Université de Liège
Faculté des Sciences appliquées
Département d'aérospatiale et mécanique
Design of Turbomachines

Development of machine learning based wall shear stress models for LES in the presence of adverse pressure gradients and separation

Margaux BOXHO

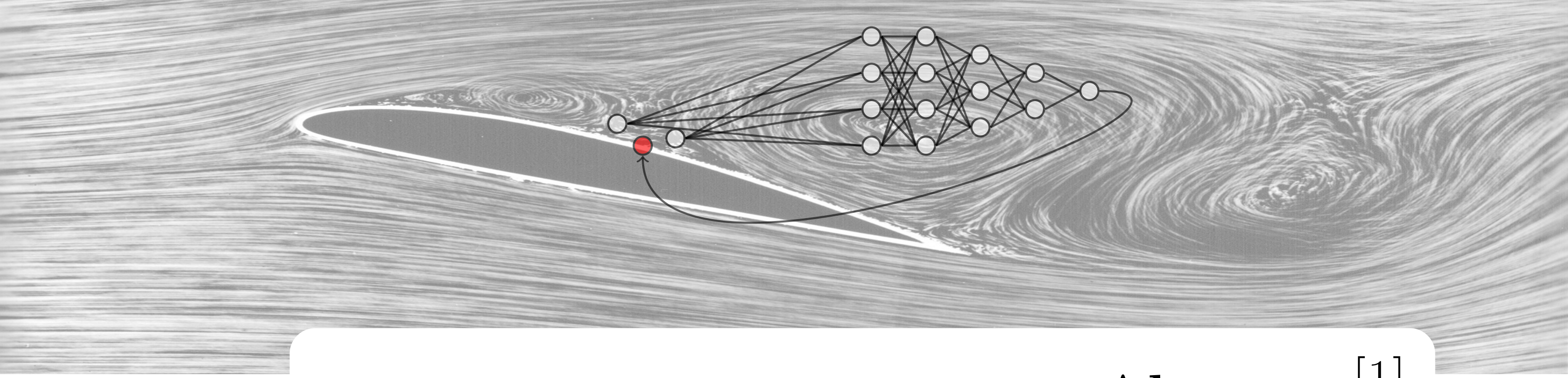
Composition du jury:

Pr. Koen Hillewaert (Promoteur)	Université de Liège, Belgique
Pr. Grégoire Winckelman (Co-promoteur)	Université catholique de Louvain, Belgique
Pr. Christophe Geuzaine (Président du jury)	Université de Liège, Belgique
Pr. Vincent E. Terrapon	Université de Liège, Belgique
Pr. Maria Vittoria Salvetti	Université de Pise
Pr. Gilles Louppe	Université de Liège, Belgique
Dr. Grégory Dergham	Safran Tech, France
Dr. Thomas Toulorge	Cenaero, Belgique

Liège, 21st of February, 2024

En collaboration avec UCLouvain, Safran et Cenaero

UCLouvain SAFRAN Cenaero



Abstract[1]

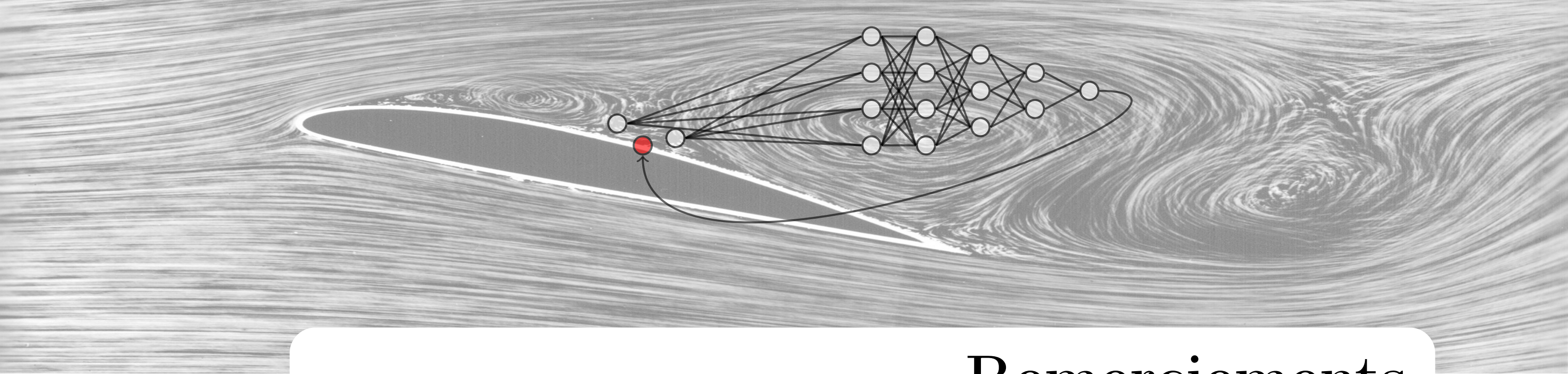
The optimization of jet engines continues to be a prominent area of interest, particularly with the new environmental standards in place. The need for reduced carbon emissions requires engineers to redesign the rotatory parts with fewer components to reduce weight while maintaining a high level of efficiency. These modifications increase the blade loading resulting in large adverse pressure gradients on the suction side of the airfoil. For instance, under these conditions and given the low Reynolds numbers at the last stages of a low-pressure turbine (LPT) and even more on low-pressure compressor (LPC), the boundary layer may separate on the rear portion of the blade suction side. This phenomenon can have a significant impact on the overall efficiency of the turbomachines. To address this issue during the design process, it is essential to *a priori* assess the size of the recirculation bubble and attempt to minimize it as much as possible.

The industry standard for simulating the stage-scale flow is the Reynolds-Averaged Navier-Stokes (RANS) method. However, RANS frequently fails at off-design conditions due to its inherent modeling assumptions. As an alternative, Large Eddy Simulation (LES) reduces the modeling assumptions by accurately resolving a significant part of the unsteady flow but remains costly at high Reynolds numbers. Wall models reduce the computational cost of LES by modeling the near-wall energetic scales and enabling the application of LES to complex flow configurations of engineering interest. However, most wall models assume that the boundary layer is fully turbulent, at equilibrium, and attached. While these models have proven successful in turbulent boundary layers under moderate adverse pressure gradients, when the adverse pressure gradient becomes too strong, and the boundary layer separates, equilibrium wall models are no longer applicable. To overcome this limitation, this study proposes the use of Machine Learning (ML) and Deep Learning (DL) tech-

[1] Photograph of a airfoil in a wind tunnel, showing separated flow over the top surface. DLR, CC-BY 3.0

niques to develop wall shear stress models thanks to the recent hardware advances combined with the exponential generation and accumulation of data. The Mixture Density Network (MDN), originally developed to predict uncertainty, is employed as a wall shear stress (WSS) model for the first time in the context of wall-modeled Large Eddy Simulations (wmLES) of separated flows. Such a network does not predict the mean wall shear stress conditioned by the inputs, but instead predicts the wall shear stress probability distribution, assuming that any distribution can be approximated by a linear combination of Gaussian distributions. The focus on the accurate prediction of the first two statistical moments is crucial for separated flows.

Before training the MDN, intensive work on the database has been conducted to study the near-wall physics and select suitable model input features. To study the near-wall physics in different flow configurations, the relationship between the instantaneous wall shear stress, velocity field, and pressure gradients is evaluated using space-time correlations. These correlations are extracted from two wall-resolved LES: a channel flow at a friction Reynolds number Re_τ of 950 and the two-dimensional periodic hill at a bulk Reynolds number Re_b of 10,595. This latter test case features a separation from the top of the hill leading to the development of a free shear layer that reattaches further downstream, creating a large recirculation bubble. The analysis of the correlations highlights that no instantaneous and local correlation is observed in the vicinity of the separation. The domain of high correlation appears to be shifted downstream. Therefore, the model inputs are the velocity field, the pressure gradients, and the curvature extracted at a given wall distance for multiple streamwise positions (taking both upstream and downstream from the prediction location). These inputs are carefully non-dimensionalized using the density, viscosity, and wall-model height, for better generalizability. An MDN is trained on turbulent channel flows at various friction Reynolds numbers and on the two-dimensional periodic hill at the bulk Reynolds number of 10,595. The model is then evaluated *a priori* on synthetic data generated from the law of the wall and shows good generalizability to higher friction Reynolds numbers. The relevance of the MDN-model is subsequently evaluated *a posteriori* by performing wmLES using the in-house flow solver Argo-DG, on two channel flows and the separated flow of the two dimensional periodic hill. The novel WSS model outperforms the existing data-driven WSS models in the literature on turbulent channel flows. The prediction is significantly improved compared to the WSS model based on Reichardt's velocity profile on a turbulent separated boundary layer. Nonetheless, the size of the recirculation bubble is still underpredicted, indicating a direction for future research.



Remerciements

La thèse ce n'est pas juste un chercheur qui reste assis derrière son écran toute la journée. La thèse, c'est aussi et surtout une aventure qui s'étale sur quatre ans avec ces hauts et ses bas, ses nombreux rebondissements, ses riches discussions et ses rencontres inoubliables. Tout comme le vol en solitaire d'Amy Johnson en 1930 qui relia l'Angleterre au port Darwin en Australie en compagnie de son biplace surnommé *Jason*, un De Havilland DH.60 G. Gipsy Moth, une thèse est rythmée par divers évènements depuis la recherche du sujet et de son financement jusqu'à son accomplissement, en passant par quelques embuches qui mettent le moral à rude épreuve. Cette formidable aventure, je le dois avant tout au Professeur Grégoire Winckelmans, professeur à l'Université Catholique de Louvain-la-Neuve, qui a cru en moi et m'a proposé ce sujet de thèse, entièrement financé par Safran Tech. Je voulais vous remercier Professeur pour tout le savoir que vous m'avez transmis à vos cours et cette proposition de thèse qui fut l'opportunité de rencontrer une équipe formidable à Cenaero.

Le 2 septembre 2019, je réalisais mes premiers pas (ou plutôt mes premiers battements d'aile) à Cenaero, un centre de recherche en aéronautique basé à Gosselies, juste en face de l'aéroport de Charleroi, quoi de mieux comme localisation pour étudier les moteurs d'avion. C'était officiel, la thèse venait de démarrer. Les premiers mois sont comparables à la préparation de l'avion, que je considère dans cette analogique comme la manifestation physique du mental du thésard, et de la planification de la trajectoire à suivre. Il faut aussi s'assurer d'un excellent copilote. Autant Amy Johnson réalisa sa traversée en solo, autant une thèse est aussi le résultat de solides collaborations. Je tiens à remercier Professeur Koen Hillewaert, professeur à l'Université de Liège, et mon directeur de thèse, pour m'avoir assurée durant ces longues heures de vol.

Quatre-vingt-neuf ans auparavant, le 5 mai 1930, Amy Johnson s'élançait dans sa folle aventure de traversée en solitaire et fit une première escale à Vienne. La thèse est également pavée d'escales qui prennent la forme de conférences: lieu de rencontre, de discussion et de présentation des résultats. Ce sont aussi et surtout des moments d'échange qui permettent d'alimenter le savoir et les idées. Bien que le plan de vol soit fixé au démarrage de la thèse, ces discussions peuvent faire évoluer la thèse dans une autre direction. Amy rencontra aussi quelques problèmes techniques lors de son périple. En effet, dans sa route vers Bagdad, un violent orage la poussa à atterrir de force dans le désert ce qui endommagea le train d'atterrisage de son biplace. Elle put compter sur les mécaniciens d'une base locale de la Royale Air Force. De mon côté, j'avais aussi une solide équipe de mécaniciens, toujours prêts à m'épauler, à m'aiguiller mais aussi à me questionner pour pousser ma réflexion et améliorer mes résultats. Je tiens à remercier mes collègues, le Docteur Thomas Toulorge et le Docteur Michel Rasquin, tous deux ingénieurs de recherche à Cenaero. Mes deux aiguilleurs du ciel qui n'ont pas failli à leur tâche. Je vous remercie tous les deux pour l'aide technique apportée et aussi pour vos encouragements qui m'ont permis de voir mes résultats sous un autre jour. Au sein de l'équipe, je pouvais aussi compter sur le Docteur Grégory Dergham, chercheur à Safran Tech et mon encadrant côté Safran. Je tiens à te remercier Grégory pour le temps que tu m'as consacré, parfois des après-midi entières, à discuter de réseaux de neurones, de métriques, de termes d'adimensionalisation et j'en passe.

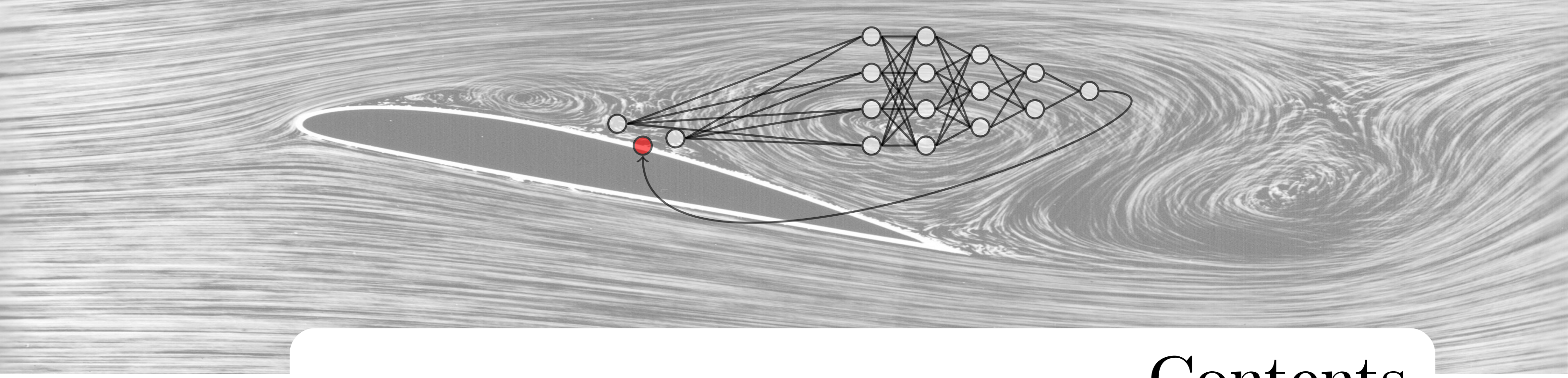
Lors d'une thèse, il arrive que on perde parfois l'objectif long terme de vue en restant trop longtemps dans des développements techniques trop *micro*. Un peu comme Amy, qui le 13 mai 1930 fut aveuglée pendant plus de deux heures par les conditions météorologiques exécrables, la forçant à atterrir à plus de quinze kilomètres de Rangoon. J'ai eu la chance de pouvoir plusieurs fois discuter avec le Docteur Tariq Benarama, ingénieur de recherche à Cenaero, sur la portée long terme de mes développements techniques, effaçant les intempéries et me permettant de repartir de plus belle. Tariq, je te remercie pour ces discussions, toutes aussi passionnantes les unes ques les autres, qui m'ont éclairée.

Après 19 jours de vol, Amy atterrit finalement à Darwin en Australie, heureuse d'avoir réussi cet exploit en solitaire malgré tous les rebondissements qu'elle a vécus avec son biplace *Jason*. Pour une thèse, on ne parle pas en jour, ni en mois mais en année. Quatre ans, ça peut sembler long mais j'étais déterminée à terminer cette thèse car une fois en vol, il n'est pas question de faire demi-tour. Au final, ces quatre ans sont passés assez rapidement car elles furent riches en interaction. Lors de cette aventure, j'ai pu rencontrer le Professeur Gilles Louppe, professeur à l'Université de Liège, et je le remercie pour ses nombreux conseils en Machine Learning et Deep Learning qui me furent extrêmement précieux. Je remercie aussi le Docteur Renaud Mercier, ingénieur de recherche à Safran Tech et la Docteur Sophie Mouriaux, qui travaille désormais à Dassault Systèmes, pour m'avoir suivie pendant mes deux premières années de doctorat, vos conseils et votre réflexion sur mon travail furent d'une grande aide.

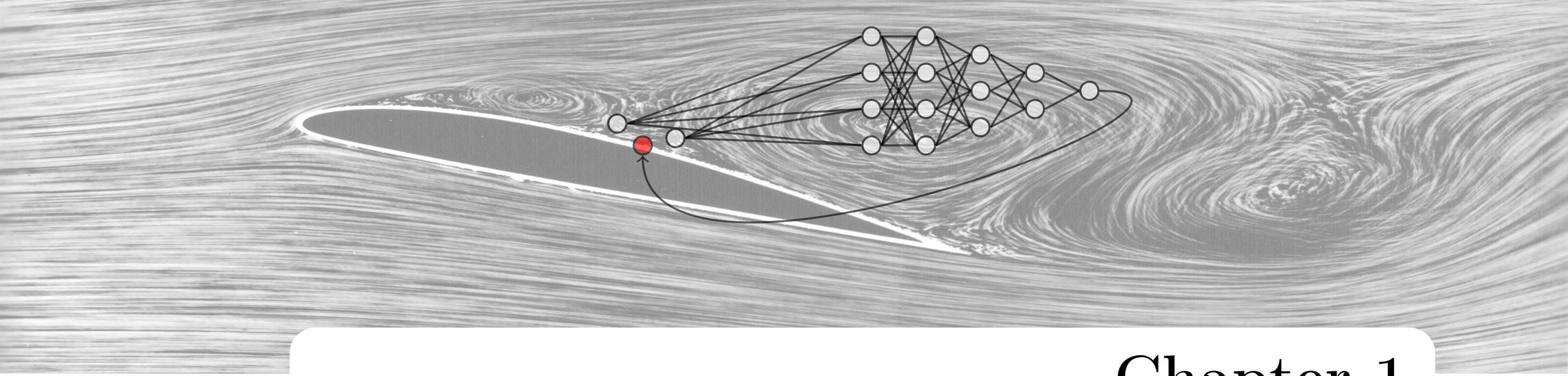
Je voulais aussi remercier Safran Tech pour avoir entièrement financé cette thèse et Cenaero pour m'avoir fait confiance pour prendre en main ce sujet.

"Success is not final, failure is not fatal: It is the courage to continue that counts."

– Winston Churchill



Contents



Chapter 1 Introduction

Have you ever thought about how artificial intelligence (AI) might modify the world? Some of you may think about autonomous vehicles used to reduce the fatality rate of driving and have a better carbon emission due to more efficient driving patterns. Some of you may think about Siri and Alexa, the two famous intelligent assistants, that you may have encountered in your daily life. Some of you may think about robots that have emerged in many fields (see Figure 1.1), including army (e.g., DARPA Robotics Challenge), medicine (e.g., surgical robots, prostheses), industry (e.g., automotive manufacturing), aeronautics (e.g., drones). Some of you may be concerned about the dark side of AI such as the generation of fake images, newspapers, videos of politicians, and cybersecurity issues. As with any new technology, there are potential drawbacks, and our modern society must address them. Since the start of this thesis, these changes are already visible, with their positive and negative aspects. However, this question, with all its concerns, must be examined before working on any Machine Learning (ML) and Deep Learning (DL) techniques. ML and DL are relatively new paradigms used to address a variety of problems across different fields. These techniques can form a new field of research on their own by examining various models at the most fundamental mathematical level, understanding them in terms of learning theory, and developing fundamental approaches for actual learning. This science lays the foundations of knowledge and theory to be extensively used in other fields. Parts of these fields can be grouped under the label of Machine Learning Engineering, in which researchers want to achieve better performance by scaling up the system and making it practically work and other are concerned by applications. Over the past few decades, numerous applications have been developed across various fields. For example, business applications require the analysis of customers' behavior by predicting when they will change their subscription or service provider. Computer vision is well-known for image reconstruction, segmentation, tracking, and labeling. Bioinformatics has reused computer vision tools to classify images in medicine to be applied for diagnostics and pathology. ML techniques are also utilized in physics and chemistry to develop new drugs, generate images from CAD (Computer Aided Design) layers, and simulate galaxies and the universe. These applications would not have been possible without two fundamental aspects. The first aspect is the access to large (labeled) databases made possible by more powerful hardware capable of recording a massive amount of data. The second aspect is the use of Graphical Processing Units (GPUs). GPUs are great at tasks that can be run in parallel and neural networks are specifically designed to run in parallel. The development of GPUs and their intensive use led to the resurgence of DL techniques in 2012.

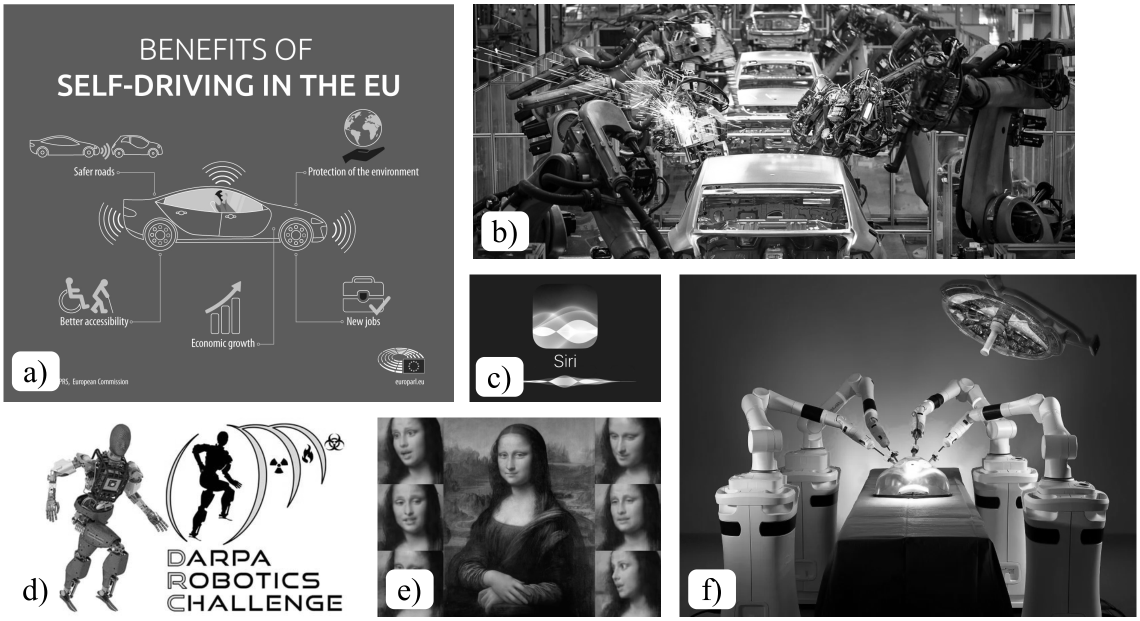
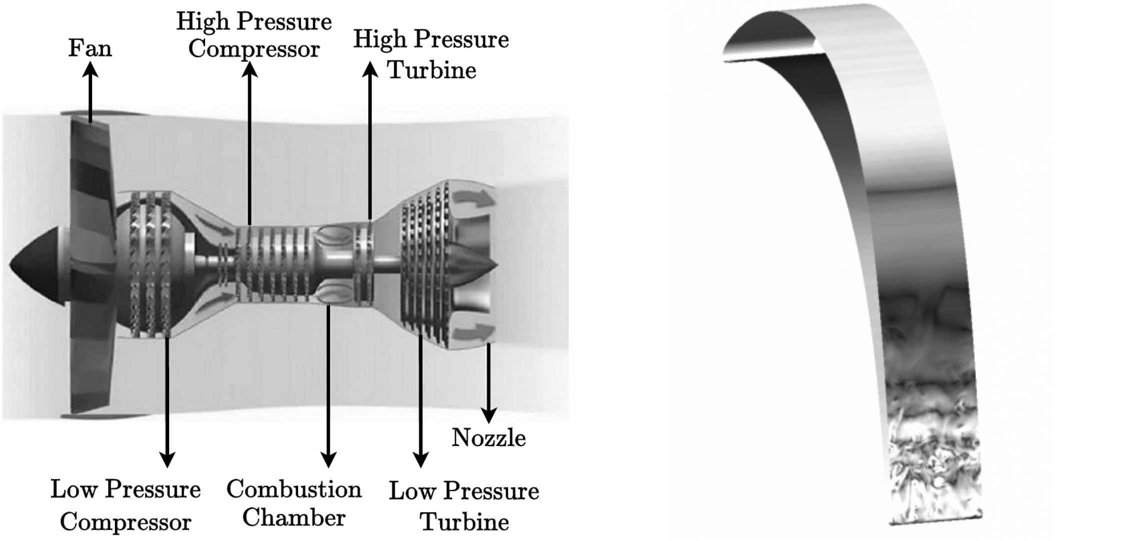


Figure 1.1: (a) Self-driving cars (European Parliament, 2019); (b) Image for the automotive industry by Boris Fojtík, Economic Analyst at Tatra banka; (c) Apple intelligent assistant Siri; (d) Darpa Robotics Challenge; (e) Deepfake extracted from `https://www.eqengineered.com/insights/what-is-a-deepfake`; (f) Surgical robot system (Kent, 2020).

AI is a powerful tool that can have significant effects on humans, cultures, societies, and the environment. While it has the potential to be an ally, it can also be a dangerous enemy if not handled carefully. Our society must therefore use AI carefully and follow political, economic, ethical, and legal guidelines to prevent misuse. However, there are specific areas where the use of AI can be well controlled with minimal risk to our society, where ML and DL tools can be successfully employed to reduce the complexity of a computational chain, for instance. In the aerospace industry, engineers are always in search of the best performances for the new engines that will equip our future aircraft. Designers work on multiple aspects, such as the request to reduce fuel consumption and noise, the components' strength and weight, the thrust generated by the engine, and many other parameters for a safe and comfortable flight. In this context, ML can be used to improve modeling.

As we delve into the aerospace industry, it is interesting to understand how aircraft actually fly. Newton's third law of action and reaction is responsible for propelling an aircraft forward. To achieve this motion, the jet engine (see Figure 1.2a) must produce high thrust at its exit. Thrust is typically proportional to the product of the mass flow and the velocity difference between the inlet and outlet. For a turbofan, thrust is usually produced by a high mass flow, whereas for a military engine it is produced by a high velocity. In a turbofan, the exhaust is obtained as a combination of two flows. Some of the incoming air passes through the core of the engine, and the rest of the incoming air passes through the fan, bypassing the interior assemblies. This part of the air generates the majority of the thrust (up to 80%) in a separate nozzle. The ratio of these two flows is called the bypass ratio. For example, the A380, the world's largest commercial aircraft, has a bypass ratio of 8. The main objective of a turbofan engine is to rotate the fan that drives the bypass air. To achieve this, a series of components are needed. The key is to compress it to a high pressure (e.g., 40 bar) and to heat up the incoming air to a high temperature (e.g., 1850 K). For this purpose, a series of compressor stages is used to bring the air to a higher pressure. The rotating blades of the compressor add energy to the fluid, increasing its (temperature and) pressure to levels adequate for the targeted cycle efficiency. The majority of temperature increases occur in the combustion chamber, which is situated immediately after the compressor. The turbine, located immediately after the combustion chamber, provides energy for rotation and is connected to the compressor by a common shaft. The high-energy fluid leaving the combustion chamber drives the turbine blades. The turbine blade has a specific airfoil shape (see Figure 1.2b) that creates lift and causes it to rotate. By transferring its energy to the turbine, the fluid pressure drops. This release of energy drives the compressor and the fan.



(a) Schematics of a turbofan engine, image extracted from the website `https://mechanicalboost.com/turbofan/`.

(b) Low Pressure Turbine blade (T106C), visualization of the skin friction in gray scales, Cenaero credits.

Figure 1.2: Illustration of a turbofan engine, which produces a high-speed jet at its exit to propel the aircraft forward; the engine consists of a series of compressor stages (low and high) followed by the combustion chamber; the very hot air produced by combustion is then injected into a series of turbine stages (high and low) to recover energy from it thanks to the specific shape of the blades, as shown on the right.

This brief explanation indicates that there are many components and sub-components that affect the engine behavior and thus its thrust, making the engine conception a complex process. Among all these components, the compressor and the turbine are crucial in the engine design as they bring the air to specific states to ensure proper engine operation. During the conception, engineers take specific care when designing these two components by optimizing the number of stages, blades per row, and blade shape. For example, reducing the number of blades per stage is a way of reducing the engine weight and, therefore, its fuel consumption, which is of particular importance given the new environmental standards in place. By decreasing the number of blades on each row while maintaining the same amount of work per stage, the loading per blade is increased. This is the well-known concept of high lift design (Gier *et al.*, 2010). This high lift results in an intense adverse pressure gradient on the suction side of the blades. At cruising altitude, the Reynolds number is relatively low, which promotes the formation of a laminar boundary layer on the suction side. Such a boundary layer can not withstand the large pressure gradient and may separate. Downstream of the separation point, the free shear layer transitions (Horton, 1968) due to instabilities, wake interaction, and free-stream turbulence. Then, in the most favorable case, the turbulent boundary layer may reattach. In the case of very low Reynolds numbers, the boundary layer may never reattach, leading to a massive laminar separation bubble that causes high losses and decreases the engine efficiency. Ultimately, understanding and predicting the flow behavior on blades is crucial for designing an engine. The aforementioned explanation applies to Low Pressure Turbine (LPT), but a similar conclusion can be drawn for Low Pressure Compressor (LPC).

Computational Fluid Dynamics (CFD) can be considered as a branch of fluid mechanics that uses numerical algorithms to solve and analyze problems involving fluid flows. CFD involves various phases such as geometry preparation, meshing, solution generation, and post-processing. It also requires the choice of a discretization method (e.g., Finite Element method, Finite Volume method, Finite Difference method, Spectral Element method, Lattice Boltzmann method, to cite but a few). Although CFD is an excellent tool for predicting the flow behavior for realistic geometry, LPT aerodynamics remains a challenging configuration for CFD tools because it requires the prediction of a separation-induced transition, a phenomenon that is not yet fully understood. Moreover, in the particular case of turbulent flows, special attention to near-wall grid resolution and turbulence model is required. Indeed, a turbulent flow is unsteady, chaotic, and highly three-dimensional, composed of a wide range of scales. These scales are illustrated (see Figure 1.3) with the example of a car driving on a motorway. As the car moves, it creates a wake. This wake is a turbulent flow composed of vortices whose size is comparable to the car width L up to vortices whose size is equal to the Kolmogorov scale η (Pope, 2000). Each scale has its own amount of energy, with the largest eddies containing the highest energy. These eddies break down into smaller ones of lower energy and themselves transfer their energy to even smaller eddies through an inviscid process. At the smallest scales, the energy is dissipated through viscous action. This process of transferring energy from one scale to another is known as the *energy cascade*, first described in 1922 by Richardson in his book (Richardson, 2007). Figure 1.3 illustrates this energy cascade composed of the energy-contained, inertial, and dissipation range.

The energy is expressed as a function of the wave number κ. Therefore, the eddies of large scale L have a small wave number, while the eddies of small scale η have a high wave number. The accuracy of the discretization method will depend on its capabilities to capture the entire energy spectrum and thus to resolve all the turbulent scales or a part of it, up to a cutting wave number κ_c.

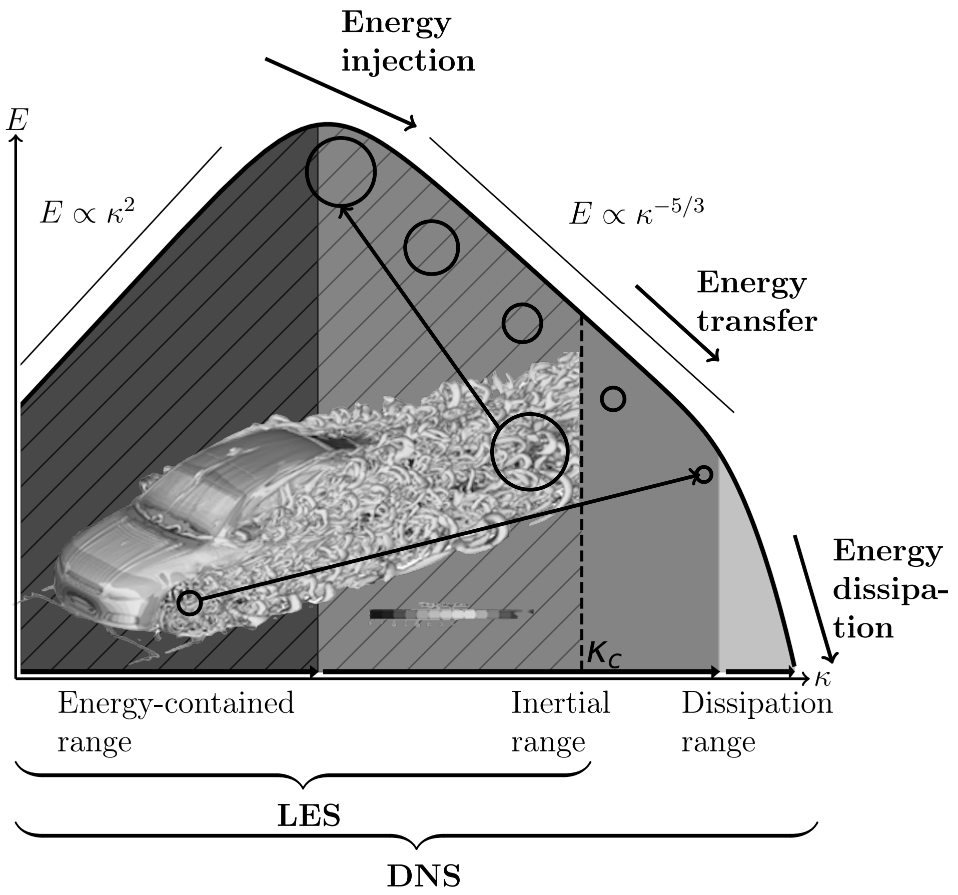


Figure 1.3: Diagram of the energy spectrum combined with an image of the instantaneous turbulent structures around a car (Doutrelant and Wong, 2021).

In wall-bounded flows, a high-shear region develops in the near-wall region bounded by the wall on one side and the top of the boundary layer on the other. A turbulent boundary layer can be divided into four parts: the viscous sublayer, the buffer layer, the log-law region, and the wake region (see Figure 1.4). In the **viscous sublayer**, the viscous shear stress dominates the turbulent shear stress. Indeed, close to the wall, in this thin layer, the turbulent fluctuations are damped. On the other hand, in the **log-law layer**, the turbulent shear stress predominates over the viscous shear stress, which can then be neglected. In between, in the **buffer layer**, the viscous shear stress and the turbulent shear stress are of equal magnitude, and neither can be neglected. Most of the turbulence is generated in this region. The **wake region** is also called the velocity-defect layer (Coles's Law of the Wall). In this latter region, the velocity profile deviates from the logarithmic law. This deviation is even more pronounced in non-equilibrium boundary layers with (adverse) pressure gradients. The end of the log-law region, which also corresponds to the beginning of the wake region in wall units y^+ (i.e., the inner scaling of the boundary layer, refer to Eq. 3.2), depends on the friction Reynolds number. The inner layer, which regroups the viscous sublayer, the buffer layer, and the log-law region, ends approximately at $y/\delta \simeq 0.15$, where δ is defined as the boundary layer thickness. The theory behind boundary layers is further explained in Chapter 3. To properly resolve the boundary layer, both the viscous and turbulent shear stresses need to be captured by the numerical scheme.

On the one hand, Direct Numerical Simulation (DNS) is the best solution for describing accurately laminar, transition, and turbulent flows because it resolves the entire spectrum. However, this method remains computationally intractable at the high Reynolds numbers (Choi and Moin, 2012) typical for most turbomachinery passages. On the other hand, one may find turbulence modeling approaches. The two major approaches are Reynolds-Averaged Navier-Stokes (RANS) and Large-Eddy Simulation (LES). RANS and Unsteady RANS (URANS) methods are mainly used for industrial applications due to their computational efficiency. RANS aims to solve directly for the mean quantities, which significantly reduces the computational cost. Nonetheless, in off-design conditions, the flow inside turbomachinery components is dominated by complex features, which are outside the range of reliability of RANS turbulence models. LES reduces the number of modeling assumptions and enables high-resolution simulations of transitional and turbulent flows by resolving accurately a part of the inertial range of the energy spectrum (see Figure 1.3), up to a cutting wave number. LES resolves the large scales and models the interaction of the small unresolved scales on the large resolved ones using sub-grid scale (SGS) models. These models can be explicit (e.g., Sagaut (2006)) or implicit (e.g., Grinstein *et al.* (2007)). The latter is called implicit LES (iLES). When employing implicit

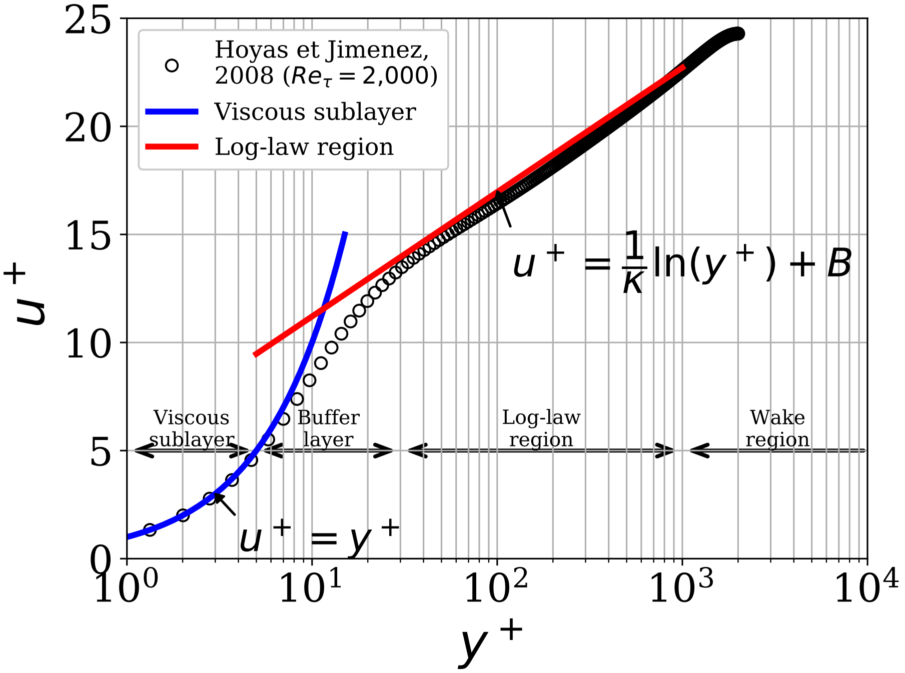


Figure 1.4: Universal Law of the Wall.

filtering, the discretization errors are adjusted to substitute the subgrid scale effects (Beck, 2015). Unfortunately, wall-resolved LES (wrLES) methods, where the wall-nearest grid point is within the viscous sublayer, remain prohibitively expensive at high Reynolds numbers (Choi and Moin, 2012). As approaching the wall, the energy containing eddies become smaller and a fine grid resolution is required in the near-wall region to properly resolve these structures. Even at moderate Reynolds numbers, the resolution of this near-wall region requires significant computational resources as illustrated on Figure 1.5. The cost of resolving a high Reynolds number turbulent boundary layer flow using DNS, wrLES, and wmLES can be evaluated with the formula of Choi and Moin (2012):

$$N_{\text{DNS}} \sim Re_{L_x}^{37/14}, \quad N_{\text{wrLES}} \sim Re_{L_x}^{13/7}, \text{ and } \quad N_{\text{wmLES}} \sim Re_{L_x}, \tag{1.1}$$

where Re_{L_x} is defined as $Re_{L_x} = u_e L_x/\nu$, with u_e the external velocity, L_x the length of the flat plate and ν the kinematic viscosity.

It motivates the development of wall-modeled LES (wmLES), where the wall-nearest region is coarsened to reduce the computational cost, which moves the first grid point away from the viscous sublayer. The effect of the unresolved near-wall physics is then usually represented by a shear stress boundary condition computed from a wall model that depends on flow quantities inside the computational domain. The advantage of wall-modeled LES for practical flows of aeronautical interest is highlighted by the formula 1.1, and even more at high Reynolds numbers. This observation is reinforced in Figure 1.5, where an estimated computational cost (i.e., the cost per one-time unit required on an AMD Opteron cluster) for wmLES and wrLES of a turbulent flow over flat plates is plotted as a function of the Reynolds number. As indicated, wrLES is computationally challenging and requires massively parallel computing resources. With the use of wmLES, the computational cost is significantly reduced and passes below 5 days according to Alam and Fitzpatrick (2017). As a consequence, new approaches have been proposed over the last decades to develop wall models and improve their reliability for more complex flows. Furthermore, their application to industrially relevant flows is still to date a topic of research.

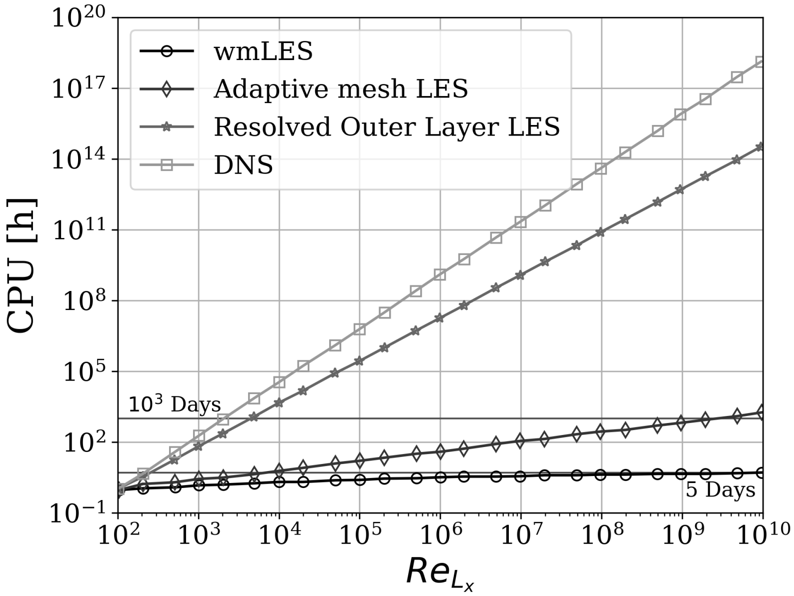


Figure 1.5: Example of comparison of computational cost of LES (Alam and Fitzpatrick, 2017).

The present work builds upon the thesis of Frère (2018), which developed cost-effective wall-modeled LES techniques for high Reynolds number turbulent flows in a high-order flow solver. Her study identified limitations in the wall shear stress (WSS) model based on Reichardt's velocity profile, particularly its inadequacy in accurately capturing flow separations. As such, the present thesis aims to improve wall models for predicting flow separations and reattachments that are encountered in industrially relevant flows such as turbomachinery. Among the different possibilities to enhance WSS models, DL techniques have been selected. Deep Neural Networks (DNNs) are employed when there exists a structural relation, fully or partially unknown, between the inputs and outputs. The current research aims to exploit the approximation capabilities of deep neural networks (DNNs), first described by (Hornik, 1991), to establish a more general model for the complex relationship between instantaneous flow fields, geometric parameters as input, and wall shear stress as output using DNS or wrLES databases.

1.1 Objectives and structure of the thesis

The main objective of this thesis is to enhance wall models using Deep Learning techniques to address flow separation and reattachment. The development of data-driven wall models is a relatively recent research topic that raises many challenges. For instance, there are still many open questions regarding interpretability, robustness, generalizability, and convergence. Moreover, deciding which neural networks to utilize, how to train them, and what type of data to consider are significant challenges on their own. To cope with these challenges, this thesis is divided into three *main* parts.

The first part focuses on the generation, analysis, and preprocessing of wrLES databases. The simulations are carried out using the in-house flow solver Argo-DG, developed at Cenaero. This code was initially developed during the thesis of Hillewaert (2013) and it has been further validated during the thesis of Carton de Wiart (2014) on various academic benchmarks such as the DNS of the Taylor-Green vortex (Carton de Wiart *et al.*, 2013a), and the LES of homogeneous isotropic turbulence and channel flow (Carton de Wiart *et al.*, 2013b). The code has been shown to produce results with similar accuracy as those obtained with academic ones. Moreover, it has been used to run DNS and LES of industrial configurations featuring transitional flow (Carton de Wiart and Hillewaert, 2012; Hillewaert *et al.*, 2014). Since a WSS model can be seen as a non-linear function establishing a relation between flow quantities extracted in the computational domain and the wall shear stress, space-time correlations between these quantities are evaluated for the various databases generated with Argo-DG. These correlations provide physical insights into the possible relationships between the variables and act as feature selections for the neural network.

The second part is dedicated to the training and the implementation of the novel data-driven WSS model in a high-order flow solver. Based on the analysis of the weaknesses of the existing WSS models, the thesis proposes a novel approach: the use of neural networks that can predict statistical distributions rather than point-wise estimates. These networks are not trained with the standard loss functions but instead with the negative log-likelihood, which accounts for the first two statistical moments when the underlying distributions are Gaussians. However, this type of network raises questions about the reconstruction of the WSS structures.

The third part concentrates on validating the proposed model on different test cases, including flows with turbulent boundary layers and those featuring separation. This part also evaluates the robustness, interpretability, and generalizability of the novel WSS model. The robustness is assessed by evaluating the stability of the simulations and the quality of the results. The interpretability is evaluated using the SHAP values (**SH**apley **A**dditive ex**P**lanations), and the generalizability is evaluated *a priori* using synthetic data.

