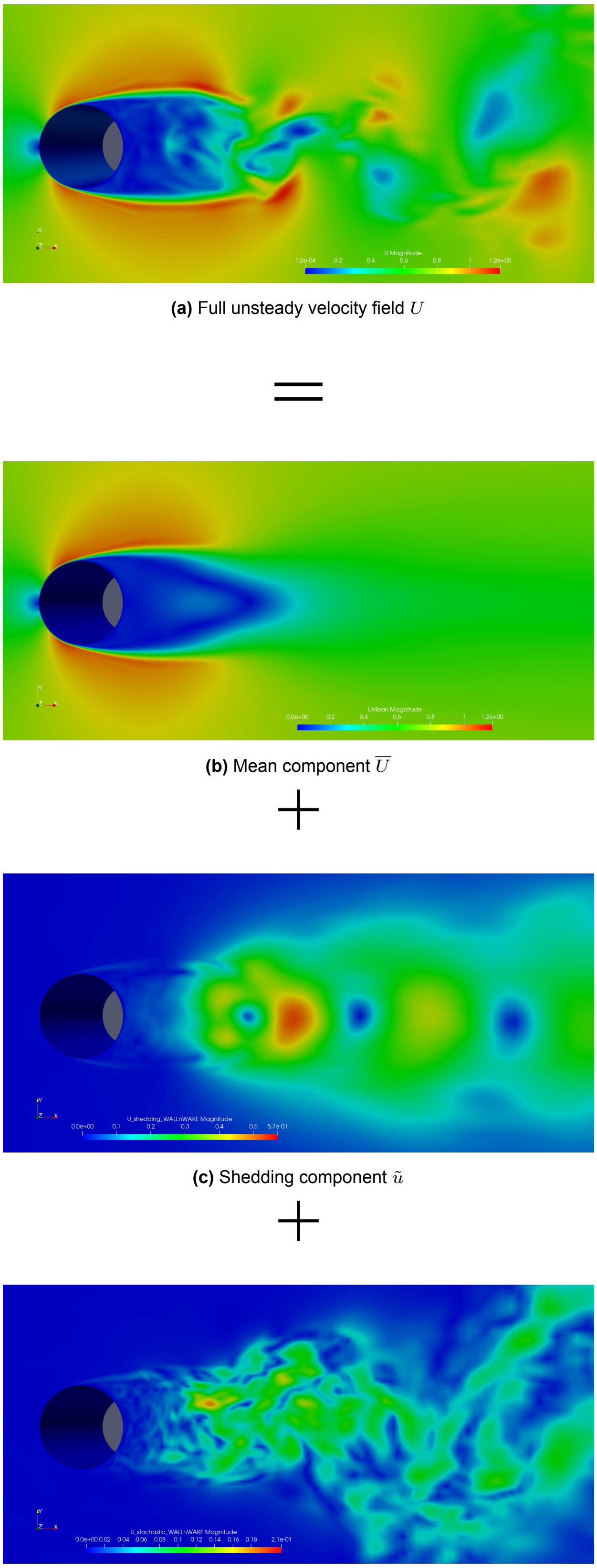


Figure 6.4: Main frequencies of stochastic component.

In figure 6.5 all the components at a time instant are displayed. The full velocity field is decomposed as in equation (2.17).



(a) Full unsteady velocity field U

(b) Mean component $\overline{U}$

(c) Shedding component $\tilde{u}$

(d) Stochastic component $u^{''}$

Figure 6.5: Velocity components after Proper Orthogonal Decomposition.

6.3. Phase and Ensemble Averaging

The next procedure that takes place, after the decomposition of the velocity, is the averaging of the quantities involved. By definition, the Reynolds stresses are an average of the velocity fluctuations. They can be averaged either over a period, and in this case it will be referred to as *phase-averaged* Reynolds stresses, or they can be treated as the result of an *ensemble* average over all the vortex shedding periods considered.

As a first attempt, we are aiming to answer how the resolved coherent velocity field can improve the anisotropy tensor prediction (Question Q1.1). The objective is to obtain a model where the inputs are functions of the shedding velocity $\widetilde{U}$, and the target is the stochastic anisotropy b''_{ij}. Since $\widetilde{U}$ varies as a periodic function, the fluctuating stochastic quantity b''_{ij} has to be averaged over the same period. The anisotropy is constructed from equation (3.16), which is dependent on the considered Reynolds stresses τ''_{ij}.

When answering Question Q1.1, the following *phase-averaged* quantities will be used:

$$\overline{\widetilde{U}}(\boldsymbol{x},\ \theta) = \frac{1}{p}\sum_{j=1}^{p} \widetilde{U}(\boldsymbol{x},\ \theta + j * N_\theta) \tag{6.6}$$

$$\overline{\tau''_{ij}}(\boldsymbol{x},\ \theta) = \frac{1}{p}\sum_{j=1}^{p} \tau''_{ij}(\boldsymbol{x},\ \theta + j * N_\theta) \tag{6.7}$$

where p is the number of periods considered, and N_θ is the number of time steps corresponding to one period. The quantities above are then defined per each phase step over a single (averaged) period.

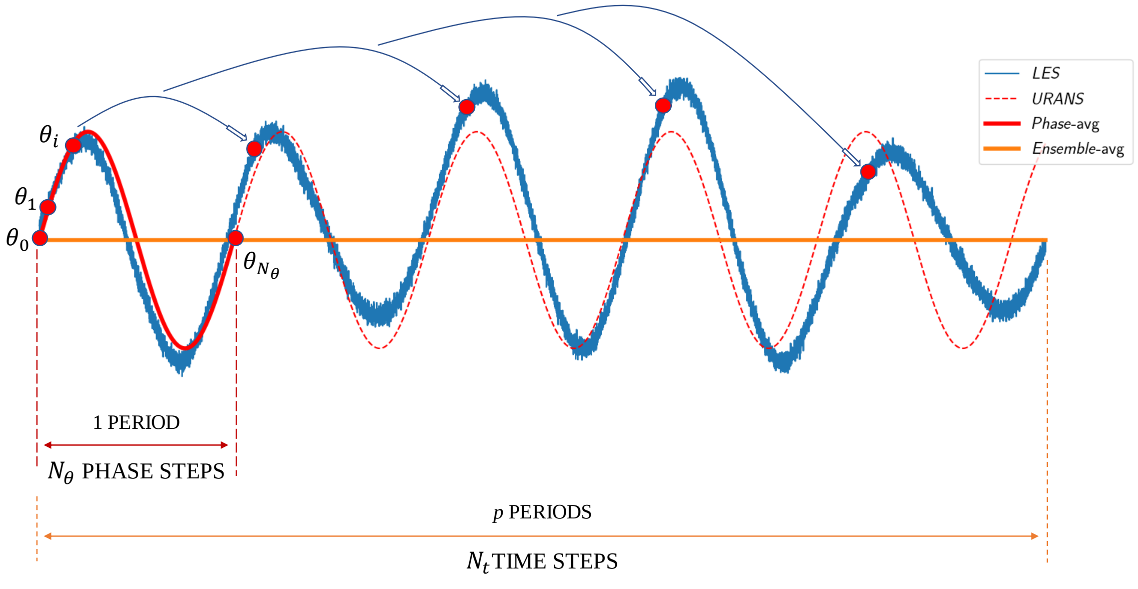


Figure 6.6: Time evolution of LES (blue line) and URANS (dashed red line) quantities. The orange line is the ensemble-averaged quantity, while in red (continuous line) is the phase-averaged one.

A simplified procedure is then applied when addressing Question Q1.2, so how SpaRTA can be used for modelling the relationship between the mean flow quantities and the anisotropic Reynolds stress tensor b''_{ij}. The intention is to find a model where the inputs are functions of the mean velocity $\overline{U}$, and the target is the stochastic anisotropy b''_{ij}, again constructed from τ''_{ij}. As anticipated, the quantities are averaged over all the time steps from the High-Fidelity data, and they are then referred to as *ensemble* averaged. It follows this formalization:

$$\overline{U}(\boldsymbol{x}) = \frac{1}{N_t} \sum_{j=1}^{N_t} \widetilde{U}(\boldsymbol{x},\ j) \tag{6.8}$$

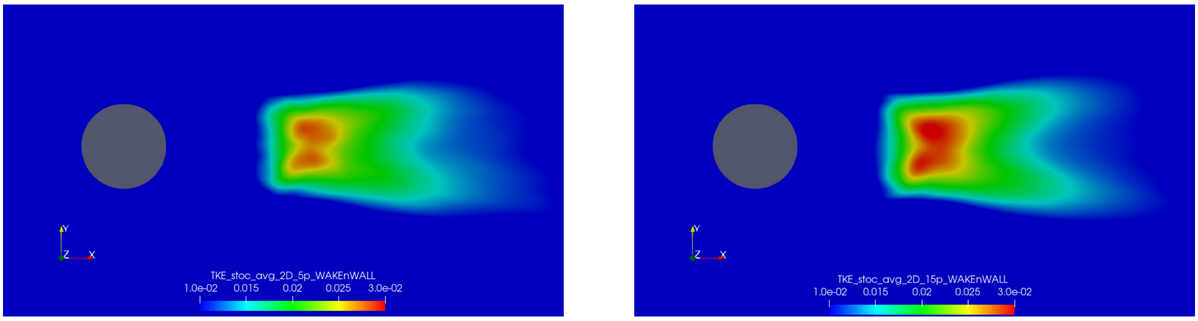
$$\overline{\tau''_{ij}}(\boldsymbol{x}) = \frac{1}{N_t} \sum_{j=1}^{N_t} \tau''_{ij}(\boldsymbol{x},\ j) \tag{6.9}$$

where N_t is the total number of time steps. The main difference in the notations is that, as can be observed, the ensemble-averaged quantities are not dependent on the phase θ but they are only a function of the location x. The representation of these two different averaging procedures is shown in Figure 6.6.

6.3.1. Influence of number of vortex shedding periods

To obtain the phase/ensemble-averaged quantities, we first have to assess how many periods are necessary to obtain converged statistics.

In Figure 6.6, the averaging procedure is presented over $p = 5$ periods. However, it is observed that 5 periods are insufficient. Indeed, the profiles of turbulent kinetic energy in Figure 6.7a appear to be less symmetric and more ”noisy” than the profile in Figure 6.7b. The latter is the result of data collected and averaged over 15 periods. After also comparing quantities averaged over 10 and 20 periods, we decided to use data averaged over 15 periods for our modelling. It was evaluated that 15 periods were a good compromise between accuracy and memory requirements.



(a) Turbulent kinetic energy averaged over 5 periods

(b) Turbulent kinetic energy averaged over 15 periods

Figure 6.7: Comparison of turbulent kinetic energy profiles when averaging over 5 versus 15 periods.

6.4. Unsteady SpaRTA

After averaging the velocity and the Reynolds stresses, as explained in Section 6.3, we can now produce the input and target quantities of our modelling. In this Section, we will not differentiate between *phase* and *ensemble* average, as the following procedure is the same for both approaches.

The **target** is the stochastic anisotropy, which is directly computed from equation (3.1), by substituting either the phase or ensemble-averaged Reynolds stresses and turbulent kinetic energy.

Concerning the **inputs**, for constructing the modelling features and tensors, a High-Fidelity value for the *specific dissipation rate* ω is needed. By injecting the LES values for τ''_{ij}, k'' and $\overline{U}$ in equation (3.7), we obtain an equivalent RANS-consistent estimate of ω. In fact, even if this study is just a *a priori* assessment, it is aimed at generating models which could, later on, be implemented in a $k-\omega$ URANS solver. For this reason, we want ω to be computed the same way as it would be calculated in the solver.