Each part of this project has a specific objective that contributes to the overall goal. The first objective is to define the test cases, which must be computationally feasible and include the necessary features for developing innovative data-driven WSS models. It is important to include at least one test case with separation to gain insight into this phenomenon. The long-term goal is to create a model that can generalize on any flow configuration, but there is debate within the scientific community about whether a universal model is achievable, or even necessary. It may be better to create multiple models that can switch according the different phenomena during the simulations. The second objective of the first part is to select relevant features for the training of the data-driven WSS model. The second part's objective is to identify the most appropriate neural networks for the problem at hand. Deep Learning offers various neural network architectures that are specific to certain tasks and can be combined to create a new network with multiple embedded invariants (i.e., *the internal properties that are encoded in the network architecture*).

The implementation of wall models in Argo-DG has been successful during the thesis of Frère (2018), but the challenge remains to implement data-driven wall models in a lightweight version that will not slow down simulations. The use of data-driven models also comes with the cost of generating labeled databases and the challenge of properly training the neural network. The third part's objective is to validate the novel model on test cases with separation, evaluate the improvement with respect to wall models already implemented in Argo-DG, and the existing data-driven WSS models in the literature. The rest of the thesis is organized as follows.

Chapter 2: Background in Deep Learning
Because the present work aims to use Deep Learning techniques, a rapid review of well-known neural network is performed in Chapter 2.

Chapter 3: State of the art in wall modelling
Chapter 3 presents a review of the data-driven wall models that have been developed over the last decade. We will also briefly review more standard wall models that are widely used in the wmLES community. A list of constraints and limitations on the implementation of the novel wall models is specified, taking into account the existing models and what can be achieved in the high-order flow solver.

Chapter 4: Analysis of space-time correlations in representative turbulent flow configurations
Chapter 4 addresses the question of the databases. Neural networks can deal with large databases to be able to understand the underlying physics. In the framework of wall models, the goal is to model the complex relation between instantaneous volume quantities (e.g., velocity, pressure gradient), geometrical parameters (e.g., curvature, relative positions), and the wall-parallel components of the wall shear stress. Moreover, this relation is sought for separated flows. hence, a test case containing a separation of a turbulent boundary layer is essential. Because the training will be performed on high-quality data extracted from wrLES, the two-dimensional periodic hill at a bulk Reynolds number Re_b of 10,595 is selected. This test case is sufficiently challenging for a wall model and can be run on modern clusters. However, relying on a single test case is insufficient. When training a neural network on a single configuration, it will overfit (i.e., *to be too specific to the studied configuration*) on that configuration and will not be able to generalize to other geometries. Therefore, two turbulent channel flows are added to the databases as

well as synthetic data extracted from the law-of-the-wall (LOTW). This chapter also focuses on analyzing these databases through space-time correlations (i.e., Pearson and distance correlations) that act as feature selection for the neural network. This work, presented at the conference ETMM13, has been selected for a special edition in the journal "Flow, Turbulence and Combustion" (Boxho *et al.*, 2022).

Chapter 5: Development of a new data-driven wall shear stress model
Chapter 5 presents the different neural network (NN) architectures used to develop novel wall shear stress models. As mentioned earlier, there is a wide variety of NN architectures that have their own embedded invariants. It is crucial to identify the specific invariant that should be embedded in the architecture to enable the network to extract the underlying physics. This chapter focuses on what a neural network is, how it is constructed, and how it is trained. Furthermore, the network must conform to the implementation constraints outlined in Chapter 3. While it may seem that everything is possible in the *a priori* implementation, practical limitations may arise in the production environment. Therefore, it is important to evaluate these constraints and integrate them into the design of the neural network. Additionally, proper preprocessing of the database (e.g., cleaning, splitting, and non-dimensionalizing) is essential for successful neural network training. Therefore, the importance of nondimensionalization is highlighted for generalization purposes.

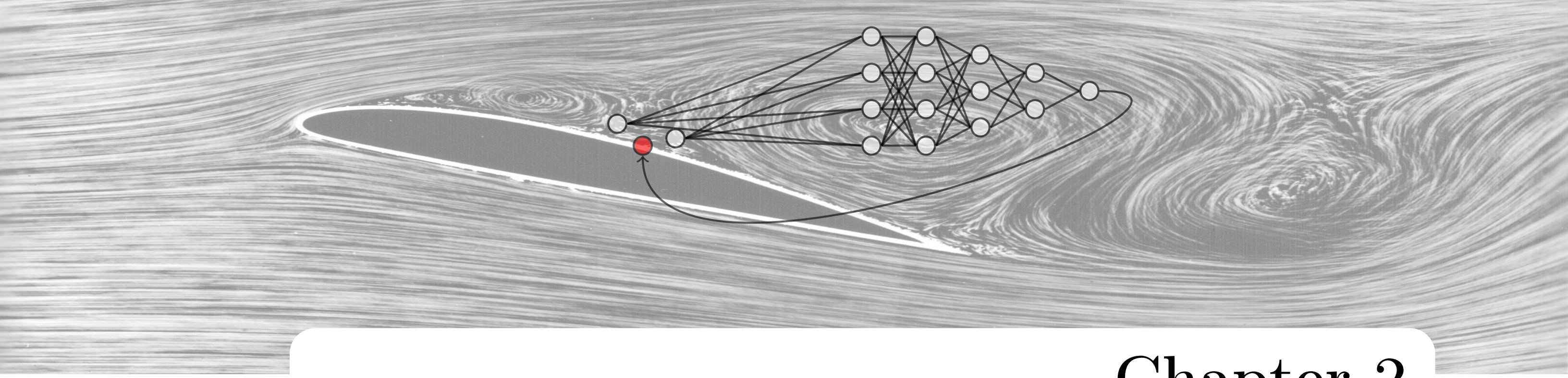
Chapter 6: Pathway to the successful network architectures for wall modeling
Chapter 6 reviews the history of neural network training, which has led to the establishment of the most successful networks for the development of wall shear stress models. It discusses the first non-dimensionalization tested at the beginning of the thesis, which was discarded to adopt the one described in Chapter 5 because it was too sensitive to the polynomial order. Although the space-time correlations, performed in Chapter 4, have indicated the *optimal* stencil size, several stencils have been tested to validate the conclusion drawn from the feature selection. The chapter closes with a discussion on the overfitting and underfitting of the trained network and how to improve upon the predictions.

Chapter 7: Fully developed turbulent flow in channel configuration
Chapter 7 presents the training of a neural network on the channel flow configurations only. This chapter sets the basis of the methodology for developing data-driven

WSS models. The model is first tested *a priori* on an unseen channel flow configuration and the synthetic database. Then, the model is evaluated *a posteriori* on a turbulent channel flow at two friction Reynolds numbers. In the Deep Learning terminology, the *a posteriori* test evaluates the model performance in the production environment. Therefore, in this context, it corresponds to a wmLES using the trained data-driven WSS model.

Chapter 8: Separated flow in the two-dimensional periodic hill flow configuration

Chapter 8 focuses on the main objective of this thesis: the application of the data-driven WSS model to a separated flow. As in Chapter 7, the model is initially trained on databases featuring separation, coupled with a turbulent channel database, to eventually aim at a more universal model. The trained model is tested *a priori* on the two-dimensional periodic hill. Finally, a wmLES is performed on the same test case using the data-driven WSS model to evaluate the model's ability to capture the separation and reattachment phenomenon in the production environment. It should be noted that the production environment can not be incorporated during the training process. A model that performs well *a priori* may struggle to accurately reproduce the underlying physics of the wall when used in practice due to grid resolution, numerical transient, and feedback loop between the predicted wall shear stress and the volume data, to name just a few. This work, presented as a keynote lecture at the ECCOMAS 2022 conference, has been submitted to the journal "Physics of Fluids" and is currently under review.



Chapter 2
Background in Deep Learning

This chapter introduces the main concepts of Machine Learning (ML) and Deep Learning (DL) techniques (algorithms), from the presentation of the simplest neural network architecture to its *training* with a *loss function* whose gradient is computed using *backpropagation*. To an uninitiated reader, these words may seem unfamiliar, but by the end of this chapter, the DL nomenclature should no longer be a secret.

Section 2.1 presents the DL and ML techniques under a probabilistic view. Section 2.2 focuses on supervised learning and discusses the generation of a database as samples from a joint probability and how a model can infer this data by approximating a conditional probability. This discussion leads to the definition of the loss function for regression problems. Section 2.2.1 introduces the various concepts that build the training process on the simplest neural network architecture: the multilayer perceptron (MLP). Section 2.2.2 introduces the convolutional neural network (CNN) and how they embed the knowledge that nearby pixels are typically related to each other, compared to MLP. The concept of feature selection is presented in Section 2.3. Section 2.4 focuses on the interpretability of a neural network and introduces the SHapley Additive exPlanations. The chapter ends with a few words about `PyTorch`, the open library we used to train our data-driven wall models.

2.1 Probability

DL and ML are focused on making predictions and are fundamentally probabilistic. For instance, researchers aim to predict the *probability* of a patient having a heart attack in the next year based on his clinical history. In anomaly detection, engineers want to assess how *likely* a set of measurements from a turbojet engine would be if it were operating normally. In the field of economics, economists aim to estimate the *likelihood* that a specific customer will purchase a particular book. Ultimately, it is necessary to use the language of probability. To illustrate the necessity to use probability, we consider two cases: (i) distinguishing images of horses and cows and (ii) weather forecasting.

Horses and cows Distinguishing between images of horses[1] and cows[2] can be a challenging task, especially when the resolution of the images is low (see Figure 2.1). At higher resolutions, a human can easily distinguish between the two animals. However, at lower resolutions, it can be difficult or even impossible to identify which image depicts the horse and which depicts the cow. If one is absolutely certain that the picture depicts a cow, then the probability that the corresponding label y is "cow" is denoted as

$$\Rightarrow P(y = \text{"cow"}) = 1\,.$$

If one has no evidence about the corresponding label of the image, then the probability is written as

$$\Rightarrow P(y = \text{"cow"}) = P(y = \text{"horse"}) = 0.5$$

(i.e., the two probabilities are equally likely). If one is reasonably confident but not quite sure that the image is a horse, then the probability satisfies,

$$\Rightarrow 0.5 < P(y = \text{"horse"}) < 1\,.$$

Weather This second case concerns the prediction of the probability that it will rain tomorrow in Belgium. If you are a native Belgian, you know this probability is high in winter.

[1] Image extracted from `https://ihearthorses.com/chestnut-horse/`

[2] Image extracted from `https://www.pbs.org/wnet/story-jews/video/episode-1/attachment/brown-cow/`

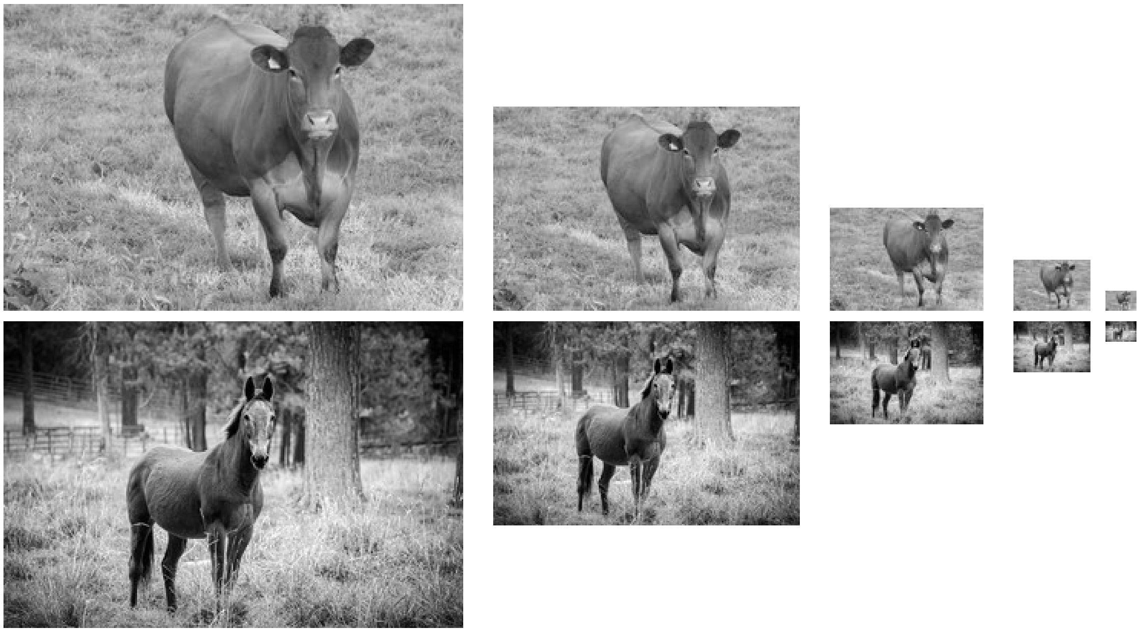


Figure 2.1: Grayscale images of horse and cow at different resolutions (from left to right: 300 × 300, 200 × 200, 100 × 100, 50 × 50, and 20 × 20 pixels).

These two examples highlight that probability provides a formal way of expressing the degree of certainty (or uncertainty). *Probability* is a function that maps a set to a real value. Formally, the probability of an event $\mathcal{A}$ in a given sample space $\mathcal{S}$ is $P(\mathcal{A})$. An *event* is a set of outcomes from a given sample space. In the simple example of throwing a die, the sample space is discrete and is $\mathcal{S} = \{1, 2, 3, 4, 5, 6\}$. If the die is not tainted, the probability of rolling a 2 is 1/6. Formally the event is denoted as $\mathcal{A} = 2$ with probability $P(2) = 1/6$.

$P(\mathcal{A})$ satisfies the following properties,

Axioms of Probability Theory

- For any event $\mathcal{A}$, its probability is never negative, i.e., $P(\mathcal{A}) \geq 0$;
- Probability of the entire sample space is 1, i.e., $P(\mathcal{S}) = 1$;
- For any countable sequence of events $\mathcal{A}_1, \mathcal{A}_2, \ldots$ that are *mutually ex-*

clusive ($\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ for all $i \neq j$), the probability that any happens is equal to the sum of their individual probabilities,

$$P(\bigcup_{i=1}^{\infty} \mathcal{A}_i) = \sum_{i=1}^{\infty} P(\mathcal{A}_i)\,.$$

The concepts of event and sample space have been defined, but the notion of a *random variable* has not yet been introduced. A random variable, denoted by X, is a measurable function from a sample space to a measurable space. It can be any quantity and is not deterministic. The distribution over the random variable X is denoted by $P(X)$, while $P(X = a)$ represents the probability of X taking the value a.

In fluid dynamics, a turbulent flow has a *statistical description*. For instance, the velocity field $\boldsymbol{u}(\mathbf{x}, t)$ (where $\boldsymbol{x}$ and t denote space and time, respectively), in a turbulent flow, is *random*. If a fluid flow experiment can be repeated many times under a given set of conditions, we can consider the event $\mathcal{A}$, such as $\mathcal{A} = \{u_i < 10\,\mathrm{m/s}\}$, where u_i is the i^{th} component of the velocity extracted at a given time and location. The event $\mathcal{A}$ may or may not occur, and $\mathcal{A}$ is therefore random. In this example, u_i is a random variable, and it only means that it does not have a unique value (i.e., its value is not the same value each time the experiment is repeated under the same set of conditions).

There is a problem of consistency between the random nature of turbulent flows and the deterministic nature of classical mechanics embodied in the Navier-Stokes equation. Why are the solutions of deterministic equations of motion random? This question can be answered by two observations. Firstly, there are unavoidable perturbations in the initial conditions, boundary conditions, and material properties. Secondly, turbulent flows are highly sensitive to such perturbations (Gleick, 1988) (especially at high Reynolds numbers). The Lorenz equations provide one of the simplest examples of a chaotic system.

Similarly to images containing millions of pixels, where each pixel can be considered a random variable, the input (e.g., the three components of the velocity field) of a data-driven wall shear stress model is also a random variable or, more precisely, a random vector. When dealing with multiple random variables, there are several quantities of interest.

- **Joint probability**, noted as $P(A = a, B = b)$, answers the question: *what is the probability that $A = a$ and $B = b$ simultaneously?*
- **Conditional probability**, noted $P(A = a|B = b)$ is the probability of $A = a$, konwing that $B = b$.
- **Bayes' theorem** is one of the most useful equations in statistics:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}\,. \tag{2.1}$$

- **Marginalization** is the operation of determining $P(B)$, the marginal probability or the marginal distribution (i.e., the result of the marginalization) from $P(A, B)$:

$$P(B) = \sum_{A} P(A, B)\,.$$

- **Independence** is an interesting property to examine. *If the random variable A is independent of the random variable B, then the occurrence of one event of A does not reveal any information about the occurrence of an event of B.* From this observation, we get that $P(B|A) = P(B)$. Applying Bayes' theorem, we can directly deduce that $P(A|B) = P(A)$.

The probability distribution of the random variable X can be further characterized by measuring its *expectation*

$$\mathbb{E}[X] = \sum_{x} xP(X = x), \tag{2.2}$$

and its *variance*

$$\mathbb{V}\text{ar}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]\,. \tag{2.3}$$

2.2 Supervised learning

First of all, there are three classes of learning: *supervised*, *unsupervised* and *semi-supervised*. In supervised learning, the data is explicitly labeled. In unsupervised learning, the data is not labeled. The DL models will detect the underlying patterns that distinguish one set of data from another (e.g., clustering problems). Finally, semi-supervised learning is an intermediate class. Labeling data can be costly. In semi-supervised learning, a small part of the database is explicitly labeled. A DL model is trained to predict these known labels. The trained model is then used to label the rest of the database. Only the labels with sufficiently high confidence are kept to retrain the model.

In the previous section, the basics of probability theory were reviewed. Based on this, the notion of supervised learning can be clearly posed. Consider an unknown joint probability distribution $p_{X,Y}$, where X represents the input (i.e., usually a p-dimensional vector of features or descriptors) and Y, the output (i.e., a real value or a category). The training data is drawn from this distribution as,

$$(\boldsymbol{x}_i, y_i) \sim p_{X,Y} \tag{2.4}$$

where $\boldsymbol{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \ldots, N$. The training data is generated identically and independently distributed (i.i.d.) with a finite size N. It is important to note that in practice, there is no prior information available about this joint probability.

Under supervised learning, there are usually two types of inference problems.

- **Classification:** Given $(\boldsymbol{x}_i, y_i) \in (\mathcal{X} \times \mathcal{Y}) = \mathbb{R}^p \times \triangle^C$, for $i = 1, \ldots, N$, we want to estimate for any new $\mathbf{x}$,

$$\arg\max_y p(Y = y | X = \boldsymbol{x}) \,. \tag{2.5}$$

- **Regression:** Given $(\boldsymbol{x}_i, y_i) \in (\mathcal{X} \times \mathcal{Y}) = \mathbb{R}^p \times \mathbb{R}$, for $i = 1, \ldots, N$, we want to estimate for any new $\mathbf{x}$,

$$\mathbb{E}[Y | X = \boldsymbol{x}] \,. \tag{2.6}$$

In other words, inference involves estimating, for any new $(\boldsymbol{x}, y)$, the conditional probability $p(Y = y | X = \boldsymbol{x})$.

Consider a function $f : \mathcal{X} \to \mathcal{Y}$ produced by a learning algorithm (e.g., polynomial fitting, random forest, or neural networks). The prediction of this function can be evaluated through a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ so that $\mathcal{L}\left(y, f(\boldsymbol{x})\right) \geq 0$ measures how close the prediction $f(\boldsymbol{x})$ is to y.

By modeling the conditional probability with a Gaussian distribution, then we get,

$$p(y|\boldsymbol{x}, f) = \mathcal{N}(y|f(\boldsymbol{x}), \sigma^2)\,, \tag{2.7}$$

where $f(\boldsymbol{x})$ describes the expected value function. To fit such a model, we maximize the likelihood of the data over f to get,

$$\begin{aligned}
\arg\max_f p(y|\boldsymbol{x}, f) &= \arg\max_f \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, f) \\
&= \arg\max_f \prod_{i=1}^{N} \mathcal{N}(y_i|f(\boldsymbol{x}_i), \sigma^2) \\
&= \arg\max_f \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - f(\boldsymbol{x}_i))^2}{2\sigma^2}\right)\,.
\end{aligned}$$

As maximizing a function is similar to maximizing its logarithm, we obtain,

$$\begin{aligned}
&\arg\max_f \left[\sum_{i=1}^{N} -\frac{(y_i - f(\boldsymbol{x}_i))^2}{2\sigma^2} - \log\left(\sqrt{2\pi}\sigma\right)\right] \\
&= \arg\min_f \underbrace{\frac{1}{2\sigma^2}\sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i))^2 + \log\left(\sqrt{2\pi}\sigma\right)}_{\geq \sum_{i=1}^{N}(y_i - f(\boldsymbol{x}_i))^2}
\end{aligned}$$

To obtain a loss that is independent of the size of the training data, the last expression is normalized by N to get,

$$\mathcal{L}\left(y, f(\boldsymbol{x})\right) = \frac{1}{N}\sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i))^2\ . \tag{2.8}$$

Equation 2.8 is the Mean Square Error (MSE) loss employed for regression problems. For classification problems, the cross-entropy loss is obtained similarly.

In fluid dynamics, the joint probability distribution can refer to a fluid-flow experiment (e.g., experimental or numerical) conducted under specific operating conditions, such as initial and boundary conditions, as well as the Reynolds number. The training data consists of realizations of this experiment. The present work belongs to the supervised learning class, where data is explicitly labeled by performing Direct Numerical Simulations of multiple test cases to obtain pairs of inputs (i.e., velocity, pressure gradients, etc.) and outputs (i.e., the wall shear stress). Chapter 4 covers the generation of the database in detail. A WSS model has to predict the wall shear stress as a continuum value. Therefore, a WSS model assisted by neural networks falls into the category of regression problems. Neural networks, known as universal approximators, are selected to solve this regression problem using a very large labeled dataset. These points are treated in chapters 3 and 5.

2.2.1 Multi-layer Perceptron

A neural network is a tool for developing new engineering models. To construct a mathematical tool that can infer predictions in a production environment, one needs a database, a model composed of parameters to be fitted, and an optimization procedure. The entire training process (i.e., including the definition of the network architecture, handling data, specifying the loss function, and training the model) is covered by considering the **multi-layer perceptron** (MLP), which is the simplest deep neural network.

For the remainder of this section, the focus is set on regression problems as the main goal is to develop a wall model that will capture the relationship between variables extracted in the computational domain and the wall shear stress. The primary objective is not only to characterize the relationship between the inputs and outputs but also to make predictions.

2.2.1.1 Architecture

An MLP (Fig.2.2) consists of several layers: input, hidden, and output layers. Each layer has a number of neurons. The number of input neurons depends on the input size, while the number of output neurons depends on the output size. The number of neurons in a hidden layer is a hyperparameter set by the user or optimized by Bayesian optimization.

As illustrated in Figure 2.2, each layer is connected to the next layer, and this connection has a certain weight. The state transferred from one layer to the next is

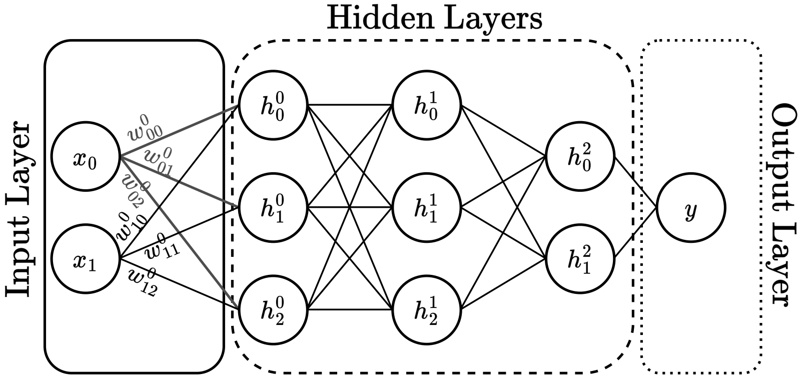


Figure 2.2: Sketch of an MLP with three hidden layers.

evaluated by a matrix-vector product:

$$\begin{bmatrix} h_0^{(0)} \\ h_1^{(0)} \\ h_2^{(0)} \end{bmatrix} = \begin{bmatrix} w_{00}^{(0)} & w_{10}^{(0)} \\ w_{01}^{(0)} & w_{11}^{(0)} \\ w_{02}^{(0)} & w_{12}^{(0)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \rightarrow \boldsymbol{h}^{(0)} = \boldsymbol{W}^{(0)} \boldsymbol{x} \,. \tag{2.9}$$

To establish a connection with a well-known model, a linear regression can be performed by a neural network without hidden layers, meaning that the input layer is directly connected to the output layer through an affine transformation. This simplified architecture is based on the hypothesis that the relationship between the input and output is linear. To overcome this limitation, one may deal with "*a more general class of functions*" by incorporating one or more hidden layers. However, stacking hidden layers on top of each other requires tracking and updating additional parameters without improving the results. The composition of an affine function is itself an affine function, meaning that the model remains a linear model.

To fully utilize the benefits of multilayer architecture, a non-linear *activation function σ* is applied to each hidden unit following the affine transformation. This activation function is the key ingredient to bring the non-linearity required to reconstruct any function and to consider neural networks as universal approximators. Activation functions are really central in deep learning. They should be differentiable operators that add non-linearity. To summarize, incorporating non-linearities helps to build expressive neural network architectures. Figure 2.3 presents four activation functions with their corresponding formulas. Note that ReLU, which stands for *rectified linear unit*, and its counterpart Leaky ReLU are piecewise linear and are not differentiable at $x = 0$, where x represents the abscissa of the graph. By default, the left-hand-side derivative is applied when the input is zero. The sigmoid and the Tanh activation functions are both squashing functions because they squash any input in the range $(-\infty, +\infty)$ to a value between 0 and 1 (or -1 and 1).

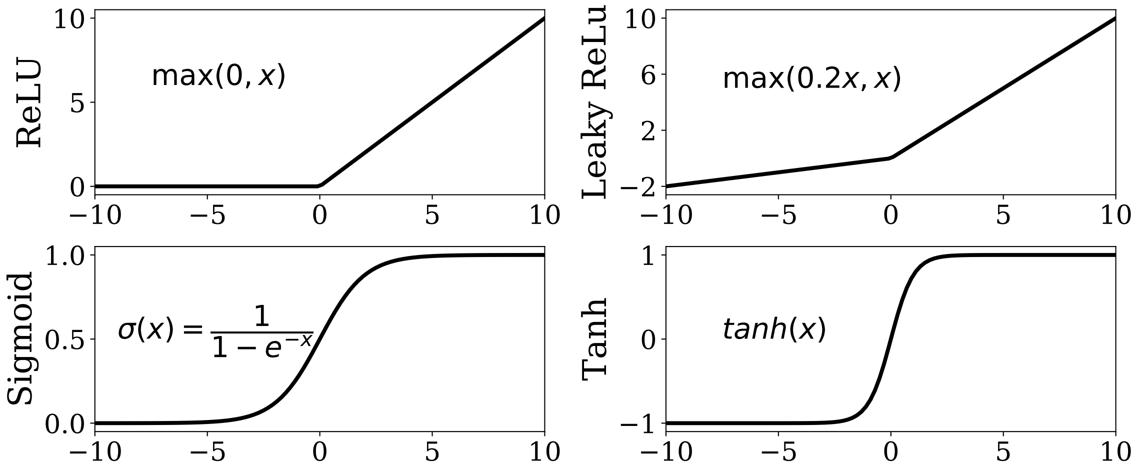


Figure 2.3: Activation functions and their formula.

Considering an activation function σ, the state $h^{(k)}$ at layer k can be written as,

$$\boldsymbol{h}^{(k)} = \sigma\left(\boldsymbol{W}^{(k)}\boldsymbol{h}^{(k-1)}\right) . \tag{2.10}$$

Base on Equation 2.9 and the notation defined in Figure 2.2, the prediction of the neural network $\hat{y}$ is written as a function of the the input $\boldsymbol{x}$, the weights $w_{ij}^{(k)}$ and the activation function as,

$$\hat{y} = f(\boldsymbol{x}; \boldsymbol{W}) = \boldsymbol{W}^{(3)}\,\underbrace{\sigma\left(\boldsymbol{W}^{(2)}\,\underbrace{\sigma\left(\boldsymbol{W}^{(1)}\,\underbrace{\sigma\left(\boldsymbol{W}^{(0)}\boldsymbol{x}\right)}_{=\boldsymbol{h}^{(0)}}\right)}_{=\boldsymbol{h}^{(1)}}\right)}_{=\boldsymbol{h}^{(2)}}$$

No activation function is used in the last layer because a regression problem aims at predicting a continuum of values. For classification problems, a softmax function is usually imposed at the output layer. Different activation functions can be incorporated into the same neural network. Activation functions are not optimized during the training process as the user defines them. Only the weights and bias are parameters that are fitted.

2.2.1.2 Gradient descent

Training a neural network is nothing more than optimizing over a very large parameterized space. The model parameters $w_{i,j}^{(k)}$ are fitted through an optimization procedure based on the definition of an appropriate objective function which is a measure of *fitness*. Such a concept has already been introduced as the loss function in Section 2.2. The loss function is defined with respect to the model parameters and depends upon the data. The learning process consists of minimizing the loss evaluated on the *training set* over and over to get the *"best"* model parameter values. For a regression problem, the loss function is the MSE and is given by Equation 2.8. Thanks to the definition of the neural network, there is an exact evaluation of the gradient of the loss with respect to each model parameter ($\partial\mathcal{L}/\partial w_{ij}^{k}$). The gradient indicates the direction of the steepest slope to be followed to reach the minimum of the loss: this optimization algorithm is the gradient descent (GD). The gradient is evaluated on a subset of the training database for better convergence. This method is called mini-batch stochastic gradient descent (SGD). Such a method visits the samples in mini-batches (of size B) and updates the parameters using the following rule:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \gamma \boldsymbol{g}_t \quad \text{where} \quad \boldsymbol{g}_t = \frac{1}{B}\sum_{b=1}^{B} \nabla\mathcal{L}\left(y_b, f(\boldsymbol{x}_b; \boldsymbol{W})\right) , \tag{2.11}$$

where γ is the learning rate, a hyperparameter which determines the step size (at each epoch) in the optimization algorithm. The learning rate can be modified manually or automatically to avoid getting stuck in local minima. However, as the gradient landscape can be very complicated, convergence to a global minimum is not guaranteed. The gradient of the loss is computed using *backpropagation*. The chain rule is the main idea behind the backpropagation. The gradient is computed from the output to the input layer in a reverse accumulation approach. This approach is cheaper than the forward accumulation because the output size is generally smaller than the input size.

2.2.1.3 Data splitting

The database is subdivided into three distinct sets: the *training*, *validation*, and *test* sets. The construction of the database and its deep analysis using space-time correlation are presented in Chapter 4.

The **training set** is used to fit the model parameters by backpropagation. Moreover, the training set is subdivided into mini-batches of size B to apply the mini-batch SGD defined by Eq. 2.11. The training set contains the data used to fit the learnable parameters $\boldsymbol{W}$.

The **validation set** contains data used to evaluate the performance of the current parameters. This set is a part of the database that is not seen by the network (i.e., not used to fit the model parameters) but is used to monitor the training: (i) to evaluate the loss and (ii) to check if the model is overfitting. Indeed, a model can perform well on the training data, but there is no guarantee that it will do as well on unseen data. In *"good"* training, the training and validation losses should be close to each other and decrease with the number of epochs. However, at some point, the validation loss will start to increase. At that point/epoch, the training is stopped: this is called *early stop*. Beyond this point, the neural network overfits the data: the model starts to get very close to the training data but does not generalize well. This behavior should be avoided, otherwise, the network is unlikely to work properly in production. Note that the validation set is also used to tune the *hyperparameters*. Hyperparameters are external configuration variables such as the number of hidden layers, the kernel size, the learning rate, and the activation function, to cite a few.

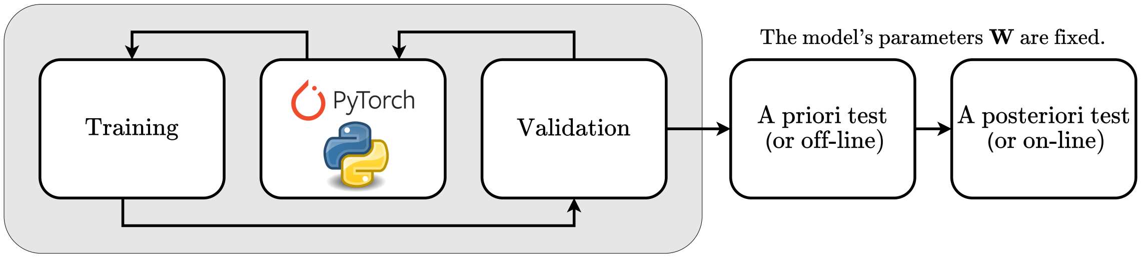

The model's parameters **W** are fixed.
Training
PyTorch
Validation
A priori test (or off-line)
A posteriori test (or on-line)

Figure 2.4: Proper evaluation protocol with three sets of data: a training set to fit the model parameters, a validation set to tune the hyperparameters of the model, and a test set to evaluate the performance of the model.

The **test set** is a completely different set that is neither used to train the model nor to monitor the training. It is instead used to give an unbiased estimate of the final tuned model performances. Figure 2.4 illustrates the proper evaluation protocol.

The *a priori* test (or offline evaluation) is performed in the same environment as the training one, but the network is evaluated on the test set. In contrast, the *a posteriori* test (or online evaluation) is when the model is actually implemented in a solver and starts to make predictions and interact with the solver environment.

2.2.1.4 Training process

The entire training process is illustrated in Figure 2.5. The training process requires a loss function (Eq. 2.8), a model (see Section 2.2.1.1), an optimization procedure (or rule), and a database (see Section 2.2.1.3). The pseudocode reads as follows. There are two loops, one over the number of epochs and one over the mini-batches (of size B). One epoch corresponds to one entire cycle through the training data. The learnable parameters $\mathbf{W}$ are updated on each mini-batch $(\mathbf{x}_b, y_b)$ using the rule of Equation 2.11, where the predictions, used to evaluate the loss, are computed based on the *forward pass*. For example, if the training data consists of 1,000 samples and that $B = 100$, then there will be 10 updates of the learnable parameters for one epoch before moving on to the next.

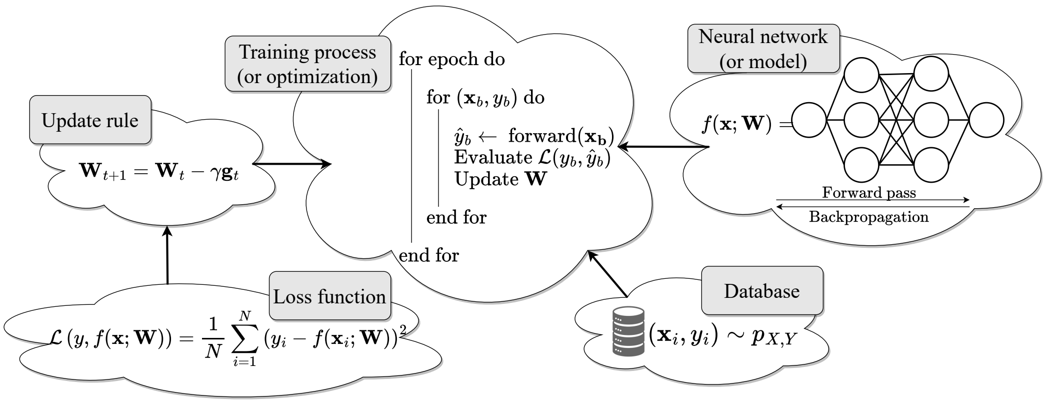

Update rule
$\mathbf{W}_{t+1} = \mathbf{W}_t - \gamma \mathbf{g}_t$
Training process
(or optimization)
for epoch do
for $(\mathbf{x}_b, y_b)$ do
$\hat{y}_b \leftarrow$ forward$(\mathbf{x_b})$
Evaluate $\mathcal{L}(y_b, \hat{y}_b)$
Update $\mathbf{W}$
end for
end for
Neural network
(or model)
$f(\mathbf{x}; \mathbf{W}) =$
Forward pass
Backpropagation
Loss function
$\mathcal{L}\left(y, f(\mathbf{x}; \mathbf{W})\right) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f(\mathbf{x}_i; \mathbf{W})\right)^2$
Database
$(\mathbf{x}_i, y_i) \sim p_{X,Y}$

Figure 2.5: Schematic of the entire training process, including the network architecture, database, loss function definition, and optimization or training process.

A few words about regularization The aim of a neural network is to learn how to map the inputs onto the outputs (i.e., to learn mapping functions). The *capacity* of a network corresponds to the range of the types of mapping functions that the model can approximate/learn. The capacity is controlled by the hyperparameters of the learning algorithm. For a polynomial fit, the model capacity is directly related to the interpolant degree. For a neural network, these hyperparameters can be the number of layers, neurons per layer, training iterations, and the regularization terms. If the capacity is too small, the model will underfit the data. If the capacity is too high, the model will overfit the data. To limit the overfitting, regularization terms can be added to the loss function. The L_2 regularization or weight decay is applied to the learnable parameters $\boldsymbol{W}$ with a given coefficient λ and added to the loss function. This term is now part of the optimization procedure. Weight decay is a technique widely employed to regularize parametric machine learning models.

2.2.2 Convolutional Neural Network

Up to now, the only way to deal with rich structures such as two-dimensional images was to discard the spatial structure of the image by flattening it into a one-dimensional vector before feeding it to an MLP (discussed in Section 2.2.1), as shown in Figure 2.6. However, such an approach is unsatisfactory because the knowledge that nearby pixels are typically related to each other is lost.

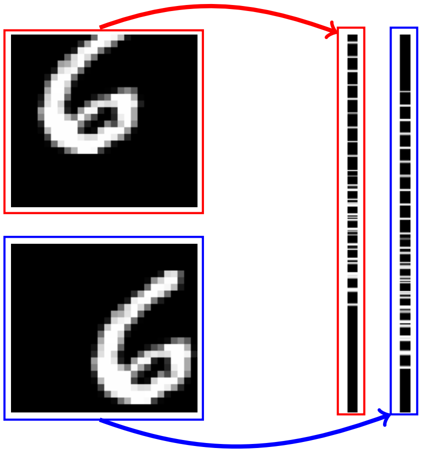


Figure 2.6: Images of the digit ’6’, translated vertically and horizontally and then flattened to obtain the one-dimensional vectors plotted on the right.

In Figure 2.6, the same digit '6' is translated vertically and horizontally and then flattened to obtain the one-dimensional *unstructured* vector on the right. Although the two images contain the same information, their flattened versions are not similar, and if fed to an MLP, the prediction could change. An MLP is thus not invariant to translation. Convolutional Neural Networks (CNNs) exploit spatial invariance to learn useful representations with fewer parameters. Since convolutions can be easily parallelized across GPUs and require fewer parameters than fully connected architectures, CNNs are more computationally efficient.

Instead of applying a matrix-vector product, where each input entry is assigned a different weight, a CNN performs a matrix-matrix dot product, where the coefficients do not change from one input entry to another. In other words, a convolutional layer applies the same linear transformation locally everywhere while preserving the signal structure. For one-dimensional tensors, given an input vector $\boldsymbol{x} \in \mathbb{R}^K$ and a convolutional *kernel* $\boldsymbol{u} \in \mathbb{R}^k$, the discrete convolution, denoted as $\boldsymbol{x} \circledast \boldsymbol{u}$, is a vector of size $K - k + 1$, defined as,

$$(\boldsymbol{x} \circledast \boldsymbol{u})\,[i] = \sum_{m=0}^{k-1} x_{m+i} u_m \,.$$

Figure 2.7 embodies this operation on a 6 × 4 image, using a kernel of size 2 × 2.

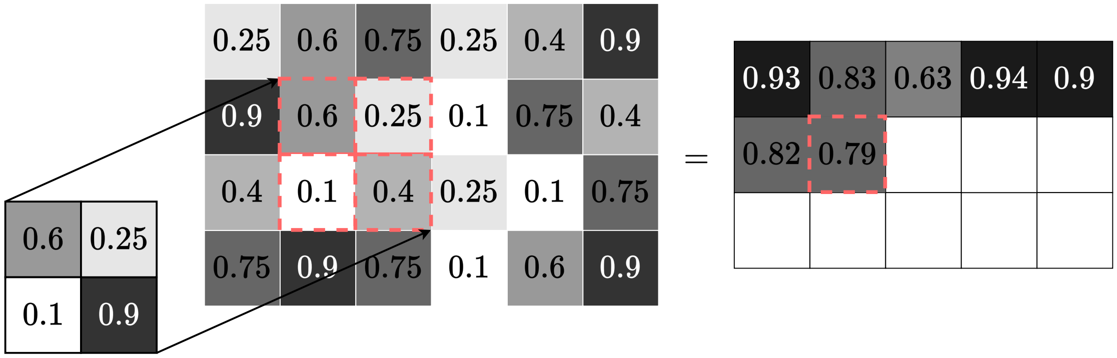


Figure 2.7: Two-dimensional convolution layer with a kernel size of 2 × 2.

Figure 2.8 shows the application of 3 × 3 convolutional kernels are applied to the dog image. We can try to interpret what the filter does to the image. The first kernel detects the edges of the dog while ignoring the background. The second kernel seems to detect the dog's whiskers, which are highlighted in white. The third kernel creates a negative of the original image. Kernels extract features from the images. The more complex an image is, the more convolutional layers are required.

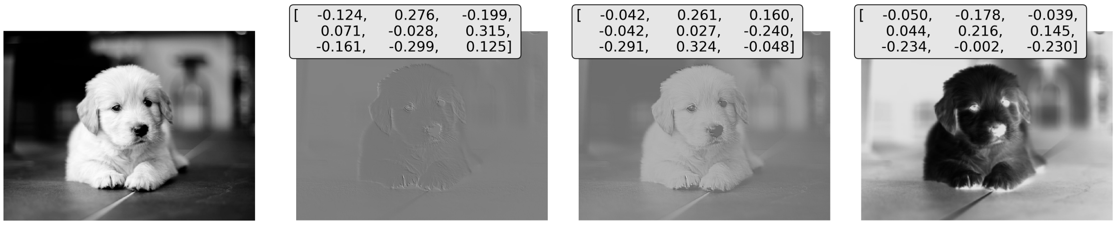


Figure 2.8: 3×3 convolutional kernels are applied to a dog image; the value of the kernel is specified on the top left.

A convolution layer has three additional hyperparameters.

- The *padding* specifies the number of extra fill "pixels" (or zeros) to be added around the input signal edges, thereby increasing the effective size of the signal.
- The *stride* defines the step size when moving the kernel across the signal.
- The *dilation* modulates the expansion of the filter without increasing the number of parameters.

These three hyperparameters are explained, in detail in the `PyTorch` documentation (Paszke *et al.*, 2019).

Each pixel is associated with one or multiple numerical values, depending on whether it is a greyscale or color image. These numerical values are encoded as channels in a CNN. For a color image, there will be three channels: one for red, one for green, and one for blue. In the context of wall shear stress models, these input channels are quantities such as the velocity and pressure fields extracted at the matching location. The selection of these fields is of high importance. The input and output need to be highly correlated for the network to detect an existing (non-)linear relationship. The selection of these inputs is discussed in Chapter 4.

Finally, the training procedure using `PyTorch` is similar to the one presented in Section 2.2.1.4, where the parameters to be fitted are the convolutional layers kernels.

2.3 Feature Selection

Prior to training the selected machine learning model on the collected data, it is desirable to evaluate the relevance of the input variables with respect to the target variable. This step, known as feature selection, can reduce the computational cost of the training and improve the model's performance. In predictive modeling, databases may contain a large number of variables that can slow down the development of the model and require a larger memory system. Furthermore, the model's performance may deteriorate when irrelevant (or redundant) input variables are included in the training. When developing a predictive model, it is important to perform feature selection, which aims to reduce the number of input variables. Feature selection is also useful to improve the model interpretability, and to help in scenarios with many features but few training examples (i.e., curse of dimensionality).

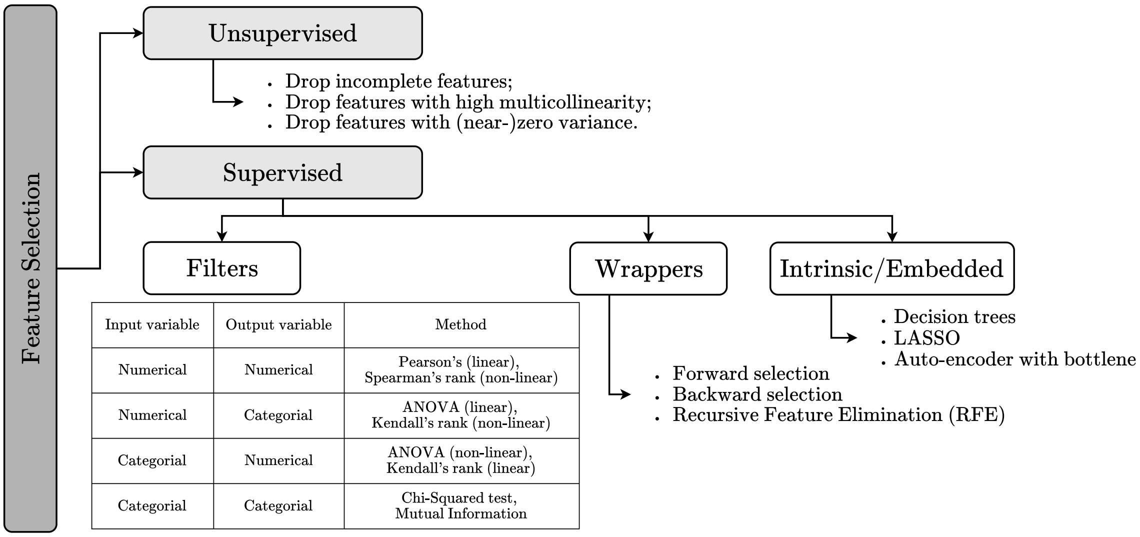
Feature selection can be decomposed into (i) *supervised* and (ii) *unsupervised* methods. Supervised feature selection techniques make use of the target variable (e.g., to identify the irrelevant variable), while the unsupervised ones ignore the target variables (e.g., methods that remove redundant variables using correlation). The remainder of this section focuses on supervised methods.