6.4.1. Non-linear eddy-viscosity model for b''_{ij}

In order to discover a model for the stochastic part of the anisotropy, a modelling assumption is needed. The symbolic regression targets to find specific expressions as a substitute for ordinary turbulence closure models. In [30], Pope proposes a nonlinear generalisation of the linear eddy viscosity concept. The main assumption made by Pope is that the anisotropy of the Reynolds stress b''_{ij} depends both on the strain rate tensor $\boldsymbol{S_{ij}} = \tau\frac{1}{2}(\partial_j U_i + \partial_i U_j)$ and on the rotation rate tensor $\boldsymbol{\Omega_{ij}} = \tau\frac{1}{2}(\partial_j U_i - \partial_i U_j)$ where τ is the timescale equal to $1/\omega$, with ω being the *specific turbulence dissipation rate*. We can then write the general form of the stochastic anisotropy as

$$b''_{ij}(S_{ij}, \Omega_{ij}) = \sum_{n=1}^{N} T_{ij}^{(n)} \alpha_n(I_1, ..., I_5) \tag{6.10}$$

with ten nonlinear base tensors $T_{ij}^{(n)}$ and five corresponding invariants I_m. This is necessary when the flow is 3-dimensional, and the definitions of such functions are reported in Appendix B. However, as anticipated in section 3.1, when the flow case is statistically 2-dimensional, this set of tensors and invariants can be simplified and reduced to the following:

$$\begin{aligned} T_{ij}^1 &= S_{ij} & I_1 &= S_{mn}S_{nm} \\ T_{ij}^2 &= S_{ik}\Omega_{kj} - \Omega_{ik}S_{kj} & I_2 &= \Omega_{mn}\Omega_{nm} \\ T_{ij}^3 &= S_{ik}S_{kj} - \frac{1}{3}\delta_{ij}S_{mn}S_{nm} & & \end{aligned} \tag{6.11}$$

6.4.2. Library of candidate functions

One of the first steps we must take for generating our model is the creation of a library of candidate functions, from which a model can be inferred by constructing a linear combination of such candidates. In order to obtain our library we start from a set of base tensors and invariants. As we already introduced before, we base our modelling on the nonlinear eddy viscosity concept and we aim at finding models for α_n in equation (6.10). As primitive input features we will take the invariants I_1 and I_2, as it was done in previous studies [34][36][21]. In addition, other features are used in our library, as proposed in [40]. The scalar quantities selected (noted $q_\bullet$), their description and formulation are reported in Table 6.1. To further extend the discovery possibilities of our regression approach, these quantities were subject to functional transformations. Specifically, their *abs*, *exp* and *tanh* were computed and also used as inputs. Last but not least, the exponential functions of these quantities are also included, such that if x is the considered feature, also $\sqrt{x}$, x^2, $\frac{1}{x}$, $\frac{1}{x^2}$ and $\frac{1}{\sqrt{x}}$ will be part of the library.

After multiplying each feature with each other, and also introducing a constant function, the resulting vector $\boldsymbol{B}$ will look like this:

$$\boldsymbol{B} = \left[1,\ I_1,\ I_2,\ I_1^2,\ I_2^2,\ \frac{1}{I_1},\ \frac{1}{I_2}, ...,\ q_Q,\ q_Q I_1,\ q_Q I_2, ...,\ q_{TI} I_2^2,\ q_{TI}^2 I_2^2\right]^T \tag{6.12}$$

The full library for regressing the models for $b^{''}_{ij}$ is then obtained by multiplying each function in $\boldsymbol{B}$ by each base tensor $T^{(n)}_{ij}$, in our case the three first ones, which leads to the library of tensorial candidate functions:

$$\boldsymbol{C}_{b''_{ij}} = \left[T^{(1)}_{ij},\ T^{(2)}_{ij},\ T^{(3)}_{ij},\ I_1T^{(1)}_{ij},\ I_1T^{(2)}_{ij}...,\ q^2_{TI}I^2_2T^{(3)}_{ij}\right]^T \tag{6.13}$$

In the table below, the mentioned features are described:

Table 6.1: Additional features used for model discovery.

Feature	Description	Formula
q_Q	Q-Criterion	$\frac{1}{2}\left(\lVert\Omega\rVert^2 - \lVert S\rVert^2\right)$
q_T	Turbulent to mean strain time scale	$\lVert S\rVert\frac{k}{\epsilon}$
q_γ	Shear parameter	$\lVert\nabla\boldsymbol{u}\rVert\frac{k}{\epsilon}$
$q_{\tau kB}$	Total to normal Reynolds stresses Boussinesq	$\frac{\lVert\tau_B\rVert}{k}$
q_ν	Viscosity ratio	$\frac{\nu_T}{100\nu}$
q_{TI}	Turbulence intensity	$\frac{k}{\frac{1}{2}\lVert\boldsymbol{u}\rVert^2}$

6.4.3. Model selection and inference using sparsity-promoting regression

Now that both the input and target quantities are computed and available to us, we can build a system of equations to regress the target data b''_{ij}, through a coefficient vector Θ. Mathematically, it will look like:

$$\boldsymbol{b}''_{ij} = \boldsymbol{C}_{b''_{ij}}\Theta \tag{6.14}$$

which represents a large, overdetermined system of equations, meaning that there are more equations than unknowns. That is why it was chosen to apply a sparse regression technique in this context. In fact, the objective is to obtain a model that with just a few terms it can accurately describe our target data.

The library $\boldsymbol{C}_{b''_{ij}}$, introduced in equation (6.13), will be a matrix where on each column every candidate function is evaluated for all the input data at the different instants in time (in the case of phase-averaged approach) or simply at the averaged condition. Such matrix will then have $n \times m$ lines ($n =$ number of cells), each being a tensor, and as many columns as the number of candidate functions. It will look as follows:

$$\boldsymbol{C}_{b''_{ij}} = \begin{bmatrix} \boldsymbol{T}^{(1)}_1 & \boldsymbol{T}^{(2)}_1 & & (\boldsymbol{q}^2_{TI}\boldsymbol{I}^2_2)_1\boldsymbol{T}^{(3)}_1 \\ \boldsymbol{T}^{(1)}_2 & \boldsymbol{T}^{(2)}_2 & & (\boldsymbol{q}^2_{TI}\boldsymbol{I}^2_2)_2\boldsymbol{T}^{(3)}_2 \\ \boldsymbol{T}^{(1)}_3 & \boldsymbol{T}^{(2)}_3 & ... & (\boldsymbol{q}^2_{TI}\boldsymbol{I}^2_2)_3\boldsymbol{T}^{(3)}_3 \\ \vert & \vert & & \vert \\ \boldsymbol{T}^{(1)}_m & \boldsymbol{T}^{(2)}_m & & (\boldsymbol{q}^2_{TI}\boldsymbol{I}^2_2)_m\boldsymbol{T}^{(3)}_m \end{bmatrix} \tag{6.15}$$

The whole linear model in equation (6.14) for regressing the data will then be as below:

$$\begin{bmatrix} \boldsymbol{b}^{''}_1 \\ \boldsymbol{b}^{''}_2 \\ \boldsymbol{b}^{''}_3 \\ \vert \\ \boldsymbol{b}^{''}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}^{(1)}_1 & \boldsymbol{T}^{(2)}_1 & & (\boldsymbol{q}^2_{TI}\boldsymbol{I}^2_2)_1\boldsymbol{T}^{(3)}_1 \\ \boldsymbol{T}^{(1)}_2 & \boldsymbol{T}^{(2)}_2 & & (\boldsymbol{q}^2_{TI}\boldsymbol{I}^2_2)_2\boldsymbol{T}^{(3)}_2 \\ \boldsymbol{T}^{(1)}_3 & \boldsymbol{T}^{(2)}_3 & ... & (\boldsymbol{q}^2_{TI}\boldsymbol{I}^2_2)_3\boldsymbol{T}^{(3)}_3 \\ \vert & \vert & & \vert \\ \boldsymbol{T}^{(1)}_m & \boldsymbol{T}^{(2)}_m & & (\boldsymbol{q}^2_{TI}\boldsymbol{I}^2_2)_m\boldsymbol{T}^{(3)}_m \end{bmatrix} \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \\ \Theta_4 \\ \Theta_5 \\ \vert \\ \Theta_B \end{bmatrix} \tag{6.16}$$

For clarity, the vector $\boldsymbol{T}^{(1)}_1$ is the column vector containing the components of the tensor $T^{(1)}_{ij}{}_1$ at every cell, so it is size (n, 3, 3), with n the number of cells.

Depending on how many functions we want to employ for our discovery, the vector of coefficients, which has the size of the number of candidate functions (B in equation (6.16)), can be quite large. However, the system is still over-determined, since the number of equations is consistently larger than the number of unknowns, which is the number of coefficients.

If we were to solve this type of system by using ordinary least-squares regression we would obtain a dense coefficient vector $\boldsymbol{\Theta}$, leading to overly complex models, which would overfit the training dataset. In order to avoid this, we will then introduce sparsity-promoting regularisation of the underlying least-squares optimisation problem. The steps in finding the most suitable model are then the following:

1. *model selection*
2. *model inference*

For the step of **model selection**, the elastic net formulation [34] is used

$$\boldsymbol{\Theta} = \underset{\hat{\theta}}{\text{arg min}} \left\| \boldsymbol{C}_{b''_{ij}} \hat{\boldsymbol{\Theta}} - \boldsymbol{b''_{ij}} \right\|_2^2 + \lambda\rho \left\| \hat{\boldsymbol{\Theta}} \right\|_1 + 0.5\lambda(1-\rho) \left\| \hat{\boldsymbol{\Theta}} \right\|_2^2 \tag{6.17}$$

which combines l_1- and l_2-norm regularisation through the mixing parameter $\rho \in [0,1]$ and the regularisation weight λ, to promote the sparsity of $\boldsymbol{\Theta}$. l_1-norm regularisation is also known as *lasso* regression, while l_2- is referred as *ridge*. The elastic net method then finds an estimator in a two-stage procedure: first, for each fixed λ it finds the ridge regression coefficients and then does a lasso-type shrinkage, which tends to zero as many coefficients as possible.

The choice of the ρ and λ values is made using a grid search algorithm based on the training data and target values. It can be observed that when λ = 0, equation (6.17) is equivalent to an ordinary least square. Also, when ρ = 0, we are solving a ridge regression, while if ρ = 1, it coincides with lasso.

After the model selection, a set of different candidate models is obtained. At this point, the second step of model discovery takes place, which is **model inference**. Similarly to the model selection, a regression is implemented, which is though only driven by ridge regularisation. By doing this, the coefficients are further shrunk to smaller values. In fact, it was found in [42][43] that the greater the coefficients, the more difficult it is to obtain a converged solution when using a model in a CFD solver. The inference step is then implemented with the goal of overcoming such limitation.

6.5. *A priori* study

At this point, different candidate models are available, each composed of a different number of functions and leading to different accuracy in targeting the anisotropy tensor b''_{ij}. For the evaluation of their performance, different parameters can be considered.

One way, for example, is through the Pearson coefficient, defined as:

$$\rho_{RS} = \frac{b_{ij}^{ref} b_{ij}^{mod}}{\sqrt{b_{mn}^{ref} b_{nm}^{ref} b_{pq}^{mod} b_{qp}^{mod}}}$$

which basically represents the normalised inner product of the modelled anisotropy with the reference anisotropy. A completely valid model would exhibit perfect alignment with a value of 1.0. However, this has some limitations, as a model can be aligned with the target, but have large differences in magnitude.

For this reason, when we will be selecting our ”best” model, two other values will also be considered:

- $\boldsymbol{MSE}$, which is the *mean squared error* and as the name suggests, it measures the average of the squares of the errors, hence the average squared difference between the estimated values and the actual value.
- $\boldsymbol{R^2}$ score, which is defined as $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$ where SS_{res} is the sum of squares of the residuals, and SS_{tot} is the total sum of squares (proportional to the variance of the data)

Realizability of Reynolds Stress Tensor

Another quantity that will be considered for assessing the performance of the obtained models, compared to the anisotropy computed from Boussinesq, is the realizability and componentiality of the turbulence states. The topic of representing the Reynolds stress tensor, and in particular the anisotropy tensor, has been faced by multiple research projects. Lumley [25] introduced the following three independent invariants of the anisotropy tensor

$$| = b_{ii}, \quad || = b_{ij}, \quad ||| = b_{ij}b_{in}b_{jn} \tag{6.18}$$

Based on the eigenvalues of the anisotropy tensor, Lumley also introduced a mapping to the second and third invariants. Thanks to this, it is possible to represent the realizable states in a ||-||| map [26]. The realistic states all fall within a triangular shape, also known as the Lumley triangle.

Later on, this concept was expanded by Banerjee [5] to form the barycentric map. This map, that can be observed in Figure 6.8, is based on the fact that the anisotropy has three limiting states. These are: the one-component turbulence corresponding to one non-zero eigenvalue of the anisotropy tensor, two-component turbulence corresponding to two non-zero eigenvalues and three-component turbulence with three non-zero eigenvalues. In this work, the componentiality reflects the number of non-zero velocity fluctuations u_i'. Any anisotropy tensor b_{ij} can then be expressed as a convex combination of the limiting states (one-component, two-component, three-component) as

$$b_{ij} = C_{1c}b_{1c} + C_{2c}b_{2c} + C_{3c}b_{3c} \tag{6.19}$$

where $\{C_{1c}, C_{2c}, C_{3c}\}$ are the coordinates of the anisotropy tensor b_{ij} in the tensor basis $\{b_{1c}, b_{2c}, b_{3c}\}$, which are the basis matrices for the three limiting states. The coordinates and normalizing coefficients $\{C_{1c}, C_{2c}, C_{3c}\}$ are all larger or equal to zero, they sum up to 1, and are defined through the eigenvalues of the anisotropy tensor as follows

$$\begin{aligned} C_{1c} &= \lambda_1 - \lambda_2 \\ C_{2c} &= 2\left(\lambda_2 - \lambda_3\right) \\ C_{3c} &= 3\lambda_3 + 1. \end{aligned} \tag{6.20}$$

The eigenvalues are ordered such that $\lambda_1 \leq \lambda_2 \leq \lambda_3$.

In conclusion, any anisotropy tensor can then be represented in the barycentric map of figure 6.8. If the Reynolds stresses are realizable, which means that the tensor is positive semi-definite, then the corresponding anisotropy mapping will fall inside such a triangle.

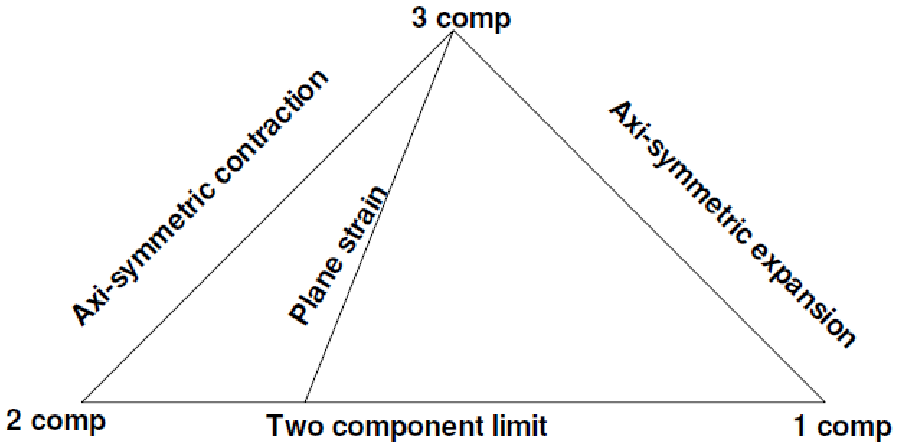


Figure 6.8: Barycentric map, identifying different componentialities of turbulence (taken from [5]).

7 Results

In this Chapter, the *Research Questions* introduced in Chapter 4 are going to be answered and the results commented on. It will first be graphically assessed whether a correlation between shedding and stochastic components is present (Section 7.1). Successively, Question Q1.1b will be addressed, hence answering whether an accurate model can be obtained when using the *phase*-averaged approach, which aims at targeting the stochastic anisotropy at different time steps inside a period using the coherent components of the velocity field. On the other hand, Section 7.3 presents the results of modelling when the *ensemble*-averaged methodology is applied. As this has shown promising performance, the obtained model is also tested on a different dataset (at $Re = 6000$) in order to evaluate its generalization capabilities.

7.1. Correlation between shedding and stochastic quantities

This Section addresses Question Q1.1a, therefore it examines whether there is a correlation between $\widetilde{U}$ and the stochastic quantities k'', τ''_{ij} and b''_{ij}.

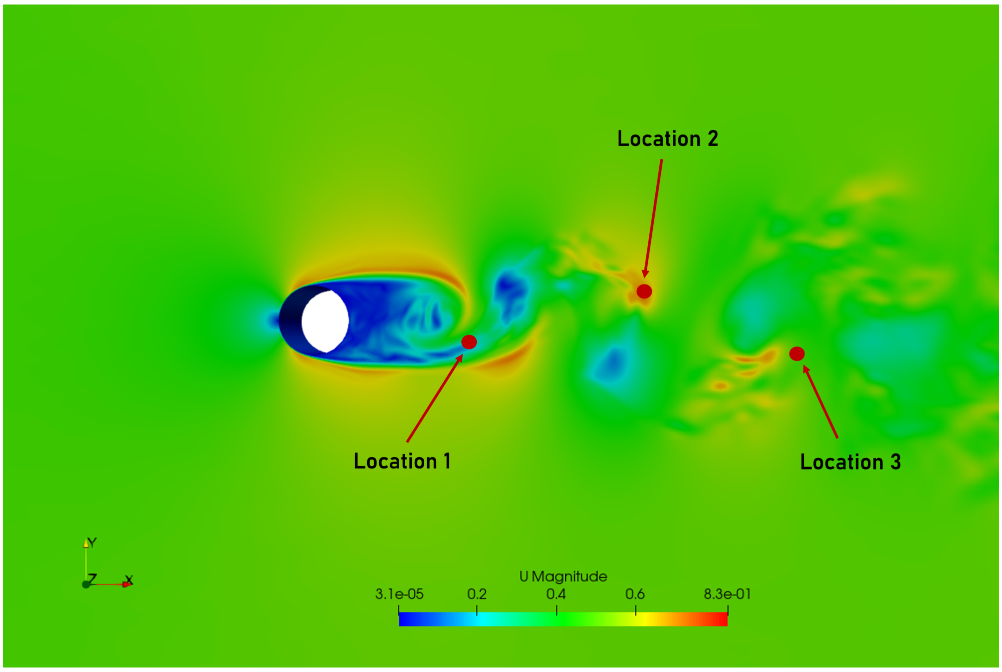
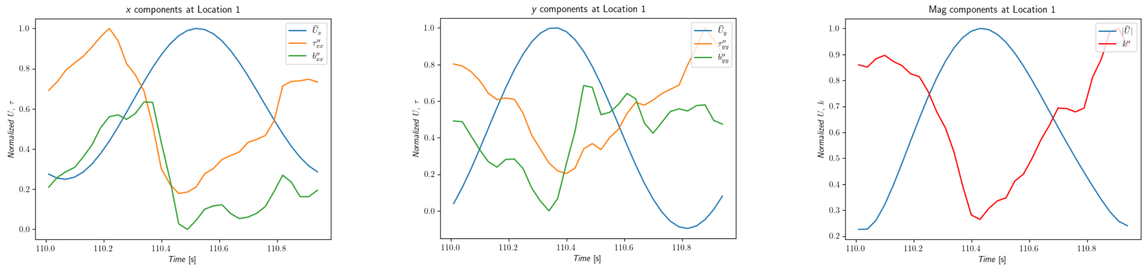


Figure 7.1: Locations where the mentioned quantities were probed.

The locations chosen for this analysis (indicated in Figure 7.1) are: at the end of the recirculation bubble (Location 1), and in the wake, in the proximity of two vortex cores (Location 2 and Location 3). The locations were picked because they are close to recurring phenomena, in particular vortex shedding, which happens at a regular frequency.

In Figures 7.2 to 7.5, the time series of the magnitude of normalized URANS and stochastic quantities are reported, in particular $\widetilde{U}_x$ to b''_{xx}, $\widetilde{U}_y$ to b''_{yy}, and $|\widetilde{U}|$ to k'', to analyze graphically whether a correlation is present.

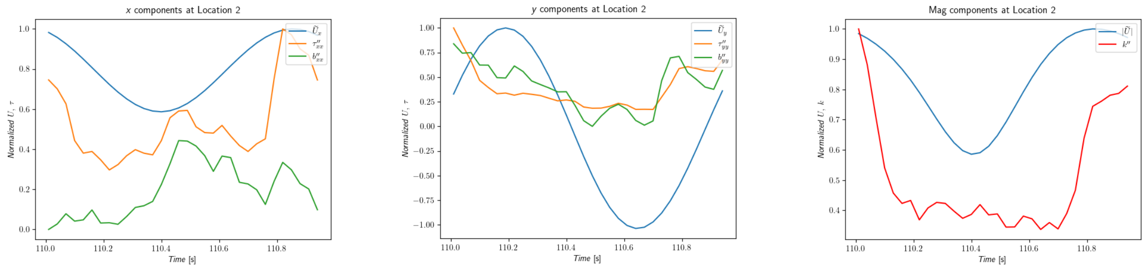
Location 1



(a) $\widetilde{U}_x$, τ''_{xx} and b''_{xx} **(b)** $\widetilde{U}_y$, τ''_{yy} and b''_{yy} **(c)** $|\widetilde{U}|$ and k''

Figure 7.2: Comparison of shedding and stochastic quantities at Location 1.

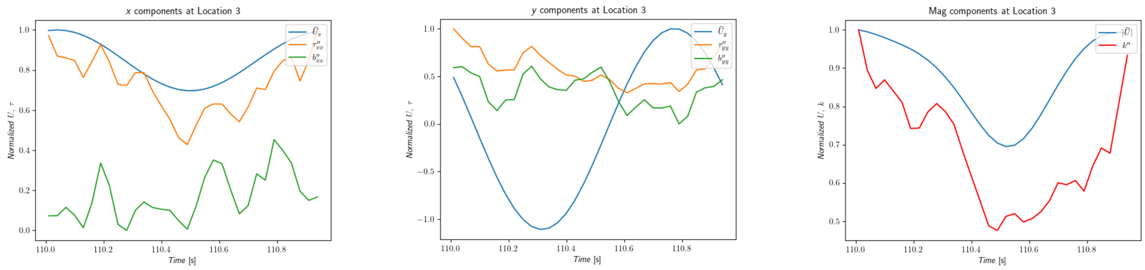
Location 2



(a) $\widetilde{U}_x$, τ''_{xx} and b''_{xx} **(b)** $\widetilde{U}_y$, τ''_{yy} and b''_{yy} **(c)** $|\widetilde{U}|$ and k''

Figure 7.3: Comparison of shedding and stochastic quantities at Location 2.