As illustrated in Figure 2.9, the supervised approach can be further divided into the wrapper, filter, and intrinsic methods.

Wrapper feature selection methods generate multiple models using all possible combinations of features and select the features that produce the best-performing model based on a performance metric. The subsets are formed using greedy approaches: forward selection or backward elimination. In the former, input features are added one by one until performance no longer improves. In the latter, input features are removed one by one until performance begins to deteriorate. One good example of backward elimination is the Recursive Feature Elimination (RFE). Although the wrapper method is computationally expensive, it is not concerned about the variable type (i.e., numerical or categorial).

Filter feature selection methods employ statistical techniques to evaluate the relationship between the target variable and each input variable. The retained input variables are those with the "strongest" relationship. Even though these methods are effective and fast, the choice of the statistical measures is highly dependent on the type of input and output variables (e.g., numerical or categorial). The table in Figure 2.9 lists the different statistical-based methods used to filter/select the input features. Considering a regression problem with numerical input and output variables, correlation coefficients are widely employed. Pearson's correlation coefficient is used for linear correlations, while rank-based methods are used for non-linear correlations. Since the statistical measures are computed one input at a time with the target variable, they refer to univariate statistical measures. Therefore, the interaction between input features is not considered.

Finally, *intrinsic (or embedded) methods* are machine learning algorithms that perform feature selection automatically during the learning process of the model. They combine the qualities of the wrapper and filter methods to produce the best possible subsets of features. These methods include penalization regression models such as LASSO and decision trees.


Feature Selection
Unsupervised
. Drop incomplete features;
. Drop features with high multicollinearity;
. Drop features with (near-)zero variance.
Supervised
Filters
Wrappers
Intrinsic/Embedded
. Decision trees
. LASSO
. Auto-encoder with bottlene
. Forward selection
. Backward selection
. Recursive Feature Elimination (RFE)

Input variable	Output variable	Method
Numerical	Numerical	Pearson's (linear), Spearman's rank (non-linear)
Numerical	Categorial	ANOVA (linear), Kendall's rank (non-linear)
Categorial	Numerical	ANOVA (non-linear), Kendall's rank (linear)
Categorial	Categorial	Chi-Squared test, Mutual Information

Figure 2.9: Overview of Feature Selection Techniques.

2.4 SHapley Additive exPlanations

Machine learning models are often described as "*black-box*", which makes their interpretation difficult. Single metrics, such as accuracy, are insufficient for describing all the necessary measures for most real-world applications. Therefore, Explainable ML techniques are necessary to comprehend the main features that affect model output and why a model makes a particular prediction. Roscher *et al.* (2020) distinguished between transparency, interpretability, and explainability in the literature on explainable ML techniques. Transparency involves understanding the ML approach. Interpretability is about the ML model together with data. Explainability involves understanding both the model, the data, and human involvement.

> **Transparency** If the model design and structure can be described and motivated, then it is considered transparent.
>
> **Interpretability** If we know what features the algorithm bases its decision on, then the model is considered interpretable.
>
> **Explainability** If the algorithm's decision can be made consistent with contextual information, then the model is explainable.

Although various methods have been recently suggested (e.g., Saliency Maps, Activation Maximization), it remains unclear how these techniques are related and when to use one over another. Lundberg and Lee (2017) proposed the SHapley Additive exPlanations (SHAP) values, rooted in cooperative game theory, as a unified approach for interpreting model predictions. The SHAP values are the Shapley values of a conditional expectation function of the original model. The Shapley value computation involves averaging the marginal contributions of each feature across all potential feature permutations. This computation has to assess every possible combination of features and determine the effect each feature has on the model prediction when included in these combinations. A balanced and interpretable evaluation of the importance of each feature in the model prediction is achieved by averaging the contributions across all possible feature arrangements. To summarize, the concept of SHAP values is to assign a specific value to each input feature, indicating its contribution to a particular prediction. Although SHAP helps to understand how individual features impact a model output, it does not assess the prediction quality.

The question is what is the average marginal contribution of a feature (AMC) to the outcome? SHAP estimates Shapley values for single input features. Shapley values provide a statistic for the AMC of a feature. Let us assume that we have F features and train an ML model with a real-valued outcome on every possible subset of those, we get thus 2^F models. To compute the AMC of the feature f, take all $(2F-1)$ models trained without f and compute the difference between the outputs of those models and the corresponding model with f added and average them appropriately.

The main question is, what is the average marginal contribution (AMC) of a feature to the outcome? The Shapley values provide a statistic for the AMC of a feature, and SHAP estimates the Shapley values for single input features. Considering F features, we have to train 2^F ML models (i.e., a model with a real-valued outcome), one for every possible subset of those F features. The AMC of feature f is evaluated as follows. The difference between the output of the $(2F-1)$ models trained without f and the corresponding model with f added is computed. These differences are averaged appropriately.

Let us consider an example of an ML model with three features, denoted U, V, and W. To evaluate the SHAP values of these three features, one would need to train 8 models with the following sets: ∅, U,V ,W ,UV ,UW , VW, UVW. The 8 models and their outcomes are summarized in Table 2.1.

Table 2.1: Example of a model with three inputs features U, V, and W.

	Models with/out U				**Models with/out V**				**Models with/out W**			
In.	∅	V	W	VW	∅	U	W	UW	∅	U	V	UV
Out.	30	35	25	35	30	50	25	60	30	50	35	60
In.	U	UV	UW	UVW	V	UV	VW	UVW	W	UW	VW	UVW
Out.	50	60	60	50	35	60	35	50	25	60	35	50
Diff.	20	25	35	15	5	10	10	−10	−5	10	0	−10

To compute the SHAP value of feature U, the variation of the output while considering or not this feature as an input of the model is evaluated by computing the difference between the model with entry U and the one with no entry at all (i.e., ∅). Averaging is performed in a weighted manner: (i) the sum of all weights is one, and

(ii) the sum of all weights is constant for models with a fixed number of features.

$$\begin{aligned}
\text{SHAP(U)} &= 20\left(\frac{1}{3}\right) + 25\left(\frac{1}{6}\right) + 35\left(\frac{1}{6}\right) + 15\left(\frac{1}{3}\right) = \frac{65}{3}\\
\text{SHAP(V)} &= 5\left(\frac{1}{3}\right) + 10\left(\frac{1}{6}\right) + 10\left(\frac{1}{6}\right) - 10\left(\frac{1}{3}\right) = \frac{5}{3}\\
\text{SHAP(W)} &= -5\left(\frac{1}{3}\right) + 10\left(\frac{1}{6}\right) + 0\left(\frac{1}{6}\right) - 10\left(\frac{1}{3}\right) = \frac{-10}{3}
\end{aligned}$$

From this simple example, one may conclude that the feature U has the largest Shapley value. This result translates as "*the AMC of U to the outcome is the largest*". Shapley values are generally only meaningful for the outputs of a single input data point, i.e., Shapley values are only an instance-based statistic.

Note that the above process becomes very challenging and cost-effective as the number of features increases. However, Lundberg and Lee (2017) combined insights from several additive feature attribution methods to get an approximation of the SHAP values. The most notable algorithms developed to calculate SHAP values are KernelSHAP, TreeSHAP, and DeepSHAP.

SHAP values have three essential properties: efficiency, symmetry, and additivity.

Efficiency The sum of all SHAP values indicates the collective impact of the model features on its prediction. SHAP values explain the model output by quantifying how each feature deviates the prediction from the average.

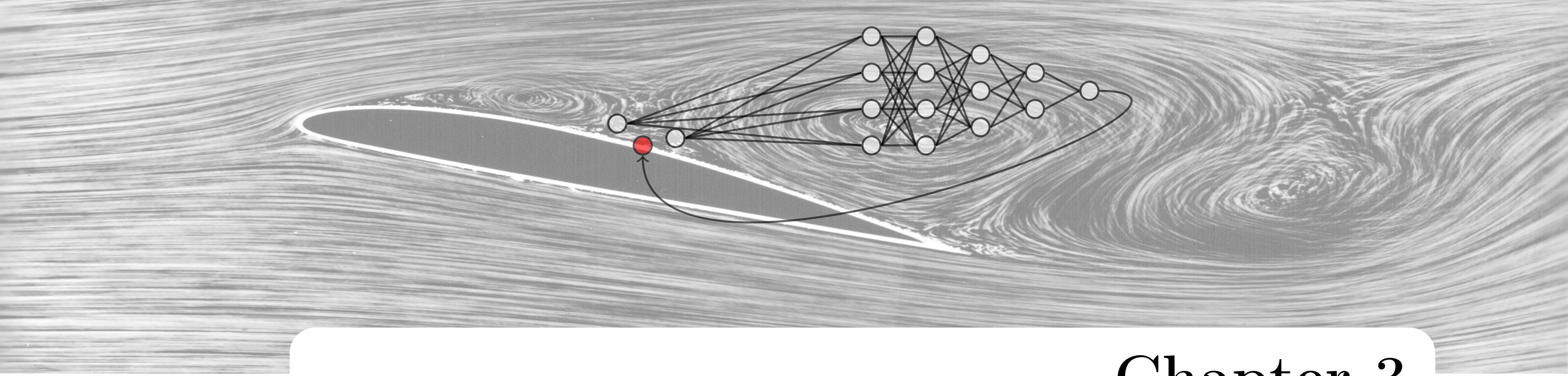
Symmetry Two features with equal contribution to a prediction will have the same SHAP values. This property guarantees fairness in assigning importance to features.

Additivity To understand the combined effect of multiple features on the model output, the joint contribution of several features to a prediction can be evaluated by adding the corresponding SHAP values.

The SHAP values will be employed to interpret the data-driven wall shear stress model trained over a separated flow in Chapter 8 to understand the relative impact of the averaged pressure gradient compared to the velocity field as input feature. SHAP values are evaluated using the SHAP package of Python that implements the KernelSHAP algorithm.

2.5 A few words about PyTorch

Among the available Deep Learning libraries, `PyTorch` is selected as the development platform of our novel data-driven wall models. `PyTorch` is an open-source software (machine learning framework), released under the modified BSD license, dedicated to both research environment and production deployment. It is a GPU-accelerated Python tensor computation package. It is designed to offer great flexibility and increase the speed of implementation of deep neural networks. It is currently one of the most popular libraries for AI researchers and practitioners worldwide, in academia and industry. `PyTorch` is the popular choice for fast experimentation and prototyping. It is widely used for image recognition and language processing, for instance. Its closest competitor is TensorFlow. They both have their pros and cons, but the major difference between the two is that `PyTorch` creates a dynamic computational graph behind the network, which makes it scalable to different dimensional inputs and easy to debug. For reproducibility, the trainings are performed using `PyTorch` version 1.12.1 with `Python` 3.10.4, `numpy` version 1.23.1, and CUDA 11.8.



Chapter 3
State of the art in wall modelling

This chapter reviews both standard wall models (Section 3.2) and new data-driven wall shear stress (WSS) models that have emerged in the last decades (Section 3.3). Indeed, the scientific community has already started to investigate the use of Machine Learning (ML) and Deep Learning (DL) techniques to address various challenges in fluid dynamics, flow control, and optimization with (experimental and numerical) data (Brunton *et al.*, 2020; Duraisamy *et al.*, 2019), and more recently, in the development of wall models. However, before going into the details of the existing wall models, the basic concepts of boundary layers are briefly reviewed in Section 3.1. The chapter concludes with the definition of a list of implementation constraints that will dictate the permissible architecture for the neural networks (Section 3.5).

3.1 Boundary layer nomenclature

When a flow passes an object at rest, the fluid is disturbed and forced to bypass it. From a microscopic point of view, boundary layer flow is described by the interaction between free molecules and the surface. The molecules closest to the wall are subject to diffuse reflections with the solid wall, in which the molecules are reflected randomly and isotropically. As a result, these molecules lose their momentum. The molecules above them also lose momentum in their collisions with the molecules closest to the surface, which then reduce the momentum of the molecules above them, and so on (see left-hand graph in Figure 3.1). The development of this thin layer, called the boundary layer, is directly related to the presence of the wall, which has an impact up to a certain distance in the fluid. This explanation does not hold for rarefied gas (i.e., the gas is not dense enough to ensure a high collision rate between molecules).

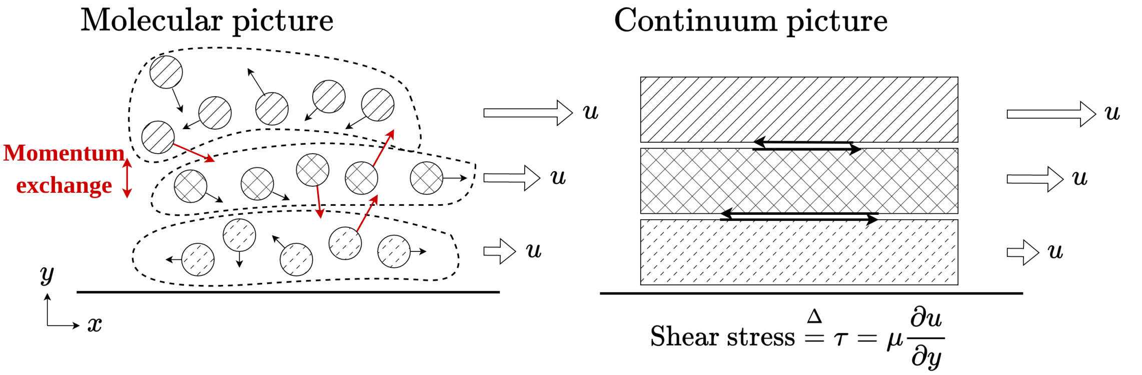

Molecular picture
Continuum picture
Momentum exchange
u
u
u
u
u
u
y
x
Shear stress $\overset{\Delta}{=} \tau = \mu \frac{\partial u}{\partial y}$

Figure 3.1: Molecular versus continuum picture of the boundary layer, where y is the wall-normal direction.

From a continuum point of view, molecules can be grouped into layers of fluid, as shown in the right-hand diagram in Figure 3.1. As molecules exchange momentum, the interaction between two layers of fluid is materialized by a force (i.e., the rate of change of momentum). This force per unit area is called the shear stress and is directly proportional to the velocity gradient,

$$\tau = \mu \frac{\partial u}{\partial y}, \tag{3.1}$$

where μ is the fluid dynamic viscosity. Under this continuum perspective, in the near-wall region, the flow changes from zero (in the case of a no-slip boundary condition for stationary objects) to the free-stream velocity $\boldsymbol{u}_\infty$.

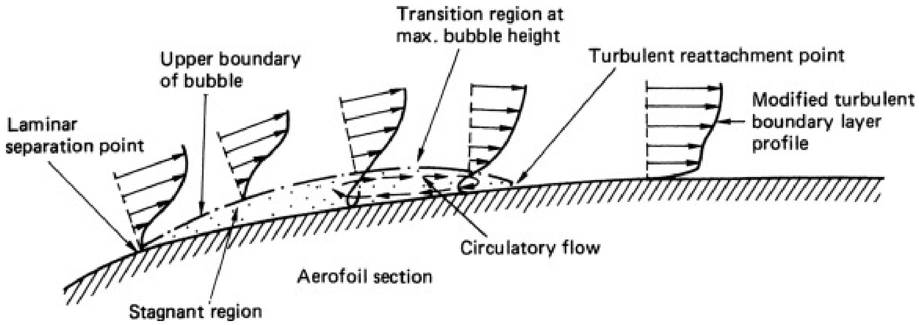

Transition region at max. bubble height
Turbulent reattachment point
Upper boundary of bubble
Modified turbulent boundary layer profile
Laminar separation point
Circulatory flow
Aerofoil section
Stagnant region

Figure 3.2: Laminar separation transition, extracted from Carlton (2019).

A boundary layer can be characterized by three different states: **laminar**, **transitional**, and **turbulent**. A boundary layer is considered laminar if it is a *smooth* and *ordered* flow that causes less skin friction drag than a turbulent boundary layer which is viewed as a *disordered* flow containing chaotic flow features known as eddies. Under certain conditions, the laminar boundary layer transitions to a turbulent boundary layer, or vice versa, known respectively as a transition and relaminarization. Transition is a complicated mechanism, and even today, it is not yet fully understood. The classification of transition is differentiated according to whether the flows are attached or separated. For attached flows, there exist three main types of transition. The **natural transition** occurs at relatively low free-stream turbulence. The development of two-dimensional Tollmien-Schlichting instabilities initiates the transition, followed by the rapid development of three-dimensional instabilities leading to characteristic Λ-structures. These structures break down into smaller vortices, which results in the appearance of turbulent spots. According to linear stability theory, one or more modes of the system are continuously amplified. The system is *unconditionally unstable.* The **transient amplification** occurs when the eigenvectors of the system are ill-conditioned, and the system is said to be *conditionally unstable.* This type of transition requires a certain level of perturbation to trigger the turbulent boundary layer. Finally, the **bypass transition**, according to Morkovin (1969), is caused by large disturbances in the external flow. This transition is quite common in gas turbine engines at high Reynolds numbers. For separated flows, the **separated-flow transition** or **separation-induced tran-**

sition, as shown in Figure 3.2, occurs in the free shear layer above a separation bubble. The laminar boundary layer, which is more sensitive to adverse pressure gradients, separates to form a free-shear layer. This free shear layer may transition and reattach further downstream as a turbulent boundary layer. Transition may also occur during the reattachment.

As already mentioned in Chapter 1, a turbulent boundary layer can be, according to the law-of-the-wall, decomposed in four regions: the viscous sublayer, the buffer layer, the log-law region and the wake region. Figure 1.4 plots the mean velocity profile in wall units u^+ as a function of the wall-normal distance from the solid surface in wall units y^+. This scaling is known as the boundary layer inner scaling. This scaling will be used throughout the thesis. The friction velocity u_τ is one fundamental parameter for the mathematical treatment of the boundary layer. It is defined as $u_\tau = \sqrt{|\tau_w|/\rho}$ where τ_w is the wall shear stress (Eq. 3.1 evaluated at $y = 0$, where y is the wall-normal direction) and ρ is the fluid density. Using this velocity and the kinematic viscosity ν, the viscous length scale ν/u_τ is obtained. These two characteristic scales are the inner scaling of the boundary layer and they non-dimensionalize the tangential velocity u and the wall distance y as

$$u^+ = \frac{u}{u_\tau} \quad \text{and} \quad y^+ = \frac{y u_\tau}{\nu} \,. \tag{3.2}$$

Under these scalings, self-similarity of the inner layer is observed. Another important dimensionless number is the friction Reynolds number Re_τ, which is also based on the wall shear stress and is defined as $Re_\tau = u_\tau \delta/\nu$, where δ is the boundary layer thickness. For external flows, both laminar and turbulent boundary layers grow as they move downstream (i.e., δ is thus a function of the streamwise direction).

The present study is mainly concerned with internal flows. However, the theory of the boundary layer is comparable to that of external flows, with the exception that the growth of the boundary layer is restricted in the wall-normal direction. Additionally, this study concentrates on the separation of turbulent flow, disregarding the concept of transition. The aim is to improve the understanding of how the pressure gradient affects separation and how it can be integrated into a data-driven WSS model.

3.2 Standard wall-models

In essence, wall-modeled LES accurately solves the outer layer solution while modeling the transport of momentum and heat transfer from the inner layer to the outer layer. In wall shear stress (WSS) models, the streaks, quasi-streamwise vortices, peaks of production and dissipation, and other dynamics of the inner layer are now represented by a single vector (value): the (magnitude of the) wall-shear stress. Due to the coarse mesh in the near-wall region, the standard no-slip boundary condition cannot be applied. As mentioned earlier, on a coarse mesh, the estimation of the velocity gradient will be incorrect, leading to an incorrect prediction of the velocity profile and, hence, the growth dynamics of the boundary layer. Approaching the wall, the velocity-fluctuation fields are characterized by smaller scales, and a coarse mesh cannot resolve the fine structures present in this near-wall region. Therefore, the wall model should act as a driver for boundary conditions on the wall shear stress. The correct wall shear stress (or total shear) needs to be found directly at the wall using the information available in the outer layer and the modeling of the inner layer.

The wall models operate under the assumption that applying the correct Neumann wall boundary condition in the wall-parallel velocity components is necessary to obtain an accurate outer layer solution. This assumption is justified for attached flows, as a constant-stress layer exists near the wall (De Graaff and Eaton, 2000). This existence implies that imposing the correct wall stress guarantees the correct transport of the streamwise momentum in the wall-normal direction. This is essential in the overall momentum balance in the boundary layer. However, for pressure gradient flows or separating flows, this argument does not hold as there is no equilibrium or constant-stress layer.

Deardoff (1970) made the first attempt to implement a wall model by simulating a channel at an infinite Reynolds number. A few years later, Schumann (1975) as working on channel flows at finite Reynolds numbers established conditions that directly link the velocity in the core to the wall shear stress components. In his approximated boundary condition, the mean stress was set equal to the given pressure gradient. Grötzbach (1987) rather considered the logarithmic law from which the mean stress was computed iteratively. He extended Schumann's boundary condition to flow configurations where the pressure gradient was not known in advance. Piomelli *et al.* (1989) introduced a downstream displacement to its model called the shifted boundary conditions, by requiring the wall shear stress to be correlated with the instantaneous velocity. This enhancement was based on the inclination of elongated structures near the wall studied by Rajagopalan and Antonia (1979). Piomelli *et al.* (1989) also proposed the ejection model based on the observation that the high-velocity fluid motion towards or away from the wall occurring during sweep-eject events significantly affects the wall stress.

To overcome the assumption that the logarithmic law-of-the-wall only holds in the mean, Mason and Callen (1986) focused on enforcing the logarithmic law locally and instantaneously while imposing the alignment of the wall shear stress with the outer horizontal velocity. According to their results, the validity of this assumption depends on the size of the averaging volume. Werner and Wengle (1993) used a power law to compute the local stress while Hoffmann and Benocci (1995) and Wang (1999) integrated the boundary layer equations with an algebraic turbulence model. However, Nicoud *et al.* (2001) noted that most models performed poorly at high Reynolds numbers, even in channel flows. Instead, they used suboptimal control theory to force the outer LES towards a desired mean velocity profile. To reduce the computational cost of this method, tables of correlations between the outer velocity and wall stress that accounted for numerical and SGS errors were generated. They finally derived a wall model from linear stochastic estimation (LSE) that produced encouraging results up to Re_τ 20,000. With the use of generalized additive models (GAM) and nonlinear, nonparametric regression, Abel *et al.* (2006) generalized Schumann's model, Piomelli *et al.*'s model, the ejection model, and the gradient model. The authors analyzed the near-wall physics and discovered that the pressure gradient had a strong influence in the viscous sublayer.

Most of the above methods relying on the law of the wall, which is valid for attached flow at moderate pressure gradients only, are limited in their ability to handle complex, unsteady and non-equilibrium flow features, including secondary flows and separation. Moreover, pressure gradients and other effects relevant to separation are not considered in these models. The Two-Layer Model (TLM) (Balaras *et al.*, 1996) and Detached Eddy Simulation (DES) (Spalart *et al.*, 1997a) can be considered as two hybrid RANS/LES approach that attempt to address these issues. The general feature of DES and its modification, delayed DES (DDES) (Spalart *et al.*, 2006), is that a large part of the attached boundary layer is treated by RANS, while the separated flow regions are processed by LES. Nonetheless, the TLM and the DES still have their own limitations. Although the TLM was used to numerically simulate square ducts, rotating channels, and backward-facing steps, to name a few, the method still suffers from two problems: the log-layer mismatch and the resolved Reynolds stresses inflow. Furthermore, the TLM still struggles to handle strong pressure gradients and separation. Breuer *et al.* (2007) used the approach of artificial viscosity to build a simple analytical model if an appropriate definition of the relative thickness of the viscous sublayer is obtained. Their analytical model was compared to different variations of Werner-Wengle's model on the two-dimensional periodic hill at $Re_b = 10595$ and encouraging results were obtained. Cadieux *et al.* (2016) addressed the separation problem using an integral wall model for LES with additional non-equilibrium terms. They obtained an analytically tractable integral formulation and successfully applied it to a flat plate subjected to an adverse pressure gradient. Krank *et al.* (2019) used a turbulent boundary layer velocity profile model to enrich the Discontinuous Galerkin (DG) solution in the near-wall region. This approach was successfully applied to channel and periodic hill flows. Although much progress has been achieved in this field, most existing equilibrium wall models (Piomelli, 2008; Bose and Park, 2018) are still unable to predict flow separations and reattachments.

All these models, and more generally all wall models, can be split into two categories according to Larsson *et al.* (2016): the hybrid LES/RANS and the wall-stress models. In the first category, the LES is defined above a certain interface, $y \leq y_{in} > 0$, explained in Section 3.2.1. For the second category, LES extends all the way down to the wall at $y = 0$ (Section 3.2.2). We can also consider wall models developed and implemented for high-order methods. Most of the wall models have been developed for finite volume methods but their transposition from one numerical discretization to another is not straightforward. Sometimes it is necessary to adapt the formulation because of the coupling between the physical model of the wall and the numerical method adopted. In her thesis, Frère (2018) discusses the implementation in the code Argo-DG of the analytical wall shear stress model based on Reichardt's velocity profile and other models, such as the two-layer model (TLM) and the Generalized law-of-the-wall of Shih *et al.* (1999). These quasi-analytical wall models are presented in Section 3.2.2. The WSS model based on Reichardt's velocity profile is one of the most commonly used wall models, especially in the industry, because it is cheap to evaluate. In the same vein, one can mention the work of Krank *et al.* (2019), who developed a novel approach to hybrid RANS/LES wall modeling based on function enrichment. This new model was implemented in a high-order discontinuous Galerkin scheme.

For a more detailed literature review on wall models, the reader is referred to Piomelli and Balaras (2002), Piomelli (2008), Larsson *et al.* (2016), Sagaut (2006), and Heinz (2020).

3.2.1 Hybrid RANS/LES

The hybrid RANS/LES methods can be seen as a mixture of the equations governing the Reynolds-averaged (RANS) and spatially-filtered Navier-Stokes equations (LES). The main idea is to modify the existing Reynolds-averaged closures to provide a scale-resolving capability in the computational domain where the grid resolution is sufficient to resolve the turbulent structures. These methods can be classified into two sub-categories: the *zonal* methods, which maintain a fixed matching location (i.e., h_{wm}) during the grid-refinement to be able to reach a grid-converged state, and the *seamless* methods, where the interface location depends on the grid and on the solution. Both methods suffer from the *log-layer mismatch.* At the interface, the mean velocity profile is not correctly maintained. This issue has been treated by Piomelli *et al.* (2003), Davidson and Dahlström (2005) who introduced small-scale forcing. Although the approach gives interesting results, the method presents robustness issues. Another approach, proposed by Choi *et al.* (2009), is to adjust the blending function between the LES and RANS eddy-viscosity.

3.2.1.1 Zonal methods

The two main classes are the non-equilibrium wall model (NEQWM) and its simplified version, the equilibrium wall model (EQWM).

Non-equilibrium wall model (NEQWM) This method solves the unsteady three-dimensional Navier-Stokes equations with RANS closures on a separated near-wall mesh to provide the Neumann wall boundary condition (viscous stress and heat flux) to the coarse LES. The kinematic non-penetration condition, $u_n = 0$, is maintained in the LES. The LES provides Dirichlet data, such as density, velocity, and pressure, for the top boundary at the wall model layer. Figure 3.3 illustrates the dynamics between the primal LES grid and the wall model layer with its own computational domain. The Navier-Stokes equations have the same complexity as in the primal LES grid but with some RANS parameterization for the turbulence within the wall model. Due to the large tangential spacing of the inner-layer wall-model

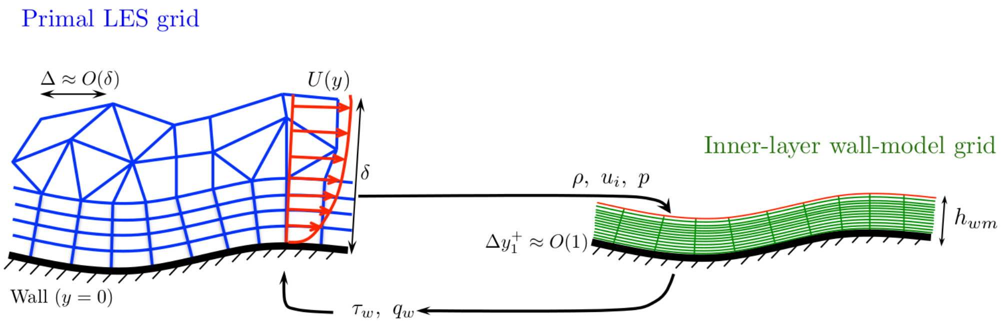


Figure 3.3: Illustration of the dynamics between the Primal LES Grid and the unsteady RANS separated near-wall mesh presented by Park and Moin (2016).

grid, we can only consider the average effect of near-wall vortices.

The Two-layer Model of Balaras *et al.* (1996), which applies to incompressible turbulent boundary layers, is a widely used NEQWM. The full TLM equations are derived by performing a Reynolds average of the NS equations and assuming that the boundary layer height is small compared to the characteristic length of the flow. This latter assumption makes it possible to neglect a few terms and to obtain a constant pressure through the boundary layer height. It yields to,

$$\frac{\partial}{\partial x_2}\left((\nu+\nu_t)\frac{\partial u_i}{\partial x_2}\right) = \underbrace{\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j)}_{=F_i}, \quad \text{for } i = 1,3 \text{ and } j = 1,2,3 \tag{3.3}$$

where ν_t is the eddy viscosity (found by using RANS-type models which may vary among authors), p is the pressure field, t is the time, (x_1, x_2, x_3) are the streamwise, wall-normal, and spanwise direction, and (u_1, u_2, u_3) are the corresponding velocity components. Full details of the equations can be found in Kawai and Larsson (2013).

Equilibrium wall model (EQWM) The simplified TLM is obtained by neglecting the time dependence and the convective terms while conserving the pressure gradient (i.e., $F_i = \frac{1}{\rho}\frac{\partial p}{\partial x_i}$). However, despite this simplification, solving the problem remains more complex than analytical laws because of the need to resolve an ODE on the auxiliary grid. This wall model yields satisfactory results in various cases, such as channel flow, backward step, and squared duct. However, there are ongoing discussions regarding the balance between the convective terms and the pressure gradient in the inner layer above the viscous sublayer. According to Larsson *et al.* (2016), the convective terms can be large and of the same order of magnitude as the pressure gradient. Therefore, in non-equilibrium cases, it is recommended to use the TLM under equilibrium conditions (i.e., $F_i = 0$) instead of solely disregarding the convective terms. Thus, the EQWM is the simplified version of the NEQWM obtained by neglecting the unsteady convection and pressure gradient terms. It has the effect of removing the tangential derivatives, leading to a significant simplification. The wall-normal viscous diffusion is not neglected. This model automatically assumes that the total shear stress is in equilibrium with the wall-shear stress by assuming the constant-stress layer. This strong assumption raises a concerning question: could this model potentially fail catastrophically? In wmLES, the LES domain covers about 80% of the boundary layer. In this part of the boundary layer, non-equilibrium effects are captured by the LES. The instantaneous information from this accurate outer layer is then fed into the wall model, whether or not the wall model accounts for non-equilibrium effects. It is possible that the wmLES may still be able to handle some non-equilibrium effects even though it was not originally intended to do so. The model also assumes that the wall shear force is aligned with the LES velocity at the matching location. This assumption can completely break down in many realistic flows, including in turbomachinery, where there is a misalignment of the velocity with the wall stress direction.

3.2.1.2 Detached Eddy Simulation

Spalart *et al.* (1997b) introduced a new hybrid RANS/LES approach called Detached Eddy Simulation (DES) to address high Reynolds number, massively separated flows. In DES, a RANS closure is used near solid walls where the flow is attached, while, for separated flow regions, a grid resolution-dependent SGS closure is employed. Roughly speaking, DES treats the boundary layers with RANS and the massively separated regions with LES. DES was originally developed for external aerodynamic flows (Spalart, 2009). DES encapsulates the general mechanics of wall modeling where RANS logic is applied close to the wall (i.e., in the region where the wall distance is much smaller than the boundary layer thickness but very large in wall units). The first attempt to use DES for wall modeling was made by Nikitin *et al.* (2000), who placed the switching location between RANS and LES in the logarithmic region. Nikitin *et al.* (2000) were the first to report that the DES also suffers from the log-layer mismatch (LLM), which is a deviation of the velocity profile that leads to the underestimation of about 15% of the skin friction.

To overcome this deficiency, the improved delayed DES (IDDES) was proposed by Shur *et al.* (2008). They proposed to create a unified set of formulas for both natural DDES and wmLES applications. This would allow for the automatic treatment of different regions within a single simulation over a complex geometry. The delayed DES (DDES) incorporates molecular and turbulent viscosity information into the switching mechanism, delaying this switching in boundary layers to handle *ambiguous* grids more effectively than the original method. However, IDDES, in its primal version, did not eliminate LLM as reported by Peterson and Candler (2011), who used IDDES as a wmLES for supersonic boundary layer flows.

3.2.2 Wall-stress models

These models are all based on the physical principle of momentum conservation in a nearly parallel shear flow. These wall models can be split into two categories: mathematics-based models (control, filters, etc.), and physics-based models (generally RANS-like models). Hybrid RANS/LES can be expensive and complex. The idea is to come back to simpler models but still take into account the non-equilibrium effects. With these new models, there are no more ODE or PDE to solve in space, thus no wall-model grid is required. In this section, three (physics-based) wall-stress models are presented: the integral wall model, the dynamic wall slip-velocity model, and the quasi-analytical wall model.

Integral wall models This method comes from the integration in the wall-normal direction of the x-momentum equation. The wall-shear stress is obtained as a function of the external pressure gradient and the direction of the streamwise momentum in the normal direction at the matching location h_{wm},

$$\tau_w = (\mu + \mu_t)\frac{\partial u}{\partial y}\Big|_{y=h_{wm}} - \frac{\partial L_x}{\partial t} - \frac{\partial p}{\partial x}h_{wm} - \frac{\partial L_{xx}}{\partial x} + \rho uv\Big|_{y=h_{wm}},$$

where, p is the pressure, ρ is the density, and $L_x = \int_0^{h_{wm}} \rho u \mathrm{d}y$ and $L_{xx} = \int_0^{h_{wm}} \rho u^2 \mathrm{d}y$ are two unknown terms because the velocity inside the boundary layer is unknown. The introduction of modeling assumptions on the inner layer scaling leads to the definition of the lifted virtual-wall model (Cheng *et al.*, 2015). A temporal ODE governing the wall shear stress is derived from the vertically-integrated momentum equation, by assuming a local-inner-layer scaling based on wall variables. In the model developed by Yang *et al.* (2015), the inner layer velocity scaling is approximated, and an analytical expression is assumed for it. They further integrate a linear perturbation to the log-law to model the advection and the pressure gradient more accurately. This modeling introduces five unknowns. The model consists of five non-linear equations to be solved. For more complex flows on a smooth wall, the number of unknowns rises to nine. Their model was tested over the wall-mounted three-dimensional cube at Reynolds 3850. The flow over the cube is well predicted, better than with a standard analytical model.

Dynamic wall slip-velocity model This model developed by Bose and Moin (2014) is free from the RANS legacy or RANS component that all the other models have. This model answers the question: *what boundary condition do the filtered scales satisfy at the wall?* The key component of their approach is the use of a specific differential filter of Germano (1986),

$$\overline{\phi} - \frac{\partial}{\partial x_k}\Big(\alpha \frac{\partial \overline{\phi}}{\partial x_k}\Big) = \phi \quad \alpha(\boldsymbol{x}) = 0 \text{ for } \boldsymbol{x} \in \partial\Omega, \tag{3.4}$$

where α is a parameter of the filter. Due to its analytical expression, the evolution of the raw data ϕ and its filtered version $\overline{\phi}$ can be examined when approaching the boundary. By expanding the expression of the filter (Eq. 3.4) and evaluating it at the wall, Bose and Moin (2014) obtained a Robin boundary condition,

$$\overline{u}_i|_w - \underbrace{\alpha|_w}_{=0} \frac{\partial^2 \overline{u}_i}{\partial x_k \partial x_k} - \underbrace{\frac{\partial \alpha}{\partial n}\Big|_w \frac{\partial \overline{u}_i}{\partial n}\Big|_w}_{(1)} = u_i|_w. \tag{3.5}$$

In Eq. 3.5, the term $\partial\alpha/\partial n$ is strongly related to the filter width. In a wall-resolved LES where a no-slip velocity is imposed at the wall, the filter width tends to zero at the wall and we are left with $\overline{u}_i|_w \approx u_i|_w = 0$. In a wall-modeled LES, the filter width is finite and since the term (1) is of order one, we end up with a Robin (or mixed) boundary condition that an under-resolved LES should satisfy at the wall,

$$\overline{u}_i|_w - C\Delta_w \frac{\partial \overline{u}_i}{\partial n}\Big|_w = u_i|_w = 0\,,$$

where $C\Delta_w$ is the slip-length. This constant is dynamically determined using the Germano identity.

Quasi analytical wall models Quasi analytical wall models are derived from the boundary layer equations which are simplified according to the following assumptions: incompressible, steady, two-dimensional flows, and no pressure gradient. From these assumptions, the logarithmic law-of-the-wall valid in the near-wall turbulent region is obtained. This law can directly predict the wall shear stress given the velocity at a wall-normal position, called the matching location or the wall-modeled height (h_{wm}). However, this location is very constrained by the validity of the law across the boundary layer. A smoothing or blending procedure between the viscous sublayer and log region is required to remove this constraint and obtained a more robust numerical implementation. Several authors such as Schlichting and Gersten (2017) or Spalding (1961), proposed this smoothing. For instance, the Reichardt's law-of-the-wall is given by:

$$u^+ = \frac{1}{\kappa}\log(1+\kappa\, y^+) + D\left(1-\exp\left(\frac{-y^+}{11}\right) - \frac{y^+}{11}\exp\left(\frac{-y^+}{3}\right)\right) , \qquad (3.6)$$

where $D = C - 1/\kappa\,\log(\kappa)$, $\kappa = 0.38$, and $C = 4.1$, which are the most recent constant suggested by Österlund *et al.* (2000) allowing for the best match with DNS. The advantage of these wall models is that they are easy to implement. The computational cost associated with their evaluation is low. They are based on the theory of a boundary layer subjected to a constant edge velocity $u_e(x)$. This hypothesis can be restrictive in the case of pressure gradients. In this case, the velocity $u_e(x)$ is no longer constant. The boundary layer can be accelerated in a favorable pressure gradient or decelerated in an adverse pressure gradient. The pressure gradient and the velocity are determined by the outer flow and depend on the Reynolds number, the blade geometry, and the angle of attack. This quasi-analytical law was implemented in Argo-DG during the thesis of Frère (2018) and validated on the channel flow at various friction Reynolds numbers (Frère *et al.*, 2017) and tested on the NACA4412 configuration (Frère *et al.*, 2018) at Re =1.6×10^6 where a discrepancy with the wrLES was observed due to the inherent modeling assumptions cited previously. She highlighted that capturing the adverse pressure gradient is a significant challenge in wall modeling since the latter required taking momentum flux into account.

3.2.3 Summary

Table 3.1 summarizes the different standard wall models presented in the previous sections.

Table 3.1: Summary of the different wall model classes.

Hybrid RANS/LES		**Wall-stress models**		
Two-layer zonal models	*Detached Eddy Simulation*	*Integral wall model*	*Wall-slip velocity model*	*Quasi-analytical wall models*
τ_w is obtained from RANS solved on a separated near-wall mesh (e.g., NEQWM and EQWM)	Major part of the attached BL is treated using RANS, while LES is only utilized in the separated flow regions	Use the vertically integrated momentum equation for determining the wall-shear stress	Robin boundary condition of the LES equation using a differential filter	Based on algebraic models
References				
Balaras *et al.* (1996), Cabot & Moin (2000), Diurno *et al.* (2001), Wang & Moin (2002)	Spalart *et al.* (1997), Shur *et al.* (2008), Han *et al.* (2020)	Chung & Pullin (2009), Cheng *et al.* (2015), Yang *et al.* (2015)	Bose & Moin (2014)	Deardorff (1970), Schuman (1975), Piomelli (1989), Werner and Wengle (1993), Gotzbach (1987)

3.3 Data-driven wall models

With the recent advances in computational power, the emergence of TPUs (**T**ensor **P**rocess **U**nits), combined with the exponential generation and accumulation of high-quality data, it is now possible to train more and more deep neural networks that, for example, help generate new engineering models. Neural Networks (NNs) of at least three layers are at the core of Deep Learning (DL) techniques, which is itself a subset of Machine Learning (ML). A NN is mainly a black box composed of thousands of parameters that can learn complex relations between its inputs and outputs and automatically detect features through data assimilation. The first neural network was designed by F. Rosenblatt in 1957. Over time, significant efforts have been dedicated to train neural networks efficiently and create new architectures that are tailored to specific tasks. In the last decade, the fluid mechanics community has progressively used ML and DL techniques for dimensionality reduction, closure (e.g., RANS closures and wall models) and subgrid-scale (SGS) models, flow control, uncertainty quantification and optimization with (experimental or numerical) data (Duraisamy *et al.*, 2019; Brunton *et al.*, 2020; Garnier *et al.*, 2021; Sharma *et al.*, 2023; Lino *et al.*, 2023). Data-driven wall models have emerged to complement existing wall shear stress models, reviewed in Section 3.2, by attempting to address their weaknesses.

Yang *et al.* (2019) developed a physics-informed data-driven wall model for the channel flows by training MLPs on filtered DNS data of the channel flow at a friction Reynolds number of 1000. They inject physical insights into the input data via a scaling inspired by the vertically integrated thin-boundary-layer equations and the eddy population density scaling. They argue that their scaling allows a less aggressive extrapolation to simulate the channel at $Re_\tau = 10^{10}$. Radhakrishnan *et al.* (2021) focused on a wall-shear stress model for channel flows using Gradient Boosted decision trees (XGBoost). Two-channel flows composed the database: one at $Re_\tau = 180$ and one at $Re_\tau = 1{,}000$. This database was then supplemented by a set of synthetically generated data obtained by rescaling the velocity field using the viscosity ν and the half-height of the channel δ. They tested three sets of features (i.e., set of input/output pairs): (i) primitive, (ii) scale-invariant, and (iii) dimensionless features. Their model does not directly predict the wall shear stress but the friction velocity instead. This choice may not be the best for separated flows. For separated flows, the wall shear stress tends to zero and therefore the friction velocity u_τ tends to infinity or to an undefined value. The set (iii) gives the best performances, and the obtained model was tested *a posteriori* on the channel at Re_τ = 2,005 and the flow over a wall-mounted hump. Their model performs correctly on the channel and is as good as the equilibrium wall model (EQWM). Regarding the more complex (the wall-mounted hump) case, their ML model and the EQWM fail to predict the streamwise velocity and stress profiles. Jamaat and Hattori (2023) *a priori* assess the performance of convolutional neural networks for predicting the wall shear stress in channel flows. The database consists of filtered DNS data at four friction Reynolds numbers. The input data set is selected based on an analysis of the thin boundary layer equation combined with the evaluation of the correlation coefficient between the wall shear stress predicted by the CNN and the one obtained from the filtered DNS data. The model predicted the wall shear stress with high correlation coefficients but for very small h_{wm}^+ values, which makes the model unsuitable for real configurations. Moreover, this non-local approach cannot be easily generalized to real geometries. A better idea would be to use local 2D patches.

These wall shear stress models target the channel flow configuration and attempt to equal or outperform the standard law-of-the-wall. Recent research also aims to overcome the existing weaknesses of standard wall models (e.g., strong adverse pressure gradients, separation, ...) using ML and DL techniques. Zhou *et al.* (2021) targets the turbulent separation by training a feed-forward neural network on the DNS of multiple geometries of the periodic hill (i.e., modification of the streamwise length). Although they obtained satisfactory *a priori* results, the *a posteriori* test fails completely in predicting the mean velocity profile on the nominal geometry at Re_b = 10,595. The work of Lozano-Durán and Bae (2021) is based on the hypothesis that any complex flow can be decomposed as a non-linear combination of simpler flows, called *building-block flows.* They developed a wall-flux-based wall model for LES using a self-critical machine-learning approach which was successfully trained on DNS data (e.g., flow over a flat plate, in a channel, in a turbulent duct, or separated flow at various Reynolds numbers). However, when applied to the NASA Juncture Flow, the model fails to predict the separation correctly. The lack of separated flows in the training data may be the source of this failure.