Location 3



(a) $\widetilde{U}_x$, τ''_{xx} and b''_{xx} **(b)** $\widetilde{U}_y$, τ''_{yy} and b''_{yy} **(c)** $|\widetilde{U}|$ and k''

Figure 7.4: Comparison of shedding and stochastic quantities at Location 3.

After looking at the plots of shedding and stochastic contributions, which are here shown over a phase-averaged period (as explained in Section 6.3), the following considerations can be drawn:

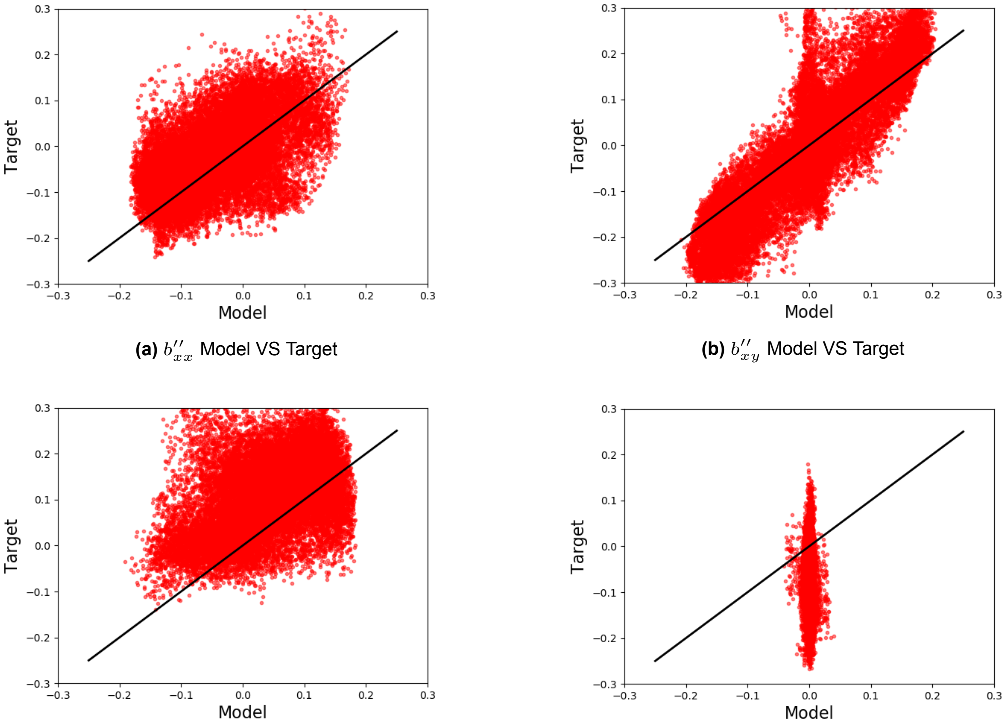
- there is a clear correlation between the magnitude of the velocity and the stochastic turbulent kinetic energy. However, this happens to be both anti-correlated (Figure 7.2c) and positively correlated (Figures 7.3c and 7.4c).
- At Location 1, the velocity $\widetilde{U}_x$ seems to be delayed compared to the Reynolds Stresses τ''_{xx} and anisotropy b''_{xx}. On the contrary, for the components in y, τ''_{yy} and b''_{yy} have a lag when compared to the velocity $\widetilde{U}_y$.
- At Location 2, a slight correlation can be spotted for the x components, but the same cannot be said for the y contributions.
- At Location 3, it is visible in Figure 7.4a, that there is quite a good correspondence between $\widetilde{U}_x$ and τ''_{xx}. However, b''_{xx}, τ''_{yy} and b''_{yy} show quite a stochastic trend, without following any sinusoidal time evolution.

Since the time evolutions showed some patterns of correlation between the shedding and stochastic quantities, and as these findings suggest a possible link between the resolved coherent velocity field and the stochastic anisotropy, the unsteady phase-averaged SpaRTA was implemented. The objective is to obtain a model for the stochastic anisotropy, in terms of quantities available in URANS simulations. In the next Section, the results obtained are presented.

7.2. Phase-averaged modelling from $\widetilde{U}_i$ to b''_{ij}

In the previous Section, the correlations between URANS and stochastic quantities were analyzed graphically. The next step is to test the capabilities of the phase-averaged SpaRTA framework. As explained in Section 6.3, the components of velocity are averaged over 15 vortex-shedding periods, and the derived quantities are interpolated over a single period.

In Figure 7.5a to 7.5d, the resulting regressions obtained with the best SpaRTA model are shown. Specifically, the figures show the *scatter plot* of the predicted anisotropy (on the *x*-axis) in relation to the true anisotropy value (on the *y*-axis) computed from equation (3.1). If the modelling was successful, we would expect most of the points to lie along the diagonal black line. However, it is not the case here.



(a) b''_{xx} Model VS Target

(b) b''_{xy} Model VS Target

(c) b''_{yy} Model VS Target

(d) b''_{zz} Model VS Target

Figure 7.5: Regression of anisotropy components using *phase*-averaged quantities.

The plots above clearly show that only the regression of b''_{xy} yields adequate results. We were though aiming at obtaining a good fit for all the other components. The overall bad performance is also confirmed by the R^2 score value obtained, which is around 0.4.

Since it was observed in Figures 7.2 to 7.5 that for some components the target quantities showed a **lag**, a different approach to modelling was also attempted. In fact, it appears as though there is an asynchronous association between shedding and stochastic components. The inputs and targets of the elastic net regression were then shifted by different phase amounts, aiming at finding better accuracy. However, no improvements were obtained from this investigation.

The phase-averaged methodology was therefore found to be ineffective for anisotropy modelling. As an alternative, the procedure proposed by Sandberg [21] was followed, hence targeting an ensemble-averaged stochastic anisotropy tensor.

7.3. Ensemble-averaged modelling from $\overline{U}_i$ to b''_{ij}

In this Section, we try to build a model between candidate functions based on mean velocity $\overline{U}$ and the ensemble-averaged stochastic anisotropy, computed using equation (6.9). The results are then discussed, by comparing barycentric maps and the performance at regressing specific components of the anisotropy.

After the model discovery using SpaRTA, different candidate models were obtained. Among all of them, the following criteria were considered when choosing the ”best” model:

- **Accuracy of the model**. This was evaluated in terms of mean squared error (MSE), or R^2 score, defined in Section 6.5. It is of course highly desired that the model does a good job at regressing the target quantity. For this assessment, the R^2 score gives an idea of the correlation between the model output and the quantity that is being modelled. The Pearson coefficient, which is also an indicator of correlation, was computed and used as a value for the comparison of different models.
- **Number of functions**. It is the number of terms that characterize the model and it was evaluated with some engineering sense and comparing to other data-driven anisotropy models from the literature. Both in [34] and [21], models up to 18 functions were selected as promising. That was then set as the maximum number of functions for choosing a candidate model. However, it was decided to lower this value, so as to try to obtain a more generalizable model. In fact, as a rule of thumb, the more functions used, the more accurate the regression, but at the same time, this can be an indication of overfitting of the training dataset. This means that if we test the same model on a different dataset, it is more prone to performing badly. This will later be investigated in Subsection 7.3.2.
- **Complexity of the functions**. A model is complex when too many functions are used. However, also the complicatedness of the functions themselves is a parameter to consider (for example, a high-order polynomial is more complex than a low-order one). For this reason, the degree of multiplication was limited to a second order. Each base tensor ($\boldsymbol{T_1}$, $\boldsymbol{T_2}$, $\boldsymbol{T_3}$) is then multiplied by a sum of functions which are the result of the product of maximum 2 different features (from Table 6.1) or invariants (I_1 and I_2).

The model selection step, hence the SpaRTA methodology, started with a total of 11026 candidate functions. After removing the functions which give *inf* and *nan* values, mean zero values and those functions which are highly correlated with each other, 1323 functions were left. We applied the SpaRTA framework to this limited set of functions and a model of 11 functions stood out, which were evaluated as a good compromise between accuracy and complexity. The criterias presented above drove the choice of the final selected model. The latter is reported in equation (7.1).

$$
\begin{aligned}
&\boldsymbol{b''_{ij}} \\
&= \\
&\frac{\boldsymbol{T_1}}{0.3145}\left(-0.0104\frac{q_\nu}{q_{TI}} - 0.025\frac{q_\gamma}{I_1} - 0.0072\frac{q_\nu}{\sqrt{q_{TI}}} - 0.0008\frac{q_{nu}^2}{q_{TI}} + 0.009\frac{tanh(I_2/0.8328)}{q_\gamma^2} + \frac{0.0448}{q_{\tau kB}\sqrt{q_{TI}}}\right) + \\
&\frac{\boldsymbol{T_2}}{0.4996}\left(-0.0177\frac{q_Q}{q_{TI}} + \frac{0.002}{I_1^2 q_{TI}}\right) + \\
&\frac{\boldsymbol{T_3}}{0.1670}\left(\frac{0.0038}{I_1} + \frac{0.0045}{q_T\sqrt{I_1}} + \frac{0.0087}{I_1\sqrt{q_{TI}}}\right)
\end{aligned}
\tag{7.1}
$$

It is here reported just one model, the one selected as most appropriate in terms of accuracy and potential generalizability. However, as anticipated, multiple models were found. In Appendix A, a matrix reporting the best performing models made of up to 30 functions is attached.

In Figure 7.6, values of $(1-R^2)$ are plotted for models with various numbers of functions. As the reader could expect, when increasing the number of functions, a higher R^2 score is achieved. It is observed that there is a large drop in the $(1-R^2)$ value when 11 terms are used instead of 7. When further increasing the number of functions, the error still drops but the rate of decrease is limited. For this reason, it was decided to choose the model in equation (7.1) as the ”best” one, as being the one with lower error while using a limited number of terms. We will refer to it as $M1$.

Note that in Subsection 7.3.2 we will also compare the model $M1$ to another model noted $M2$ (expression provided in Appendix C) which is characterized by a similar R^2 score (see Figure 7.6) but which is composed of more functions (25 instead of 11). The analysis of the performance of the model is also available in Appendix C.

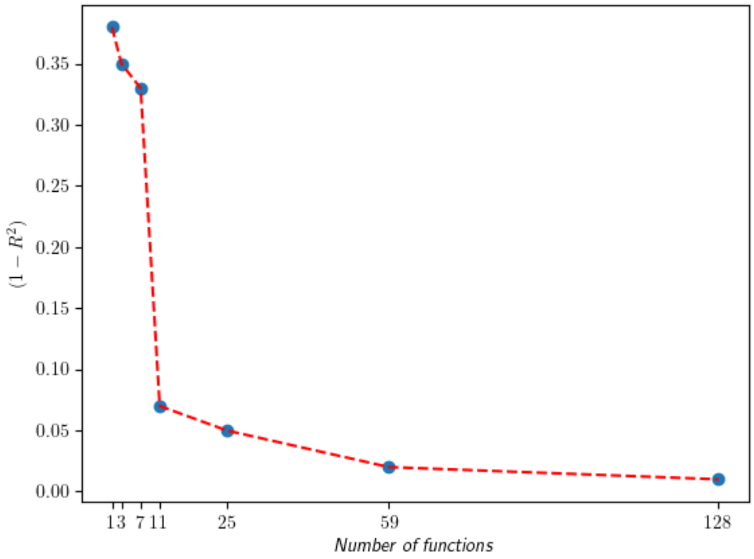


Figure 7.6: Variation of performance of the modelling when using a different number of functions.

7.3.1. Comparison of Boussinesq approximation and modelled anisotropy

After having assessed the good performance of the model in terms of R^2 score, we will now compare it to the modelling provided by the Boussinesq assumption, hence by the following relation:

$$b''_{ij} = -\frac{1}{2\omega''}\left(\frac{\partial \overline{U_i}}{\partial x_j} + \frac{\partial \overline{U_j}}{\partial x_i}\right) = -S_{ij} \tag{7.2}$$

where S_{ij} is the mean strain rate tensor, divided by ω'', and coincides with the tensor T^1_{ij} introduced in Subsection 6.4.1.

First, the barycentric maps are compared from Figures 7.7 to 7.9. For the sake of clarity, not all the points from the training dataset are plotted in these maps. Points representing a certain density are substituting the whole dataset of points. The colour map on the right of each figure is an indicator of the density (number of points) in each area. Figure 7.7 shows that the componentiality of the target turbulence is highly three-dimensional. In fact, the points lie on the upper right side of the barycentric triangle, which corresponds to the area of 3 components turbulence. Such a trend is poorly predicted by the Boussinesq assumption. This is confirmed by Figure 7.8, which represents a turbulent state that mimics the *plane strain* turbulence from Figure 6.8.

With success, the componentiality of the anisotropy tensor modelled through SpaRTA is in line with the target anisotropy. When comparing Figure 7.9 with Figure 7.7, it is clear that the points lie in a very similar region. They are displaced slightly differently, for example, the dots predicted by our model are denser towards the three-component turbulence, but overall they are in good agreement.

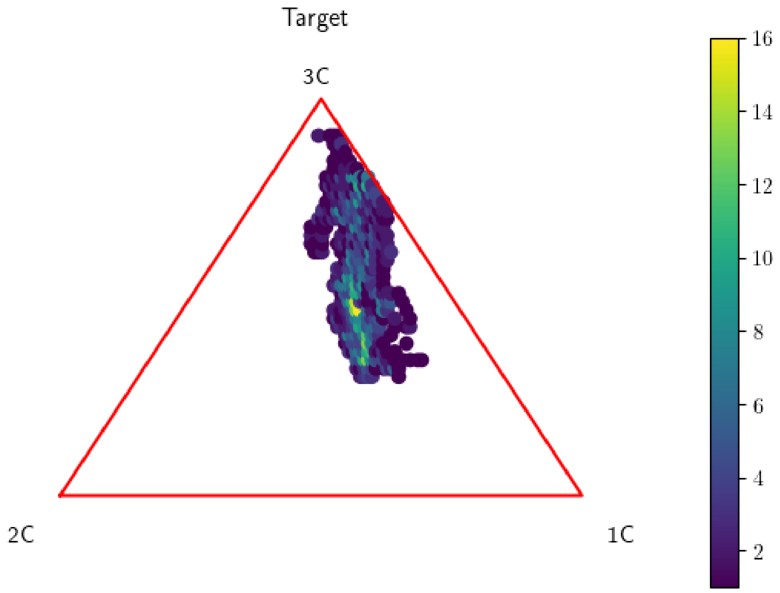


Figure 7.7: Barycentric map of target anisotropy.

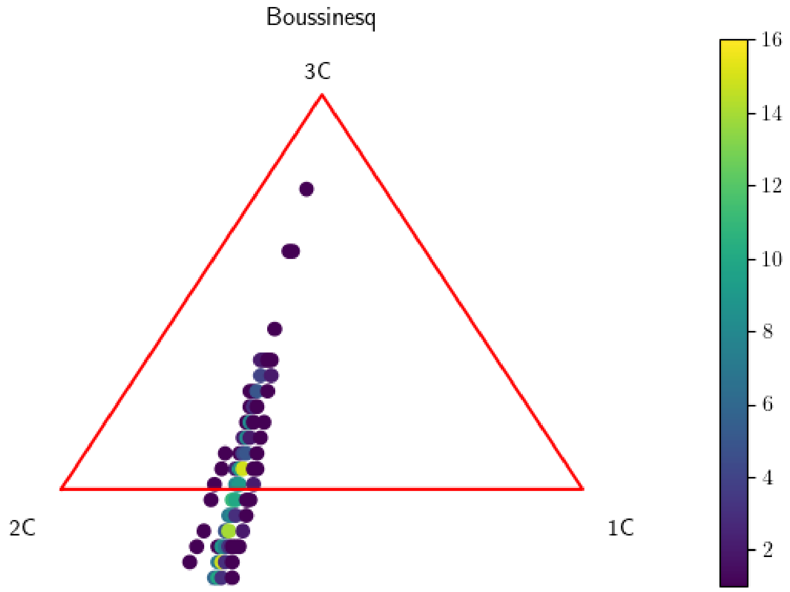


Figure 7.8: Barycentric map of anisotropy from Boussinesq.

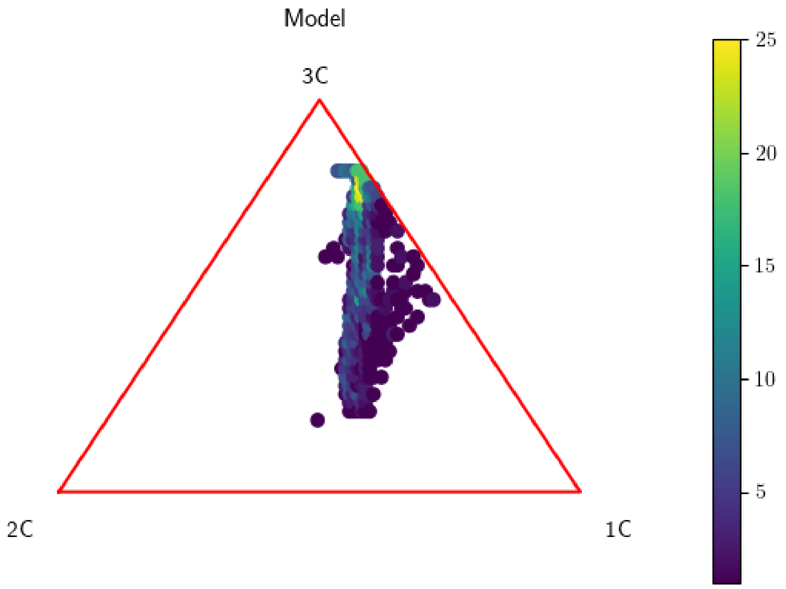
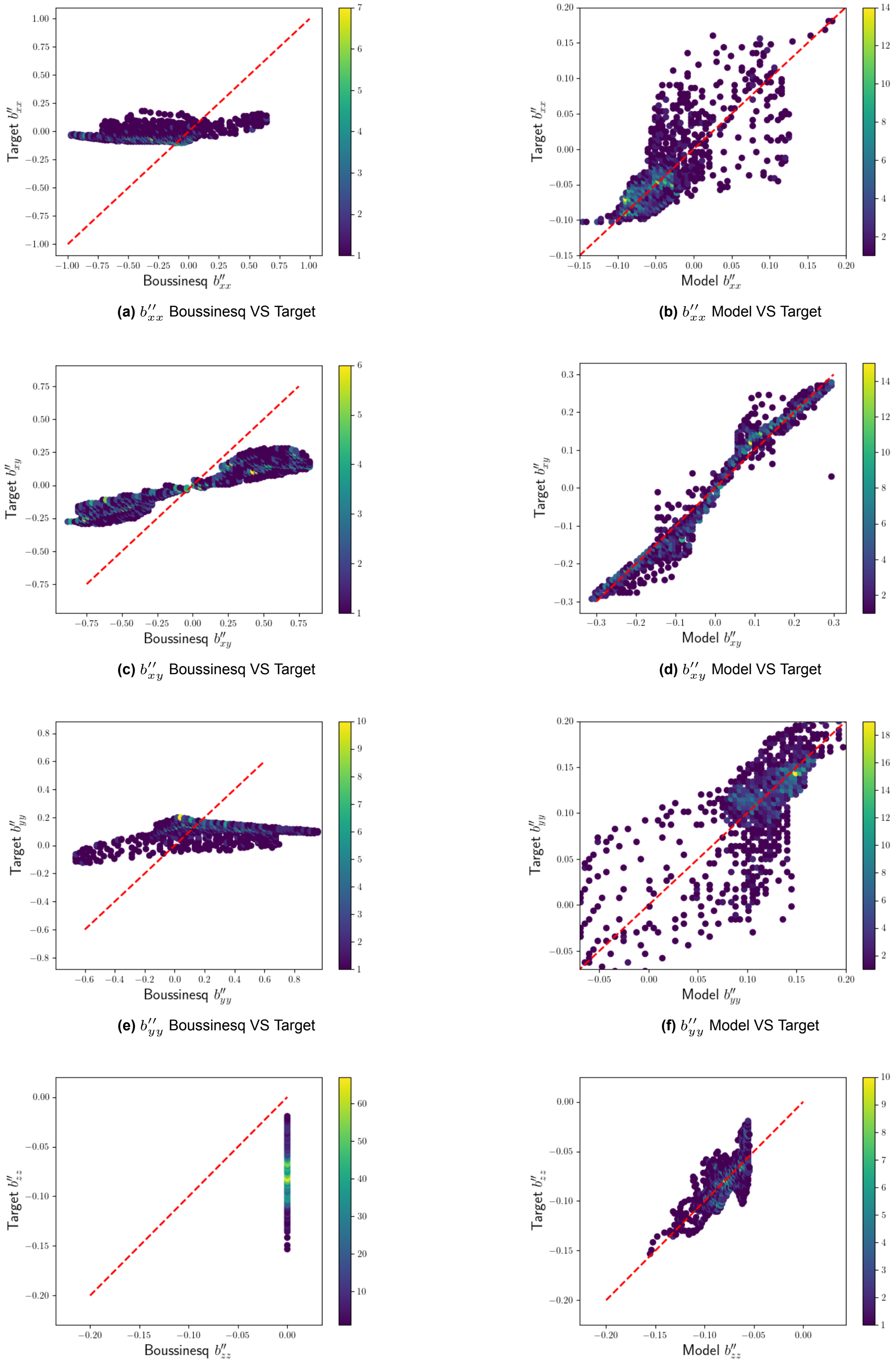


Figure 7.9: Barycentric map of modelled anisotropy.

After having analysed the barycentric maps, hence the type of turbulence that is predicted by Boussinesq and by our model, as opposed to the true turbulence, the components of the anisotropy tensor are now compared. In Figures 7.10a to 7.10h, the charts on the left display the results from Boussinesq assumption (equation (7.2)), while those on the right the regression capability of our model (equation (7.1)). For all the charts, the y-axis corresponds to the target anisotropy (equation (3.1)).