Dupuy *et al.* (2023a) focuses on data-driven wall models for separated flows by training a multi-layer perceptron (MLP) on filtered and sampled (to the LES formalism of Leonard) DNS data of two turbulent channel flows and the three-dimensional diffuser of Stanford. They made their model: Mach number invariant, Galilean invariant, and rotationally invariant. They showed that increasing the spatial information (i.e., the size of the input stencil) helps to discriminate between separated and non-separated flow regions. The model is tested *a posteriori* on the backward step, which is a geometry with flow separation from a geometric discontinuity. The model performs better than the standard law of the wall, but there is still a misprediction in the recirculation bubble. Dupuy *et al.* (2023b) continued their work on developing new wall shear stress (WSS) models using DL techniques for separated flows. They addressed the problem using graph neural networks (GNN) which are considered efficient tools for encoding unstructured data (e.g., data extracted from an unstructured mesh). The DNS data of three channel flows, a three-dimensional diffuser, a backward-facing step, an adverse pressure gradient test case, and a NACA blade at two angles of incidence were filtered onto coarser meshes, i.e., representative of LES meshes. In addition to be Mach number invariant, Galilean invariant, and rotationally invariant (Dupuy *et al.*, 2023a), the model is also orthogonal invariant, equivariant under reflection, and independent of the coordinate system. The authors observed that $N \geq 3$ message passing steps[1] are necessary in order to correctly predict wall shear stress and, hence, discriminate among the various flow configurations. This observation is consistent with the analysis of space-time correlations (Boxho *et al.*, 2022). The backward-facing step is simulated using their GNN-WSS model. Although the predictions are improved by increasing the message passing steps, the reattachment location is still underestimated. Moreover, the assumption used in the present shear-stress paradigm saying that the wall shear stress vector is aligned with the tangential velocity is unrealistic at the reattachment location, which may lead to a significant misprediction of this location. Finally, they applied their model to the NACA blade at 7° angle of incidence. The model shows a clear improvement in reducing the WSS prediction compared to the standard law-of-the-wall.

Unlike many other papers on data-driven wall models, Lee *et al.* (2023) took a different direction in defining the input data for the model. According to the authors, the input variable extracted from the law-of-the-wall can lead to poor performance when the configuration features complex flow physics. To bypass this observation, they employed the Fukagata-Iwamoto-Kasagi (FIK) identity to predict the skin friction coefficient coupled with an Artificial Neural Network (ANN). FIK can be employed

[1] Neural Message Passing is an important concept in Graph Neural Networks that enables information exchange and aggregation among nodes in a graph. It allows model dependencies and interactions in graph data to be taken into account.

to estimate skin friction in complex flows. However, this identity leads to a misprediction of skin friction in wmLES due to the lack of near-wall flow contribution. The authors used ANN to reconstruct skin friction in the absence of near-wall information. The generalizability of this technique is questionable because the FIK identity decomposition depends on the flow physics. *A posteriori*, the mean velocity profile agrees well with DNS results at the lowest Re_τ. At the higher Reynolds number, a log-layer mismatch is observed at $y^+ < 300$. This mismatch becomes even more pronounced when the mesh is coarsened. The model is also applied to the separated turbulent boundary layer flow in a periodic domain where the upper wall is subjected to blowing and suction. The model shows relatively good agreement regarding the separation and reattachment points compared to the DNS reference.

Zhideng *et al.* (2023) trained an MLP on DNS data of the two-dimensional periodic hill and synthetic data to predict the two-wall parallel components of the wall shear stress. The synthetic data are extracted from the law-of-the-wall for a large set of friction Reynolds numbers and wall-normal heights. The *a priori* test shows a clear benefit in the wall shear stress prediction for the periodic hill and synthetic data compared to the training on the periodic hill data alone. *A priori*, the model generalizes to two slight variations of the baseline geometry at $Re_b = 10{,}595$. The model captures the rapid change of friction coefficient amplitude on the lower wall, which is even more significant in the shorter domain. *A posteriori*, only the channel flows ($Re_\tau = 10^3$ to 1.2×10^8) are simulated with the data-driven wall model. They showed that as a standard wall model, the behavior of a data-driven wall model is also influenced by the subgrid scale models.

The above WSS models are developed using decision trees or neural networks (e.g., MLP, CNN, and GNN). There is another subset of machine learning: reinforcement learning (RL). It is one of the three basic machine learning paradigms, along with supervised learning and unsupervised learning. Bae and Koumoutsakos (2022) adopted RL, initially used for flow control, to develop a wall model for zero-pressure gradient turbulent boundary layers, mainly channel and flat plates. RL is a different approach than standard ML and DL techniques. With RL, the neural network is trained in the wmLES environment (also called the production environment). The model is, therefore, not just a wall model but also compensates for the numerical errors. The inputs/outputs are nondimensionalized with the viscosity and the modeled friction velocity at the previous time step, and the reward is based on the existence of a log-layer near the wall. The model is then successfully tested on multiple channel flows and a flat plate. RL has then been adopted to develop new wall models for separated flows in Zhou *et al.* (2022). Instead of directly modeling the wall shear stress, which appears to have a serious drawback, according to the authors (i.e., a directional inconsistency between the predicted direction of the wall shear stress and the direction of velocity at a given height affects the size of the recirculation bubble), the model predicts the eddy-viscosity as the boundary condition. Their inputs are the instantaneous wall-parallel velocity, the wall-normal location of the agent[2], and the turbulence strain rate, which are nondimensionalized with $u_{\tau,p}$ proposed by Duprat *et al.* (2011). Their agents are trained on a low Reynolds number flow over the two-dimensional periodic hills and then applied to higher Reynolds numbers. A reasonable agreement with the wrLES results is observed. There remains the question of the ability of such a model to extrapolate to other flow configurations, such as a 3D diffuser.

3.4 Positioning of the present research work

The present research aims at exploiting the approximation capabilities of deep neural networks (DNNs), described first by Hornik (1991), to establish a more general model for the complex relationship between instantaneous flow fields, geometrical parameters, and wall shear stress using DNS or wrLES databases. In the data-driven wall models reviewed, the statistical notion of the turbulent wall shear stress is never addressed. All these WSS models are treated as a regression problem using the Mean Square Error (MSE) loss. However, most of the NNs trained with the MSE makes the hypothesis that the underlying distribution is a Gaussian with a fixed variance. Consequently, many authors find that the prediction has less variance than the actual or filtered DNS values. In our work, we aim to account for the statistics of the wall shear stress by predicting the probability distribution instead of a point estimate. This statistical notion is even more crucial for separated flows, as stated by Zhou *et al.* (2022). In the separation vicinity, the wall shear stress has a large variance, which is rarely captured by standard neural networks. For this purpose, a Mixture Density Network (MDN) is trained to predict K Gaussian distributions, each characterized by a mean and a standard deviation. To the authors' knowledge, this work is the first attempt at creating a statistics-based wall shear stress model.

[2] An agent is the learner and the decision maker of a reinforcement learning model. It intercats with its environement and takes actions based on a current state and reward.

3.5 Wall shear stress model requirements

The targeted application is wall shear stress modeling, where the predicted wall shear stress is imposed as a boundary condition. A wall model aims to model the near-wall physics to reduce the computational cost of high Reynolds number configurations. Therefore, the wall model should be inexpensive to evaluate and, at the same time, easy to implement in a flow solver. The implementation in the flow solver Argo-DG, developed at Cenaero, is further discussed in Section 5.3.2. The selection of the neural network and the input features is dictated by the physics and design constraints (or limitations) prescribed for developing new wall models in high-order flow solvers.

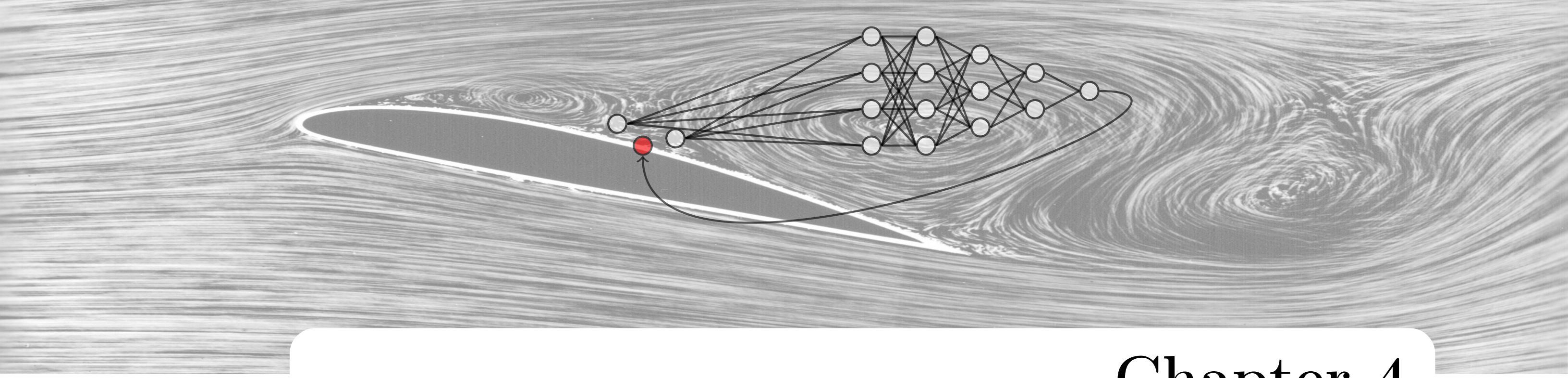
The **first constraint** is to create an instantaneous model, where the prediction of the wall shear stress at time t solely relies on the flow quantities extracted in the computational domain at the exact same time step t. This restriction eliminates the need for storing flow history, which can be expensive (in high-order flow solvers). This constraint could be relaxed to a few time steps to allow for time correlation.

The **second constraint** is to produce a model that is local, meaning it should only use the immediate surroundings of the prediction point to infer the wall shear stress. When using neural networks, this constraint is restrictive. Indeed, in a brute force approach, the network can be fed with all the data available at time t to infer the wall shear stress. However, this approach would require a large amount of communication between partitions (i.e., MPI ranks) and large storage in highly scalable flow solvers. Therefore, there is a balance to find between the implementation constraints and the model accuracy.

The model should perform local predictions. This **third constraint** is directly related to the previous one. One might ask why the wall shear stress is not predicted globally, i.e., on the entire solid wall. Firstly, this approach is not generalizable to other configurations. Secondly, it will require many communications to send the predictions to each quadrature point (see Section 5.3.2). The wall shear stress is not predicted on a small batch because we will face some discontinuities at the interfaces. Therefore, the wall shear stress is predicted at each position on the surface.

The **fourth constraint** is related to the wall model robustness. Most wall models predict the magnitude of the wall shear stress and then align it with the velocity extracted at the matching location (h_{wm}). To avoid this manipulation, the model should explicitly predict the two wall-parallel components of the wall shear stress. This last constraint is of importance for skewed and separated boundary layers.

The **fifth constraint** is that the model must be independent of the matching location or the wall-model height. The wall model should be also invariant to the coordinate system, rotationally invariant, Mach number invariant, and Galilean invariant, as stated in the work of Dupuy *et al.* (2023a). The Mach number invariance has not been tested as the test cases have a low Mach number and are nearly incompressible.



Chapter 4
Analysis of space-time correlations in representative turbulent flow configurations

This chapter is based on the paper *"Analysis of space-time correlation to support the development of wall-modeled LES"* published in Flow, Turbulence and Combustion (Boxho *et al.*, 2022).

In this chapter, the generation of databases using a high-order flow solver (Section 5.3.1) and its analysis using space-time correlations (Section 4.1.1) for new wall model development assisted by neural networks are discussed. The preprocessing of this database, such as cleaning and non-dimensionalization are not presented in the present chapter. These steps are covered in detail in Chapter 5.

Database generation is critical to correctly train the data-driven WSS model and will determine its capability to discriminate between different flow physics. The difficulty is to identify which test cases are the most representative of the complexity of a fluid flow, which may include a laminar or turbulent boundary layer, transition, shock, and separation. The long-term goal may be to develop a model that can discriminate between a wide variety of flow behaviors. As a starting point, this thesis focuses on wall models for the fully turbulent separation phenomenon. For this purpose, the database is composed of two test cases, which include both turbulent boundary layers (at equilibrium and subjected to moderate adverse pressure gradient) and separated regions. These test cases are the turbulent channel flow and the two-dimensional periodic hill. The former is analyzed in Section 4.2, and the latter is studied in Section 4.3.

In Sections 4.2.3 and 4.3.4, the analytical wall shear stress model based on the Reichardt's velocity profile (which is abreviated as AWSSR) is evaluated on the instantaneous wrLES data of both cases to observe the deviation from the *true* wall shear stress. This study aims to investigate the existing weaknesses of a specific analytical WSS model and derive guidelines for new wall models. This particular WSS model (i.e., AWSSR) is selected for comparison because it was implemented in the high-order flow solver Argo-DG during the thesis of Frère (2018).

4.1 Method and parameters

As mentioned, the wall models for separated flows using deep neural networks should be instantaneous and local. To enforce locality, feature selection (described in Section 2.3) is applied to reduce the problem dimensionality and remove irrelevant and redundant features from the dataset. In the supervised methods, the filter feature selection methods are considered. Although they use univariate statistical measures that do not consider the interaction between the input features, they are fast and can provide insights into the physical problem under study. The idea is to select input/output pairs that are (highly) correlated to ensure that the model will detect a possible non-linear relationship during the training process. Therefore, two types of correlations are considered: Pearson correlation and distance correlation, and evaluated on two test cases: a turbulent channel flow and the two-dimensional periodic hill. It is important to recall that correlation is not causality. This approach may seem counter-intuitive because, in fluid dynamics, we are sometimes more interested in causality than in correlation.

The correlations evaluated in the two test cases aim to identify the size of the input stencil. Although analytical WSS models can account for a streamwise displacement, most of them extract the velocity at a single wall-modeled height. Therefore, the extension of this idea to more points in space and time is investigated for other flow physics (i.e., separated flow). The two correlation formula are presented in Section 4.1.1, while Section 4.1.2 introduces the parameters of each test case.

4.1.1 Space-time correlations

Although turbulent flows are described by deterministic equations and often have statistically stationary solutions, they appear to be highly disorganized and unpredictable in their exact behavior, as discussed in Section 2.1. Consequently, any turbulent flow quantity can be decomposed into a mean and fluctuations that (may) vary in space and time. The flow behavior can be further characterized by searching for relationships (i.e., linear and non-linear) between the velocity, the pressure gradient, and the wall shear stress in space and time. Moreover, this analysis of space-time correlations will help to define the input of our data-driven wall model. These correlations are initially examined on a well-known geometry: a turbulent channel flow. The analysis is then extended to a more complex flow physics involving separation and reattachment from a curved wall: the two-dimensional periodic hill. This geometry is divided into the upper and lower solid wall. The lower wall presents a massive separation that generates a free shear layer that reattaches further downstream. This free-shear layer creates a contraction in the domain which affects the pressure gradient on the upper wall as well as the skin friction.

The standard Pearson correlation (Edelmann *et al.*, 2021) is the most popular correlation coefficient. A correlation coefficient is a single number that condensates the strength of the dependence. Pearson correlation measures the strength of linear dependencies between random variables. If two random variables are independent, the correlation coefficient is zero. However, the reverse implication is false. This drawback can be avoided by using alternative correlation coefficients, such as the distance correlation coefficient, proposed by Székely *et al.* (2007).

Pearson correlation. Pearson correlation is extended to handle space shift $\delta\xi$ (see Figure 4.13) and time delay δt as follows

$$R(\delta t, \delta\xi) = \frac{\mathbb{E}_{t,z}\left[u(\mathbf{x} + (\delta\xi)\hat{\boldsymbol{e}}_\xi, t_0 + \delta t)\tau_w(\mathbf{x}, t_0)\right]}{\sqrt{\mathbb{E}_{t,z}\left[u^2(\mathbf{x}, t_0)\right]}\sqrt{\mathbb{E}_{t,z}\left[\tau_w^2(\mathbf{x}, t_0)\right]}}, \tag{4.1}$$

where $\mathbf{x}$ is the tuple of coordinates (ξ, η, z) (as defined in Figure 4.13) and $\hat{\boldsymbol{e}}_\xi$ is the unit vector in the ξ-direction. The fluctuations of tangential velocity and wall shear stress taken in the wall parallel direction are u and τ_w, respectively. In both cases, the spanwise direction is periodic and homogeneous. Moreover, the data are extracted once a statistically converged simulation is obtained. Therefore, the correlations can be accumulated along the spanwise direction and in time. Note that the forcing term used to drive the simulation is extracted from the pressure gradients. In the remainder of this chapter, the velocity (or pressure gradients) will refer to the spatial variations of the temporal fluctuations of the velocity (or the spatial variations of the temporal fluctuations of pressure gradient).

Distance correlation. To account for space displacement and time delay, a similar exercise is applied to the sample distance correlation. Assuming that $X = u(\mathbf{x} +$

$\delta\xi, t_0{+}\delta t)$, $Y = \tau_w(\mathbf{x}, t_0)$ are two realizations of size n, the sample distance covariance measured at $(\delta\xi, \delta t)$ is defined as

$$\mathrm{dCor}(X,Y) = \frac{\mathrm{dCov}(X,Y)}{\sqrt{\mathrm{dVar}^2(X)\,\mathrm{dVar}^2(Y)}}, \tag{4.2}$$

where $\mathrm{dCov}(X,Y)$ is the distance covariance between X and Y computed as

$$\mathrm{dCov}_n^2(X,Y) = \frac{1}{n^2}\sum_{j=1}^{n}\sum_{k=1}^{n} A_{j,k}B_{j,k}\ . \tag{4.3}$$

In this last expression, $A_{j,k}$ and $B_{j,k}$ are two matrices of size $n \times n$. The matrix $A_{j,k}$ is computed as $A_{j,k} := a_{j,k} - \overline{a}_{j.} - \overline{a}_{.k} + \overline{a}_{..}$ where $\overline{a}_{j.}$ is the j^{th} row mean, $\overline{a}_{.k}$ is the k^{th} column mean and $\overline{a}_{..}$ is the grand mean of the distance matrix. $a_{j,k} = \|X_j - X_k\|, \quad j,k = 1,2,\cdots,n$ is the distance matrix of X, where $\|.\|$ denotes the Euclidean norm. The matrices $b_{j,k}$ and $B_{j,k}$ are evaluated in the same way by replacing X with Y in the above definitions. Finally, the sample distance variance is defined as: $\mathrm{dVar}^2(X) := \mathrm{dCov}_n^2(X,X) = \frac{1}{n^2}\sum_{k,l} A_{k,l}^2$. The brute force implementation of the distance correlation has a quadratic complexity $\mathcal{O}(n^2)$. Chaudhuri and Hu (2019) proposed a reduced algorithm complexity of $\mathcal{O}\left(n\log(n)\right)$, which is implemented for this study.

4.1.2 Parameters

As mentioned above, the main objective is to detect high correlations between the wall shear stress, which is the output of our novel data-driven WSS model, and volume fields (e.g., velocity, and pressure gradients). Therefore, Pearson and distance correlations are evaluated on two distinct test cases: a turbulent channel at $Re_\tau = 950$ and the two solid walls of the periodic hill at $Re_b = 10,595$. Correlations in the streamwise (or spanwise periodic) wall-parallel direction provide two parameters for the establishment of a new data-driven wall model: a time delay δt and a space displacement $\delta\xi$ (or δz). These two parameters are scaled differently, depending on the test case. For the channel flow (Section 4.2), the scaling is the known wall unit normalization, assimilated to the subscript +. For the periodic hill (Section 4.3), the scaling uses a length scale h (i.e., the hill height) and a velocity scale u_b (i.e., the bulk velocity). Table 4.1 lists all the possible correlations. Among all these correlations, only significant correlations are examined.

Table 4.1: List of all combinations (and notation used in the present chapter) of correlations between velocity (or pressure gradients) and wall shear stress, where, for instance, $U0T0$ stands for the correlation (distance or Pearson) between the velocity component u_ξ and the wall shear stress component $\tau_{w,\xi}$.

Wall shear stress	**Velocity**			**Pressure gradients**		
	u_ξ	u_η	u_z	$\frac{\partial p}{\partial \xi}$	$\frac{\partial p}{\partial \eta}$	$\frac{\partial p}{\partial z}$
$\tau_{w,\xi}$	$U0T0$	$U1T0$	$U2T0$	$P0T0$	$P1T0$	$P2T0$
$\tau_{w,z}$	$U0T2$	$U1T2$	$U2T2$	$P0T2$	$P1T2$	$P2T2$

Unless otherwise stated, contours at 85% are drawn as thin black lines to highlight the domain of high correlation, noted $\mathcal{D}$, for fair comparison between the channel and the periodic hill. White dashed lines in correlation maps are drawn to highlight the presence of anti-correlations. These contours are defined as:

$$\mathcal{C}(f(\delta t, \delta\xi)) = \alpha \max\left(|f(\delta t, \delta\xi)|\right) \quad \text{where } f(\delta t, \delta\xi) = \begin{cases} R(\delta t, \delta\xi) \\ \mathrm{dCov}(\delta t, \delta\xi)\,, \end{cases} \tag{4.4}$$

where $\alpha = 0.85$, R, and dCov are given by the equations 4.1, and 4.2, respectively.

4.2 Channel flow configuration

The space-time correlations are first evaluated on a turbulent channel flow at $Re_\tau = 950$ (TC950). The geometry and the flow statistics acquired with Argo-DG are presented in Section 4.2.1. Instantaneous snapshots of velocity, pressure gradients, and wall shear stress are interpolated at regular time intervals on structured probes. These data are then used to evaluate Pearson and distance correlations (Section 4.2.2) for different time delays and space shifts.

Since the boundary layers of a channel flow are fully turbulent, attached, and at equilibrium, existing wall models work well. In many wall models (Schumann, 1975; Grötzbach, 1987), the prediction of $\boldsymbol{\tau}_w$ is performed with the velocity extracted above the prediction point at a given wall modeled height. In the work of Piomelli *et al.* (1989), the authors accounted for the correlation by extracting the velocity downstream along a given structure. Therefore, as a preliminary consideration, the correlations should have a slight space-time lag, which may depend on the height at which the correlation is evaluated.

The assumptions about the channel boundary layer (i.e., fully turbulent, attached, and at equilibrium) are valid on average but may be invalid instantaneously. The AWSSR is evaluated *a priori* on the channel at different heights, and the predictions are compared with the wrLES wall shear stress in Section 4.2.3. The variance of the wrLES $\boldsymbol{\tau}_w$ is expected to be higher than the variance of the predictions due to local instability and instantaneous effects.

4.2.1 Flow statistics

A channel flow can be seen as two infinite plates separated by a height 2δ (Hoyas and Jimenez, 2008), where δ is the half height or the boundary layer thickness. These infinite plates are numerically encoded as periodic and homogeneous in the streamwise and spanwise directions. The domain sizes in these directions are $L_x/\delta = 2\pi$ and $L_z/\delta = \pi$, respectively. A uniform pressure gradient (i.e., the forcing term) drives the flow. For fair comparisons with the incompressible flow references, the Mach number is set to a low value of $M = 0.1$. The mesh resolution is comparable to a wrLES: $\Delta x^+ \simeq 90$ and $\Delta z^+ \simeq 46$, where the superscript $+$ denotes the wall unit normalization. The effective resolution is $\Delta x^+ \simeq 30$ and $\Delta z^+ \simeq 15$ for Lagrange polynomials of order $p = 3$. The first grid cell has an effective resolution of $\Delta y^+ \simeq 1$ on average. To move forward in time, an implicit integration scheme is used to overcome the restrictive acoustic-induced CFL condition in the near-wall region, where a very small y^+ is required to get a good wrLES. The convective CFL is kept at about 0.3 to ensure the proper resolution of the turbulence-related time structures. The statistics have been accumulated for about $13,870\, t^+$, after evacuating the numerical transient, where t^+ is defined as $t\, u_\tau^2/\nu$. This dimensionless time corresponds to ~ 46 flow-through times (t_c), defined as $t\, u_b/L_x$, where u_b is the bulk velocity.

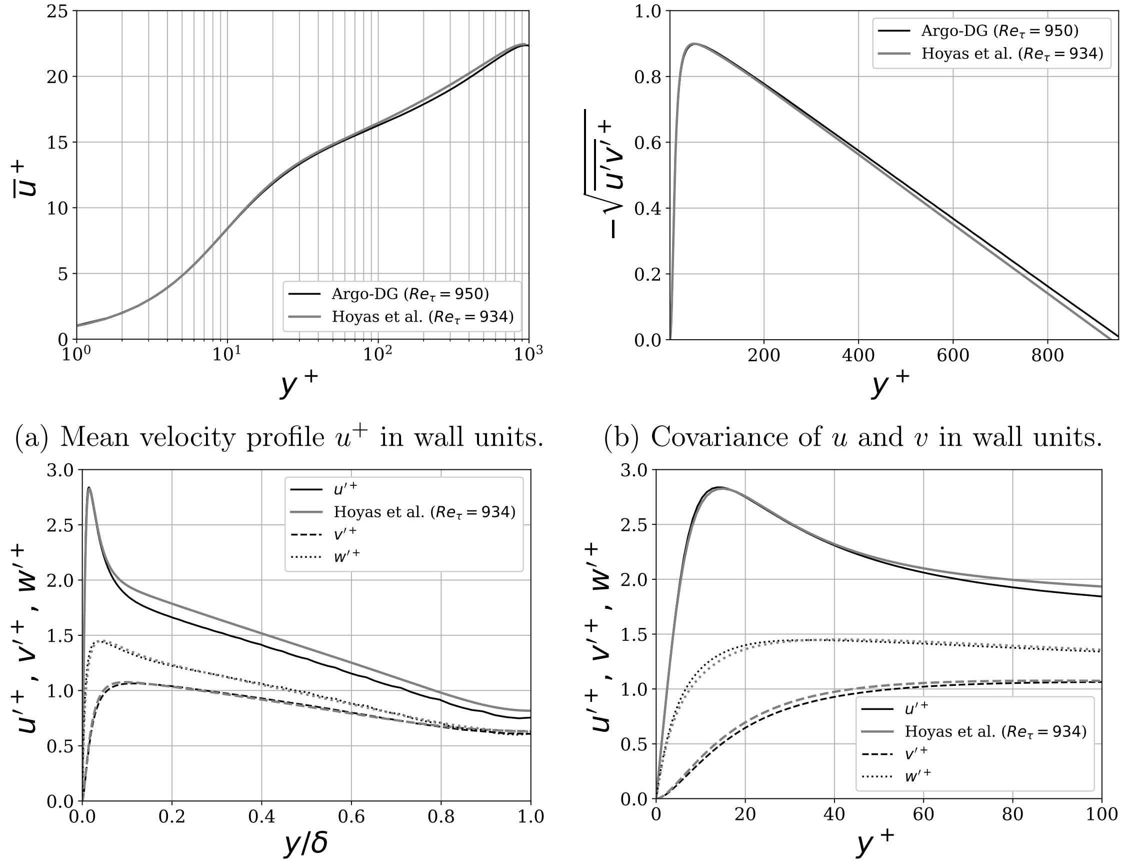
Figure 4.1 shows the mean velocity profile and Reynolds stresses in wall units. The mean velocity profile is almost perfectly superimposed on the DNS reference of Hoyas and Jimenez (2008) (in gray), with a good agreement in the near-wall region (Figure 4.1a). The covariance between u and v (Figure 4.1b) is a straight line, as expected for wall-bounded flows. However, the two curves diverge slightly for larger y^+ because the two frictional Reynolds numbers are not identical. The profile obtained with Argo-DG (in black) crosses the horizontal axis at $y^+ \simeq 957$. Concerning the RMS velocity profiles, a fair agreement with the DNS reference of Hoyas and Jimenez (2008) is obtained in the near-wall region (Figure 4.1d) while in the center of the channel, the fluctuations u'^+ are lower than the reference (Figure 4.1c).

4.2.2 Space-time correlations

The analysis is divided into the streamwise and spanwise directions. First, Pearson and distance correlations are evaluated along the streamwise direction. Although this direction is homogeneous and periodic, it is aligned with the convection velocity. The correlation is expected to be non-symmetric in the $(\delta\xi, \delta t)$-frame. Secondly, the correlations are calculated in the spanwise direction. On average, there is no convection in this direction. Therefore, the correlation should be symmetric.

4.2.2.1 Pearson and distance correlations in the streamwise direction

The channel at $Re_\tau = 950$ has almost no log-layer. According to Marusic *et al.* (2013), the log-layer bounds are $y^+ = 92.5$ and $y^+ = 142.5$. Therefore, the correlations are computed at a wall-normal distance of $y^+ \simeq 100$, which is still close enough to the wall to feel its interaction but large enough to be in the logarithmic layer. The convergence of the statistics is accelerated by averaging multiple realizations of the correlations in the streamwise and spanwise directions.



(a) Mean velocity profile u^+ in wall units.

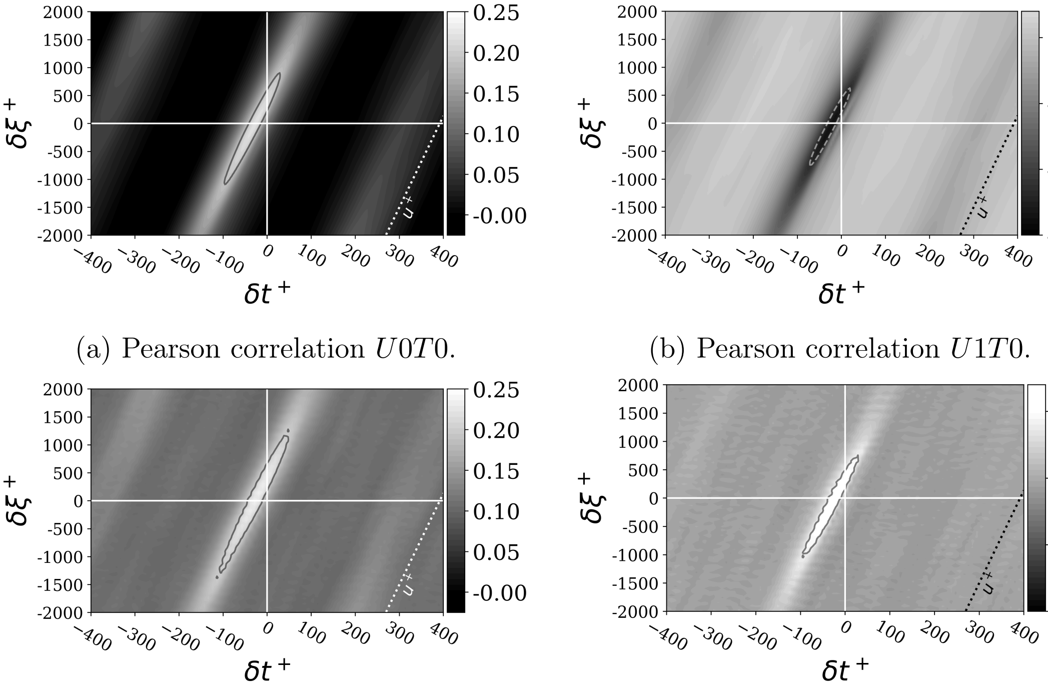
(b) Covariance of u and v in wall units.

(c) RMS velocity profiles in global coord.

(d) Velocity RMS profiles in wall units.

Figure 4.1: Standard flow statistics of the channel flow at $Re_\tau = 950$, compared to the results of Hoyas and Jimenez (2008).

For a channel flow, the most significant correlations are $U0T0$ and $U1T0$ (as defined in Table 4.1). No correlation has been detected between one of the component of the velocity field and the spanwise wall-shear stress. Besides, no relation has been found between the pressure gradients and the wall shear stress at $y^+ = 100$. The work of Abel *et al.* (2006) confirms this observation. He only found a correlation between the pressure gradient and the wall shear stress in the viscous sublayer. This correlation quickly decreases with the increase of the wall distance. Consequently, it is mainly the velocity field that triggers the wall shear stress.



(a) Pearson correlation $U0T0$.

(b) Pearson correlation $U1T0$.

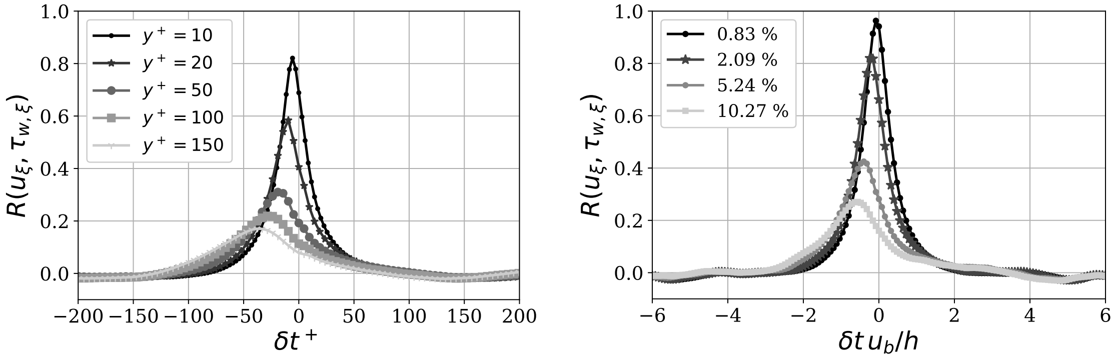
(c) Distance correlation $U0T0$.

(d) Distance correlation $U1T0$.

Figure 4.2: Space-time correlations in the streamwise direction evaluated on the channel at $Re_\tau = 950$ at a wall-normal distance of $y^+ = 100$, the local averaged streamwise velocity u^+ is drawn in dotted line.

Correlation $\boldsymbol{U0T0}$. At zero displacement, the correlation shows a time delay of $-30\delta t^+$. Due to convection, this delay corresponds to a dimensionless downstream displacement of about 450. This result is consistent with experiments which give $\delta\xi^+ = |y^+|cot(13°) \simeq 433$ for large distances from the wall (Piomelli and Balaras, 2002). The correlation aligns with the mean streamwise velocity u^+, which corresponds to the local convection (see Figure 4.2a). A similar domain of high correlation $\mathcal{D}$ is obtained with the distance correlation (see Figure 4.2c). Its low amplitude may indicate a non-linear relation between u_ξ and $\tau_{w,\xi}$. To conclude, if local information is used, a time delay needs to be considered for u_ξ. While, if instantaneous information is used, a space displacement has to be considered due to the streamwise convection of the near-wall structures. The correlation has some repetition (or periodicity) in time. The spacing between two peaks is approximately $320\delta t^+$, which corresponds to one flow-through time ($\sim 1.04t_c$). This pattern is probably due to the imposed periodicity in the streamwise direction. This repetition is also visible for Pearson and distance correlation and the correlation $U1T0$.

Correlation $\boldsymbol{U1T0}$. An anti-correlation between $\tau_{w,\xi}$ and u_η is found in Figure 4.2b. When the wall-normal velocity increases, the wall shear stress decreases and the friction reduces. This observation is consistent with flow ejections acting on the wall by reducing the wall shear stress as explained by Piomelli *et al.* (1989) in their ejection model. Since the distance correlation is based on a measurement of a distance in the L_2-norm (Eq. 4.2), it does not indicate an anticorrelation. Nonetheless, a similar domain of high correlation is obtained. The correlation is also shifted in time and space and aligns with the local convection velocity. This example shows how Pearson correlations and distance correlations complement each other. With these observations, similar wall models' best practices are drawn.



(a) For the channel at $Re_\tau = 950$, average over every streamwise positions.

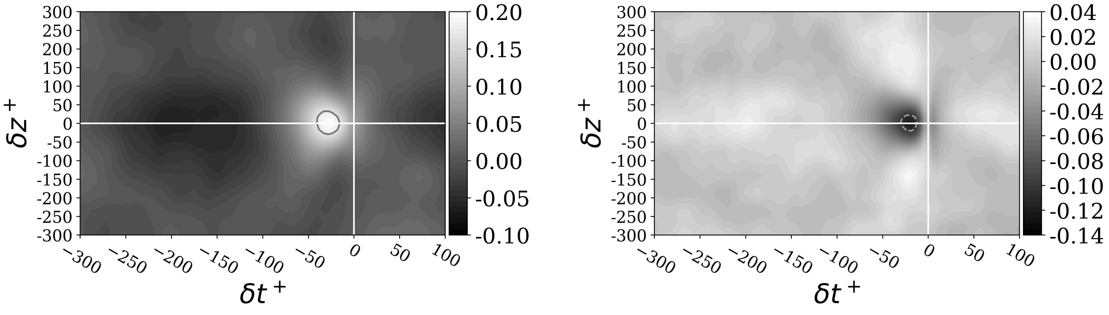
(b) For the periodic hill at $Re_b = 10,595$ at $x/h \simeq 9.0$ on the upper solid wall.

Figure 4.3: Evolution of the time correlation (for $\delta\xi = 0$) as going away from the solid wall (a) in wall units and (b) in % of the hill height h.

Wall-normal evolution of $\boldsymbol{U0T0}$. The correlation $U0T0$ shows a time delay at $y^+ = 100$ but how this delay is evolving as moving closer to the wall. Figure 4.3a shows the wall-normal evolution of $U0T0$ without any streamwise displacement (i.e., $\delta\xi = 0$). Close to the wall (i.e., at the beginning of the buffer layer), the two variables are highly correlated with almost no time delay $\delta t^+ \simeq 0$. With increasing distance from the wall, the correlation decreases in amplitude and the peak is shifted in $\delta t < 0$. According to Colella and Keith (2003), the time delay increases with distance from the wall due to the existence of an angle of inclination of structures convected along the wall. The decay of these convected structures is at the origin of the amplitude decrease. Figure 4.3b shows the same graph but for the upper wall of the periodic hill. Although this test case has not yet been presented, a similar behavior is observed. The correlation $U0T0$ decreases in amplitude and is shifted in $\delta t < 0$ as moving away from a wall that is subjected to a moderate pressure gradient (see Section 4.3.2).

4.2.2.2 Pearson correlations in the spanwise direction

In the spanwise direction, the two significant correlations at $y^+ = 100$ are also $U0T0$ and $U1T0$. Other correlations such as $U2T0$, $U0T2$, and $U1T2$ are visible at lower y^+ values (not shown here for brevity) and become insignificant at the current wall-normal height. These cross correlation are symmetric around $\delta z = 0$ with two lobes (one positive and one negative).



(a) Pearson correlation $U0T0$.

(b) Pearson correlation $U1T0$.

Figure 4.4: Space-time correlations in the spanwise direction evaluated on the channel at $Re_\tau = 950$ for a wall-normal distance of $y^+ = 100$.

Correlations $\boldsymbol{U0T0}$ and $\boldsymbol{U1T0}$. The spanwise wall shear stress $\tau_{w,z}$ is zero on average since there is neither a skewed boundary layer nor a spanwise pressure gradient in the channel flow. Therefore, the domain $\mathcal{D}$ in Figure 4.4a is narrow and symmetric w.r.t. $\delta_z^+ = 0$. The correlation map $U0T0$ extends over twenty wall units in space and shows a time delay equivalent to that observed in the streamwise correlations. The correlation map $U1T0$ shown in Figure 4.4b has the same characteristics as $U0T0$. However, an anticorrelation is observed as for the streamwise correlation $U1T0$.

4.2.2.3 Correlations on the channel: conclusion

Analysis of space-time correlations has revealed that the streamwise velocity has a strong impact on the wall shear stress, while the wall-normal velocity is anticorrelated with $\tau_{w,\xi}$ as explained by ejection/sweep theory. Both the streamwise and spanwise correlations present a time delay ($\delta t^+ < 0$) that increases with distance from the wall. This observation is consistent with the increase in the mean velocity, when moving away from the wall, that convects structures across the domain. This phenomenon promotes the inclination of structures as explained by Colella and Keith (2003) and earlier by Rajagopalan and Antonia (1979). The local mean velocity links the time delay and the space displacement, as shown in Figure 4.5. According to this observation, the wall model can remain local if a time delay is taken into account. This procedure holds if the numerical time step of the wall model δt_{wmLES} is smaller than the delay measured in these correlations. Otherwise, a downstream shift can compensate for the delay. This method has already been used in the shifted boundary condition model of Piomelli *et al.* (1989).

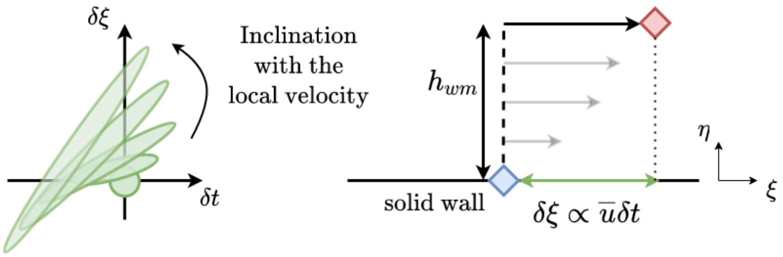
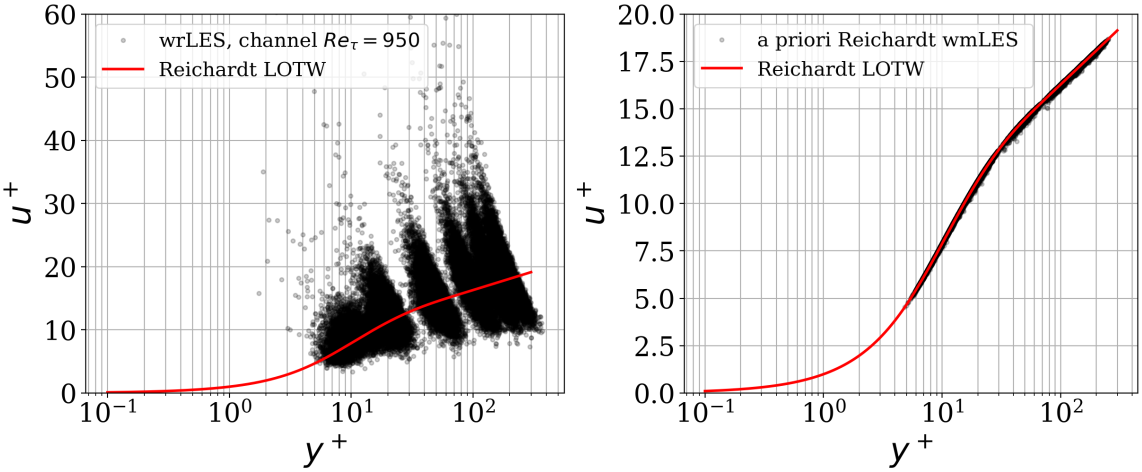

$\delta\xi$
δt
Inclination with the local velocity
h_{wm}
solid wall
$\delta\xi \propto \overline{u}\delta t$
η
ξ

Figure 4.5: Explanation of the inclination of the correlations and the time delay experienced when moving away from the wall.

4.2.3 Deviation from the WSS model based on Reichardt's velocity profile

In previous sections, the link between potential neural network entries and the wall shear stress is evaluated in space and time. On the one hand, this analysis gives information about the input stencil size: how many points in the streamwise (or spanwise) direction should be considered? On the other hand, it also provides information about which input/output pairs are relevant for the network. As mentioned, feature selection using filter methods (based on univariate statistical measures) is performed on a turbulent channel flow. If the distance correlation does not reveal any relationship, the network will have some difficulties finding one either.

Another interesting analysis is to observe how *good* the *a priori* predictions of a quasi-analytical WSS model are, compared to the exact wrLES wall shear stress. For this analysis, the WSS model based on Reichardt's velocity profile (AWSSR) is considered. This model and its implementation in Argo-DG have been validated during the thesis of Frère (2018). On average, the model can retrieve the mean velocity profile and the Reynolds stresses. The quality of the predictions is not only evaluated based on averages but also on their distribution and statistical moments, such as variance and skewness.



(a) wrLES.

(b) *a priori* Reichardt wmLES.

Figure 4.6: Distribution of the instantaneous scaled streamwise velocity u^+ as a function of the scaled distance to the wall y^+ for TC950.

This section assesses the deviation between the wall shear stress predicted by the AWSSR and that obtained from a wrLES. Figure 4.6 shows the distribution of the instantaneous scaled streamwise velocity u^+ as a function of the scaled wall-normal distance y^+ for the turbulent channel at $Re_\tau = 950$ (TC950). The instantaneous wrLES data (Fig. 4.6a) are scattered around Reichardt's profile indicating that the equilibrium assumption does not hold instantaneously. Figure 4.6b is obtained by applying the AWSSR *a priori* based on the available wrLES data at six wall-normal heights. The wall shear stress prediction can be seen as the projection of the instantaneous data onto Reichardt's profile. The second (i.e., the standard deviation) and third (i.e., the skewness) statistical moments are evaluated for the reference data and the corresponding predictions made at $h^+_{wm} = 100$, which is within the limits of the logarithmic layer according to Marusic *et al.* (2013) for $Re_\tau = 950$. The standard deviation of $\tau_{w,wrLES}$ is 0.42 while the standard deviation predicted by Reichardt drops to 0.19. Due to convection along the streamwise direction, $\tau_{w,wrLES}$ has a positive skewness of 1.02. However, the predictions exhibit no skewness at that wall-normal height. By extracting data closer to the wall at $h^+_{wm} = 10$, the skewness of the prediction rises to a value of 0.52.

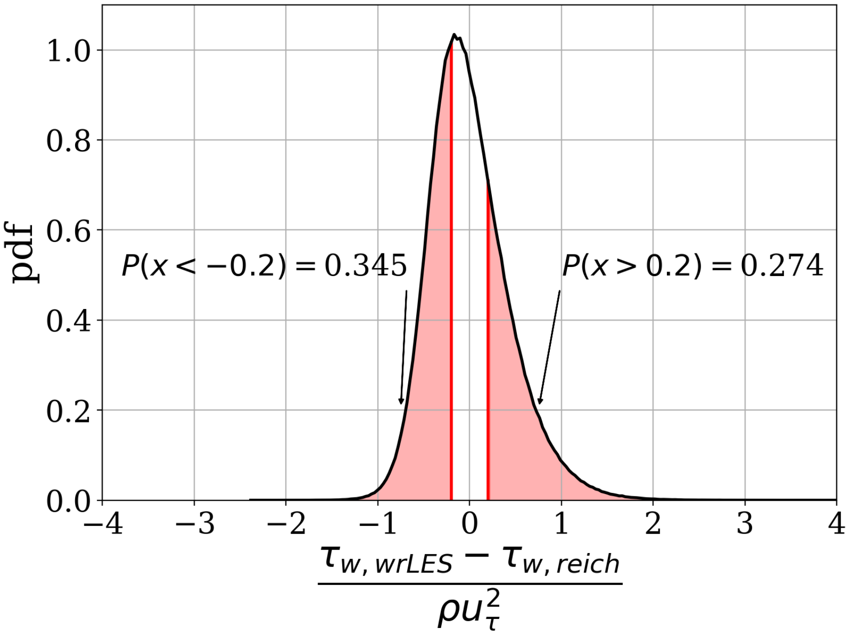


Figure 4.7: Non-dimensional deviation between the wrLES and predicted wall shear stress using a Reichardt model for TC950, where u_τ is the mean friction velocity.

Figure 4.7 shows the deviation between the wrLES wall shear stress and the pre-

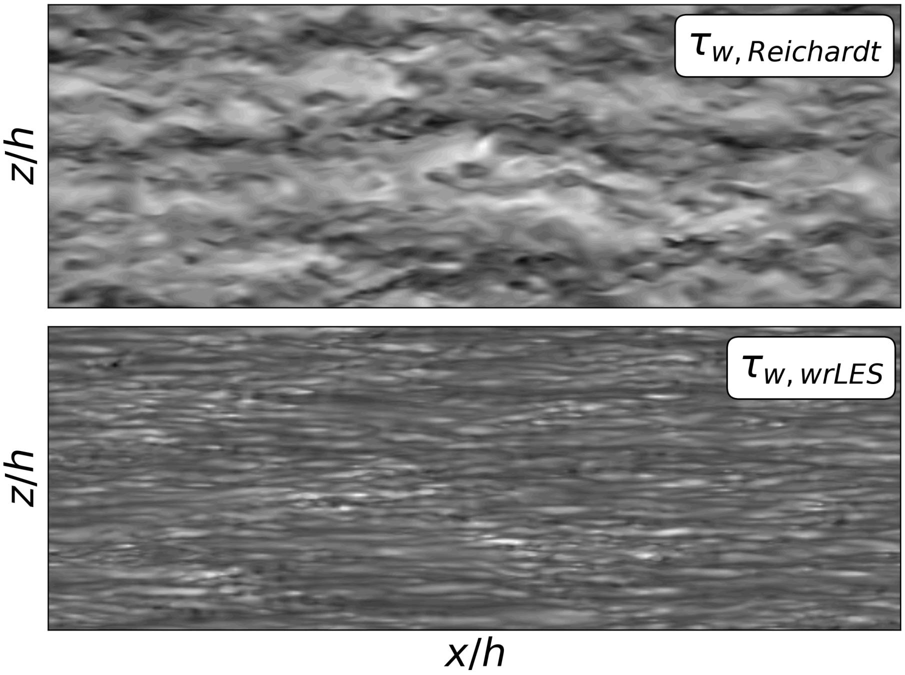


Figure 4.8: Instantaneous *a priori* prediction of the AWSSR for a velocity field extracted at $h^+_{wm} = 100$ compared to the wrLES WSS in the streamwise direction.

diction from the AWSSR. The shaded area indicates the probability of having a deviation greater than 0.2 and smaller than -0.2. These probabilities are 0.274 and 0.345, respectively. The latter probability is higher because of the non-negligible skewness of $\tau_{w,wrLES}$, which is not present in Reichardt's predictions. One would expect the skewness of the velocity field to be transferred to the predicted wall shear stress via the non-linear relationship. However, the velocity field at $y^+ = 100$ has a negative skewness of amplitude 0.2. The logarithm reduces the skewness even further. Such a non-linear transformation is used to reduce the skewness of a given random variable.

Figure 4.7 shows the distribution of the deviation, but says nothing about the spatial correlations of the predicted wall shear stress (i.e. the two-dimensional structures of the wall shear stress). An instantaneous *a priori* predicted wall shear stress is shown in Figure 4.8 and compared with the wrLES WSS at a given time step t. The wrLES WSS structures are long streaks stretched in the streamwise direction, whereas they are narrow in the spanwise direction. The predicted structures are much larger than the wrLES structures. The model transfers the two-dimensional structures of the velocity field to the predicted wall shear stress.

The AWSSR has proven its ability to retrieve the mean flow behavior for decades now. Nevertheless, the generated predictions do not have the statistical moments of the instantaneous wrLES data. This capability of retrieving the variance is even more important for separated flows as discussed in Section 4.3. As noted by Zhou *et al.* (2022), the imposition of the mean is not sufficient, for separated flow. For this reason, new wall models are developed using networks capable of predicting a distribution rather than pointwise estimates. These types of neural networks are called Mixture Density Networks (MDN) and are presented in Section 5.1.1.

4.3 Periodic hill flow configuration

The two-dimensional periodic hill can be divided into the upper wall (Section 4.3.2), which behaves similarly to a channel wall, except that it is subjected to a moderate pressure gradient, and the lower wall (Section 4.3.3), which is characterized by a massive separation from the curved wall followed by a reattachment to the flat lower part of the wall. Therefore, instead of having a single flow physics to analyze, two walls with different physics are available. Note, however, that the two walls are connected. Indeed, if the separation is not properly captured by the simulation, it will affect the free shear layer and the reattachment location. The former will modify the pressure gradient and thus, the friction on the upper wall.

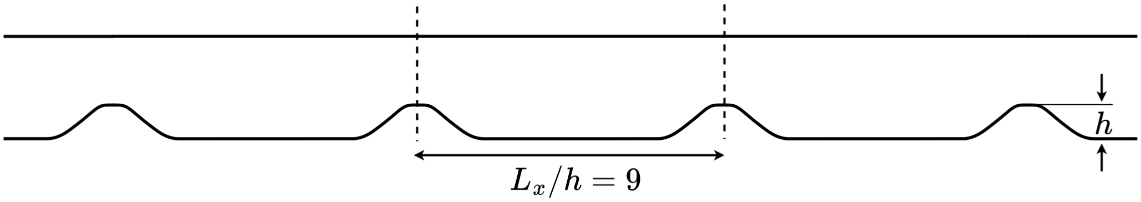


Figure 4.9: Two-dimensional periodic hill geometry illustrated as a succession of hills in the streamwise direction; the periodic length is L_x/h, where h is the hill height.

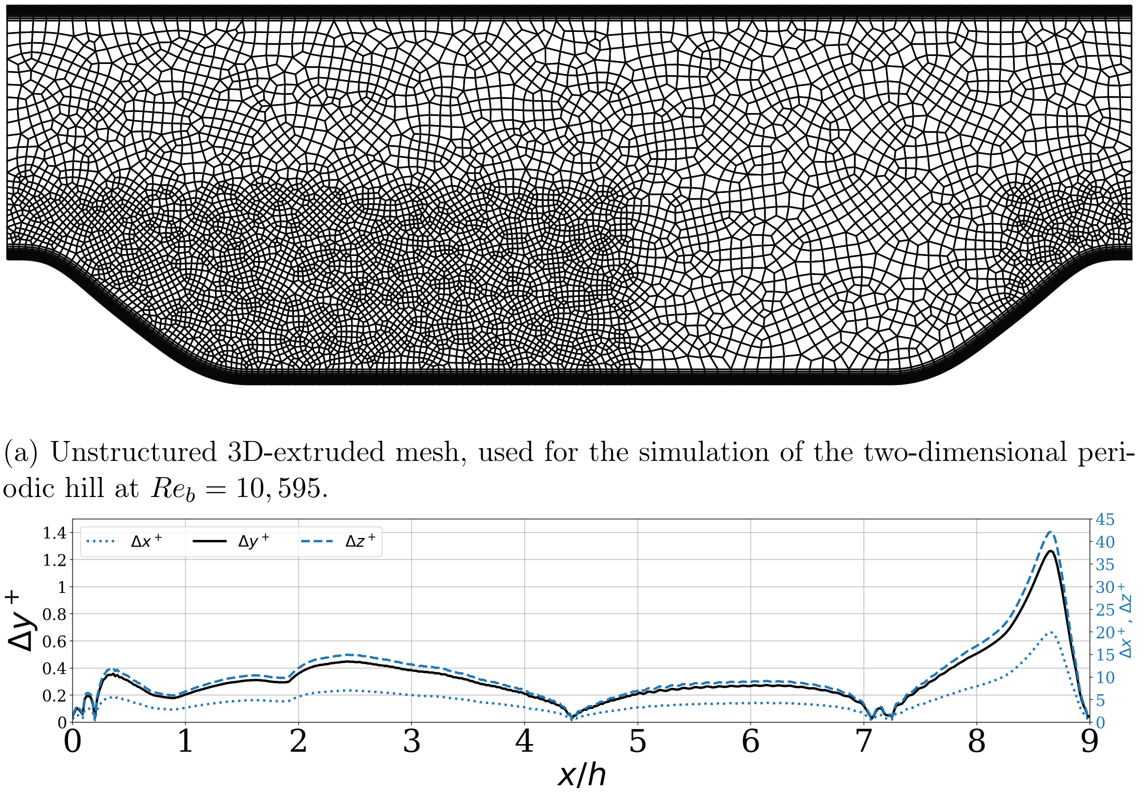
4.3.1 Flow statistics

Figure 4.9 shows the two-dimensional periodic hill geometry. This test case consists of a succession of streamwise constrictions separated by a distance $L_x/h = 9.0$, where h is the hill height. The numerical representation of this test case is the part from one hilltop to the next, marked by the dashed line in Figure 4.9. To numerically force the infinite number of hills, a periodic condition is imposed in the streamwise direction. Since the test case is three-dimensional, the spanwise direction is also a periodic direction, with a size fixed to $L_z/h = 4.50$. The periodic hill geometry has been carefully designed to allow the flow to separate from the hill crest, reattach to the flat lower surface, and reaccelerate before the next hill. The flow presents a massive recirculation bubble, bounded by the free shear layer generated by the separation. This test case is part of the ERCOFTAC KB wiki and has been extensively studied both experimentally by Song and Eaton (2004), and numerically (Gloerfelt and Cinnella, 2015; Breuer *et al.*, 2009). The reader is referred to Fröhlich *et al.* (2005) for a description of the 3D extruded geometry and a detailed discussion of the flow behavior.

The flow is simulated at a bulk Mach number of $M_b = 0.1$, and statistical data is accumulated over more than 38 flow-through times after evacuating the numerical transient for 45 flow-through times. A constant pressure gradient drives the flow, and its magnitude is controlled to match the bulk Reynolds number ($Re_b = \frac{u_b h}{\nu}$, where u_b is the bulk velocity) using the procedure proposed by Benocci and Pinelli (1990), with minor modifications introduced by Carton de Wiart *et al.* (2015) to account for compressibility effects.

The unstructured mesh shown in Figure 4.10a is composed of 445,005 hexahedra. A refinement is applied from the hilltop to the average reattachment location to better capture the separation and resolve the free shear layer. The separation is very sensitive to the mesh resolution and, as mentioned above, a misprediction of the separation will affect the entire domain. The near-wall region (representing 10% of the hill height) on the upper and lower walls is refined with a boundary layer structured mesh using a geometric progression to impose a first cell size of $y^+ = 1$. Figure 4.10b shows the averaged grid refinements in the streamwise, wall-normal, and spanwise directions. The simulation is performed with Lagrangian interpolants of order $p = 3$ to give a total of 28 million degrees of freedom.

Considering the bulk velocity u_b, the spatial resolution of the mesh near the separation, the polynomial order of the DG method, and the imposed time step, the convective CFL is kept at about 0.3, which ensures that turbulence-related time structures are properly resolved for the computation of space-time correlations.



(a) Unstructured 3D-extruded mesh, used for the simulation of the two-dimensional periodic hill at $Re_b = 10,595$.

(b) Averaged grid refinements $(\Delta x^+, \Delta y^+, \Delta z^+)$ in wall units computed from the averaged wall shear stress.

Figure 4.10: Information relative to the mesh refinement.

The standard statistical data (i.e., mean velocity profile, Reynolds stresses, and friction coefficient) are compared with Gloerfelt and Cinnella (2015) and Breuer *et al.* (2009). The friction coefficient on the lower wall, defined as $\tau_w/(0.5\rho u_b^2)$, is shown in Figure 4.11. The overall behavior is similar to both references. At the hill crest, a tiny precursory separation is observed just before the main separation. This pre-

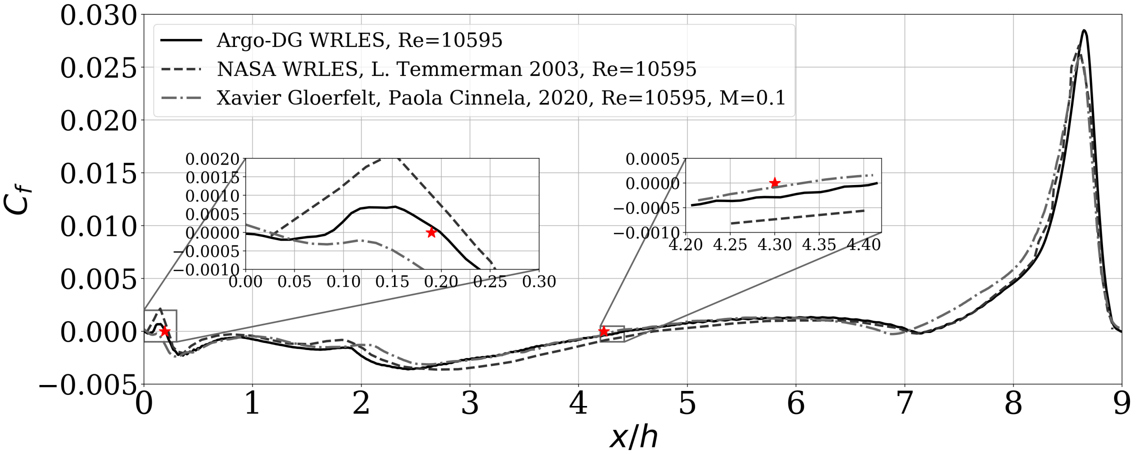


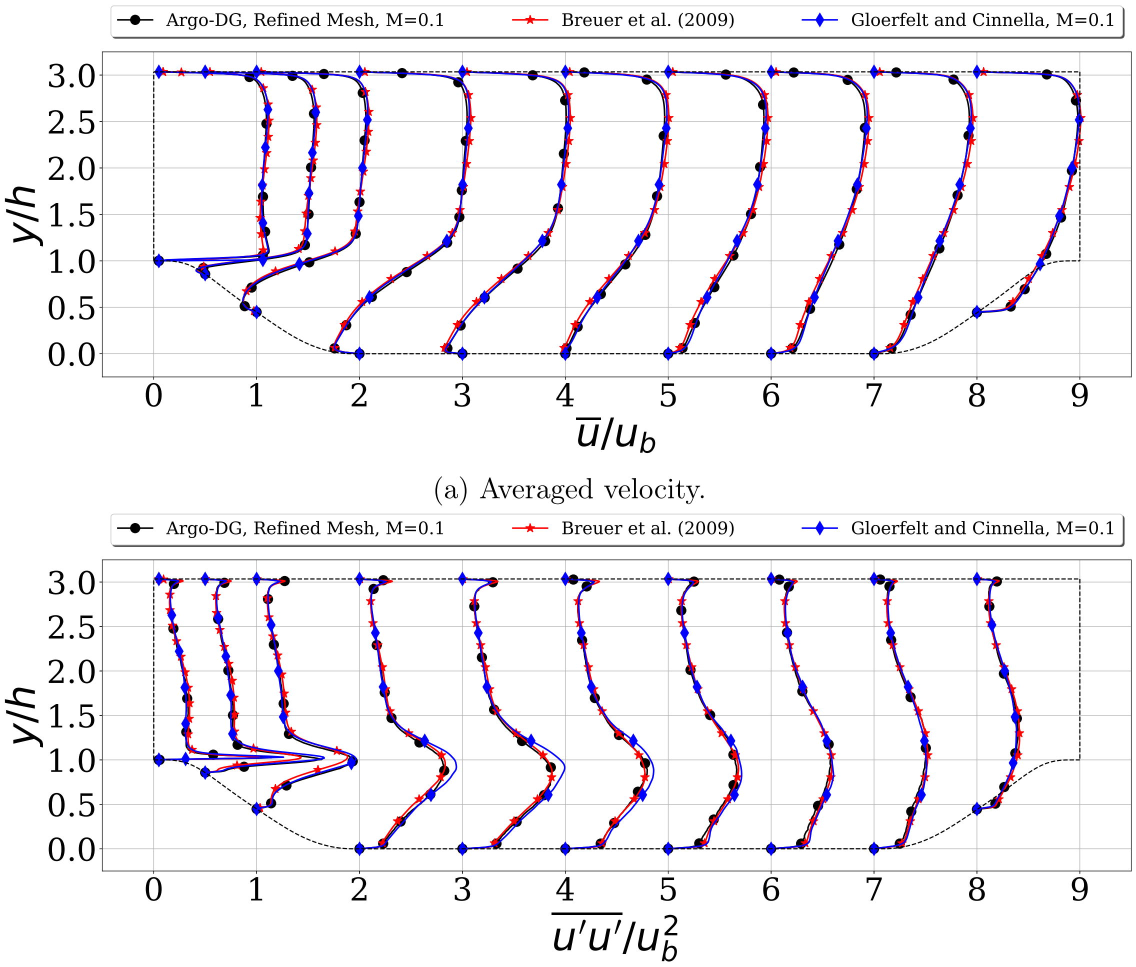
Figure 4.11: Friction coefficient C_f measured on the bottom surface of the periodic hill at $Re_b = 10,595$, compared to Gloerfelt and Cinnella (2015), and Temmerman *et al.* (2003); the red stars indicates the experimental averaged spearation and reattachment locations.

separation is explained by a high-pressure gradient region that emerges downstream of a low-pressure region located at $x/h \simeq 8.75$. Such a region produces eddies of high kinetic energy that are convected downstream and influence the separation process on the curved wall (Breuer *et al.*, 2007). No pre-separation is visible for Gloerfelt and Cinnella (2015) (i.e., the dash-dotted line), and the flow is already separated at $x/h \simeq 0.05$, while for Temmerman *et al.* (2003) (i.e., the dashed line), a large friction peak is noticed just before the main separation. The average position of the separation depends on the mesh and the flow solver, but also on the accumulation period due to the random location of the instantaneous separation point. After the separation ($x/h \simeq 0.2$), the friction coefficient is negative until the reattachment. Around $x/h \simeq 2.0$, the flow slows down and then accelerates again due to the change in curvature. The deceleration/acceleration process is slightly shifted for the dash-dotted C_f curve. Regarding the reattachment location, the three curves reattach at different locations: $x/h \simeq 4.40$, $x/h \simeq 4.75$, and $x/h \simeq 4.32$ for our wrLES, Breuer *et al.* (2009), and Gloerfelt and Cinnella (2015), respectively. It shows how sensitive the reattachment is to the numerical environment (i.e., flow solver, mesh, and turbulence models, to name but a few). After the reattachment, the flow recovers and starts to accelerate. At the windward base of the hill, the tree curves indicate a small separation around $x/h = 7.0$. Note that Gloerfelt and Cinnella (2015) predicts this second separation slightly upstream of the other two. This discrepancy with Gloerfelt and Cinnella (2015) is probably due to the geometric definition of the lower wall and, more specifically, the connection between the bottom flat part and the hill. This second separation is followed by a strong acceleration along the ascending part of the hill. Our friction peak is slightly shifted downstream compared to the two references.

When examining the mean velocity profiles and the Reynolds stresses, a good agreement with respect to the DNS references is observed (see Figure 4.12). Due to earlier reattachment in Breuer *et al.* (2009), a slight descrepancy is observed between our mean velocity profile and their at the lower wall between $x/h = 4.0$ and $x/h = 7.0$. In Figure 4.12b, the Reynolds stress component of Gloerfelt and Cinnella (2015) is slightly different from ours between $x/h = 2.0$ and $x/h = 5.0$. The peak of u' indicates the free shear layer center. Due to the latter reattachment of Gloerfelt and Cinnella (2015), the free shear layer that drives the reciculation bubble, carries more energy.

4.3.2 Analysis of space-time correlations on the top wall of the periodic hill

Although the upper wall of the periodic hill (PHU10595) geometry is similar to a channel wall in that it has no curvature, the streamwise direction is not homogeneous due to strong fluctuations in the pressure gradient caused by the complex dynamics occurring at the lower solid wall (i.e., separation, reattachment, unsteady free shear layer, recirculation bubble, turbulence recycling due to the periodicity assumption). Therefore, the correlations are computed for three regions: the separation, the reattachment, and the converging region (see blue diamonds in Figure 4.13). All correlations are computed at a wall-normal distance of $0.1h$ for the upper and lower walls. For the upper wall, such a wall-normal height corresponds to values between 50 and 70 in wall units. For the lower wall, it corresponds to values between 20 and 40 for $x/h \in [0, 8]$ and 100 along the converging part. Due to the curvature, each field is projected onto the local reference frame defined by the geometry. Therefore, for the rest of this chapter, curvilinear coordinates defined in Figure 4.13 are utilized.



(a) Averaged velocity.

(b) Reynolds stress component.

Figure 4.12: Flow statistics of the periodic hill at $Re_b = 10,595$, compared to Gloerfelt and Cinnella (2015), and Breuer *et al.* (2009).

4.3.2.1 Streamwise Pearson and distance correlations

The significant correlations between the velocity and τ_w are $U0T0$, $U1T0$, and $U2T2$. In this section, correlations with the pressure gradients are also examined. Among all possible correlations, only $P0T0$ and $P2T2$ stand out in our analysis. Even though the streamwise direction is not homogeneous, these correlations are similar in shape and amplitude at the three locations (see blue diamonds in Figure 4.13).

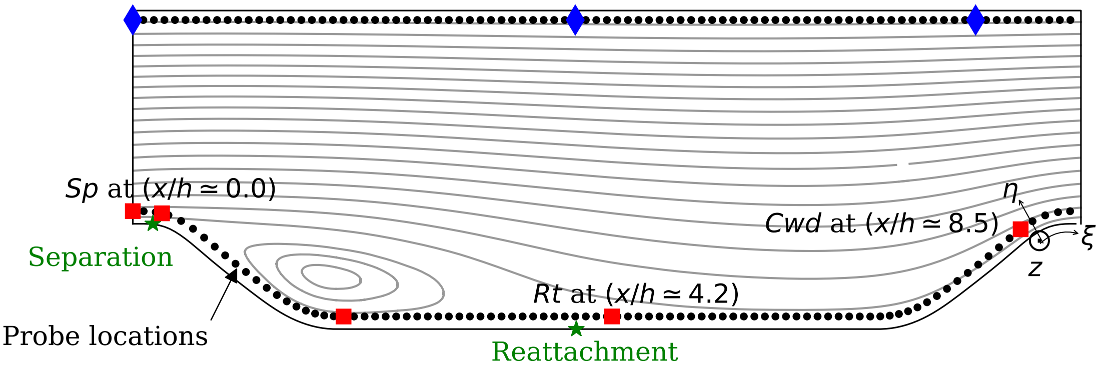
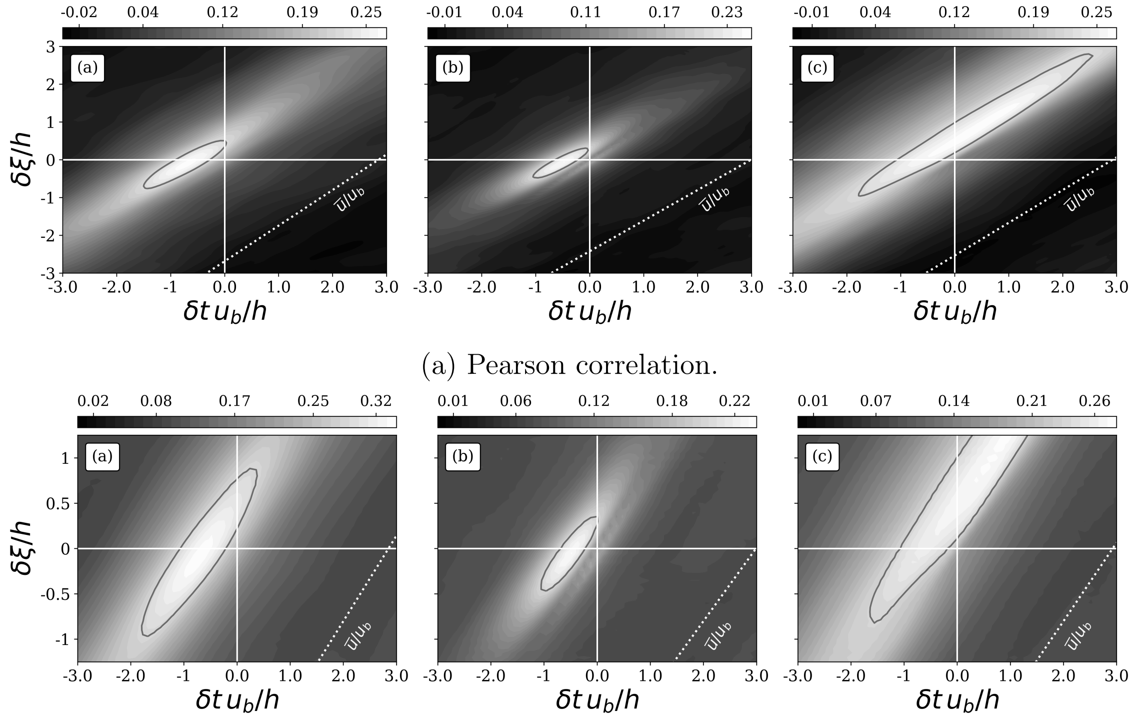


Figure 4.13: Probe positions at a given wall-normal height ($\eta/h \simeq 0.1$) (upper and lower wall) and location of the observation points along the lower wall (red square) and the upper wall (blue diamond).

Therefore, they are only presented for $x/h \simeq 0$. However, the amplitude is different at $x/h \simeq 4.20$, with a decrease in magnitude of 10 to 20%.

Correlations $\boldsymbol{U0T0}$, $\boldsymbol{U1T0}$, and $\boldsymbol{U2T2}$. Figure 4.14a and 4.14b show the Pearson and distance correlations $U0T0$, $U1T0$, and $U2T2$ (from left to right) at $x/h \simeq 0$. The amplitude of the three Pearson correlations is around 0.20-0.25. For distance correlation, $U1T0$ and $U2T2$ have approximately the same amplitude of about 0.22-0.26, whereas $U0T0$ has a higher amplitude, probably indicating a different probabilistic distribution. The domain of high correlation $\mathcal{D}$ is similar for each type of correlation. At each location, the correlation aligns with the local mean streamwise velocity $\overline{u}/u_b$ and is shifted by about $-0.5\delta t\, u_b/h$. While $U0T0$ and $U1T0$ remain in the left quadrant, $U2T2$ spreads over 2.5 time units in the upper right quadrant (Figure 4.14a(c)). This part of the correlation cannot be exploited due to causality.

Correlations $\boldsymbol{P0T0}$ and $\boldsymbol{P2T2}$ Pearson and distance correlations for $P0T0$ and $P2T2$ are shown in Figure 4.15 at $x/h \simeq 4.20$. This location is preferred for illustration as it has the highest correlation value for $P0T0$. The amplitude of the $P2T2$ correlation (Pearson and distance) barely varies from one location to another, indicating a weak variation of the spanwise pressure gradient. Pearson correlations of $P0T0$ and $P2T2$ indicate an anti-correlation around 15%. This anti-correlation is not surprising and is consistent with the theory of favorable and adverse pressure gradients in boundary layers. The magnitude of the two distance correlations is



(a) Pearson correlation.

(b) Distance correlation.

Figure 4.14: Space-time correlation (a): $U0T0$, (b): $U1T0$, and (c): $U2T2$ in the streamwise direction evaluated at $x/h \simeq 0$ on the upper solid wall of the periodic hill at $Re_b = 10,595$.

different, indicating distinct distributions, whereas Pearson suggests only a weak correlation. Compared to correlations with the velocity, the high domain of correlation $\mathcal{D}$ is narrower. Nevertheless, it still aligns with the local mean velocity. Although the domain $\mathcal{D}$ is barely shifted in space and time. This observation indicates that the pressure gradient structures are not convected similarly to the velocity structures when moving away from the wall. The time delay and space displacement can be neglected. Consequently, instantaneous and local information are enough to characterize the relation between the pressure gradients and the corresponding shear stresses. The correlations show some oscillations, but they are only observed in the streamwise, not in the spanwise direction as shown in Section 4.3.2.2.

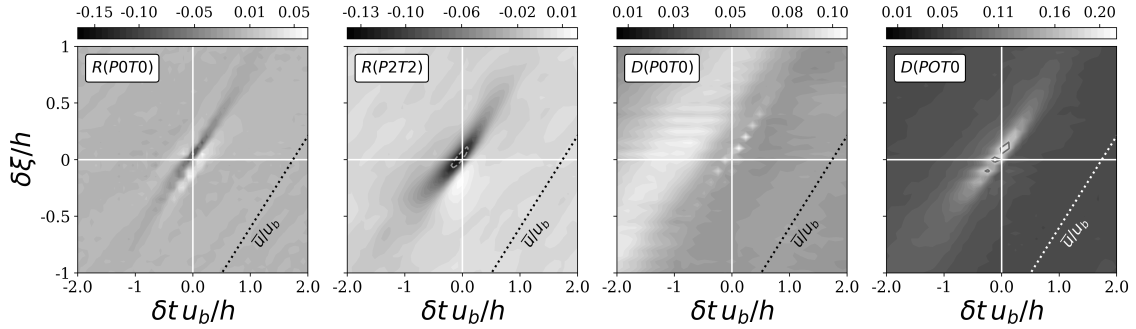


Figure 4.15: Space-time correlations $P0T0$ and $P2T2$ in the streamwise direction evaluated at $x/h \simeq 4.2$ on the upper wall of the periodic hill at $Re_b = 10,595$.

4.3.2.2 Spanwise Pearson correlations

Even though the geometry is 2D-extruded, the massive separation at $x/h \simeq 0.19$ on the lower solid wall and the reattachment of the free shear layer further downstream at $x/h \simeq 4.21$ generate hairpin vortices that are three-dimensional structures. For this reason, correlations along the spanwise direction are also evaluated to quantify this three-dimensional phenomenon. In contrast to the channel, where only two spanwise correlations were significant, on the upper wall, five non-negligible correlations with the velocity are detected: $U0T0$, $U1T0$, $U0T2$, $U1T2$, and $U2T2$. The lower height in wall units for the periodic hill may reveal more correlations than for the channel since the extraction height is smaller (i.e., closer to the wall). The correlations are similar in shape and amplitude for the three regions (see Figure 4.13), except at $x/h \simeq 4.20$ where a drop of 30% and 50% is measured in the amplitude of the correlation $U1T0$ and $U2T2$, respectively. Therefore, for the sake of brevity, only correlations at $x/h \simeq 0$ are provided. Moreover, distance correlations with the velocity field are not shown because they give a similar domain $\mathcal{D}$ with relevant magnitudes of 15% to 30%. The correlations $U0T0$ and $U1T0$ (Figure 4.16a and 4.16b) are similar to those detected in the channel, but no anti-correlation is detected for $U1T0$. Correlation $U2T2$ has the same amplitude and time delay as $U0T0$ and $U1T0$ but spreads over more spanwise points. The two cross-correlations present two lobes, one negative and one positive, of equal amplitude, arranged symmetrically around the horizontal axis.

Correlations with pressure gradients are also analyzed in the spanwise direction, and

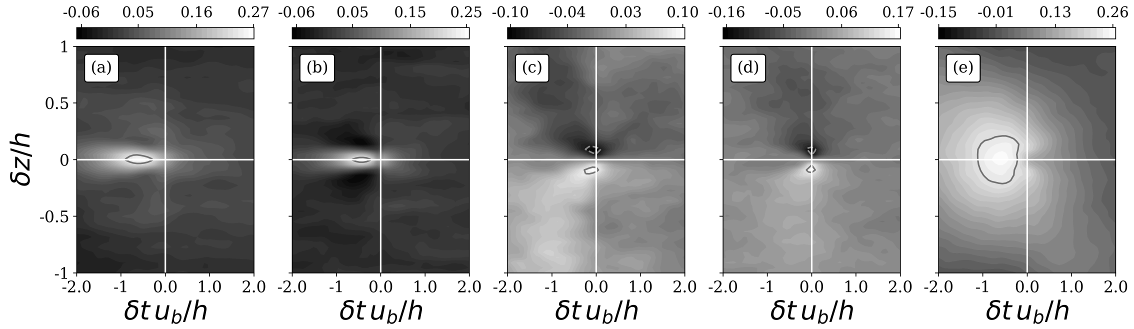


Figure 4.16: Space-time correlations $U0T0$, $U1T0$, $U0T2$, $U1T2$, and $U2T2$ (from left to right) in the spanwise direction evaluated at $x/h \simeq 9.0$ along the upper wall of the periodic hill at $Re_b = 10,595$.

only two correlations are relevant: $P1T2$ and $P2T2$ (not shown here for the sake of brevity and clarity). However, their domain $\mathcal{D}$ is so narrow that instantaneous and local information is sufficient to characterize their relationship with the spanwise wall shear stress.

4.3.2.3 Correlations on the upper wall: conclusion

Although this wall is subject to pressure fluctuations generated by the lower wall, correlations with the velocity evaluated in the streamwise direction lead to a similar conclusion as for the channel: the correlation shows a time delay ($\delta t < 0$) to compensate for the space displacement where both space and time are coupled to the local mean velocity. This observation is emphasized in Figure 4.3b, where the time delay increases with the distance from the wall.

Due to the homogeneity of the spanwise direction, the domain $\mathcal{D}$ (for the correlations evaluated in the spanwise direction) is always symmetrical about the horizontal axis. The correlation $U2T2$ shows a greater extent of the domain $\mathcal{D}$ in the spanwise direction. This observation suggests an extension of the stencil in the spanwise direction. This recommendation is even more true for the cross-correlations which have zero correlation at the origin. This statement is true, if and only if the wall model discretization $(\delta z)_{wmLES}$ is smaller than the spanwise displacement measured with the correlations.

Due to the blockage effect generated by the massive recirculation bubble, a pressure gradient acts on the upper wall, and correlations with the pressure gradient are detected. In contrast to correlations with the velocity, these correlations are mostly origin-centered and narrow. Hence, instantaneous and local information seems sufficient to characterize the relation with the pressure gradient.

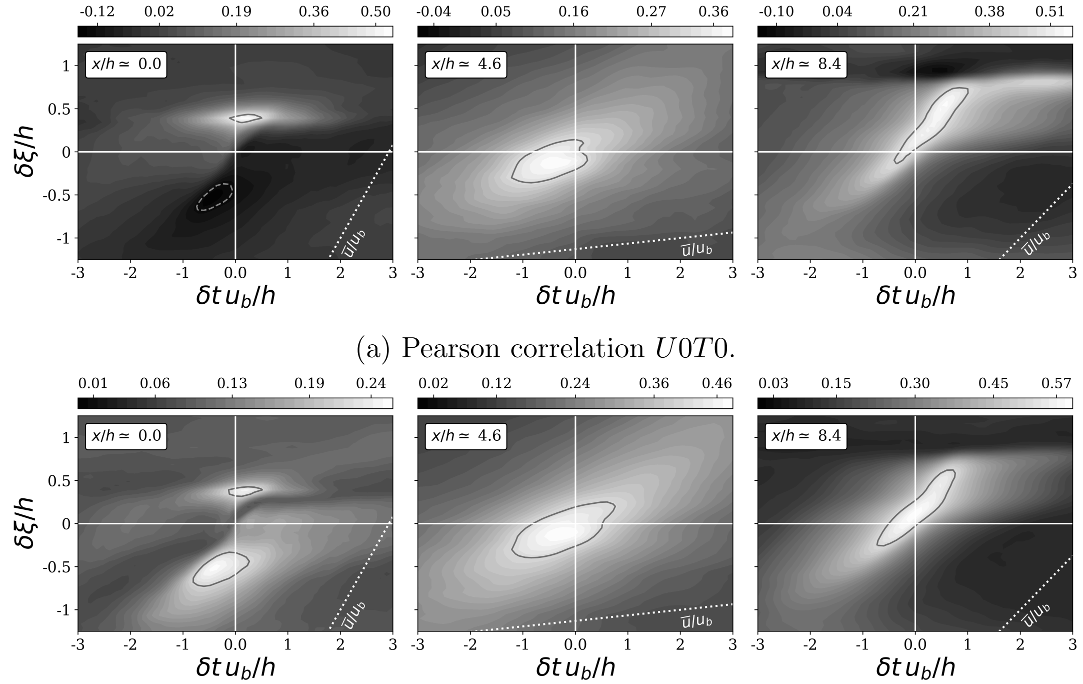
4.3.3 Correlations on the bottom wall of the periodic hill

As mentioned above, the geometry of the periodic hill (PHL10595) was carefully designed to allow the flow to separate from the curved surface and to reattach further downstream. Each streamwise position along the lower wall sees a different flow physics. Correlations must, therefore, be sought in different ξ-regions. Three regions are (i) the vicinity of the separation, (ii) after the reattachment, and (iii) on the convex windward wall of the next hill. In Figure 4.13, the first two red squares (on the left side) are located before and after the mean separation location, respectively. Remember that the instantaneous separation occurs over a large portion of the hill. The third point is located in the recirculation bubble, while the fourth dot is set just after the mean reattachment location. Due to the random nature of the separation position, the reattachment is also affected, and its instantaneous position differs from the mean position. The last red square is positioned on the convex windward wall of the next hill, where the flow undergoes a strong acceleration.

4.3.3.1 Streamwise Pearson and distance correlations

Three locations are targeted for the analysis of the correlations in the streamwise direction: (Sp), (Rt), and (Cwd), for *separation*, *reattachment* and *convex windward wall of the next hill*, and corresponding to $x/h \simeq 0$, 4.2, 8.0 in Figure 4.13, respectively. At these locations, $U0T0$, $U1T0$, and $U2T2$ are detected as significant correlations.

Correlation $\boldsymbol{U0T0}$. For the same location (see Figure 4.17), Pearson and distance correlations have the same domain $\mathcal{D}$, but with slightly different amplitudes which may indicate a non-linear relationship between u_ξ and $\tau_{w,\xi}$. However, at two locations, the correlations have nothing in common. Both shapes and amplitudes of $\mathcal{D}$ are different. Near separation, the correlation is split into two lobes; one shifted in the positive $\delta\xi/h$ with a positive amplitude, and one shifted in space (negative $\delta\xi/h$) and time with a negative amplitude (i.e., anti-correlation). There is no instanta-

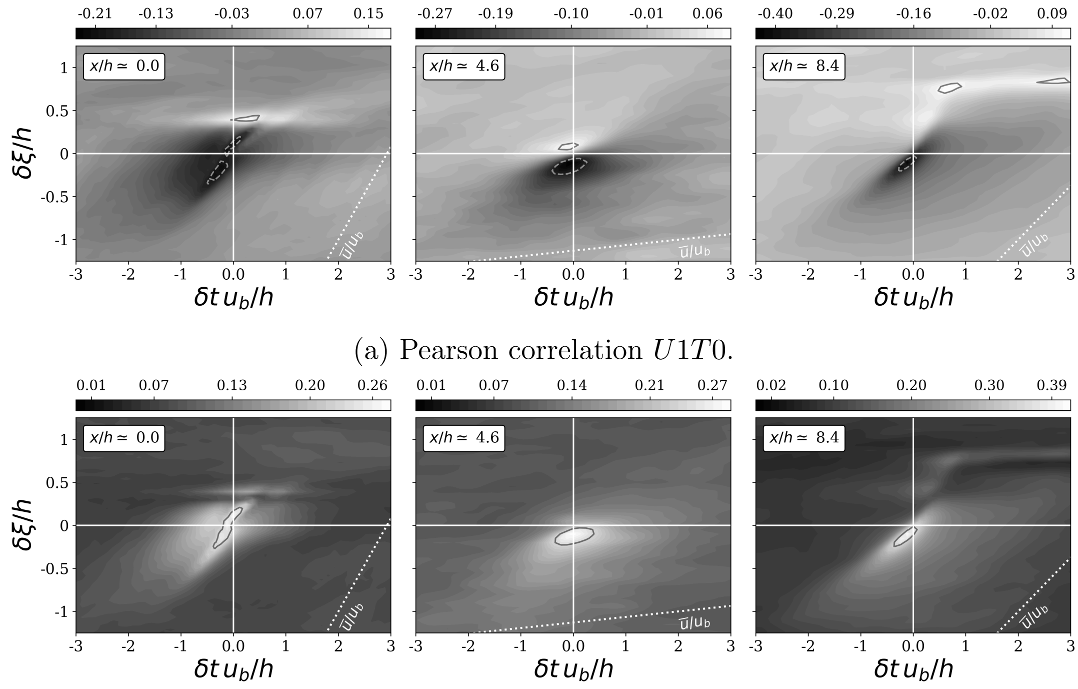


(a) Pearson correlation $U0T0$.

(b) Distance correlation $U0T0$.

Figure 4.17: Space-time correlation $U0T0$ in the streamwise direction evaluated at $x/h \simeq 0$, 4.2, 8.0 (from left to right, corresponding to Sp, Rt, and Cwd) along the lower solid wall of the periodic hill at $Re_b = 10,595$, white dotted line corresponds to $\overline{u}_\xi/u_b$ at a wall-normal distance of $0.1h$.

neous and local correlation between u_ξ and $\tau_{w,\xi}$, probably due to the sudden onset of the free shear layer, which carries energetic structures away from the wall. The anti-correlation in the lower left quadrant indicates that if the velocity decreases, the boundary layer is less prone to separation while an increase in the velocity promotes separation. On the one hand, the amplitude of this anti-correlation (measured by Pearson correlation) is five times smaller than the amplitude of the other lobe, indicating a weaker relationship. On the other hand, the distance correlation measures similar amplitude in both domains (i.e., similar joint distribution). To better characterize the separation with a neural network, it is necessary to encode upstream and downstream information in the inputs. In previous space-time correlations, the domain $\mathcal{D}$ was always aligned with the local mean velocity. However, for correlations measured in the separation vicinity, none of the lobes align with the local velocity. The lobe located at $\delta\xi/h = 0.5$ is aligned with the horizontal (i.e., zero velocity). This observation is consistent with the separation phenomenon. The lobe located at $\delta\xi/h = -0.5$ is aligned with the mean velocity measured on the convex windward wall of the next hill. For (Rt) and (Cwd), the alignment of the correlations with the local mean velocity is recovered. Interestingly, the correlations stretch as the flow accelerates. Along the convex windward wall of the next hill, the correlation is shifted in $\delta t\, u_b/h > 0$. This shift may be due to curvature effects.



(a) Pearson correlation $U1T0$.

(b) Distance correlation $U1T0$.

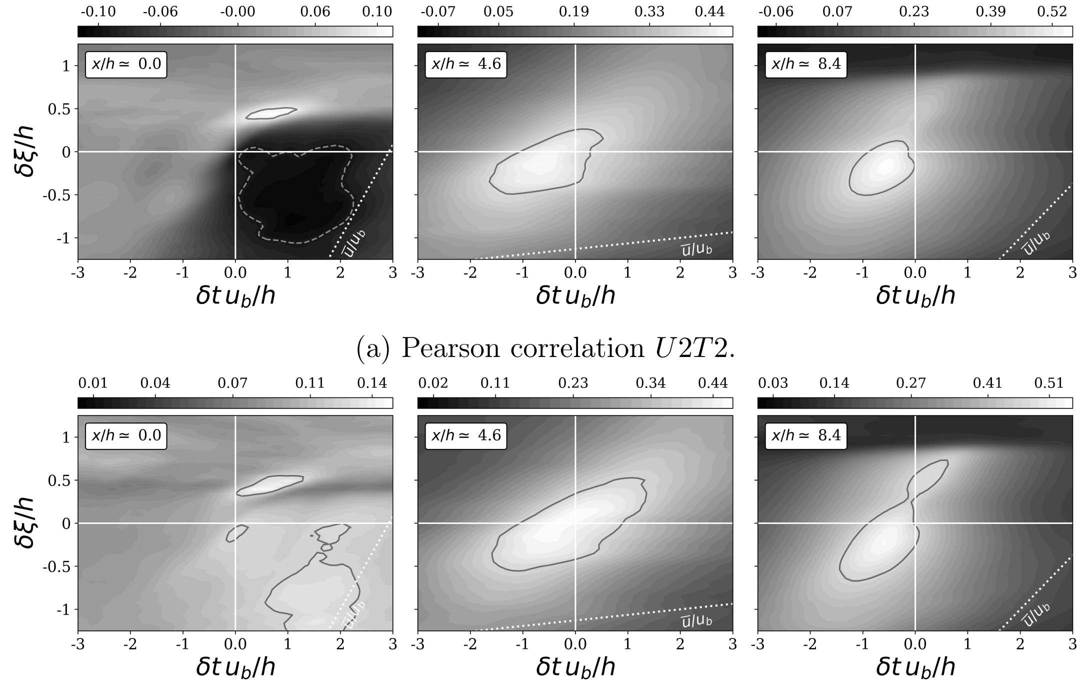
Figure 4.18: Space-time correlation $U1T0$ in the streamwise direction evaluated at $x/h \simeq 0$, 4.2, 8.0 (from left to right, corresponding to Sp, Rt, and Cwd) along the lower solid wall of the periodic hill at $Re_b = 10,595$, white dotted line corresponds to $\overline{u}_\xi/u_b$ at a wall-normal distance of 10% of the hill height.