By looking at the plots, some observations can be done:

- the predictions from Boussinesq are largely inaccurate, and for the b''_{zz} component the regression is not even possible. This is explained by the fact that Boussinesq prediction relies on equation (7.2) and hence on the mean velocity gradients. Since the z velocity component is equal to zero for a 2D simulation, also its corresponding predicted anisotropy will be zero;
- on the contrary, our model allows regressing also b''_{zz}. This is possible thanks to the tensor $\boldsymbol{T_3}$ (equation 6.11) which thanks to the term $-\frac{1}{3}\delta_{ij}S_{mn}S_{nm}$ allows for the modelling of all the components on the diagonal (b''_{xx}, b''_{yy} and b''_{zz});
- our model performs particularly well at regressing the b''_{xy} component. This is due to the fact that the true value of this component lies in a wider range ([-0.3, 0.3]) than the other components. As our elastic-net regression is driven by the minimization of the mean square error between the predicted and target quantities (equation (6.17)), the algorithm is more prone to regressing better this quantity, giving up some accuracy when predicting the other contributions. This is also reflected in the barycentric coloured map in Figure 7.13. It can be seen that when compared to Figure 7.11, there is a very good agreement of predicting the area in correspondence of the shear layers coming off the cylinder body.



(a) b''_{xx} Boussinesq VS Target

(b) b''_{xx} Model VS Target

(c) b''_{xy} Boussinesq VS Target

(d) b''_{xy} Model VS Target

(e) b''_{yy} Boussinesq VS Target

(f) b''_{yy} Model VS Target

(g) b''_{zz} Boussinesq VS Target

(h) b''_{zz} Model VS Target

Figure 7.10: Comparison of regressing target anisotropy from Boussinesq (on the left) and through modelled anisotropy (on the right).

For further analysis, the barycentric points from Figures 7.7 to 7.9, are here transferred to the actual training flow region and represented in Figures 7.11 to 7.13. It is visible how the Boussinesq assumption performs badly at predicting the different componentialities of turbulence (Figure 7.12). On the contrary, the model achieved through SpaRTA shows an overall good agreement with the target componentialities. In particular, a very accurate prediction of the area coinciding with the shear layers coming from the cylinder body. This then confirms the high correlation that was shown in the b''_{xy} component (Figure 7.10d). In the plots shown here, the blue colour is for 3 components of turbulence, green for 2 components and red for 1 component only. It should be noticed that in this representation the vertices of one component and two components are mirrored compared to the representation of Figure 6.8.

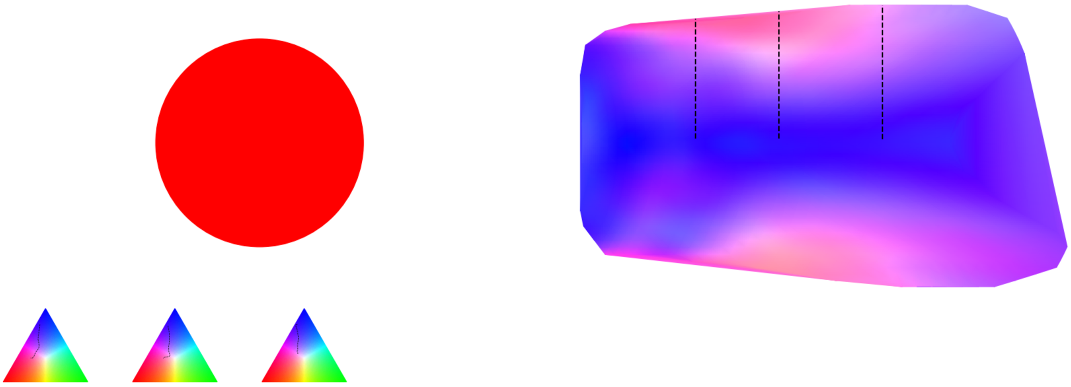


Figure 7.11: Barycentric coloured map of target anisotropy.

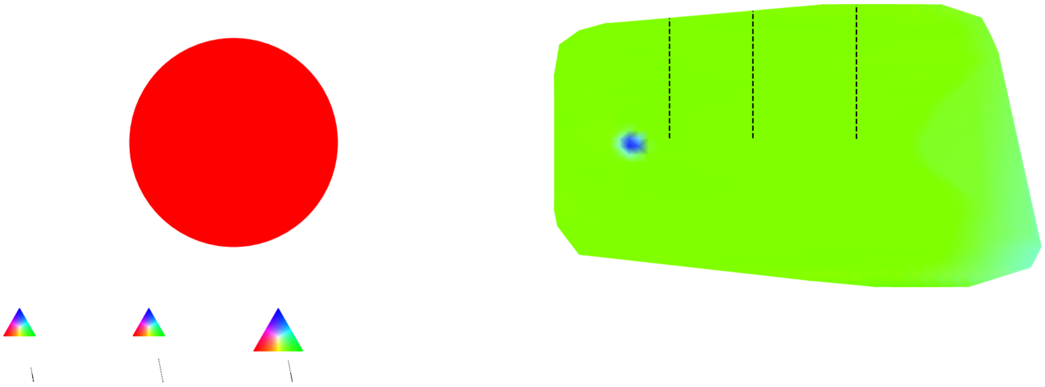


Figure 7.12: Barycentric coloured map of Boussinesq anisotropy.

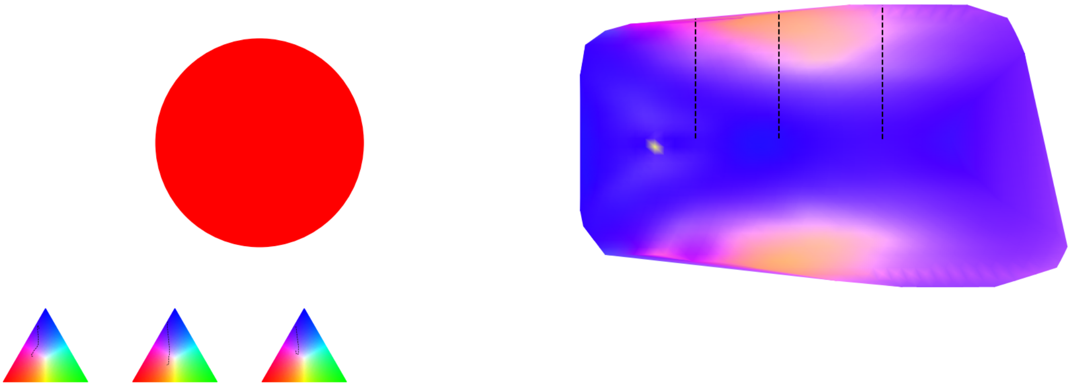
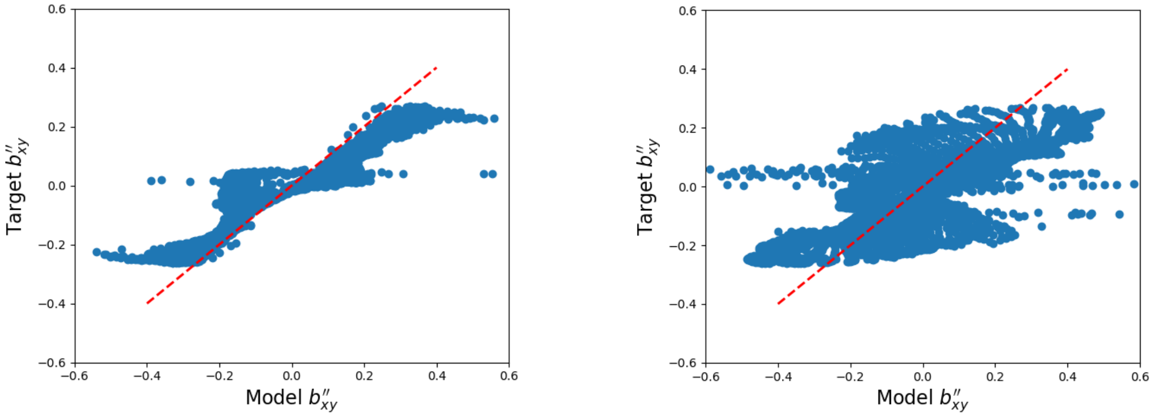


Figure 7.13: Barycentric coloured map of SpaRTA-modelled anisotropy.

7.3.2. Testing different models at $Re = 6000$

We now investigate the generalizability accuracy of models $M1$ and $M2$, by testing them on a different dataset than the one used for training. We remind the reader that $M1$ and $M2$ differ by their complexity, with $M2$ being composed of 25 functions, while $M1$ is composed of 11. The reference simulations were run at $Re = 3900$, and at this Reynolds number the models were also trained. Both models are now tested at $Re = 6000$, and the results of regressing component b''_{xy} are reported in Figure 7.14.



(a) $M1$ (with 11 functions).

(b) $M2$ (with 25 functions).

Figure 7.14: Comparison of generalizability accuracy at $Re = 6000$, when using model $M1$ (on the left) and model $M2$ (on the right).

At $Re = 3900$, while $M2$ with 25 functions performs better (see Figure 7.6), at $Re = 6000$, it can be seen that it performs worse than the model with 11 functions. This indicates that the model with 25 functions showed some overfitting.

8 Conclusions

In this Thesis work, the application of the unsteady SpaRTA methodology was extensively investigated. Different approaches were applied:

- graphical assessment of the correlation between coherent and stochastic quantities (Section 7.1);
- phase-averaged SpaRTA regression by using tensors and invariants derived from the coherent velocity field, targetting the stochastic anisotropy (Section 7.2);
- phase-averaged SpaRTA regression by using inputs and targets delayed by different amounts of phase shift (Section 7.2);
- ensemble-averaged SpaRTA regression by using tensors and invariants derived from the mean velocity field, targetting the stochastic anisotropy (Section 7.3);
- generalizability capabilities of the obtained models (Subsection 7.3.2).

Addressing the *Research Questions* that were driving this Thesis work, some conclusions can be drawn.

Is there any correlation between URANS velocity $\widetilde{U}$ and stochastic anisotropy b''_{ij}?

In Section 7.1 it was graphically assessed whether a correlation between shedding and stochastic quantities is present. The outcome is that there are certain correlations, but a common rule cannot be derived.
There are in fact quantities at certain locations that appear to be strictly correlated, such as:

- velocity and Reynolds stresses in Figures 7.3a and 7.4a, appear to be in phase with each other and follow a similar periodic behaviour;
- velocity, Reynolds stresses and anisotropy in Figures 7.2a and 7.2b occur to have the same periodic behaviour, but they are out of phase with each other. In Figure 7.2a, the velocity is in delay compared to τ''_{xx} and b''_{xx}, while the opposite happens in Figure 7.2b;
- velocity and turbulent kinetic energy are always correlated with each other, as shown in Figures 7.2c, 7.3c and 7.4c. Being the turbulent kinetic energy always positive by definition, the two quantities are positively correlated when $U > 0$, negatively correlated when $U < 0$. The turbulent kinetic energy associated with the stochastic component of the velocity has a similar contour as the velocity itself (Figure 6.5d). It looks like it follows the vortex shedding flow configuration, hence it could be expected that the URANS velocity and the stochastic TKE were correlated.

To concisely answer this question, we can say that **yes** there is a correlation between URANS velocity and stochastic components of the anisotropy, but this only happens in limited locations and not in a consistent way. It is therefore not possible to come up with a common rule which is valid in the majority of the flow field.

How can SpaRTA find such a model between $\widetilde{U}$ and b''_{ij} and how accurate is this model?

In order to answer this question, the unsteady SpaRTA framework was applied as a *phase*-averaged methodology, as explained in Section 6.3, and whose results are presented in Section 7.2. Overall, from Figures 7.5a to 7.5d, it can be deduced that such approach did not produce satisfactory models.