Correlation $\boldsymbol{U1T0}$. Although the size of the domain $\mathcal{D}$ is much smaller than $U0T0$ at each location, the positioning of $\mathcal{D}$ is quite similar to $U0T0$ (see Figure 4.18). Near the separation, Pearson correlation indicates two lobes as for $U0T0$, with one positive and one negative. However, unlike $U0T0$, the amplitude of the anti-correlation is here greater than the positive one. As energy is extracted from the boundary layer, the latter is more likely to separate downstream. The positive lobe again aligns with the horizontal due to zero convection in the separation vicinity, whereas the anti-correlation seems to almost align with the local mean velocity. Moving to the reattachment location, the correlation is shifted upstream $\delta\xi/h < 0$ for both Pearson and distance. Pearson correlation also indicates a tiny lobe downstream, but its amplitude is five times smaller than the anti-correlation magnitude. Hence, it is neglected in the analysis. For (Cwd), the correlation is also shifted upstream and an anti-correlation is predicted by Pearson. For both (Rt) and (Cwd), the domain of high correlation $\mathcal{D}$ coincides with the local mean velocity.

Correlation $\boldsymbol{U2T2}$. As for $U0T0$ and $U1T0$, the correlation $U2T2$ is characterized by two lobes, one positive, shifted downstream, and one negative, shifted upstream (see Figure 4.19). Due to causality, the correlation is not exploitable because most of the domain of high correlation is shifted *in the future* (i.e., $\delta t\, L_x/u_b > 0$). The domain of the anti-correlation is not captured in the same way, probably due to a slight statistical convergence issue in capturing this phenomenon. For (Rt) and (Cwd), the correlations converge better with clean edges and shapes. However, they do not align with the local mean velocity. They appear to be more tilted, indicating higher convection of the correlated structures.

Correlation $\boldsymbol{P0T0}$. Near the separation, in Figure 4.20, the domain $\mathcal{D}$ of $P0T0$ appears fragmented, as already observed in the streamwise correlation for the pressure gradient on the upper wall (see Figure 4.15). While Pearson correlation presents a weak anti-correlation, the distance correlation has a higher value of 21%, indicating a non-linear relationship between $\partial p/\partial\xi$ and $\tau_{w,\xi}$. As the streamwise pressure gradient increases, $\tau_{w,\xi}$ decreases, promoting separation, as expected. This observation holds at each location, as an anti-correlation is detected at Sp, Rt, and Cwd. Except at separation, the correlations align with the local mean velocity. At the separation, the correlation is aligned with a lower velocity, illustrating the deceleration induced by the separation. The effect of the pressure gradient on $\tau_{w,\xi}$ is even more pronounced when the boundary layer is accelerated on the convex windward wall of the next hill with a value of 30%.

Correlation $\boldsymbol{P1T0}$. No clear correlation is observed for $P1T0$ at the reattachment

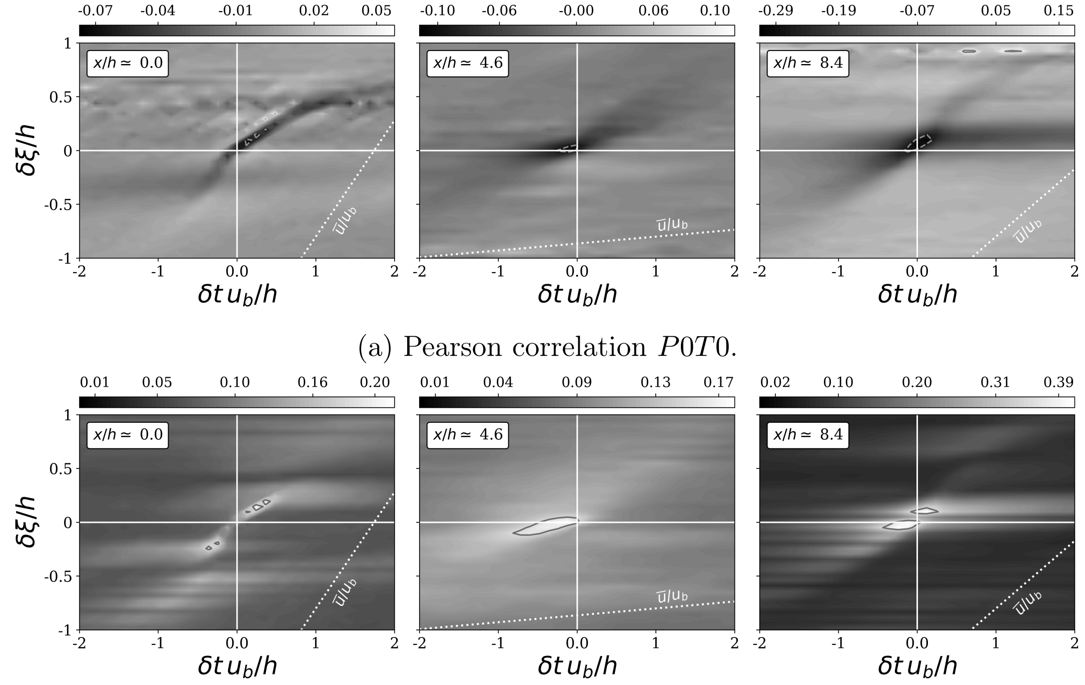


(a) Pearson correlation $U2T2$.

(b) Distance correlation $U2T2$.

Figure 4.19: Space-time correlation $U2T2$ in the streamwise direction, evaluated at $x/h \simeq 0$, 4.2, 8.0 (from left to right, corresponding to Sp, Rt, and Cwd) along the lower solid wall of the periodic hill at $Re_b = 10,595$, white dotted line corresponds to $\overline{u}_\xi/u_b$ at a wall-normal distance of 10% of the hill height.

location. Therefore, only correlations at Sp, and Cwd are shown in Figure 4.21. Near separation, the domain $\mathcal{D}$ is again fragmented. It is not clear whether this fragmentation is related to an oscillatory behavior of the streamwise pressure gradient or to a lack of statistical convergence. Distance correlation appears to be less sensitive to oscillations and provides two lobes of equal amplitude: one located in the upper right quadrant and one located in the lower left quadrant. This second lobe can only be captured if a time delay is considered in the input of the wall model. None of the lobes align with the local mean velocity, but with a reduced velocity due to the deceleration of the boundary layer while approaching the separation. In the convex windward wall of the next hill, Pearson correlation detects two lobes, while distance detects only one. The amplitude of the anti-correlation is twice as small as the amplitude of the positive lobe. Since the distance correlation does not detect this second domain, it can be neglected. Pearson correlation seems to better capture the alignment with the local mean velocity, while for the distance correlations, the domain $\mathcal{D}$ is aligned with a lower velocity.



(a) Pearson correlation $P0T0$.

(b) Distance correlation $P0T0$.

Figure 4.20: Space-time correlation $P0T0$ along the streamwise direction, evaluated at $x/h \simeq 0$, 4.2, 8.0 (from left to right, corresponding to Sp, Rt and Cwd) along the lower solid wall of the periodic hill at $Re_b = 10,595$, white dotted line corresponds to $\overline{u}_\xi/u_b$ at a wall-normal distance of $0.1h$.

Correlation $\boldsymbol{P2T2}$. In Figure 4.22, the high correlation region of $P2T2$ is less fragmented than $P0T0$. It is also less spread over time and space. More generally, at each location, $\mathcal{D}$ is mostly centered and aligned with the local mean velocity. Therefore, there is no need to consider either time delay or space displacement to correctly capture the non-linear relationship between $\partial p/\partial z$ and $\tau_{w,z}$. As for $P0T0$, an anticorrelation is observed, which is consistent with the separation phenomenon

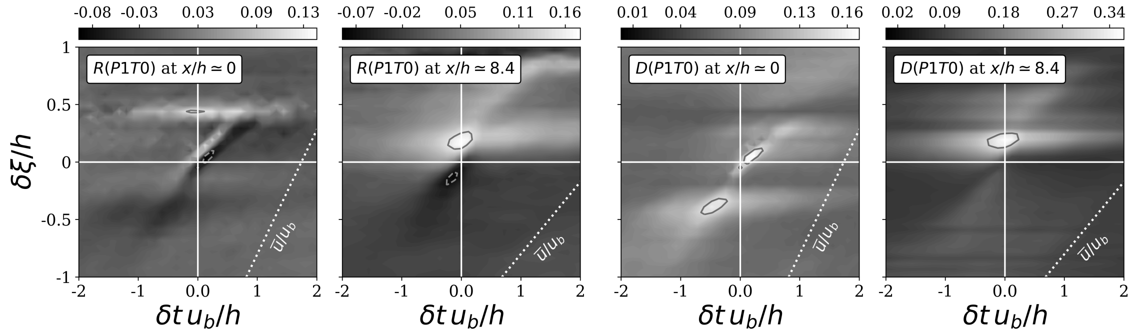


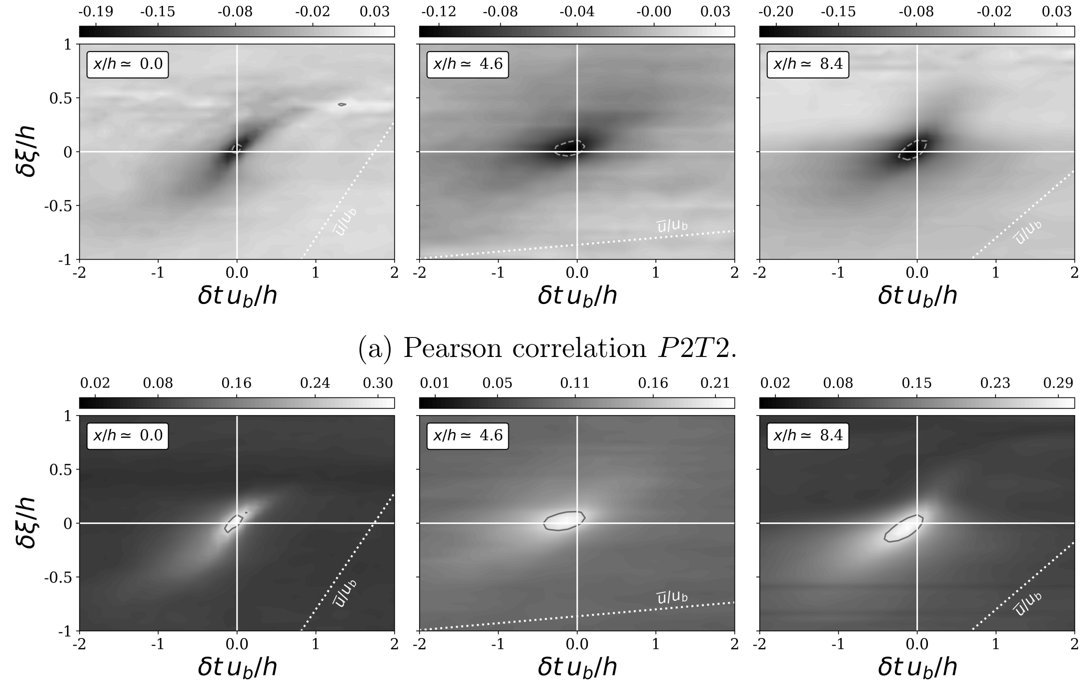
Figure 4.21: Space-time correlation $P1T0$ along the streamwise direction, evaluated at $x/h \simeq 0$, 8.4 along the lower solid wall of the periodic hill at $Re_b = 10,595$, white dotted line corresponds to $\overline{u}_\xi/u_b$ at a wall-normal distance of $0.1h$.

in the case of a pressure gradient.

4.3.3.2 Spanwise Pearson correlations

As mentioned above, turbulence is three-dimensional. Therefore, correlations are also sought in the spanwise direction. However, the correlation maps obtained in this section are not significantly different from those obtained for the channel and the upper wall. The streamwise direction seems to drive the flow physics in these three configurations. A few existing wall models using instantaneous and local velocity sometimes account for the spanwise direction by averaging data along that direction. However, they fail to predict the wall shear stress of the separated flow, which reinforces the interest in the streamwise rather than the spanwise direction.

Although spanwise correlations behave in the same way as on the upper wall, they differ in both shape and amplitude in the five regions (see red squares in Figure 4.13). For the correlations $U0T0$, $U1T0$, and $U2T2$ (where $U0T0$ is displayed in Figure 4.23), $\mathcal{D}$ is centered around the horizontal axis (i.e., symmetric about the time axis) due to the absence of convection. The direct consequence is that the correlations are shifted only in time (i.e., $\delta t < 0$) and not in space. The amplitude and shape grow from left to right, from $x/h = 0$ to $x/h = 9$. For the correlations $U2T0$, $U0T2$, and $U1T2$ (where $U2T0$ is shown in Figure 4.24), the domain $\mathcal{D}$ is split into two lobes, one positive and one negative of equal amplitude. These lobes



(a) Pearson correlation $P2T2$.

(b) Distance correlation $P2T2$.

Figure 4.22: Space-time correlation $P2T2$ along the streamwise direction evaluated at $x/h \simeq 0$, 4.2, 8.0 (from left to right, corresponding to Sp, Rt, and Cwd) along the lower solid wall of the periodic hill at $Re_b = 10,595$, white dotted line corresponds to $\overline{u}_\xi/u_b$ at a wall-normal distance of $0.1h$.

are inverted in the recirculation bubble (second and third images of Figure 4.24). In contrast to the upper wall, the spanwise extent of $\mathcal{D}$ is larger, which means that larger structures develop on the lower wall (i.e., big roll-up at the reattachment). All these correlations are, as expected, symmetrical about the horizontal axis. Some of them are not perfectly symmetrical due probably to a lack of statistical convergence. As for $U0T0$, the amplitude and shape of $U2T0$ increases from left to right. However, instead of being shifted to the left, the domain of high correlation is more and more shifted to the right (i.e., positive time delay $\delta t\, L_x/u_b > 0$).

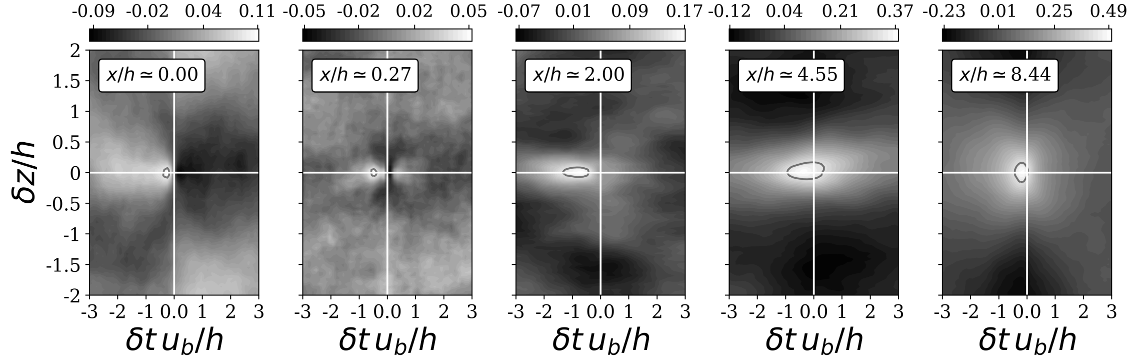


Figure 4.23: Pearson space-time correlation $U0T0$ along the spanwise direction, measured at five locations (Fig.4.13) along the bottom wall of the periodic hill at $Re_b = 10,595$.

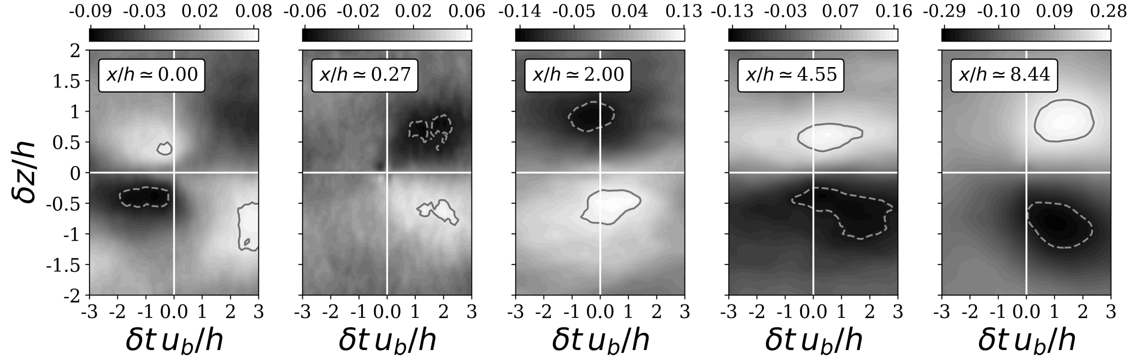


Figure 4.24: Pearson correlation $U2T0$ in the spanwise direction, measured at five locations (Fig.4.13) along the bottom wall of the periodic hill at $Re_b = 10,595$.

4.3.3.3 Correlations on the lower wall: conclusion

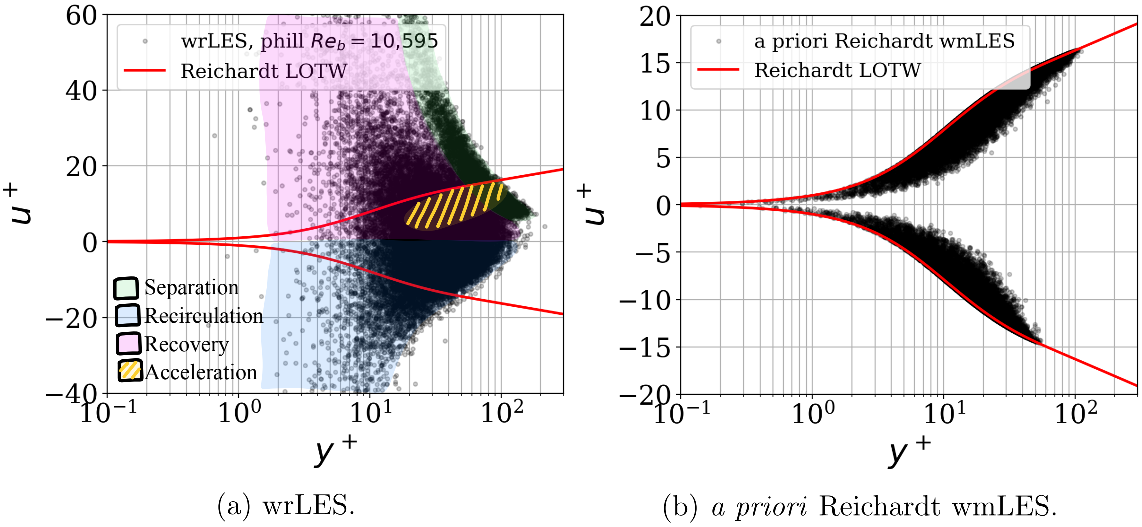
The main difference with the streamwise correlations obtained on the upper wall is the dipole detected in the separation vicinity for every correlation with the velocity field. However, this behavior is not always seen in correlation with the pressure gradient. The correlations are highly dependent on the position along the lower wall, whereas they are less sensitive to the position on the upper wall due to the minimal influence of the lower wall. However, for attached flows, correlations are similar to those observed for the channel flow.

The way the spanwise correlations behave is similar to that observed for the upper wall and the channel at lower y^+. These correlations also occur in one or two lobes that are symmetrical about the horizontal axis, reflecting the absence of convection in the spanwise direction. However, the domain of dependence is more spread out in the spanwise direction, with significant values reaching up to $\delta z/h \simeq \pm 1$. This observation suggests that it would be beneficial to extend the range of influence in the z-direction when using wall models for reattachment phenomena. Additionally, the fact that there is a 30% correlation between the spanwise velocity and the streamwise wall shear stress indicates that there are 3D effects and significant coupling in the wall-parallel direction.

4.3.4 Deviation from the WSS model based on Reichardt's velocity profile

As performed in Section 4.2.3, the *a priori* deviation of the WSS model based on Reichardt's profile (AWSSR) from the wrLES data is quantified. The assumptions on the boundary layer of such a model (i.e., fully turbulent, at equilibrium, and attached) do not allow to use it on separated flows. Even though the model works well for a turbulent channel flow, a weakness has been identified: the model predictions have a drastically reduced variance compared to the wrLES wall shear stress. The boundary layer is not instantaneously at equilibrium. However, the AWSSR forces the BL to be at equilibrium in a wmLES. Furthermore, it assumes a strict relationship between the streamwise velocity and the wall shear stress, which was shown from space-time correlations to be incorrect. This observation is even more true for separated flows, for which correlations with the pressure gradient were found. Moreover, geometric parameters, such as the curvature, may also affect the wall shear stress (directly or indirectly).

Figure 4.25 shows the distribution of the instantaneous scaled streamwise velocity u^+ as a function of the scaled wall-normal distance y^+ for the lower wall of the two-dimensional periodic hill at $Re_b = 10{,}595$ (PHL10595). The (u^+, y^+) points are even more scattered than in a channel. The lower part of the graph (in light blue) with negative velocity represents the recirculation bubble. The long region (in light green) extending to very high-velocity values indicates the separation region. For the rest of the graph, points with relatively low y^+ values (in light pink) belong to the reattachment region, while the others (in hatch yellow) belong to the accelerated



(a) wrLES.

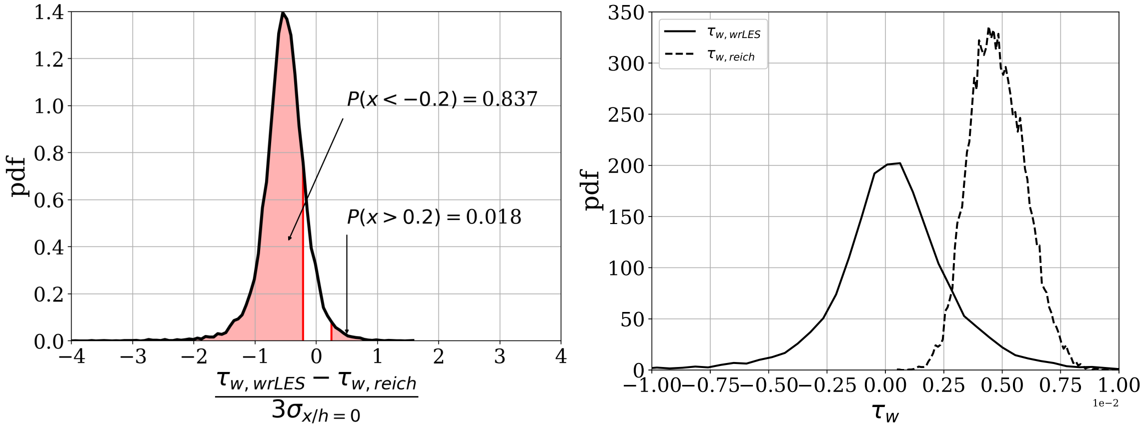
(b) *a priori* Reichardt wmLES.

Figure 4.25: Distribution of the instantaneous scaled streamwise velocity u^+ as a function of the scaled distance to the wall y^+ for PHL10595.

region. None of these points seem to fit Reichardt's profile. Figure 4.25b shows the *a priori* predictions of the AWSSR. All points are stacked below Reichardt's profile, and none of the four regions (i.e., separation, recirculation, reattachment, and accelerated) can be distinguished. Note that in this *a priori* evaluation, the feedback loop between the wall shear stress and the velocity is not quantified. The two plots have nothing in common.

To further quantify the deviation, the probability density function of the deviation between the predicted streamwise wall shear stress and the wrLES one is computed at several locations along the lower wall. Figure 4.26b shows the distribution of the wrLES wall shear stress (black line) and the distribution of the predicted wall shear stress (black dashed line). The predicted mean is strictly positive: the AWSSR predicts no separation. The gap between the two means is highlighted in Figure 4.26a, which shows the PDF of the deviation near the separation. As the mean wall shear stress is almost zero at $x/h \simeq 0$, the relative error is computed with the wrLES wall shear stress standard deviation at $x/h \simeq 0$. Most of the predictions have an error greater than 20%, which makes the model *a priori* unusable.

Moving to a point in the recirculation bubble, the mean of the prediction using



(a) Relative deviation distribution.

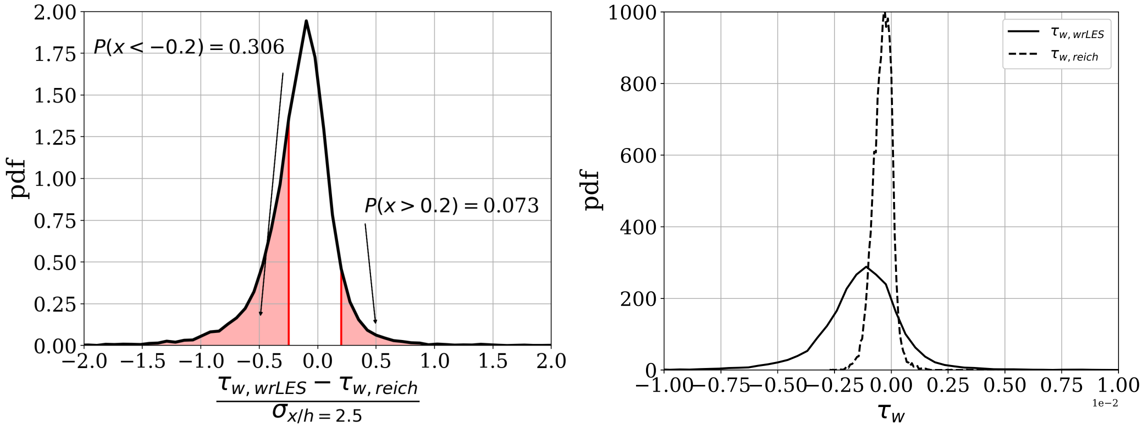
(b) Distribution of $\tau_{w,wrLES}$ and $\tau_{w,reich}$.

Figure 4.26: Non-dimensional deviation between the wrLES and predicted wall shear stress using the AWSSR on PHL10595 at $(\xi/h, \eta/h) \simeq (0, 0.1)$.

the AWSSR is negative meaning that the model has understood the reverse flow but with the wrong amplitude (see Figure 4.27b). As observed for the channel, the variance of the prediction is much smaller which further explains the very low level of dispersion in Figure 4.25b. The PDF of the relative error based on the standard deviation indicates that the model generates an error greater than 20% with a non-negligible probability of about 0.4 (see Figure 4.27a).

Figure 4.28b shows the distributions of the wrLES wall shear stress and the predictions in the recovery region. This region is located after the reattachment when the flow stabilizes and starts to reaccelerate on the flat lower surface. Both means are positive and the distribution of Reichardt's predictions has a significant skewness. Such a skewness was not observed in the *a priori* prediction distribution for a channel (Section 4.2.3). Since the local mean velocity is zero at separation, there is no convection and no skewness in the prediction. In the recirculation bubble, the velocity structures evolve at a low speed. Therefore, there is no skewness. A better overlap of the distribution is observed, and this observation is reflected in the PDF of the deviation. The model has an error greater than 20% with a probability of 0.3.

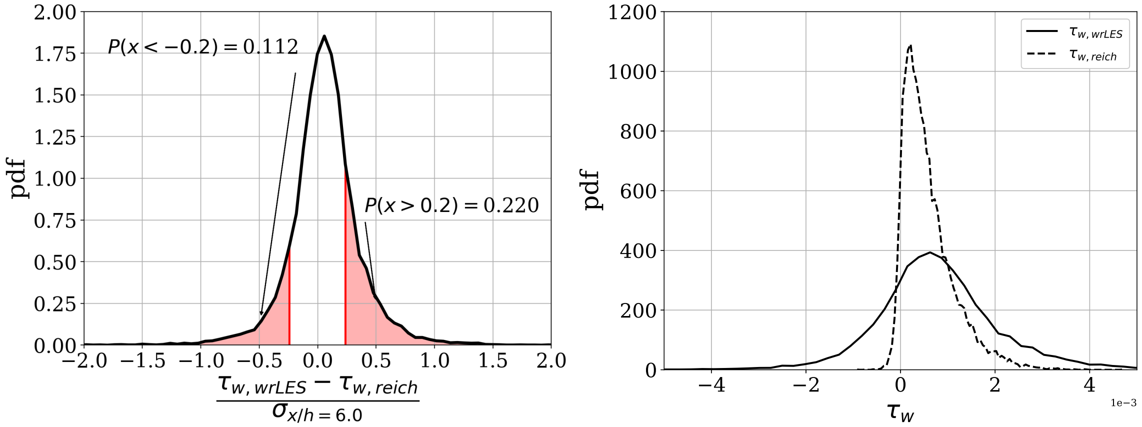
The probability of making an error greater than 20% decreases as one approaches the reaccelerated region. This observation is consistent with the assumption of Re-



(a) Relative deviation distribution.

(b) Distribution of $\tau_{w,wrLES}$ and $\tau_{w,reich}$.

Figure 4.27: Non-dimensional deviation between the wrLES and predicted wall shear stress using the AWSSR for PHL10595 at $(\xi/h, \eta/h) \simeq (2.5, 0.1)$.



(a) Relative deviation distribution.

(b) Distribution of $\tau_{w,wrLES}$ and $\tau_{w,reich}$.

Figure 4.28: Non-dimensional deviation between the wrLES and predicted wall shear stress using the AWSSR for PHL10595 at $(\xi/h, \eta/h) \simeq (6, 0.1)$.

ichardt's model. The model should perform better in regions where the boundary layer is attached and subjected to a low-pressure gradient. However, *a priori*, the WSS model based on Reichardt's profile is not suitable for separated flows, as expected. This claim is confirmed in Figure 4.29, which shows the instantaneous WSS prediction. Compared to the channel predictions, the two-dimensional structures of the predicted WSS are more coherent with those of wrLES. However, the amplitude of the predicted WSS is different, as already observed with PDFs.

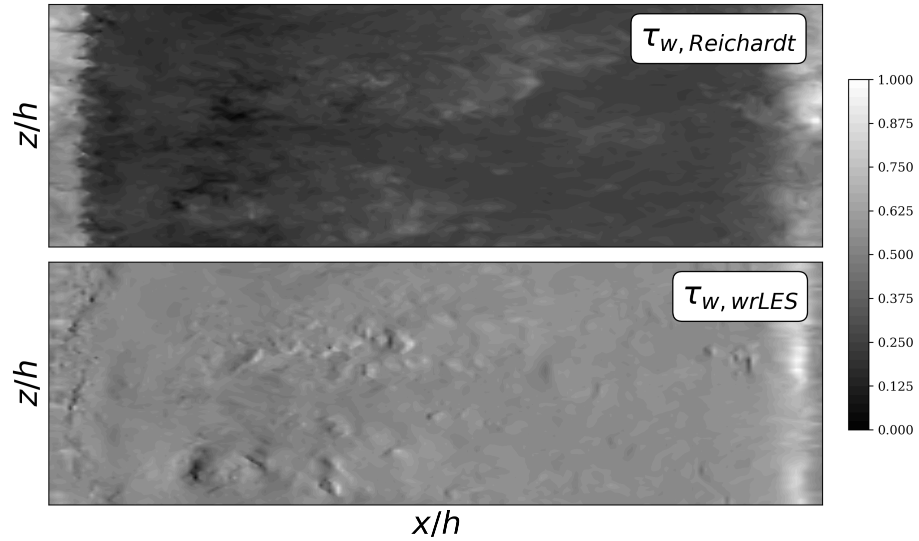


Figure 4.29: Instantaneous *a priori* predictions of the streamwise wall shear stress using the AWSSR for PHL10595, compared with the wrLES wall shear stress at a given time step t.

The new data-driven wall shear stress will have to overcome the weaknesses of the quasi-analytical WSS model based on Reichardt's profile (AWSSR) by increasing the input stencil, capturing a more complex relationship between instantaneous volume fields and the WSS, and predicting both the mean and standard deviation of the wall shear stress.

4.4 Additional databases

Additional datasets generated via other methods are joined to the current database (i.e., the turbulent channel flow at $Re_\tau = 950$ and the two-dimensional periodic hill at $Re_b = 10{,}595$). The first additional dataset is a set of synthetic (SYNTH.) data extracted from the LOTW. The sample generation is described in Section 4.4.1. This database is used for testing purposes only and will never be used to train the neural network. This approach contrasts with the work of Zhideng *et al.* (2023), who used synthetic data to train their network along with periodic hill data. The second and third supplementary datasets are two turbulent channel flows: one at $Re_\tau = 1{,}000$ and another at $Re_\tau = 5{,}200$. These databases are extracted from the Johns Hopkins Turbulent Databases website and are presented in Section 4.4.2.

4.4.1 Synthetic data extracted from the law-of-the-wall

In addition to DNS and wrLES data, synthetic data is extracted from the law of the wall (LOTW):

$$u^+ = \frac{1}{\kappa} \ln\left(y^+\right) + B \,, \tag{4.5}$$

where $\kappa \sim 0.41$ is the von Kármán constant and $B = 5.2$. The synthetic data are evaluated from this law for a wide range of friction Reynolds numbers: $Re_\tau = 180, 250, \cdots, 10^4$, equispaced for a total of 41 Re_τ. For each Re_τ, a set of wall-normal heights is created from $y^+ = 30$ (i.e., at the border of the buffer layer) to $y/h = 0.2$. The density and the friction velocity u_τ are set to 1. Hence, the kinematic viscosity is the inverse of Re_τ. For each pair (Re_τ, h^+), the velocity u is extracted from Eq. 4.5. The velocity components v and w are set to zero. This synthetic database is only representative of "*2D-extruded geometry*" where the spanwise wall shear stress is zero on average. This database is not used for training but only for testing purposes, i.e. to evaluate the capability of the neural network to extrapolate to higher Reynolds number and to assess whether the network has learned the log-law instrinsically.

4.4.2 Turbulent channel flows extracted from Johns Hopkins Turbulent Databases

Turbulent channel flow at $\boldsymbol{Re_\tau = 1{,}000}$ (TC1000). This test case is part of the Johns Hopkins Turbulence Databases (Graham *et al.*, 2016). DNS of turbulent channel flow was performed in a domain of size $8\pi h \times 2h \times 3\pi h$. The incompressible Navier-Stokes equations were solved with a pseudo-spectral method in the two periodic and homogeneous directions (x and z), while a 7^{th} order B-spline collocation method was applied in the wall-normal direction. The simulation had required $2048 \times 512 \times 1536$ nodes. The channel flow was firstly equilibrated with a bulk velocity of 1 and then switched to impose a uniform and constant pressure gradient. After accumulating the statistics, the friction velocity attained the value of 0.0499.

It corresponded to a friction Reynolds number of 999.35 with a kinematic viscosity of 5×10^{-5}. From this online database, 16 snapshots of size $112 \times 35 \times 15$ of the velocity and the wall shear stress are extracted on both the upper and lower wall, for a total of 1,881,600 pairs of input/output. The number of wall-normal position have been extended to cover a larger range of y^+ values. The streamwise and spanwise spacing are sufficiently large to obtain uncorrelated pairs. The wall shear stress is scaled with the kinematic viscosity to get a friction velocity of 1 as follows,

$$\tau_w^\dagger = \left(\frac{\nu_{new}}{\nu}\right)^2 \tau_w \, , \tag{4.6}$$

where $\nu_{new} = 1/Re_\tau$. The velocity is scaled accordingly as $u^\dagger = (\nu_{new}/\nu)u$.

Turbulent channel flow at $\boldsymbol{Re_\tau}$ = 5,200 (TC5200). This test case is also part of the Johns Hopkins Turbulence Databases (Moser and Lee, 2015). The same domain size and the same code as for the test case TC1000 are used for the present simulation. It had required $10240 \times 1536 \times 7680$ nodes. While our channels are driven by a uniform (in space and time) pressure gradient, their channel was equilibrated with a constant bulk velocity ($u_b = 1$). After accumulating the statistics, the friction velocity u_τ had reached a value of 0.0414872, leading to a friction Reynolds number of $Re_\tau = 5185.897$. The kinematic viscosity ν was set to 8×10^{-6}. The database is extracted for five snapshots, each containing of $128 \times 95 \times 15$ in streamwise, wall-normal, and spanwise directions, respectively. Finally, 1,824,000 pairs of input/output are obtained. The wall shear stress is also scaled with Eq. 4.6 to get a friction velocity of 1.

4.5 Conclusion

The main part of this chapter investigates the near-wall physics in equilibrium and non-equilibrium flow conditions using filter feature selection methods. This analysis employs space-time correlations (i.e., univariate statistical measures) to establish functional relationships between volume quantities and wall shear stress in a range of flow configurations, including a simple channel flow and a separating and reattaching flow. Throughout our investigation, two types of correlations (Pearson and distance correlations) are used to gain a more comprehensive understanding of the physics developing in each flow configuration.

In a channel flow, the wall shear stress is mostly correlated with the streamwise velocity component. No clear correlation was found with the pressure gradients in the logarithmic layer, as also noted by Abel *et al.* (2006). The correlation maps exhibit an elongated elliptical shape, extending over approximately $200\delta t^+$ units. To compensate for this delay, a downstream displacement can be considered, as proposed by Piomelli *et al.* (1989) in their shifted boundary condition model. Although cross-correlations in the z-direction were noticeable at lower values of y^+ (i.e., $y^+ < 100$), they become insignificant at $y^+ \geq 100$. Only $U0T0$ and $U2T2$ were detected at higher y^+ values. Therefore, the three-dimensional effects generated by the wall reduce as the distance from the wall increases.

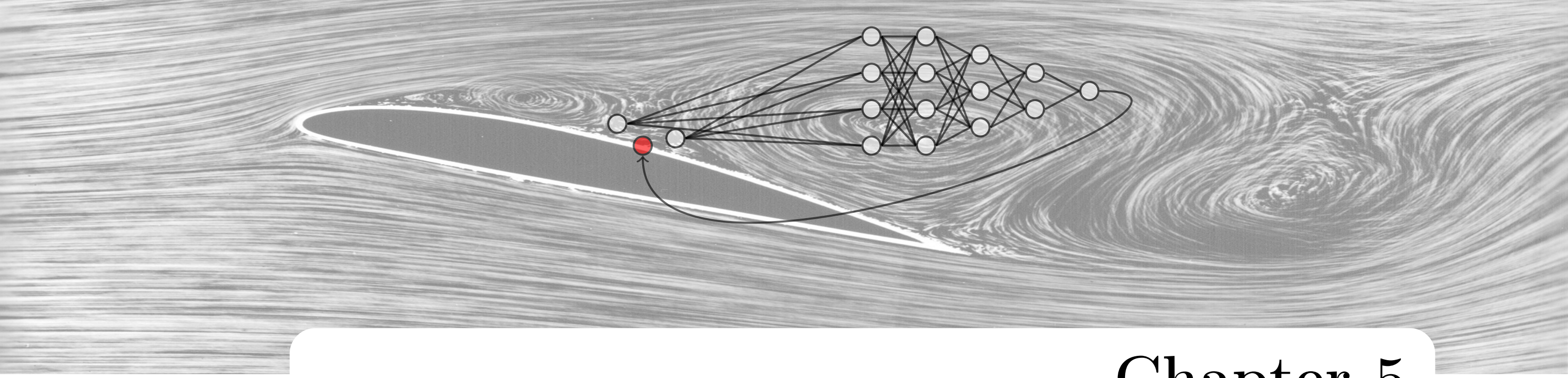
The analysis of the periodic hill is divided into the upper and lower walls. The correlations for the upper wall are similar to those seen for the channel. Despite the non-homogeneous nature of the streamwise direction, the correlations are mostly independent of the position in that direction, indicating that the lower wall has minor effects on the upper wall. Nonetheless, correlations with the pressure gradient are detected at $x/h \simeq 4.20$, in the streamwise and spanwise directions. At the lower wall, correlations are highly dependent on the streamwise position. Three locations are targeted: the separation, the reattachment, and the convex windward wall of the next hill. Similarities with the channel are observed in the reattachment and converging regions. Nevertheless, the correlations are wider at these mentioned locations. Moreover, the correlations are less spread in time, but this observation is mainly due to the lower wall-normal height. In separation and reattachment phenomena, the pressure gradient plays an important role, and correlations with it are detected. In the separation vicinity, the correlation map exhibits oscillations despite the absence of any natural frequency in the separation. The most interesting correlations are obtained in the vicinity of the separation. At this location, most of the detected correlations are divided into two sub-domains. One sub-domain is shifted downstream (i.e., $\delta\xi > 0$), indicating that the free-shear layer and the wall shear stress at the separation are highly correlated. The second sub-domain reveals an anti-correlation between the velocity measured on the convex windward wall of the next hill and the wall shear stress at the separation point. In the spanwise direction, correlations look like those obtained for the channel flow at lower y^+ values. The cross-correlations exhibit two symmetrical lobes about the horizontal axis, each with opposite signs. A change in sign occurs when measuring the correlation within the recirculation bubble.

It is important to keep in mind that the findings of this study, which specifically examine non-equilibrium conditions, may not necessarily be applicable to other flow configurations. Nevertheless, this research marks a significant first step in developing data-driven wall models to address the separation phenomenon. As a result, several recommendations are made to assist in this effort. These correlations primarily serve to guide the selection of the appropriate input stencil size, which may include a number of streamwise and spanwise positions, and a time delay.

As previously mentioned, current wall models using the instantaneous and local velocity predict correctly the wall shear stress in configuration similar to a turbulent channel flow and the upper wall of the two-dimensional periodic hill (Frère *et al.*, 2017, 2018). However, the *a priori* analysis of the WSS model based on Reichardt's velocity profile fails to correctly predict the second and third statistical moments of the streamwise wall shear stress (i.e., the standard deviation and the skewness). Furthermore, the two-dimensional wall shear stress structures are incorrect as they replicate the velocity structures.

A **first path for improvement** is to use neural networks that predict the wall shear stress statistical distribution to recover the higher statistical moments, and to account for structure reconstructions.

A **second path for improvement** is to account for a space displacement or a time delay in the input stencil to capture the high correlation between the velocity and the wall shear stress. The time delay is not considered in the current development because an instantaneous WSS model is sought as discussed in Section 3.5. The new WM has to be local (i.e., one or a few points in the surrounding area) to limit communication between MPI ranks. Because the time delay is ignored, small space displacements are considered by enlarging the input stencil size in the streamwise direction. This space displacement is closely connected to the local mean velocity and is, therefore, highly dependent on the wall-normal height. This streamwise extension of the input stencil is particularly important for separated flows where no instantaneous and local correlation was detected in the separation vicinity. Additionally, a spanwise displacement could be considered to capture the two symmetrical lobes of cross-correlations. This analysis is perhaps counter-intuitive, but for a neural network, the input/output pair must be (highly) correlated to ensure that the model will correctly capture the non-linear relationship that may exist between them. This observation can be reinforced by evaluating SHAP values (see Chapter 8).



Chapter 5
Development of a new data-driven wall shear stress model

Chapter 4 covered the limitations of quasi-analytical WSS models, outlined guidelines for novel WSS models, and generated and analyzed databases. This chapter focuses on the development of new WSS models using neural networks. More specifically, this work uses deep neural networks, which are the core element of Deep Learning (DL). DL includes two categories of supervised learning: regression problems and classification problems. Our model falls into the regression category as it predicts a continuous value, namely wall shear stress.

In the initial stages of the thesis, we first looked at regression problems that use the Mean Square Error (Eq.2.8) as the loss function to measure the fitness of the trained model. For that, several MLPs (Section 2.2.1) and CNNs (Section 2.2.2) were trained to predict the wall shear stress. As expected (Eq. 2.6), the predictions were accurate on average, but the higher statistical moments of the wall shear stress were not captured. Moreover, to obtain the MSE loss, the conditional probability $p(y|\mathbf{x}, f)$ was assumed to be a Gaussian distribution (Eq. 2.7) where the variance σ^2 is fixed. This hypothesis is very strong and appears to be incorrect for the two-dimensional periodic hill, as shown in Figure 5.1. Furthermore, Zhou *et al.* (2022) stated that imposing the mean wall shear stress on the lower wall of the two-dimensional periodic hill is too restrictive and leads to unsatisfactory results. Therefore, our attempt to predict the instantaneous behaviors of the wall shear stress using ML and DL techniques was unsuccessful. Regarding the recommendations of Chapter 4 to overcome the weaknesses of quasi-analytical WSS models, we have decided to predict the wall shear stress distribution instead of its instantaneous value. The Mixture Density Network (MDN) was chosen based on the assumption that any distribution can be represented as a linear combination of Gaussian distributions.

This chapter is divided into three sections. Section 5.1 presents the Mixture Density Network (MDN), how they can be combined to an MLP or a CNN, and how they are trained. Section 5.2 discusses the pre-processing of the generated database: data cleaning and non-dimensionalization. Section 5.3 presents the development environment Argo-DG, a high-order flow solver, developed at Cenaero and the implementation of the novel data-driven WSS in Argo-DG using `TorchScript`.

5.1 (Deep) Neural Networks for wall modeling

Chapter 2 introduced the principal concept of neural network training, which is nothing more than an optimization in a high parameter space. Based on the literature review (Chapter 3), the current wall model will be part of the data-driven wall shear stress (WSS) model, where the model predicts the wall shear stress, a scalar or vector, which is then imposed as a boundary condition.

These conventional neural networks predict the conditional average of the ground truth, conditioned by the input $\boldsymbol{x}$ (Eq. 2.6). The conditional average consists of a very limited statistic. It is therefore interesting to explore models that predict a distribution rather than providing a point estimate. The Mixture Density Network belongs is therefore introduced in Section 5.1.1.

5.1.1 Network for predicting statistics

In "*standard*" regression, the goal is to estimate the function $f(\boldsymbol{x})$ that best fits the data (see Section 2.2). Figure 5.1 illustrates the non-linear relationship between the streamwise velocity u and the wall shear stress $\tau_{w,\xi}$. Due to the separation phenomenon, the conditional probability distribution of $\tau_{w,\xi}$ for $u/u_b = 1$ has two lobes, indicating two possible behaviors of the flow: the flow is attached (i.e., positive wall shear stress) or the flow is detached (i.e., negative wall shear stress). In other words, the wall shear stress can behave differently for a similar input. Considering "*standard*" regression methods, one would obtain the red dashed line: the conditional mean of $\tau_{w,\xi}$. The other two conditional distributions, drawn in Figure 5.1, have a single peak. The spread of the three conditional distributions is different. Therefore, the variance differs from one entry to the other. Moreover, as shown in Figure 4.11, the variance is also location-dependent along the lower wall.

As stated in Section 2.2, inference involves the estimation of the conditional probability $p(Y = y|X = \mathbf{x})$ for a new $(\mathbf{x}, y)$. In the case of a "*standard*" regression problem, this conditional probability was modeled by a Gaussian distribution with a fixed variance (Eq. 2.7). Instead of considering a fixed variance, the variance can be, as the mean, a function of the input data such that,

$$p(y|\mathbf{x}, \mu, \sigma) = \mathcal{N}\left(y|\mu(\mathbf{x}), \sigma^2(\mathbf{x})\right) , \tag{5.1}$$

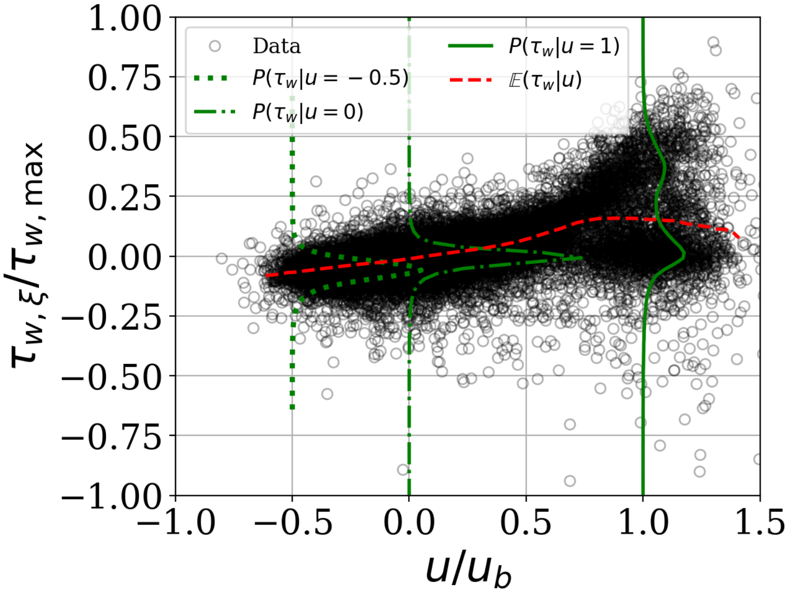


Figure 5.1: Illustration of the conditional probability of the wall shear stress knowing the streamwise velocity.

where $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ are parametric functions to be learned. For each input, the model does not predict a point estimate but a distribution of the output. The objective is to maximize the likelihood of the data over μ and σ:

$$\begin{aligned}\arg\max_{\mu,\sigma} p(\mathbf{d}|\mu,\sigma) &= \arg\max_{\mu,\sigma} \prod_{\mathbf{x}_i,y_i\in\mathbf{d}} p(y_i|\mathbf{x}_i,\mu,\sigma)\\ &= \arg\max_{\mu,\sigma} \prod_{\mathbf{x}_i,y_i\in\mathbf{d}} \mathcal{N}\left(y_i|\mu(\mathbf{x}_i),\sigma^2(\mathbf{x}_i)\right)\\ &= \arg\max_{\mu,\sigma} \prod_{\mathbf{x}_i,y_i\in\mathbf{d}} \frac{1}{\sqrt{2\pi}\sigma(\mathbf{x}_i)}\exp\left(-\frac{(y_i-\mu(\mathbf{x}_i))^2}{2\sigma^2(\mathbf{x}_i)}\right)\\ &= \arg\min_{\mu,\sigma} \sum_{\mathbf{x}_i,y_i\in\mathbf{d}} \frac{(y_i-\mu(\mathbf{x}_i))^2}{2\sigma^2(\mathbf{x}_i)} + \log(\sigma(\mathbf{x}_i)) + C\end{aligned}$$

In this last expression, there is a trade-off between the first and the second term. If $\mu(\mathbf{x}_i)$ is far away from the true value y_i, the difference in the numerator is large, and this term drives the loss. To decrease the loss, the variance, appearing at the denominator, may increase. The second term is present to avoid an infinite variance and to keep it within acceptable bounds. Note that if the variance is fixed to any positive value, the MSE of Equation 2.8 is retrieved.

Assuming that any general distribution can be approximated by a mixture of *simpler* distributions, the conditional distribution $p(y|\mathbf{x})$ can be modeled as a mixture of K Gaussian components (i.e. multi-modal Gaussian). Generally, any probability distribution can be used, but the Gaussian distribution is frequently adopted due to its good mathematical properties and computational performances. Under this assumption, $p(y|\mathbf{x})$ is written as,

$$p(y|\mathbf{x}, \mu_k, \sigma_k) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(y|\mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x})\right) \ , \tag{5.2}$$

where $0 \leq \pi_k \leq 1$ for all k and $\sum_{k=1}^{K} \pi_k = 1$. Mixture Density Network (MDN) is the neural network implementation of the Gaussian Mixture Model. An MDN, as presented in Figure 5.2, has two components: a network and a mixture model. The mixture model has one head of K-components, which produces a mean μ_k, a standard deviation σ_k, and a mixture coefficient π_k. This head can be connected to any neural network (denoted as **NN**).

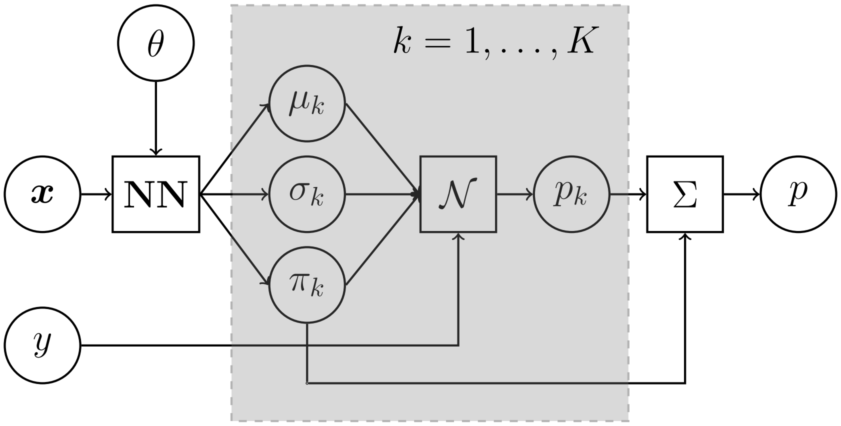


Figure 5.2: Schematic of the architecture of a Mixture Density Network with K modes, each mode predicting a mean μ_k, a standard deviation σ_k, and a mixture coefficient π_k.

To get an expression for the loss, we maximize the likelihood of the data over the network parameters θ:

$$\begin{aligned}\arg\max_{\theta} p(\boldsymbol{d}|\theta) &= \arg\max_{\theta} \prod_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} p(y_i|\boldsymbol{x}_i, \theta) \\ &= \arg\max_{\theta} \prod_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(y_i|\boldsymbol{x}_i, \mu_k(\boldsymbol{x}_i), \sigma_k^2(\boldsymbol{x}_i)\right) \right) \\ &= -\arg\min_{\theta}\ \log \left\{ \prod_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(y_i|\boldsymbol{x}_i, \mu_k(\boldsymbol{x}_i), \sigma_k^2(\boldsymbol{x}_i)\right) \right) \right\} \\ &= -\arg\min_{\theta} \sum_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} \log \left(\sum_{k=1}^{K} \frac{\pi_k}{\sqrt{2\pi}\sigma_k(\boldsymbol{x}_i)} \exp\left(-\frac{(y_i - \mu_k(\boldsymbol{x}_i))^2}{2\sigma_k^2(\boldsymbol{x}_i)} \right) \right)\end{aligned}$$

This last expression cannot be further simplified. Nonetheless, `PyTorch` implements the `torch.logsumexp` that allows to write the following expression,

$$\arg\max_{\theta} p(\boldsymbol{d}|\theta) = -\arg\min_{\theta} \sum_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} \underset{1 \le k \le K}{\operatorname{logsumexp}} \left[g(\pi_k, \mu_k, \sigma_k) \right]$$

where $g(\pi_k, \mu_k, \sigma_k)$ is defined as,

$$g(\pi_k, \mu_k, \sigma_k) = \log(\pi_k) - \log\left(\sigma_k(\boldsymbol{x}_i)\right) - \frac{1}{2}\log(2\pi) - \frac{1}{2}\left(\frac{y_i - \mu_k(\boldsymbol{x}_i)}{\sigma_k(\boldsymbol{x}_i)} \right)^2, \qquad (5.3)$$

for $k = 1, \ldots, K$. Finally, the loss function used as the fitness criterion for a Gaussian Mixture Network is defined as,

$$\mathcal{L}(y, \mu_k, \sigma_k) = -\frac{1}{N} \sum_{i=1}^{N} \underset{1 \le k \le K}{\operatorname{logsumexp}} \left[g(\pi_k, \mu_k, \sigma_k) \right]. \qquad (5.4)$$

The whole network is trained end-to-end using backpropagation applied to the Negative Log Likelihood (Eq. 5.4).

5.1.2 Input stencil size and network architecture

The number of neurons in the input layer depends on the number of input features. The inputs are the three components of the velocity field (u_ξ, u_η, u_z), the three components of the pressure gradients $(\partial p/\partial \xi, \partial p/\partial \eta, \partial p/\partial z)$ and the curvature $\mathcal{K}$, where (ξ, η, z) are the curvilinear coordinates defined in Figure 4.13. These inputs are further non-dimensionalized. This process is explained in Section 5.2. Moreover, each input feature is extracted on a given stencil. If the stencil size is 1, the input is extracted in the wall-normal direction at the matching location h_{wm}. If the stencil size is greater than 1, then several points are extracted. Figure 5.3 illustrates these two configurations.

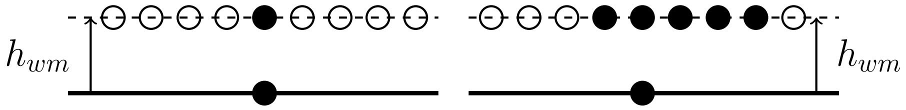


Figure 5.3: Sketch of two input stencil (with one and five points) on a flat wall.

Other sets of inputs were also tested during the thesis, such as adding the relative positions and the mean velocity. Moreover, several stencil sizes were considered, from the largest available to the smallest, with symmetric or asymmetric stencils. These tests aimed to validate the appropriate stencil size obtained from the feature selection performed in Chapter 4. Using space-time correlations, three important information were extracted: the streamwise and spanwise displacement and the time delay. Because the wall model has to be instantaneous to avoid storing a large amount of data as mentioned in Section 3.5, only the space displacement is considered. We cannot simply take one point further downstream, as proposed by Piomelli *et al.* (1989) in their shifted boundary condition model, due to the non-convection of the pressure gradient correlation (see Figure 4.22). As a consequence, more points were added to the stencil to encompass the non-linear relationship between the input features and the wall shear stress. A larger input stencil helps to discriminate between the different flow physics (i.e., turbulent, recirculating, separated, and reattached flow).

The *optimal* stencil is constructed as the union of every space-time correlation to capture all the domains of high correlation $\mathcal{D}$. Firstly, most of the correlations are shifted downstream $\delta\xi > 0$, and thus ten points downstream are added to the stencil (see Figure 4.17). Secondly, the upright point in the wall-normal direction is always part of the stencil (see Figure 4.22) to account for correlation with the pressure gradient. Thirdly, to encapsulate the correlation $U1T0$, shown in Figure 4.18, five points upstream are also added to the stencil. We ended up with an asymmetric stencil of size 16.

Note that the number of points in the stencil depends on the probe set resolution. This is one drawback of using multiple points to characterize the relationship with the wall shear stress. The *a posterior* tests will need to be performed on a grid of similar resolution or at least to keep the same spacing between two entries (i.e., if the resolution is double, then to reconstruct the stencil at a particular location, one probe over two should be taken). One possible way to deal with this drawback is to randomly select m points in a stencil of size $M + 3$ (where $M + 3 \geq m$). The "+3" is because the the most-left, most-right and the *current* points are fixed, as illustrated in Figure 5.4. Among the M red points, m are then randomly picked to construct a stencil of size $m + 3$. In addition to reducing the dependency on probe grid resolution, we also perform *data augmentation*. Indeed, for one location on the wall, we can construct $\frac{M!}{m!(M-m)!}$ stencils.

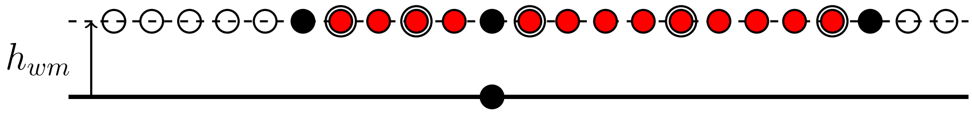


Figure 5.4: Sketch of a stencil where the three black points are fixed and the 5 red circle points are randomly selected among the 13 red points.

For the development of new WSS models, two neural networks were selected: an MLP and a CNN for the prediction of $\boldsymbol{\tau}_w$ where the output layer is replaced by Gaussian Mixture Networks or Heads (GMH). One head is selected to predict the distribution of $\tau_{w,\xi}$ and another for $\tau_{w,z}$ (i.e., the wall-parallel components of the wall shear stress). Each head has its number K of Gaussian distributions. The component $\tau_{w,\eta}$ is set to zero as in many WSS models (i.e., non-penetration condition). The subsections 5.1.2.1 and 5.1.2.2 further detail these two architectures employed to develop new data-driven WSS models.

5.1.2.1 MLP coupled with GMH

As presented in Chapter 2, the MLP is the simplest fully-connected neural network. In the present work, the output layer is removed and replaced by two Gaussian Mixture Heads (GMHs): one for the prediction of $\tau_{w,\xi}$ and one for $\tau_{w,z}$. Each head has its own number K of Gaussian distributions, and the loss function results from the sum of two Negative Log Likelihood. The tuning of the various hyperparameters (e.g., number of hidden layers, activation function, and learning rate) are treated in Section 7.1.

5.1.2.2 CNN coupled with GMH

When the input stencil is enlarged, a CNN is prefered to an MLP. As discussed in Section 5.1.2.1, the output layer of the CNN is also removed and replaced by two GMHs for predicting of the distribution of $\tau_{w,\xi}$ and $\tau_{w,z}$, respectively. The CNN is not used in its standard version as described in Chapter 2. Residual blocks are introduced in the architecture to easily train deep neural networks.

Skip connections in Deep Learning By increasing the input stencil and the number of fields, the relationship between the inputs and the outputs becomes more complex. Deep neural networks are therefore attractive because they can learn complex functions more efficiently than shallow networks. However, training them is a complicated and computationally expensive task. Additionally, as the architecture of the model becomes deeper, its performance decreases, which is known as the degradation problem. This issue is not caused by overfitting or vanishing gradient problems, but rather the challenge of optimizing a larger number of parameters. This problem has been solved by introducing skip connections in the neural network architecture (i.e., ResNet by He *et al.* (2015)). In this work, a succession of residual blocks composed of convolutional layers is implemented. The network (see Figure 5.5) is therefore composed of three convolutional blocks, two residual blocks, a linear block to switch from the convolutional part to the MDN and two Gaussian Mixture Heads (described in Section 5.1.1). The detailed implementation of these residual blocks is given in Figure 5.5 (on the right) under the nomenclature of `PyTorch`. To ensure that the data coming directly from the block entry and the two convolution layers (in the pink rounded rectangle in Figure 5.5) are of the same size, the padding (i.e., an hyperparameter of the convolutional layer presented in Section 2.2.2) is set to one, which in this case is equal to `kernel_size`//2. This padding argument corresponds to the number of *zero-padding* to be added to the edges of the data during the computation.

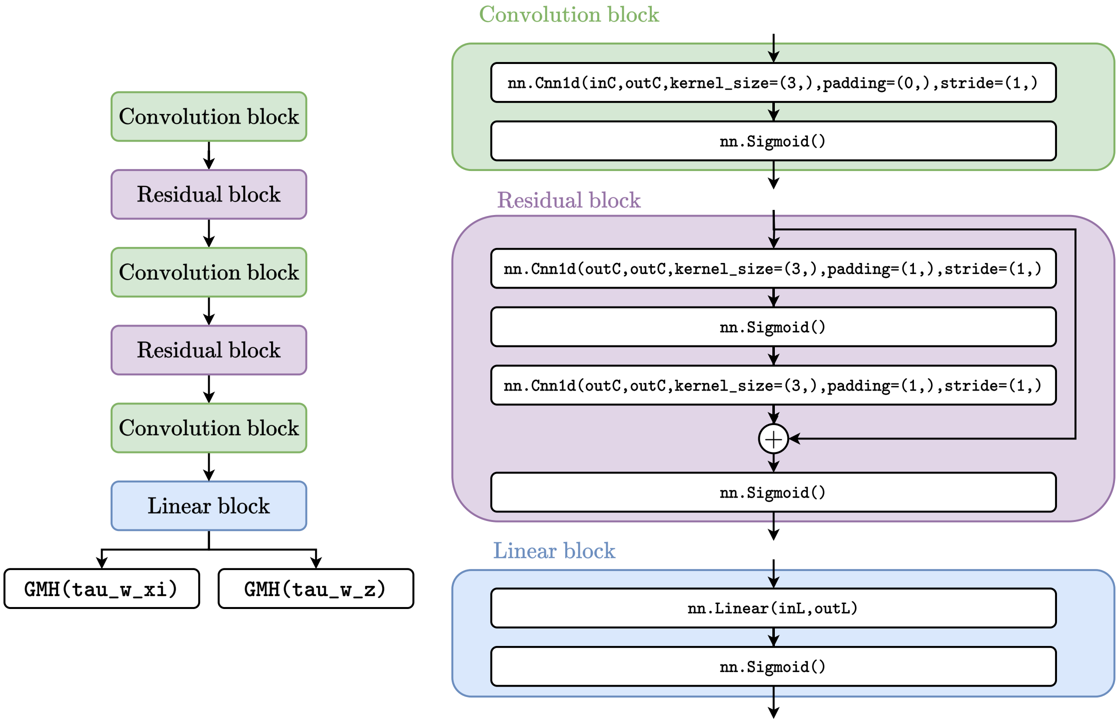


Figure 5.5: On the left, the architecture of the CNN-skip-GMH composed of residual and convolutional blocks; on the right, schematics of the individual blocks composing the CNN architecture: a residual block in light pink (i.e., skip connections), a convolutional block in light green and a linear block in light blue; architecture defined under the nomenclature of `PyTorch`; for reproducibility we use `Python` 3.10.4 with `numpy` version 1.23.1 and `torch` version 1.12.1.

5.2 Data manipulation and preprocessing

Manipulating data acquired by Direct Numerical Simulations with Argo-DG (see Chapter 4) before feeding them into the model is a crucial step. In the present case, the raw data are the velocity, pressure gradients, and wall shear stress fields extracted from the high-order solutions on structured probe grids. Database preprocessing concerns all the necessary steps that make the three-dimensional time-dependent flow fields ready to be treated by the machine learning process. This step can encapsulate filtering, cleaning, non-dimensionalization procedure, data structuring under an adapted format, and data augmentation. This preprocessing will impact the capabilities of the trained neural network to generalize to unseen configurations, and it can bring desired invariance to the model (such as Mach number invariance, and rotationally invariant, to name but a few).

Filtering is often used in wall models to ensure that the flow field will resemble the one observed during the wmLES. This step is even more frequent in data-driven approach (Yang *et al.*, 2019; Zhou *et al.*, 2021; Dupuy *et al.*, 2023a,b). However, our data are not filtered because the interpolation of the probe grid already acts as a filter. Even if the filtering is implicitly employed, the non-dimensionalization, on the other hand, is explicitly applied to both the inputs and outputs and is presented in Section 5.2.1.

Data cleaning is also a preprocessing step that aims to identify any issues or incorrect data. A first effective technique is to remove duplicate data. Indeed, duplicates can skew and confuse the resulting model. Irrelevant data can also be removed as they may slow down the training process. These irrelevant data are features that are uncorrelated to the model output. A third effective technique for cleaning data is to remove unwanted outliers. Outliers are unusual data values. They are induced by sampling errors, natural variations, and data entry errors. Outliers can alter statistics. Nevertheless, removing outliers should be carefully analyzed. Therefore, the Probability Density Functions (PDF) of the two-wall parallel wall shear stress are studied to detect outliers in Section 5.2.2.

5.2.1 Data non-dimensionalization

Selecting the proper input features is crucial in ML and DL. The non-dimensionalization of the input features is especially helpful for training a model on a limited dataset that will then be able to generalize to flows with different length scales, velocity scales, and fluid properties. In fluid dynamics, one is used to dealing with non-dimensional quantities that use relevant flow quantities, such as the boundary layer height, the freestream velocity, etc. Unfortunately, these quantities may not be the most relevant for generalization purposes and are not always available during the computation. The idea is to encode *a priori* knowledge about the underlying data distribution using physically based non-dimensionalization. The latter has to deal with quantities directly available for the the wall model during the wmLES such as the **density** ρ [kg/m^3], the **kinematic viscosity** ν [m^2/s], and the **wall**

model height h [m]. The inner scaling of the boundary layer is a great choice for the database non-dimensionalization. Nonetheless, the wall shear stress is not available to scale the input. However, the product y^+u^+ (i.e., $y^+u^+ = \frac{yu_\tau}{\nu}\frac{u}{u_\tau} = \frac{u\,y}{\nu}$) is independent of u_τ and can be used to scale the input data. Moreover, the relationship between y^+u^+ and y^+ is nearly linear for a turbulent boundary layer. Frère (2018) has used a similar trick to tabulate Reichardt's velocity profile and thus gets a faster evaluation of the model in the high-order flow solver Argo-DG. The relation y^+u^+ is illustrated in Figure 5.6 where u^+ is extracted from Eq. 3.6.

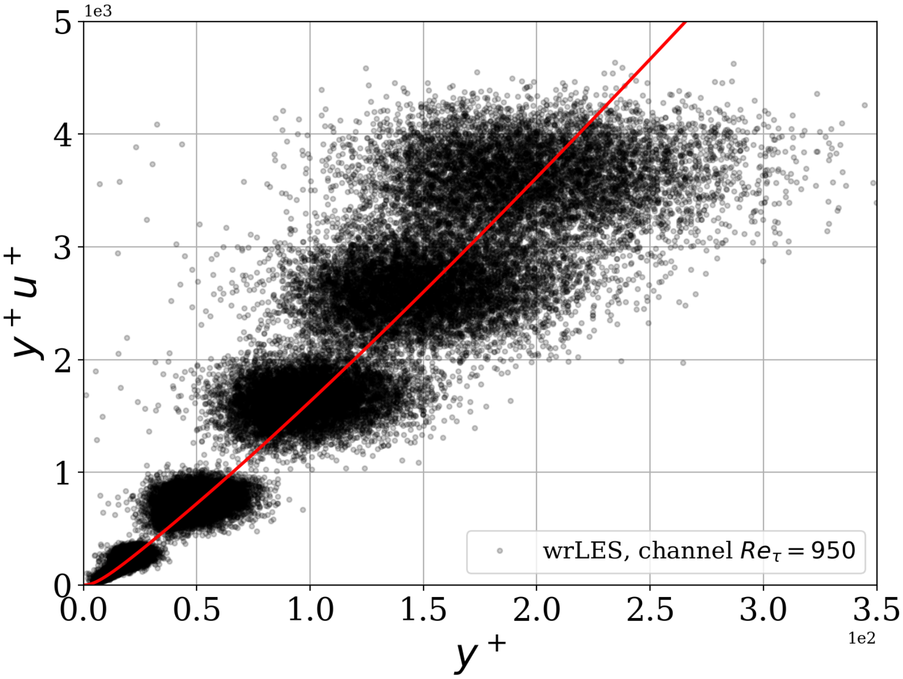


Figure 5.6: Reichardt's velocity profile multiplied by y^+ is drawn in red while the black markers represent the wrLES scaled data of the turbulent channel flow at $Re_\tau = 950$.

The product y^+u^+ is nothing else than a local Reynolds number. Hence, h_{wm} and ν serve as a velocity scale (i.e., ν/h_{wm}) to scale the velocity field accordingly. The scaled velocity $\boldsymbol{u}^*$ is defined as,

$$\boldsymbol{u}^* = \frac{\boldsymbol{u}\,h_{wm}}{\nu}\,, \tag{5.5}$$

where $\boldsymbol{u}$ is the velocity extracted at h_{wm}, and projected in the curvilinear coordinates (ξ, η, z). The wall-model height is not used as a model input because it is implicitly used in the non-dimensional velocity. Regarding the pressure gradient, it is first defined as the pressure velocity $\boldsymbol{u}_p$ and then the velocity scale ν/h is applied:

$$\boldsymbol{u}_p = \left(\frac{\nu}{\rho}\nabla p\right)^{1/3} ,$$

then this velocity is scaled as in Eq. 5.5 to get,

$$\boldsymbol{u}_p^* = \frac{\boldsymbol{u}_p\, h_{wm}}{\nu} . \tag{5.6}$$

This pressure velocity is used in algebraic wall models that use of the pressure gradient (see Eq.3.17 in the thesis of Thiry (2017)). This pressure velocity is inspired by the definition of Clauser's parameter. Separation can occur on a flat plate due to an adverse pressure gradient, or on a curved wall where both the curvature and the pressure gradient affect the boudary layer state. Therefore, the curvature appears to be an interesting input to be added to the WSS model and it is non-dimensionalized with the wall model height,

$$\mathcal{K}^* = \mathcal{K} h_{wm} , \tag{5.7}$$

where $\mathcal{K}$ is computed as

$$\mathcal{K} = \left(\frac{|\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}|}{1 + \left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)^2}\right)^{3/2} , \tag{5.8}$$

where $f(x)$ is the function that describes the wall. Concerning the outputs, the two wall-parallel components of wall shear stress are scaled as a signed version of h^+:

$$h_i^+ = \mathrm{sign}\left(\tau_{w,i}\right) \frac{h_{wm}}{\nu} \sqrt{\frac{|\tau_{w,i}|}{\rho}} , \tag{5.9}$$

where $i = \{\xi, z\}$. The interesting fact behind using a square root for the non-dimensionalization of the wall shear stress is that this variable transformation reduces the distribution skewness. Because the skewness is not an explicit output (or model distribution parameters), the GMN may struggle to predict this third statistical moment correctly. Therefore, reducing the skewness before training is of interest. Moreover, Section 5.2.2 indicates how important the skewness is to recover the expected mean.

In addition to the non-dimensionalization, the data need to be scaled or standardized for the training. The scaling will constraint a given feature ϕ to lie between 0 and 1 by using its minimum and maximum values as,

$$\phi^{\ddagger} = \frac{\phi - \min(\phi)}{\max(\phi) - \min(\phi)} \,. \tag{5.10}$$

Assuming that the feature ϕ follows a normal distribution, then this feature ϕ can be standardized using its mean μ and its standard deviation σ to rescale its distribution with a zero-mean and a unitary-variance. Figure 5.7 illustrates the different steps executed before and after the forward pass, which are dedicated to non-dimensionalization and scaling processes of the inputs and outputs.

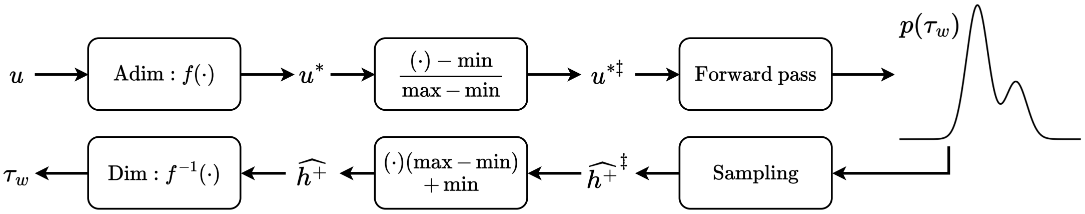


Figure 5.7: Diagram illustrating the non-dimensionalization of the input, its scaling using the minimum and the maximum, the forward pass, the sampling from the predicted distribution, the rescaling using the output minimum and maximum and the dimensionalization of the predicted wall shear stress.

Discussion on the non-dimensionalization. Table 5.1 reviews the nondimensionalized input/output pairs of existing studies on data-driven WSS models from the literature that are known to the authors. The table includes neither hybrid RANS/LES models nor the work of Lee *et al.* (2023), which used the Fukagata-Iwamoto-Kasagi identity, a decomposition of the skin friction, to predict the wall shear stress. For the sake of clarity and comparison, the input and output terms are adapted to our nomenclature. The reader is referred to the original papers for more information.

Bae and Koumoutsakos (2022) and Zhou *et al.* (2022) developed new WSS models with reinforcement learning (RL). The training approach contrasts with standard

ML and DL because the network parameters are fitted within the production environment, allowing the model to predict the WSS while also correcting for numerical errors. In this setting, the WSS predicted at the previous time step is used to nondimensionalize the input features. Using the instantaneous friction velocity for standard ML and DL is not the best choice. Although the neural network can predict the wall shear stress with a relative error of $1-2\%$, the prediction may rapidly deteriorate in the production environment due to the error being re-injected into the input. Yang *et al.* (2019) also used the friction velocity in the definition of y_0. In their paper, it is not clear how the friction velocity is extracted from the wmLES. They use the mean friction velocity, which is known *a priori* for turbulent channels. However, this approach cannot be generalised to other flow configurations where the WSS depends on the positions or is not known *a priori*.

Yang *et al.* (2019), Zhou *et al.* (2021), and Zhideng *et al.* (2023) used the logarithm of the wall distance to improve the generalizability of their models in equilibrium boundary layers. The input feature was originally proposed by Yang *et al.* (2019), where the friction velocity is employed. Zhou *et al.* (2021), and Zhideng *et al.* (2023) instead used an approximation of the friction velocity and extended its definition to the pressure gradients (i.e., $u_{\nu,p}$) to include separated boundary layers. However, in such a nondimensionalization, the velocity is scaled with *itself* through the definition of $u_{\nu,p}$. This may lead to numerical sensitivity issues (see discussion below).

Jamaat and Hattori (2023) decided not to nondimensionalize their input feature because their study focused on the *a priori* assessment of non-local WSS models for the turbulent channel flows based on two-dimensional CNNs. Their study was not aimed at generalising their model to new flow configurations.

Lozano-Durán and Bae (2021) nondimensionalized the input features with flow-specific scales that are case-dependent. Moreover, the nondimensionalization *a posteriori* over a complex geometry requires great care and knowledge about the type of flow physics appearing on each surface. As already mentioned, our nondimensionalization only uses local and instantaneous flow quantities that are directly available to the wall models during the wmLES. Moreover, it is common to every flow configuration.

Radhakrishnan *et al.* (2021) use both dimensional and nondimensional input feature to feed their network. Similarly to what is done here, they defined a local

Reynolds number. Their input feature also contained the logarithm of this local Reynolds number. They used an approximation of the friction velocity by computing the shear stress τ to non-dimensionalize the velocity. Concerning the output, their model has to predict the velocity in wall units u^+. This output is maybe not the best choice for separated flow where the friction velocity tends to zero.

Regarding the nondimensionalization of the velocity field, we have the same definition as Dupuy *et al.* (2023a), which used the local Reynolds number based on the viscosity and the wall distance, except that the velocity field is not projected onto the local frame of reference created by the wall. The authors chose not to explicitly feed derivatives to the network. They claimed that feeding the model with spatial information was sufficient for the model to compute these derivatives internally if required. The Dupuy *et al.* (2023a) model predicted the sign version of the square wall distance in wall units (i.e., $\text{sign}(h^+_{wm})(h^+_{wm})^2$), where our model directly predicts the wall distance in wall units h^+_{wm}. This choice is related to the reduction in the skewness of the predicted distribution due to the square root.

Before finding the nondimensionalization $(\cdot)^*$, other nondimensionalizations were tested for the training of the network MLP-GMH on the turbulent channel flow at $Re_\tau = 5{,}200$, and the following observations were made.

- If the nondimensionalization depends on the *wall-modeled heights*, the model will generalize to other Reynolds numbers as long as the data are extracted within the training bounds (i.e., $[\min(h_{wm}), \max(h_{wm})]$). This observation is confirmed even if h_{wm} is not an explicit model input.
- If the nondimensionalization depends on *global quantities* (such as the bulk velocity), the model will be able to generalize to new h_{wm} if the Reynolds number is kept within the training bounds (i.e., $[\min(Re), \max(Re)]$).
- If the process of nondimensionalization relies on data that is computed by the solver during the wmLES, such as the velocity u in the definition of $u_{\nu,p}$, the sensitivity of the model to the polynomial order at which it is trained will increase. This is especially true if the data used for training the network is extracted from a wrLES, rather than a pure DNS which should be independent of the numerical scheme.

Table 5.1: Review of the non-dimensionalised inputs and outputs in the literature on data-driven wall models; the notation is adapted to our nomenclature for clarity: $y_0 = \frac{\nu}{u_\tau}\exp(-\kappa B)$; $u_{\tau,p} = \sqrt{u_\tau^2 + u_{w,p}^2}$ is a friction velocity based on the wall shear stress and the pressure gradient at the wall (Duprat *et al.*, 2011) where $u_{w,p} = \left((\nu/\rho)\partial p_w/\partial\xi\right)^{1/3}$; $y^* = \nu/u_{\nu,p}$ where $u_{\nu,p} = \sqrt{u_\nu^2 + u_p^2}$ and $u_\nu = \sqrt{\nu u_\xi/h_{wm}}$; τ is the shear stress; $\tilde{u}$ is the relative velocity w.r.t. the wall; $\tilde{S}_{ij}$ is the rate of deformation tensor; $\tilde{\Omega}_{ij}$ is the rate of rotation tensor; U_∞ is the free-stream velocity; $\mathcal{U}_k$ is a velocity scale that depends on the building blocks considered (e.g., $\mathcal{U}_1 = \mathcal{U}_2 = u_\tau$, and $\mathcal{U}_3 = u_p$); $\mathcal{L}_k$ is a length scale that also depends on the building blocks considered (e.g., $\mathcal{L}_1 = h_{wm}$, $\mathcal{L}_2 = \mathcal{U}_2/\|\nabla u\|$, and $\mathcal{L}_3 = \nu/\mathcal{U}_3$); $u_\tau^m = \sqrt{\tau_w^m(x,z,t)/\rho}$ is the modeled friction velocity where $\tau_w^m(x,z,t)$ is the modeled WSS; $|u_\parallel| = (u_\xi^2 + u_z^2)^{1/2}$ is the norm of the wall-parallel velocity.

Authors	**Inputs**	**Outputs**
Bae and Koumoutsakos (2022)	$\dfrac{u_{x,y,z}}{u_\tau^m(x,z,t)}$	τ_w
Yang *et al.* (2019)	$\dfrac{\lvert u_\parallel\rvert}{h_{wm}}, \dfrac{h_{wm}/y_0}{\lvert u_\parallel\rvert}$	$\lvert\tau_{w,\xi}\rvert$
Jamaat and Hattori (2023)	$u_{x,y,z}$ **or** $u_x, \dfrac{\partial u_x}{\partial y}, u_x u_y, h_{wm}$	$\tau_{w,x}$, $\tau_{w,z}$
Lozano-Durán and Bae (2021)	$\dfrac{\tilde{u}_{\xi,\eta,z}}{U_\infty}, \dfrac{\tilde{u}_{\xi,\eta,z}}{\mathcal{U}_k}, \dfrac{\tilde{S}_{ij}\mathcal{L}_k}{\mathcal{U}_k}, \dfrac{\tilde{\Omega}_{ij}\mathcal{L}_k}{\mathcal{U}_k}, \dfrac{U_\infty h_{wm}}{\nu}, \dfrac{\mathcal{U}_k\mathcal{L}_k}{\nu}, \ldots$	$\dfrac{\boldsymbol{\tau}_w}{\rho U_\infty^2}$
Zhou *et al.* (2022)	$\dfrac{u_\xi}{u_{\tau p}}, \dfrac{h_{wm}u_{\tau p}}{\nu}, \dfrac{1}{2}\left(\dfrac{\partial u_\xi}{\partial\eta} + \dfrac{\partial u_\eta}{\partial\xi}\right)\dfrac{h_{wm}}{\nu}$	τ_w
Zhou *et al.* (2021)	$\ln\left(\dfrac{h_{wm}}{y^*}\right), \dfrac{u_{\xi,\eta}}{h_{wm}}, \dfrac{\partial p}{\partial\xi,\partial\eta}\dfrac{h_{wm}}{h}$	$\tau_{w,\xi}$
Dupuy *et al.* (2023a)	$\dfrac{\rho h_{wm}}{\mu}u_{x,y,z}$	$\rho\left(\dfrac{h_{wm}}{\mu}\right)^2\tau_w$
Zhideng *et al.* (2023)	$\ln\left(\dfrac{h_{wm}}{y^*}\right), \dfrac{u_{\xi,\eta,z}}{h_{wm}}\dfrac{\delta}{u_b}, \dfrac{\partial p}{\partial\xi,\partial\eta}\dfrac{h_{wm}}{u_b^2}$	$(\tau_{w,\xi}, \tau_{w,z})/u_b^2$
Radhakrishnan *et al.* (2021)	$u_x, h_{wm}, Re^* = \dfrac{u_x h_{wm}}{\nu}$, $\ln(Re^*)$, $\dfrac{u_x}{\sqrt{\tau}}$	u^+
Boxho (2024)	$\boldsymbol{u}^*$, $\boldsymbol{u}_p^*$, $\mathcal{K}^*$	h_ξ^+, h_z^+

5.2.2 Data cleaning

The process of extracting valid, previously unknown, and ultimately comprehensible information from large datasets is an important step before making decisions or predictions. However, this step is fraught with problems such as data redundancy, incomplete data and outliers. Duplicates are highly unlikely for turbulent flows that have a chaotic behavior. Nonetheless, highly correlated features in space or time are possible. Hence, the streamwise, spanwise, and time sampling must be chosen carefully. Outlier detection therefore remains. An outlier is a data point that differs from the rest of the data according to a particular measurement. This point may be noise or it may be an indication of abnormal system behaviour. Outlier detection is an interesting problem of data mining (Aggarwal and Yu, 2005; Bakar *et al.*, 2006). In the literature review of Chapter 3, none of the authors have discussed outlier detection. They have mostly filtered (in space) their database to train their network with a field compatible with the wmLES one. However, discussing the quality of the database is just as important as training the neural network correctly. If the database is not representative of the phenomenon we want to capture, the neural network will not be capable of recognizing the underlying relationship between the inputs and outputs. Outlier detection can be divided into three approaches: (i) statistical approach (Chatterjee *et al.*, 2021), (ii) distance-based approach, and (iii) deviation-based approach. For one-dimensional samples, the statistical approach is well-suited. As the dimensionality increases, the distance-based approach is preferred because it becomes difficult to estimate the multidimensional distributions of the data points (Ramaswamy *et al.*, 2000).

Table 5.2: Statistical moments of the streamwise wall shear stress extracted on the turbulent channel at $Re_\tau = 5{,}200$

Data	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$	$\boldsymbol{\mathcal{S}}$	$\boldsymbol{Q_{25\%}}$	$\boldsymbol{Q_{75\%}}$	$\boldsymbol{IQR/k}$	**min**	**max**
TC5200	1.005	0.444	1.153	0.683	1.241	0.559	-0.849	4.102

Outlier detection is illustrated for the turbulent channel flow at $Re_\tau = 5{,}200$ (see Table 5.2) on the streamwise wall shear stress. Because it is a one-dimensional random variable, the statistical method is selected. The *conventional standard deviation method* cannot be applied because the current distribution is not Gaussian-like. Therefore, the interquartile range (IQR) method, which is part of the statistics-based outlier detection techniques, is selected. These techniques assume that *normal* data occur in the high probability regions of a stochastic model, while outliers occur in the low probability region (Ilyas and Chu, 2019). The IQR method uses quantiles to detect the low-probability region. The cut-off limit is proportional to a factor k of the difference between the quantiles at 25 and 75%: $IQR = k(Q_{25\%} - Q_{75\%})$. The lower and upper cutting values are $Q_{25\%} - IQR$ and $Q_{75\%} + IQR$, respectively. The method gives the results summarized in Table 5.2 for the turbulent channel. For $k = 1.5$, data lower than -0.155 or greater than 2.079 are considered outliers. However, the resulting database mean ($\mu = 0.969$) does not correspond to the expected mean of 1. By eliminating these supposed outliers, the first moment of the resulting distribution has changed drastically.

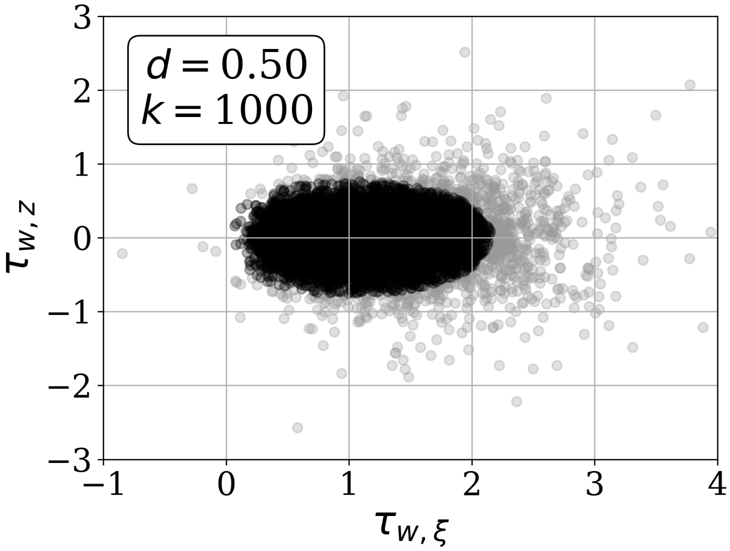


Figure 5.8: Distance-based approach using the K^{th} Nearest Neighbors on the random variable $\boldsymbol{\tau}_w$ of the database TC5200.

The distance-based approach is also applied to the two-dimensional random variable $(\tau_{w,\xi}, \tau_{w,z})$, which are the two outputs of our WSS model. A simple distance-based approach is the K-Nearest Neighbors Distance method (Upadhyaya and Singh, 2012), which measures the distance between a point and its K nearest neighbors. A threshold distance d is determined based on the maximum distance value. Then, the outlier score of a point is the number of nearest neighbors that are not more than d distance away from the given point in the dataset. This technique is based on the function NearestNeighbors from sklearn.neighbors of the library sklearn of Python. The number of nearest neighbors k is fixed to 1000. The distance to the farthest neighbors is sorted, and all points with a distance greater than $d = 0.5$ are considered as outliers. Figure 5.8 shows the outliers in light gray and the retained points in black. The resulting mean of the streamwise wall shear stress is 0.961. Again, the mean is modified by removing possible outliers. These data are part of the physics and cannot be removed.

5.3 Development environment and practical implementation

This work uses Argo-DG, the in-house flow solver developed at Cenaero, to produce both wall-resolved and wall-modeled LES. Section 5.3.1 explains the physical modeling and numerical methods employed to simulate the turbulent channel flows and the two-dimensional periodic hill. Additionally, Section 5.3.2 details the implementation of wall models in the DG discretization framework. The specific challenges and constraints of this implementation are reviewed. For a more comprehensive overview of Argo-DG, please refer to the works of Hillewaert (2013), Carton de Wiart (2014), Schrooyen (2015), and Frère (2018).