After assessing the visual correlation between shedding and stochastic quantities (Question Q1.1a), the result in Section 7.2 confirmed the general result of Section 7.1. Hence, that there might be correlations of certain components at certain locations, but no common rule can be found, and not even the elastic net regression procedure could identify a suitable model.

How can we use SpaRTA to model the relation between the mean flow quantities (for example $\overline{U}$) and the anisotropic Reynolds stress tensor b''_{ij}?

To answer Questions Q1.2 and Q1.2a, the unsteady SpaRTA framework was applied in an *ensemble*-averaged approach. The results are displayed in Section 7.3 and discussed.

The main takeaway is that the obtained model, composed by a limited number of terms (11 functions), evidently outperforms the anisotropy prediction from Boussinesq assumption. This is assessed in terms of R^2 score (see Figure 7.6), componentiality of the predicted turbulent state (Figures 7.9 and 7.13), but also in terms of performance in predicting singular components of the anisotropy (Figures 7.10a to 7.10h).

As presented in Subsection 7.3.2, the chosen best model (equation (7.1)) is also tested on a different dataset ($Re = 6000$), showing good predictive performance.

9
Future Work

After having answered the *Research Questions* and drawn final conclusions, some recommendations for future work will here be given.
They can be summarized in the following points:

- *A priori*, promising results were obtained when applying the ensemble-averaged SpaRTA methodology. A natural follow-up of this Thesis work is then *a posteriori* investigation of the obtained models. Equation (7.1) can then be directly implemented in a URANS solver as:

$$\tau_{ij}^{URANS} = 2k\frac{\delta_{ij}}{3} - 2\nu_t S_{ij} + 2kb''_{ij} \tag{9.1}$$

  It would then act as a substitute for the Boussinesq assumption, which bases the anisotropy modelling only on the mean strain rate tensor S_{ij}.
- To further extend the generalizability capabilities of the corrective models, the training of the models can be done using different geometries and flow cases, at different Reynolds numbers.
- If a large amount of computational memory is available, the phase-averaged approach could again be tried by averaging over a very large number of vortex-shedding periods. It could be that the 15 periods used in this dissertation give good averaged quantities when doing an ensemble average, but they might not be enough to obtain converged statistics when applying phase-averaging.

References

[1] H. D. Akolekar et al. “Development and Use of Machine-Learnt Algebraic Reynolds Stress Models for Enhanced Prediction of Wake Mixing in Low-Pressure Turbines”. In: *Journal of Turbomachinery* 141 (2019).

[2] H. D. Akolekar et al. “Integration of machine learning and computational fluid dynamics to develop turbulence models for improved low pressure turbine wake mixing prediction”. In: *Journal of Turbomachinery* 143 (2021).

[3] S. Arvidson. “Methodologies for RANS-LES interfaces in turbulence resolving simulations”. In: *Department of Mechanics and Maritime Sciences* (2017).

[4] P. Baj, P. J. K. Bruce, and O. R. H. Buxton. “The triple decomposition of a fluctuating velocity field in a multiscale flow”. In: *Physics of Fluids* 27 (2015).

[5] S. Banerjee et al. “Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches”. In: *Journal of Turbulence* 8 (2007).

[6] G. Berkooz, P. Holmes, and J. L. Lumley. “The proper orthogonal decomposition in the analysis of turbulent flows”. In: *Annu. Rev. Fluid Mech.* 25 (1993), pp. 539–75.

[7] R. D. Blevins. *Flow Induced Vibration*. 2nd Ed. Van Nostrand Reinhold, 1990.

[8] J. Boussinesq. *Théorie de l’écoulement tourbillonnant et tumultueux des liquides dans les lits rectilignes a grande section*. 1st ed. Paris: Gauthier-Villars et fils, 1897.

[9] S. Dey et al. “Reynolds stress anisotropy in flow over two-dimensional rigid dunes”. In: *Royal Society Publishing Proceedings A* 476 (2020).

[10] K. Duraisamy, G. Iaccarino, and H. Xiao. “Turbulence modeling in the Age of Data”. In: *Annual Review of Fluid Mechanics* 51 (2019), pp. 357–77.

[11] K. Duraisamy, Z. J. Zhang, and A. P. Singh. “New approaches in turbulence and transition modeling using data-driven techniques”. In: *AIAA SciTech* 53 (2015).

[12] J. Franke and W. Frank. “Large eddy simulation of the flow past a circular cylinder at Re_D=3900”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 90 (2002).

[13] T. B. Gatski and C. G. Speziale. “On explicit algebraic stress models for complex turbulent flows”. In: *J. Fluid Mechanics* 254 (1993), pp. 59–78.

[14] J. Hart. “A New Dynamic One-Equation Subgrid-Scale Model for Large Eddy Simulations”. In: *33rd Aerospace Sciences Meeting and Exhibit* AIAA 95-0356 (1995).

[15] J. Hart. “Comparison of turbulence modeling approaches to the simulation of a dimpled sphere”. In: *Procedia Engineering* 147 (2016), pp. 68–73.

[16] J. R. Holland, J. D. Baeder, and K. Duraisamy. “Towards Integrated Field Inversion and Machine Learning With Embedded Neural Networks for RANS Modeling”. In: *AIAA SciTech Forum* 1884 (2019).

[17] J. R. Holland, J. D. Baeder, and K.Duraisamy. “Field inversion and machine learning with embedded neural networks: physics-consistent neural network training”. In: *AIAA Aviation Forum* (2019).

[18] J. P. Huijing, R. P. Dwight, and M. Schmelzer. “Data-driven RANS closures for three-dimensional flows around bluff bodies”. In: *Computers and Fluids* 225 (2021).

[19] N. D. Katopodes. *Free Surface Flow - Environmental Fluid Mechanics*. 1st Ed. Butterworth-Heinemann, 2019.

[20] A. N. Kolmogorov. “The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers”. In: *Dokl. Akad. Nauk SSSR* 30 (1941), pp. 299–303.

[21] C. Lav, J. Philip, and R. D. Sandberg. “A new data-driven turbulence model framework for unsteady flows applied to wall-jet and wall-wake flows”. In: *Proceedings of ASME Turbo Expo 2019* GT2019-90179 (2019), pp. 1–12.

[22] C. Lav, J. Philip, and R. D. Sandberg. “Improvement in unsteady wake prediction through machine learning based RANS model training”. In: *Proceedings of the 15th International Symposium on Unsteady Aerodynamics, Aeroacoustics Aeroelasticity of Turbomachines* (2018).

[23] J. Ling, R. Jones, and J. Templeton. “Machine learning strategies for systems with invariance properties”. In: *Journal of Computational Physics* 318 (2016), pp. 22–35.

[24] J. Ling, A. Kurzawski, and J. Templeton. “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance”. In: *Journal of Fluid Mechanics* 807 (2016), pp. 155–166.

[25] J. L. Lumley. “Computational modeling of turbulent flows”. In: *Advances in Applied Mechanics* 18 (1979), pp. 123–176.

[26] J. L. Lumley and G. R. Newman. “The return to isotropy of homogeneous turbulence”. In: *Journal of Fluid Mechanics* 82 (1977), pp. 161–178.

[27] D. A. Lysenko, I. S. Ertesvag, and K. E. Rian. “Large-Eddy Simulation of the flow over a circular cylinder at Reynolds number 3900 using the OpenFOAM toolbox”. In: *Flow, Turbulence and Combustion* 89 (2012), pp. 491–518.

[28] F. R. Menter. “Performance of popular turbulence model for attached and separated adverse pressure gradient flows”. In: *AIAA Journal* 30 (1992), pp. 2066–2072.

[29] F. R. Menter. “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications”. In: *AIAA Journal* 32 (1994), pp. 1598–1605.

[30] S. B. Pope. “A more general effective-viscosity hypothesis”. In: *J. Fluid Mechanics* 72 (1975), pp. 331–340.

[31] S. B. Pope. *Turbulent Flows*. 1st Ed. United Kingdom: Cambridge Univeristy Press, 2000.

[32] O. Reynolds. “On the dynamical theory of incompressible viscous fluids and the determination of the criterion”. In: *Proc. R. Soc. Lond.* 56 (1894), pp. 40–45.

[33] A. Roshko. “Perspectives on bluff body aerodynamics”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 49 (1993), pp. 79–100.

[34] I. B. H. Saidi et al. “CFD-driven Symbolic Identification of Algebraic Reynolds Stress Models”. In: *Journal of Computational Physics* 457 (2022).

[35] R. D. Sandberg et al. “Applying machine learnt explicit agebraic stress and scalar flux models to a fundamental trailing edge slot”. In: *Journal of Turbomachinery* 140 (2018).

[36] M. Schmelzer, R. P. Dwight, and P.Cinnella. “Discovery of algebraic Reynolds-stress models using sparse symbolic regression”. In: *Flow, Turbulence and Combustion* 104 (2020), pp. 579–603.

[37] J. Smagorinsky. “General circulation experiments with primitive equations. The basic experiment”. In: *Monthly Weather Review* 91 (1963), pp. 99–164.

[38] P. R. Spalart and S. R. Allmaras. “A one-equation turbulence model for aerodynamic flows”. In: *30th Aerospace Sciences Meeting and Exhibit* (1992).

[39] D. S. K. Ting. *Basics of Engineering Turbulence*. 1st Ed. Canada: Academic Press, 2016.

[40] P. S. Volpiani et al. “Machine learning-augmented turbulence modeling for RANS simulations of massively separated flows”. In: *Physical Review Fluids* 6 (2021).

[41] J. X. Wang, J. L. Wu, and H. Xiao. “A physics informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data”. In: *Physical Review Fluids* 2 (2017).

[42] J. Weatheritt and R. D. Sandberg. “A novel evolutionary algorithm applied to algebraic modifications of the RANS stress-strain relationship”. In: *Journal of Computational Physics* 325 (2016), pp. 22–37.

[43] J. Weatheritt and R. D. Sandberg. “The development of algebraic stress models using a novel evolutionary algorithm”. In: *International Journal of Heat and Fluid Flow* 68 (2017), pp. 298–318.

[44] D. C. Wilcox. “Reassessment of the scale-determining equation for advanced turbulence models”. In: *AIAA Journal* 26 (1988), pp. 1299–1310.

[45] C. H. K. Williamson. “Vortex dynamics in the cylinder wake”. In: *Annu. Rev. Fluid Mech.* 28 (1996), pp. 477–539.

[46] M. M. Zdravkovich. *Flow around Circular Cylinders - Volume 1: Fundamentals*. 1st Ed. Oxford Science Publications, 1997.

[47] Y. Zhao et al. “RANS turbulence model development using CFD-driven machine learning”. In: *Journal of Computational Physics* 411 (2020).

[48] X. H. Zhou, J. Han, and H. Xiao. “Learning nonlocal constitutive models with neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 384 (2021).

A

Model Matrix for models from 1 to 30 functions at $Re = 3900$

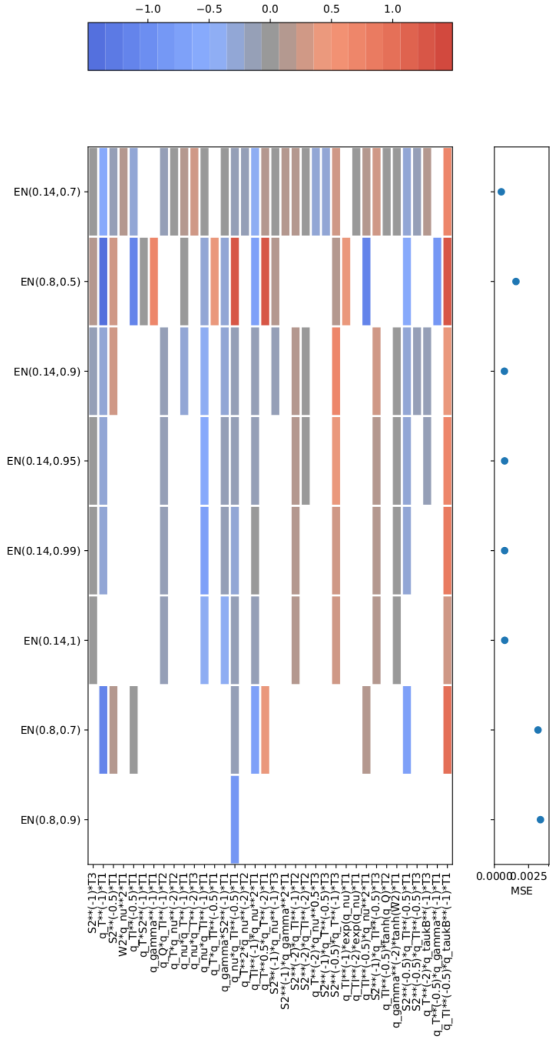


Figure A.1: Different models between 1 and 30 functions. Comparing number and type of functions with MSE.

B

3-D Tensors and Invariants

Given the bi-dimensionality of our flow case, to build the library of candidate functions when applying the SpaRTA framework, a limited set of tensors and invariants was used (equation (6.11)).

When we want to apply the proposed methodology to a 3D case, the full set proposed by Pope [30] should be used. The definitions of such quantities are here reported.
Ten tensors:

$$\begin{aligned}
T_{ij}^{1} &= S_{ij} & T_{ij}^{2} &= S_{ik}\Omega_{kj} - \Omega_{ik}S_{kj} \\
T_{ij}^{3} &= S_{ik}S_{kj} - \frac{1}{3}\delta_{ij}S_{mn}S_{nm} & T_{ij}^{4} &= \Omega_{ik}\Omega_{kj} - \frac{1}{3}\Omega_{mn}\Omega_{nm} \\
T_{ij}^{5} &= \Omega_{ik}S_{kj}S_{ji} - S_{ik}S_{kj}\Omega_{ji} & T_{ij}^{6} &= \Omega_{ik}\Omega_{kj}S_{ji} + S_{ik}S_{kj}\Omega_{ji} - \frac{2}{3}\delta_{ij}S_{ik}\Omega_{ki}\Omega_{ij} \\
T_{ij}^{7} &= \Omega_{ik}S_{kj}\Omega_{ik}\Omega_{kj} - \Omega_{ik}\Omega_{kj}S_{ik}\Omega_{kj} & T_{ij}^{8} &= S_{ik}\Omega_{kj}S_{ik}S_{kj} - S_{ik}S_{kj}\Omega_{ik}S_{kj} \\
T_{ij}^{9} &= \Omega_{ik}\Omega_{kj}S_{ik}S_{kj} + S_{ik}S_{kj}\Omega_{ik}\Omega_{kj} + \dots & T_{ij}^{10} &= \Omega_{ik}S_{kj}S_{ji}\Omega_{ik}\Omega_{kj} - \Omega_{ik}\Omega_{kj}S_{ji}S_{ik}\Omega_{kj} \\
&\quad - \frac{2}{3}\delta_{ij}S_{mn}S_{nm}\Omega_{mn}\Omega_{nm}
\end{aligned}$$

and five invariants:

$$\begin{aligned}
I_1 &= S_{mn}S_{nm} & I_2 &= \Omega_{mn}\Omega_{nm} \\
I_3 &= S_{lm}S_{mn}S_{nl} & I_4 &= \Omega_{ln}\Omega_{nm}S_{ml} \\
I_5 &= \Omega_{lm}\Omega_{mn}S_{nm}S_{ml}
\end{aligned}$$

C

Definition and Performance of $M2$ Model

The model $M2$ introduced in Section 7.3 is here displayed in equation (C.1). The 25 terms model gives a slightly better R^2 score when compared to $M1$ (equation (7.1)), but the additional complexity is not worth the little increase in accuracy. It was also demonstrated in Subsection 7.3.2, that when testing the two models $M1$ and $M2$ on a different dataset than the training one, $M2$ performs worse than $M1$. Also for this reason, $M1$ was selected overall as a better model.

$$\begin{aligned}
&\boldsymbol{b''_{ij}} \\
&= \\
&\frac{\boldsymbol{T_1}}{0.3145}\left(\frac{0.0076}{q_T} - \frac{0.0162}{\sqrt{q_{TI}}} + 10^{-5}\frac{q_\nu}{q_T} - 0.0074\frac{q_\nu}{q_{TI}} + 0.051\frac{q_\gamma}{I_1} + 0.0137\frac{q_{nu}}{\sqrt{q_{TI}}} + \ldots\right. \\
&\qquad\left. - 0.0038\frac{q_\nu^2}{q_{TI}} - \frac{0.0017}{q_T^{3/2}} - 0.0308\frac{q_\gamma^2}{I_1} - 1.2 * 10^{-5}\frac{e^{(q_\nu/0.6416)}}{q_{TI}^2} - 0.0024\frac{q_\nu^2}{\sqrt{q_{TI}}}\right) + \\
&\frac{\boldsymbol{T_2}}{0.4996}\left(-0.048\frac{q_Q}{q_{TI}} + 0.051\frac{q_T}{q_\nu^2} - 2.5\frac{q_T^2}{q_\nu^2} + \frac{0.003}{I_1^2 q_{TI}} - \frac{0.0001}{I_1^2 q_{TI}^2} + 0.007\frac{tanh(\frac{q_Q}{0.375})}{\sqrt{q_{TI}}} - 0.015\frac{tanh(\frac{q_Q}{0.375})}{q_\gamma^2}\right) + \\
&\frac{\boldsymbol{T_3}}{0.1670}\left(\frac{0.0131}{I_1} + 0.0005\frac{q_\nu}{q_T^2} - \frac{0.042}{I_1 q_\nu} - 0.0018\frac{\sqrt{q_\nu}}{q_T^2} - \frac{0.0148}{I_1\sqrt{q_T}} + \frac{0.0217}{\sqrt{I_1}q_T} + \frac{0.0017}{q_T^2 q_{\tau kB}}\right)
\end{aligned} \tag{C.1}$$

The performance of $M2$ is here analysed in terms of the capabilities at regressing the b''_{zz} and at predicting the turbulence componentiality.

When comparing Figure C.1 with Figure 7.10h, it can be observed that in Figure C.1 a larger number of points follows the diagonal line. This is an indication that $M2$ performs better at regressing the b''_{zz} component. This can be explained by the additional terms multiplying the $\boldsymbol{T_3}$ tensor in the $M2$ expression. It was explained in Subsection 7.3.1 that the b''_{zz} component can only modelled through $\boldsymbol{T_3}$. This little increase in accuracy is responsible for the increase in the R^2 score. It was in fact noticed that the other components (b''_{xx}, b''_{xy} and b''_{yy}) are equally modelled by $M1$ and $M2$.

Figure C.2 shows the predicted turbulence componentiality in the barycentric map, in the same way as it was done for model $M1$ in Figure 7.9. When comparing the two figures to the barycentric map of the true anisotropy (Figure 7.7), it can be concluded that $M2$ better predicts the spread of the point from top to bottom of the barycentric map. On the contrary, $M1$ places an excessive number of points towards the $3C$ corner.

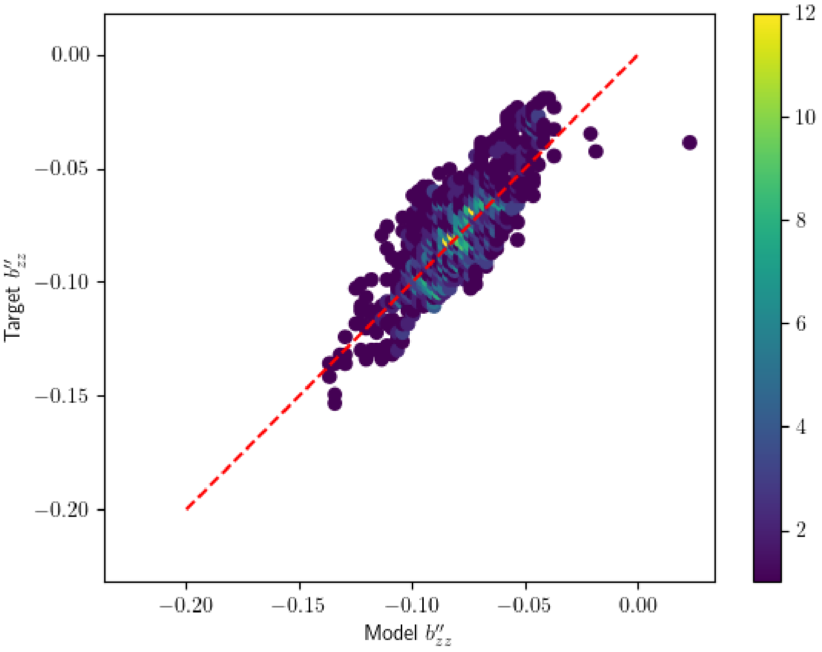


Figure C.1: Regression of b''_{zz} component when using 25 functions.

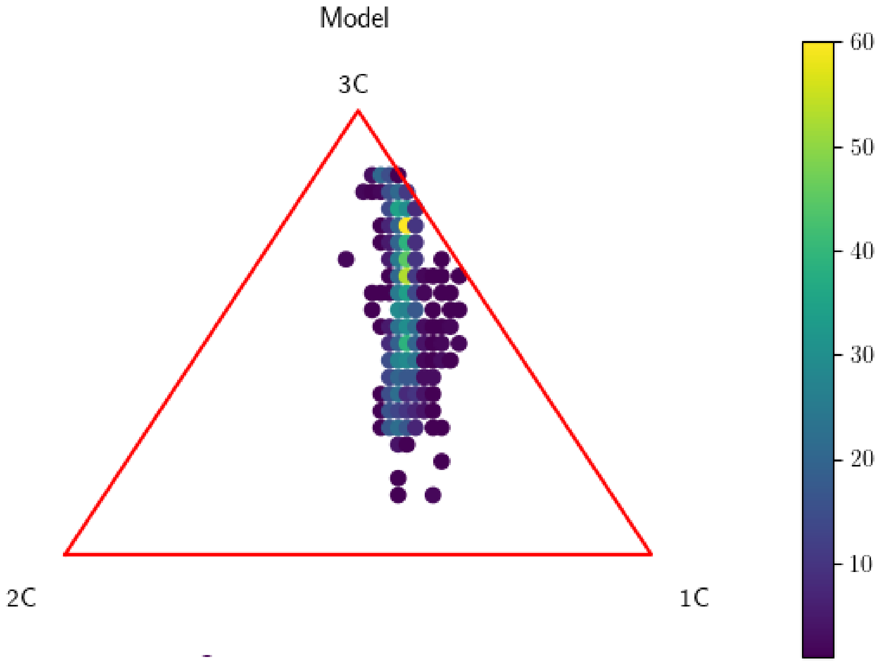


Figure C.2: Barycentric map when using 25 functions.