5.3.1 Description of Argo-DG code

Spatial discretization. The compressible Navier-Stokes equations are discretized using the discontinuous Galerkin method, which is a special class of Galerkin finite element methods (FEM). In such a method, a linear combination of N shape functions (e.g., Lagrange polynomials of degree p, as in Argo-DG) approximates each variable of the solution vector. Conversely to standard FEM, these shape functions are not required to be continuous at the element interfaces. As a result, the global problem (i.e., over the entire computational domain) can be decomposed into smaller sub-problems at each element level, and therefore, the global solution is evaluated as a sum over each of these elements independently. Unfortunately, this particularity drastically increases the number of degrees of freedom (dofs). In fact, the dofs at the interface are doubled because they are present in both the current element and its neighbor. However, this particularity makes the method highly scalable, resulting in high CPU efficiency. Once the problem has been defined at the element level, the Galerkin variational principle is used to solve the *elementwise* Navier-Stokes equations. Note that the interface fluxes do not cancel because of the discontinuities between elements. As a result, the equations include four additional terms that connect elements. To stabilize the formulation, the convective fluxes are treated using an approximate Riemann solver. For this study, Roe's flux difference splitting is used. One of those four additional terms stabilizes the diffusive scheme, while another provides a symmetric formulation that agrees with the Symmetric Interior Penalty formulation (Arnold *et al.*, 2002). The formulation is a generalized version of the boundary penalty method, which effectively enforces weak Dirichlet boundary conditions at interfaces. The variational formulation's integral terms are treated using the Gauss-Legendre quadrature rule. By applying a mapping, the problem is solved in the reference element rather than the global frame element. The motivation behind this method is that high-order methods have really low dissipation and dispersion errors, which is a very interesting property to accurately capture the energy spectrum. Moreover, the method is seen to be very robust and can handle badly-shaped mesh elements without losing accuracy (Carton de Wiart, 2014).

Implicit LES. Large Eddy Simulation (LES) is a mathematical model for turbulence that reduces the computational cost by ignoring the smallest scales. These scales are particularly computationally expensive to resolve as they necessitate finer meshing to be explicitly resolved. LES resolves a large part of the energy spectrum (i.e., energy-contained and inertial ranges) up to a cut-off wave number, defined by the effective mesh size (i.e., account for the polynomial order p in high-order flow solver). To achieve this, a low-pass filter is applied to the Navier-Stokes equations to remove the smallest scales from the numerical solution efficiently. This low-pass filter can be viewed as a time and space-averaging. To compensate for the *non-resolution* of the smallest scales, a sub-grid scale (SGS) model is introduced to model for the effects of the unresolved small scales on the resolved larger scales. Argo-DG does not implement any explicit SGS model. Instead, it is the numerical dissipation of the underlying high-order DG scheme that acts spectrally as an explicit SGS model (Carton de Wiart *et al.*, 2014) used in *traditional* LES. Therefore, Argo-DG relies on the implicit LES (ILES) approach.

Time integration. Multiple time discretizations are implemented in Argo-DG (e.g., BDF2, Runge-Kutta, Euler implicit, ...). All simulations (i.e., wrLES and wmLES) performed with Argo-DG in this work use the second-order implicit time-stepping procedure based on the Backward Differentiation Formula (BDF2). At each time step, a non-linear problem resulting from the implicit integration is solved through a Newton/GMRES method, preconditioned with elementwise block-Jacobi. Even though the time step can be *larger* than with explicit time discretization, the time step is adjusted to obtain a convective CFL lower than one for a good resolution of the time structures.

Additional information. The code can operate on large cases thanks to the high scalability of the DG method. It implements a hybrid parallelism based on *message passing interface* (MPI) and *open multi-processing* (OpenMP). It can be shown that the DG discretization achieves low dispersion and dissipation errors. Moreover, on unstructured meshes, in the L_2-norm, it offers $\mathcal{O}(h^{p+1})$ accuracy. Figure 5.9 shows the different iterative processes implemented in Argo-DG and how these processes are coupled with the data-driven wall model presented in Section 5.3.2.

5.3.2 Implementation of a data-driven WSS model in Argo-DG

Sections 5.1 and 5.2 discussed the type of neural network architecture chosen for the development of new WSS models and how to preprocess with the acquired databases. Chapters 7 and 8 present the training of these neural networks and their *a priori* and *a posteriori* tests. While the *a priori* test does not require any modifications compared to the training, the *a posteriori* test involves the implementation of the neural network in the flow solver. This implementation of the new data-driven WSS model in Argo-DG required several steps, described in Figure 5.10. It includes,

STEP 1 the **interpolation** of the input features on the volume probes,

STEP 2 the creation of the **input stencil**,

STEP 3 the **forward pass** through the network,

STEP 4 the **sampling** from the predicted distribution and the **rescaling** of the predicted wall shear stress, and

STEP 5 the **interpolation** of the predictions from the surface probes to the quadrature points.

The Python programming language is the primary interface of `PyTorch`. `Python` is known to be suitable for many scenarios that require dynamism and ease of iteration. However, there are situations where these properties are unfavorable. `C++`

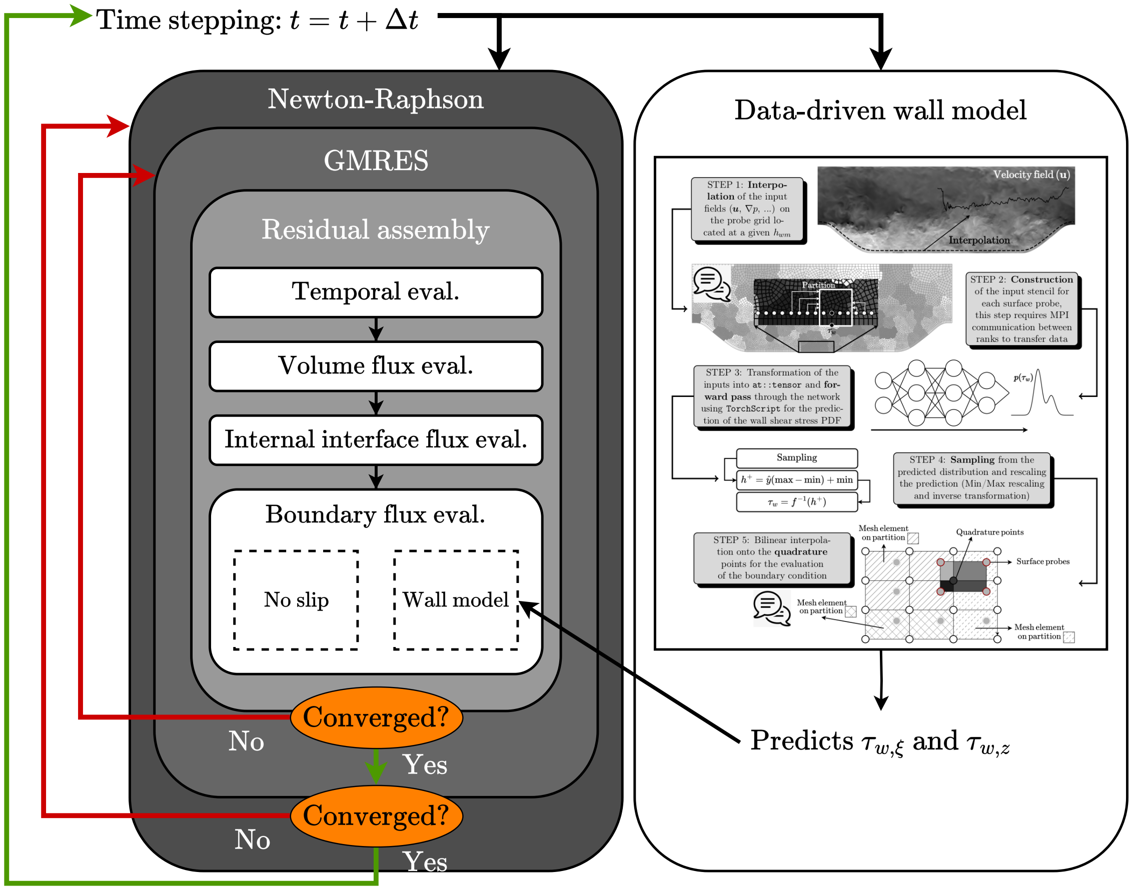

Time stepping: $t = t + \Delta t$
Newton-Raphson
GMRES
Residual assembly
Temporal eval.
Volume flux eval.
Internal interface flux eval.
Boundary flux eval.
No slip
Wall model
Converged?
No
Yes
Converged?
No
Yes
Data-driven wall model
STEP 1: Interpolation of the input fields ($\boldsymbol{u}$, ∇p, ...) on the probe grid located at a given h_{wm}
Velocity field ($\mathbf{u}$)
Interpolation
Partition
τ_w
STEP 2: Construction of the input stencil for each surface probe, this step requires MPI communication between ranks to transfer data
STEP 3: Transformation of the inputs into at::tensor and forward pass through the network using TorchScript for the prediction of the wall shear stress PDF
$p(\tau_w)$
Sampling
$h^+ = \hat{y}(\max - \min) + \min$
$\tau_w = f^{-1}(h^+)$
STEP 4: Sampling from the predicted distribution and rescaling the prediction (Min/Max rescaling and inverse transformation)
STEP 5: Bilinear interpolation onto the quadrature points for the evaluation of the boundary condition
Mesh element on partition
Quadrature points
Surface probes
Mesh element on partition
Mesh element on partition
Predicts $\tau_{w,\xi}$ and $\tau_{w,z}$

Figure 5.9: Summary of the different iterative processes used to resolve the Navier-Stokes equations in Argo-DG with or without data-driven wall model; the figure in the data-driven wall model block is further described in Figure 5.10; this diagram is inspired by the work of Frère (2018).

is often the language of choice for production scenarios. The implementation of a PyTorch's model in a C++ environment is made possible by the use of TorchScript.

The very first step to do is to create a trace of your PyTorch's model. This step removes any Python dependencies and embeds the forward pass of your model for inference in the production environment. The resulting model is saved as a binary file and kept for future use. In our model, the scaling and non-dimensionalization procedures of the input stencil are implemented in the trace and are, therefore, invisible to the user. This step reduces the amount of code in production and is less prone to errors.

Implementation of the WSS model in Argo-DG can now be reviewed. As mentioned in Section 5.3.1, this solver is highly scalable. To avoid any deterioration of the code's capabilities, the new WSS model needs to be fast to evaluate and local to reduce the amount of MPI communication between ranks. Based on space-time correlations (Chapter 4), the input stencil is enlarged to capture the correlation high peak (see discussion in Section 5.1.2) and help for the discrimination between various flows physics as also observed by Dupuy *et al.* (2023a,b).

Due to this increase in the number of entries, the number of MPI communications increases too, compared to the model implemented by Frère (2018), which uses the ghost-cells of the near-wall element to avoid MPI communication. Another difference to the model (Frère, 2018) is that the wall shear stress is not predicted directly at the quadrature points. The predictions are first evaluated on probes located on the target surface. These probes are obtained as the orthogonal projection of volume probes on the solid wall. There are two sets of probes. The volume probes are used to interpolate the input fields and create the input stencil. After the forward pass, the predictions lie on the surface probes and are interpolated to the quadrature points to evaluate the boundary condition.

Figure 5.10 schematizes the implementation of the new WSS model in Argo-DG. This implementation is divided into **five major steps**.

The **first step** is the interpolation of the required input fields (e.g., velocity, pressure gradients, density, and viscosity) onto the volume probes. These probes are located at a certain wall-modeled height h_{wm}, specified by the user, and spread on several partitions.

Once each rank (i.e., mesh partitions) knows its input fields, it communicates with its neighbors to construct the entire input stencil. This **second step** is the construction step. The second insert image of Figure 5.10 shows how the wmLES mesh of the periodic hill is partitioned. For the prediction of τ_w at the black dot, the rank knows only two downstream points and one upstream point. In this simplified example, its right neighbor communicates two points while its left neighbor communicates four points. The amount of communication is directly proportional to the input stencil size and the number of partitions. Since the mesh does not move, an initialization procedure constructs and stores a table of communications that associates the probe index to the rank to send to.

Once each rank has created its input, it can be sent to the model for inference in the **third step**. The forward pass produces the distribution of the wall-parallel components of the wall shear stress as a sum of K Gaussians distributions (as discussed in Section 5.1.1).

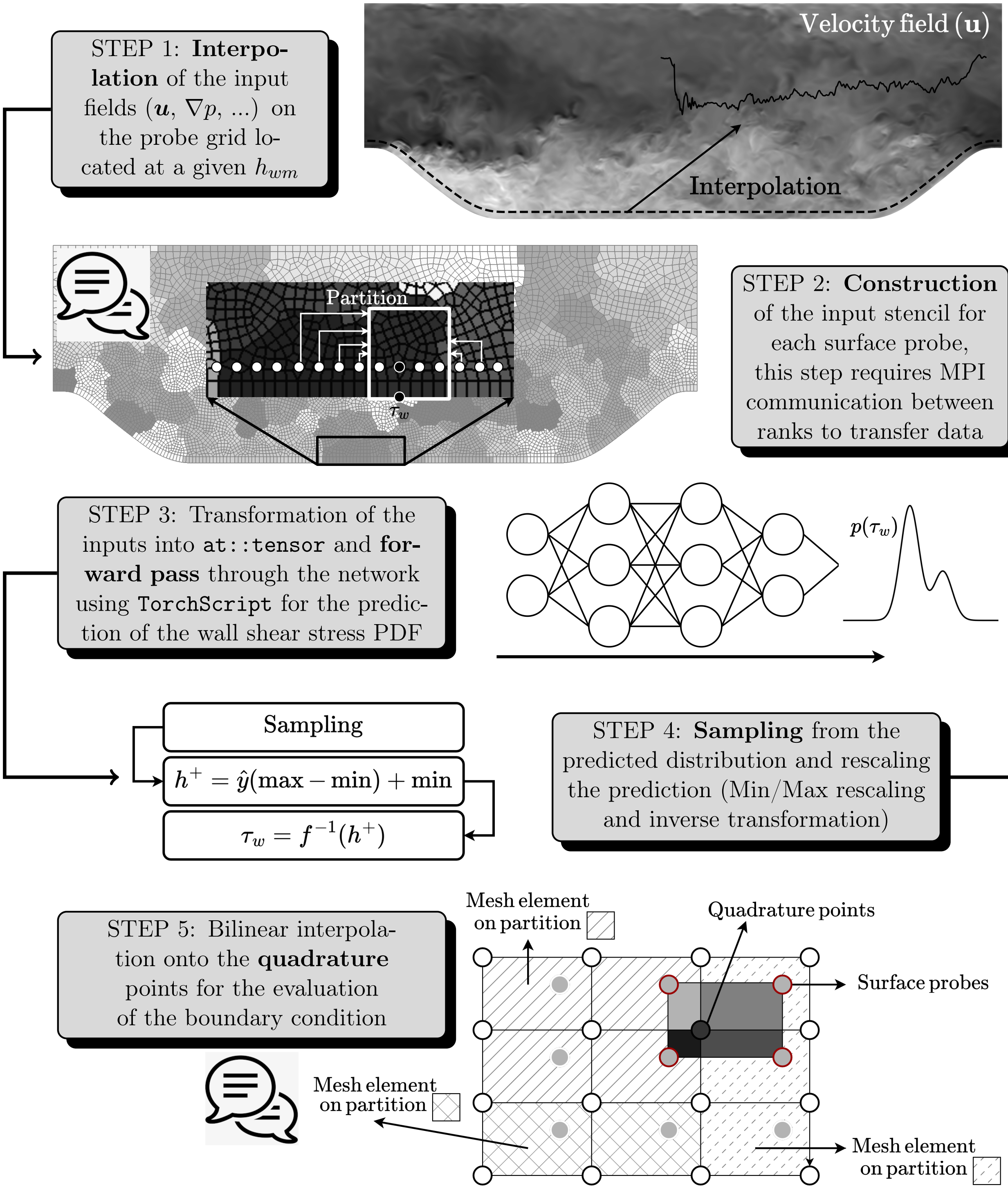
The **fourth step** is dedicated to the sampling from the predicted distribution based on a vector of K means (`mu`), K standard deviations (`stddev`) and K coefficients (`weights`) as described in the code snippet 5.1. The predictions are then rescaled according to the inverse of Equation 5.9.

Listing 5.1: Sampling from a mixture of K Gaussian distribution

```cpp
float sampling(const std::vector<float>& mu,
               const std::vector<float>& stddev,
               const std::vector<float>& weights) {
  std::random_device rd;
  std::mt19937 generator = std::mt19937(rd());
  std::discrete_distribution<size_t> w(weights.begin(),
                                       weights.end());
  std::size_t index = w(generator);
  std::normal_distribution<float> d(mu[index], stddev[index]);
  return d(generator);
}
```

Finally, in the **fifth step**, the predictions lying on the surface probes are interpolated onto the quadrature points. This step also requires MPI communications for quadrature points located at the interface of two partitions. As for the construction of the input stencil, the initialization procedure also stores these communications.

Remarks Even if the neural network is trained on a GPU, its inference in the code Argo-DG is performed on a CPU. Such a type of wall model is fully applicable to new codes developed on GPU partitions. A new implementation of the code Argo-DG on GPUs is currently under consideration at Cenaero.



STEP 1: **Interpolation** of the input fields ($\boldsymbol{u}$, ∇p, ...) on the probe grid located at a given h_{wm}

STEP 2: **Construction** of the input stencil for each surface probe, this step requires MPI communication between ranks to transfer data

STEP 3: Transformation of the inputs into `at::tensor` and **forward pass** through the network using `TorchScript` for the prediction of the wall shear stress PDF

STEP 4: **Sampling** from the predicted distribution and rescaling the prediction (Min/Max rescaling and inverse transformation)

STEP 5: Bilinear interpolation onto the **quadrature** points for the evaluation of the boundary condition

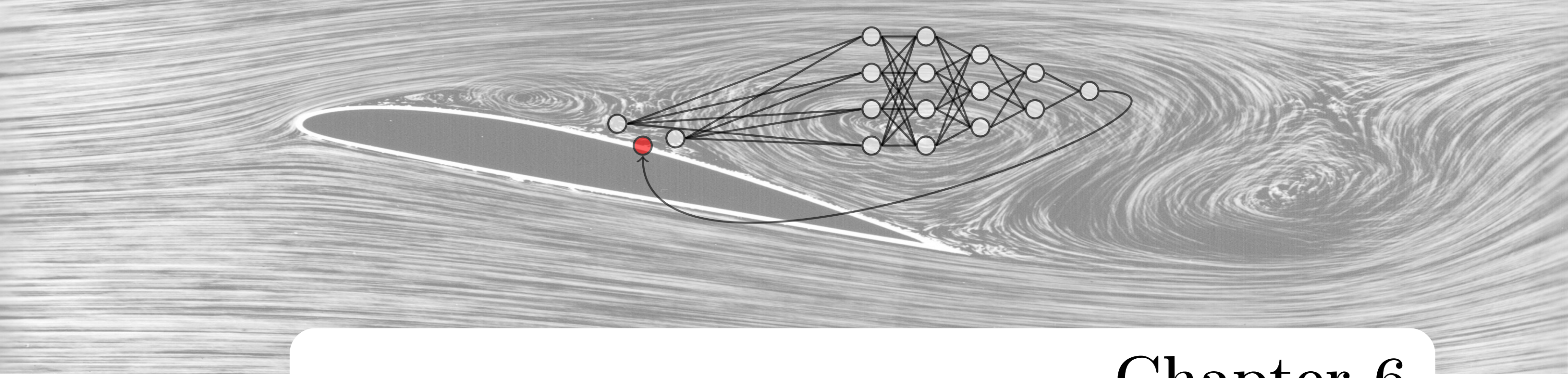
Sampling

$$h^+ = \hat{y}(\max - \min) + \min$$

$$\tau_w = f^{-1}(h^+)$$

Velocity field ($\mathbf{u}$)
Interpolation
Partition
τ_w
$p(\tau_w)$
Mesh element on partition
Quadrature points
Surface probes
Mesh element on partition
Mesh element on partition

Figure 5.10: Implementation of the data-driven WSS model in Argo-DG; the dialogue icon illustrates the need for MPI communication to exchange information across MPI ranks (i.e., mesh partitions).



Chapter 6
Pathway to the successful network architectures for wall modeling

This chapter reviews the history of neural network training, which has led to the establishment of the most successful networks for the development of wall shear stress models.

In machine learning, numerous hyperparameters (e.g., number of hidden layers, learning rate, batch size, kernel size, to cite a few) need to be adjusted to achieve optimal predictions on the validation set. On top of that, we can also modify (i) the type of network (e.g., MLP or CNN), (ii) the input features (e.g., add the curvature or not), (iii) the input stencil size, and (iv) the non-dimensionalization. Therefore, you can imagine the number of training required to test all these combinations. Open-source hyperparameter optimization frameworks, such as `Optuna`, can automate the search for neural network hyperparameters. However, finding the appropriate non-dimensionalization, network architecture, input features, and stencil size for the targeted application still requires attention.

Section 6.1 discusses the first non-dimensionalization tested at the beginning of the thesis, which was discarded to adopt the one described in Section 5.2.1 because it was too sensitive to the polynomial order. Section 6.2 discusses the size of the input stencil for the non-dimensionalization presented in Section 5.2.1. Although the space-time correlations have indicated the *optimal* stencil size, we tested several stencils (e.g., small, long, symmetric, and asymmetric) to validate the conclusion drawn from the feature selection. When working with inputs that have a spatial structure, Section 6.3 explains the advantages of considering a CNN instead of an MLP. Section 6.4 discusses the overfitting and underfitting of the trained network and how to improve the predictions.

6.1 Discussion about the non-dimensionalization

The first non-dimensionalization of the input/output pairs used to train the network architecture and presented at the conference ECCOMAS 2022 and DLES13 is inspired by the one proposed by Zhou *et al.* (2021). The networks were trained with the velocity $\boldsymbol{u}$, pressure gradient ∇p, and a length scale h_{wm}.

The non-dimensionalization is based on the length scale $y_{\nu,p}$ inspired by the near-wall scaling compatible with separation (Duprat *et al.*, 2011),

$$y_{\nu,p} = \frac{\nu}{u_{\nu,p}} \,,$$

where $u_{\nu,p}$ is a velocity scale defined as

$$u_{\nu,p} = \sqrt{u_\nu^2 + u_p^2}, \quad \text{where } u_\nu = \sqrt{\frac{\nu u_\parallel}{h_{wm}}}, \text{ and } u_p = \left| \frac{\nu}{\rho} \frac{\partial p}{\partial x} \right|^{1/3} .$$

The wall model height is scaled with $y_{\nu,p}$, the velocity is scaled with $u_{\tau,p}$, and the pressure gradient is scaled as the Clauser parameter as

$$h^{**} = \ln\left(\frac{h_{wm}}{y_{\nu,p}}\right), \quad \boldsymbol{u}^{**} = \frac{\boldsymbol{u}}{u_{\tau,p}}, \quad \text{and } \nabla p^{**} = \frac{h_{wm}}{\rho u_{\nu,p}^2} \nabla p \,. \tag{6.1}$$

The wall shear stress is non-dimensionalized as a friction coefficient,

$$\boldsymbol{\tau}_w^{**} = \frac{\boldsymbol{\tau}_w}{\frac{1}{2}\rho u_{\nu,p}^2} \,. \tag{6.2}$$

The non-dimensional quantities are denoted with a double $*$ to differentiate them from those defined in Section 5.2.1.

Using this non-dimensionalization, several convolutional neural networks were trained with the Mean Square Error (MSE) loss, defined in Section 2.2. In the early stage of the thesis, standard networks were trained in the hope of recovering the instantaneous behavior of the friction before attempting to estimate the PDF of the wall shear stress. Three main tests were conducted to evaluate the impact of the input features and the network capability to generalize using the non-dimensionalization presented in Equations 6.1 and 6.2.

6.1.1 Training a CNN on PHL10595

A convolutional neural network (CNN), introduced in Section 2.2.2, was trained on the lower wall of the periodic hill (PHL10595) and evaluated on the upper wall (PHU10595) and the channel (TC950). The architecture and the different hyperparameters employed to train the CNN are not presented to maintain conciseness.

On average, the neural network recovered the LES wall shear stress. The network captured the separation and reattachment correctly and underestimated the friction peak. The standard deviation was under-predicted at every streamwise location. This under-prediction was even more pronounced near the separation, where there was not enough richness in the neural network to characterize the instantaneous behavior, mainly due to the underlying hypothesis of the MSE.

The prediction of $\tau_{w,\xi}$ on the upper wall of the periodic hill was unsatisfactory. The mean prediction presented oscillations, and the standard deviation was overestimated, indicating the high uncertainty of the model on this configuration. Regarding the channel, the mean predictions underestimate about 10% the ground truth and also presented oscillations.

Conclusion We conclude that the oscillations originate from the velocity scale $u_{\nu,p}$ and, more precisely, from the pressure gradient. To damp these oscillations, the spatial-averaged of the velocity scale $\langle u_{\nu,p} \rangle_{\xi,z}$ was considered to non-dimensionalize the wall shear stress in the two subsequent tests.

6.1.2 Training a CNN on PHL10595 and TC950

The same CNN as in Section 6.1.1 was trained on the channel (TC950) and lower wall of the periodic hill (PHL10595) and evaluated on the upper wall (PHU10595).

The predictions on the channel were very satisfactory, and no more oscillations were present in the averaged prediction. Regarding the lower wall of the periodic hill, the friction peak was again underestimated. It seems that the network has some difficulties in discriminating the accelerating part with the rest of the domain. The prediction for the upper wall (PHU10595) was as inaccurate as the prediction from Reichardt's LOTW. The network did not correctly understand the correlation with the pressure gradient.

Conclusion To overcome the misprediction in the friction peak, we decided to introduce the curvature as a new input feature. To improve the understanding between the wall shear stress on the upper wall and the pressure gradient, the PHU10595 is added to the training data while the turbulent channel is left as the validation set.

6.1.3 Training a CNN on PHL10595 and PHU10595

The same CNN as in Section 6.1.1 was trained on the upper (PHU10595) and lower wall of the periodic hill (PHL10595) and evaluated the channel (TC950).

The prediction matches the ground truth everywhere, even at the friction peak. The prediction on the upper wall is improved, but the correlation with the pressure gradient is still not perfectly captured.

Due to the integration of the curvature in the input feature, the prediction of the wall shear stress on the channel deteriorated. The physics that is the most representative of the turbulent channel, is on the accelerating part of the periodic hill $7.0 \leq x/h \leq 8.5$. However, by adding the curvature, this part of the periodic hill is invisible to the channel, which has a zero curvature.

Conclusion The mean pressure gradient can be added to the input features to better characterize the correlation between the pressure gradient and the wall shear stress. Although the curvature is an important feature to better discriminate among the different physics on the lower wall of the periodic hill, it diminishes the network capabilities to generalize to other configurations.

6.1.4 Training a CNN on PHL10595, PHU10595 and TC950

In regard to the previous test, the CNN is trained on both the upper (PHU10595) and lower wall of the periodic hill (PHL10595), as well as the channel (TC950). The model was then evaluated on all three databases but at different time steps. However, this approach may not be the most appropriate for avoiding overfitting. The *a priori* tests yielded satisfactory results. The model was thus implemented in Argo-DG following the procedure presented in Section 5.3.2. The *a posteriori* test was first conducted on the channel flow at a polynomial order $p = 3$. The obtained results were comparable to those in the literature. To further validate the approach, the wmLES was performed at a fourth polynomial order, but the simulation never stabilized. After multiple tests, it was concluded that the non-dimensionalization presented above was too sensitive to the polynomial order. As a result, the non-dimensionalization has been redefined using only the kinematic viscosity ν and the matching location h_{wm}, instead of using *"the velocity itself to scale the velocity"*.

6.1.5 Conclusion

In conclusion, the curvature and the mean pressure gradient were added as input features to train the data-driven wall shear stress model.

The curvature helps to discriminate between the different physics on the lower wall, especially between the accelerated region from $x/h = 7.0$ to $x/h = 8.5$ and the separation region establishing instantaneously over a large part of the hill crest, since these two regions are both characterized by a high-pressure gradient and velocity.

The mean pressure gradient helps the network capture the correlation between the wall shear stress and the pressure gradient on the upper wall to avoid reproducing similar results as Reichardt's LOTW. The importance of the mean pressure gradient is further characterized using SHAP values in Section 8.2.4.

New and more physical scaling factors have been defined due to the sensitivity of non-dimensionalization to the polynomial order. This new non-dimensionalization is used for the rest of this chapter.

6.2 Discussion about the stencil size

Section 5.1.2 discusses the *optimal* size for the input stencil. Nonetheless, the optimality can be questioned by testing different stencil sizes. In the present section, the smallest (i.e., 1) and the largest stencil size (i.e., 180) are considered to train an MLP and a CNN and compared to the optimal stencil size (i.e., 16). In Section 6.1, the networks were trained with an MSE. For the rest of this chapter, every network is connected to a GMH and trained with the Negative Log-likelihood loss defined in Equation 5.4. For the smallest stencil size, an MLP is considered, while for the larger stencil, a CNN is used. The three neural networks are trained on PHL10595, PHU10595, and TC1000 using the instantaneous velocity, the instantaneous and mean pressure gradient, and the curvature. The hyperparameters summarized in Table 8.1 are employed for the three neural networks. The major difference between the three networks is the number of inputs, which directly affects the number of parameters to be fit during the training.

To evaluate the performances of the different neural networks, two metrics are introduced. The first metric is the $Q2$-criterion which evaluates the *goodness of the prediction*. This metric is also termed predicted variation and is defined as

$$Q_2 = 1 - \frac{\mathbb{V}\mathrm{ar}(y - \hat{y})}{\mathbb{V}\mathrm{ar}(y)} \,, \tag{6.3}$$

where y is the ground true and $\hat{y}$ is the corresponding prediction. The second metric is Pearson coefficient defined as

$$R = \frac{n\sum_{i=1}^{n} y_i\hat{y}_i - \left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} \hat{y}_i\right)}{\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}\sqrt{n\sum_{i=1}^{n} \hat{y}_i^2 - \left(\sum_{i=1}^{n} \hat{y}_i\right)^2}} \,. \tag{6.4}$$

Table 6.1: Summary of the experiments conducted on three neural networks, where Q2 and R are two metrics used to evaluate the performance of the network, results obtained for the lower and upper wall of the two-dimensional periodic hill are indicated by L and U, respectively.

Network	**Size**	**Min. Loss**	**Q2**		**R**	
			L	U	L	U
MLP-1	1	−3.78	0.23	−0.82	0.992	0.856
CNN-16	16	−4.14	0.42	−0.36	0.997	0.954
CNN-180	180	−4.15	0.35	−0.31	0.996	0.969

Table 6.1 summarizes the results obtained on the three neural networks where Q2 is evaluated on the instantaneous prediction and R on the averages. In both cases, the two metrics are computed for the lower (L) and upper (U) walls of the two-dimensional periodic hill. The networks trained with a larger stencil size, either 16 or 180, produce better results on both the upper and lower walls. The predictions of CNN-16 are as good as those of CNN-180. This observation confirms that the

optimal stencil size obtained with feature selection is sufficient to capture the correlations necessary for the network to perform well.

Figure 6.1 shows the mean and standard deviation of the predictions, on the upper wall, against the ground truth. The MLP-1 predictions underestimate the target wall shear stress between $5.0 \leq x/h \leq 9.0$ while the two CNNs perform well with Pearson coefficients of 0.954 and 0.969.

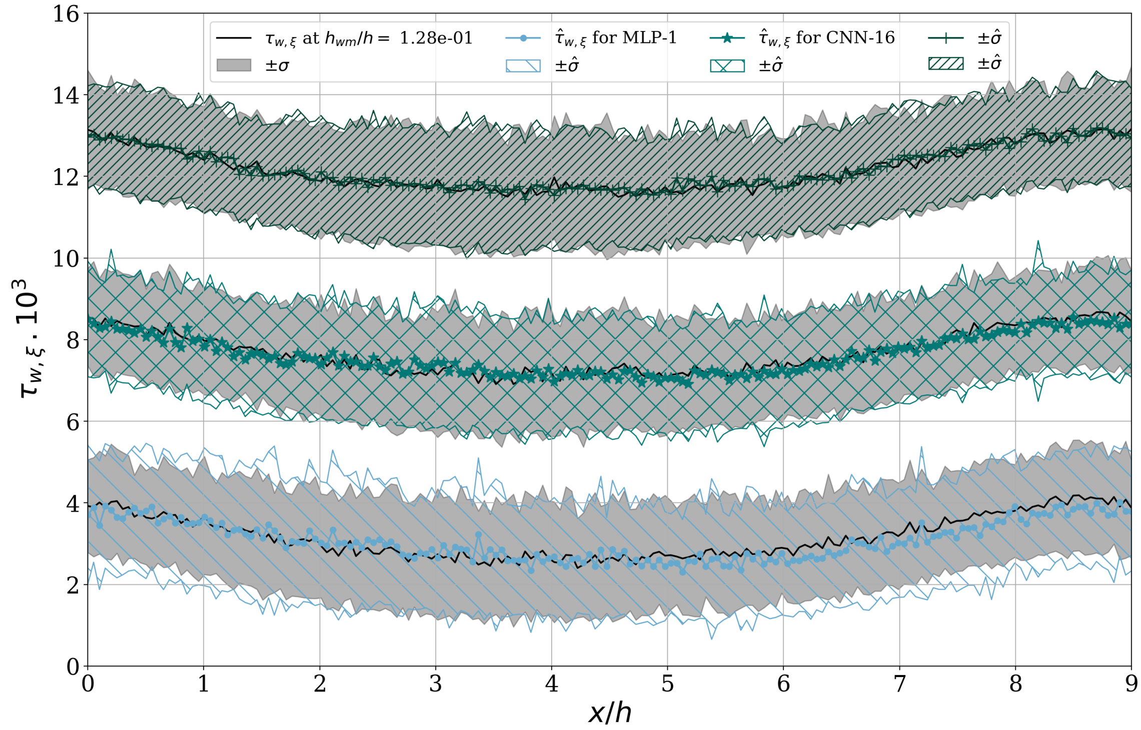


Figure 6.1: *A priori* test on PHU10595 to validate the stencil size: the black plain line is the wrLES mean wall shear stress, while the gray area represents the wrLES standard deviation; the MLP-1 predictions are drawn in dark green; the CNN-16 predictions are drawn in turquoise; the CNN-180 predictions are drawn in light blue.

Figure 6.2 shows the mean and standard deviation of the predictions on the upper wall against the ground truth. MLP-1 struggles to predict the mean wall shear stress in the separation vicinity where no local and instantaneous correlations were found (see Figure 4.17). Moreover, the standard deviation has a larger amplitude in that region, indicating greater uncertainty.

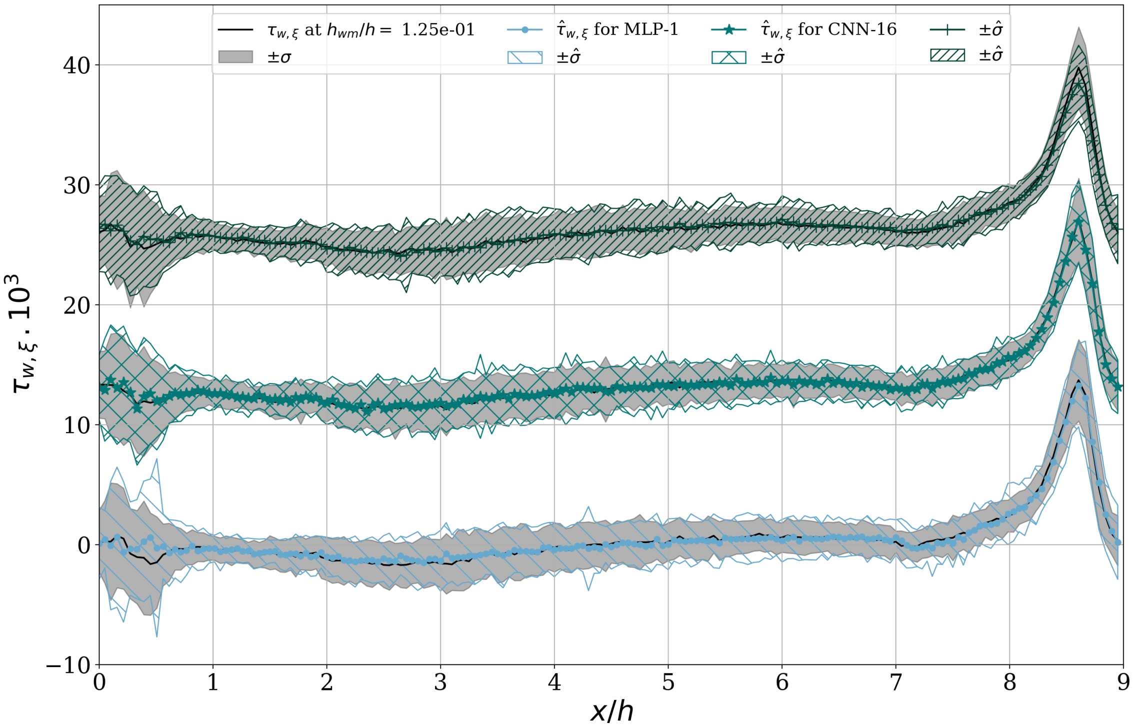
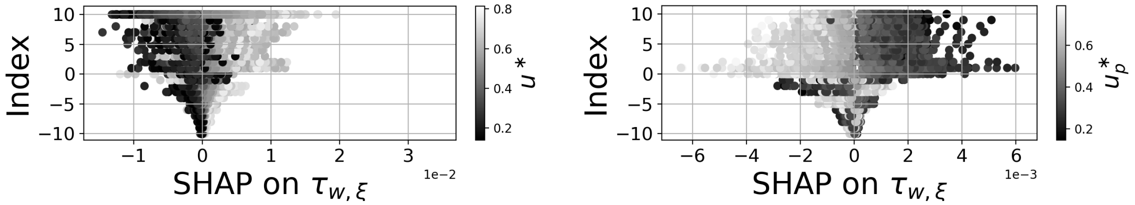


Figure 6.2: *A priori* test on PHL10595 to validate the stencil size: similar legend as in Figure 6.1.

The preceding discussion mentions the size of the stencil. However, we can also debate the location of the points: whether they should be extracted only upstream, downstream, or a balance between upstream and downstream points. To evaluate which point influences the wall shear stress the most, SHAP values are computed for a network trained over a symmetric stencil of size 21 where 10 points are extracted upstream and 10 points are extracted downstream. In this case, the network is only trained on the upper wall of the two-dimensional periodic hill. To ensure traceability of the SHAP values, the output is evaluated as $\sum_k \pi_k \mu_k$ instead of sampling from the predicted distribution.

The averaged marginal contribution of the non-dimensionalized and scaled streamwise velocity component u^* (ranging from 0 to 1) on the outcome is shown in Figure 6.3a. The vertical axis represents the different positions along the stencil. Negative indexes represent upstream points, while positive indexes represent downstream points. The streamwise velocity extracted downstream contributes more to a modification of the outcome than points extracted upstream. This observation is consistent with the space-time correlation $U0T0$ in Figure 4.14. Each point is colored by the value of the input (i.e., u^*). Light values correspond to a high velocity, while dark values correspond to low values. A positive velocity contributes to an augmentation of the wall shear stress, while a negative velocity tends to decrease the wall shear stress. Figure 6.3a confirms that the wall model correctly learns the relationship between u and $\tau_{w,\xi}$.



(a) Shapley values of the feature u^* on the outcome $\tau_{w,\xi}$.

(b) Shapley values of the feature u_p^* on the outcome $\tau_{w,\xi}$.

Figure 6.3: SHAP summary plot on the upper wall of the two-dimensional periodic; on the left, the SHAP values related to the streamwise velocity, and on the right, those related to the pressure gradients.

The averaged marginal contribution of the non-dimensionalized and scaled streamwise pressure velocity u_p^* (ranging from 0 to 1) on the outcome is shown in Figure 6.3b. Similar conclusions to those drawn for the Shapley values of u are made. Downstream points contribute more to the outcome. Moreover, points with index $0, 1, 2$ contribute more to the outcome as highlighted by the space-time correlation of Figure 4.15 where the region of high correlation is centered at $(\delta t, \delta\xi) = (0, 0)$. The anti-correlation detected in Figure 4.15 is also confirmed by the SHAP values.

Conclusion The four experiments conducted in the present section suggest that a larger stencil (i.e., greater than one) is more effective in predicting the wall shear stress in the separation vicinity and on the upper wall. The computation of SHAP values indicates that the downstream points have a greater averaged marginal contribution to the outcome. Therefore, an asymmetric stencil is recommended. The *optimal* stencil size deduced from the analysis of space-time correlation is thus confirmed by the present analysis.

6.3 Discussion about the architecture

At this point, the input features (i.e., the instantaneous velocity, the instantaneous and mean pressure gradient, and the curvature) and the stencil size (i.e., 16 points with 5 taken upstream, 10 taken downstream, and the current point) are fixed by the previously mentioned experiments. In the present section, we discuss which architecture between the MLP and the CNN is better for the targeted application. In the experiments, a CNN with two-dimensional convolution layers is also tested to see if an extension in the spanwise direction could be beneficial. The stencil was taken symmetrically because the correlations in the spanwise direction were mostly symmetrical because this direction is homogeneous and periodic. This hypothesis cannot be generalized to more complex geometries with inhomogeneous directions.

Table 6.2: Summary of the experiments conducted on three neural networks, where Q2 and R are defined by Equations 6.3 and 6.4, results obtained for the lower and upper wall of the two-dimensional periodic hill are indicated by L and U, respectively.

Network	**Size**	**# param.**	**Min. Loss**	**Q2**		**R**	
				L	U	L	U
MLP-16	16	83,772	-4.17	0.42	-0.39	0.997	0.956
CNN1d-16	16	10,212	-4.14	0.42	-0.36	0.997	0.954
CNN2d-16x16	16×16	20,162	-4.39	0.51	-0.15	0.996	0.968

Table 6.2 summarized the three architectures with their performances (Q2 and R) on both the lower and upper walls of the two-dimensional periodic hill. MLP-16 performs as well as CNN1d-16 but with eight times more parameters. The extension in the spanwise direction is beneficial, as demonstrated by the performance of CNN2d-16x16. CNN2d-16x16 has a higher Q2 criterion, and the predictions on the upper wall are improved, as confirmed by the Pearson coefficient.

Figure 6.4 shows the mean and standard deviation of the predictions on the upper wall of the two-dimensional periodic hill against the ground truth. All three networks correctly predict both the mean and the standard deviation, demonstrating their ability to comprehend the correlation between τ_w and the pressure gradients.

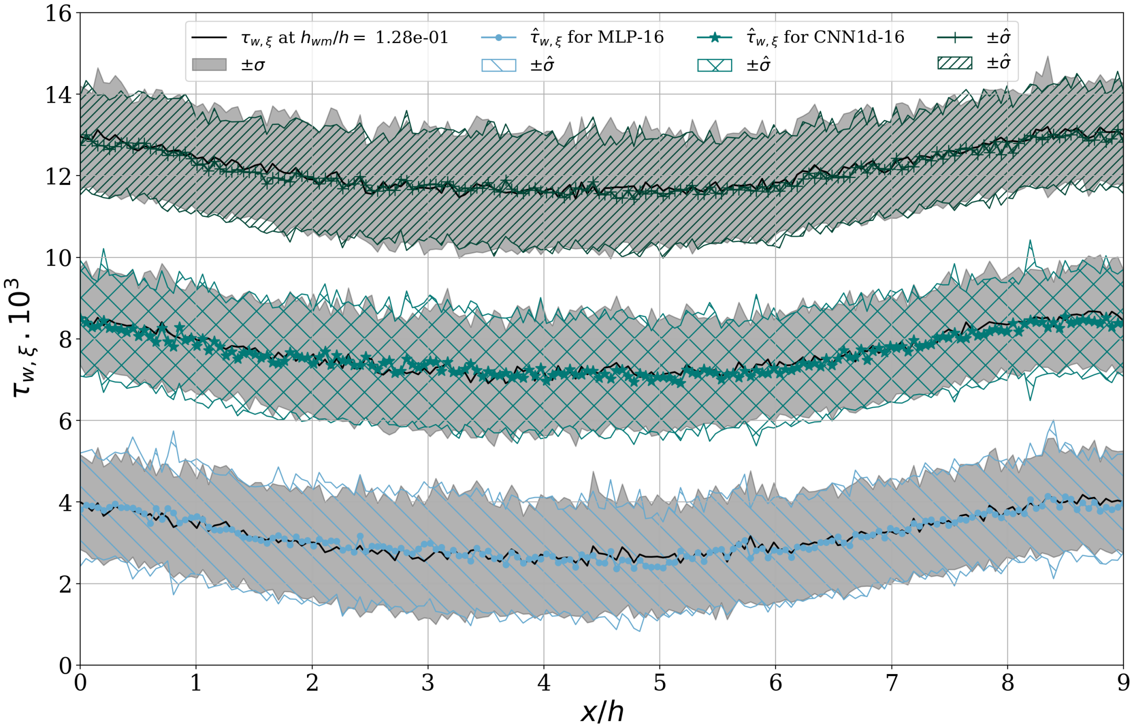


Figure 6.4: *A priori* test on PHU10595 to validate the architecture: the black plain line is the wrLES mean wall shear stress, while the gray area represents the wrLES standard deviation; the MLP-16 predictions are drawn in dark green; the CNN1d-16 predictions are drawn in turquoise; the CNN2d-16x16 predictions are drawn in light blue.

The mean and standard deviation of the predictions on the lower wall of the two-dimensional periodic hill are shown in Figure 6.5 against the ground truth. All three networks distinguish between the separation, reattachment, and accelerating region, and recover the friction peak.

SHAP values The *optimal* size in the spanwise direction has not yet been discussed. As observed in Figure 4.24, the cross-correlation $U0T2$ covers a significant space area. However, for all other correlations, the domain of high correlation remains concentrated around the time axis and does not spread much in space. It

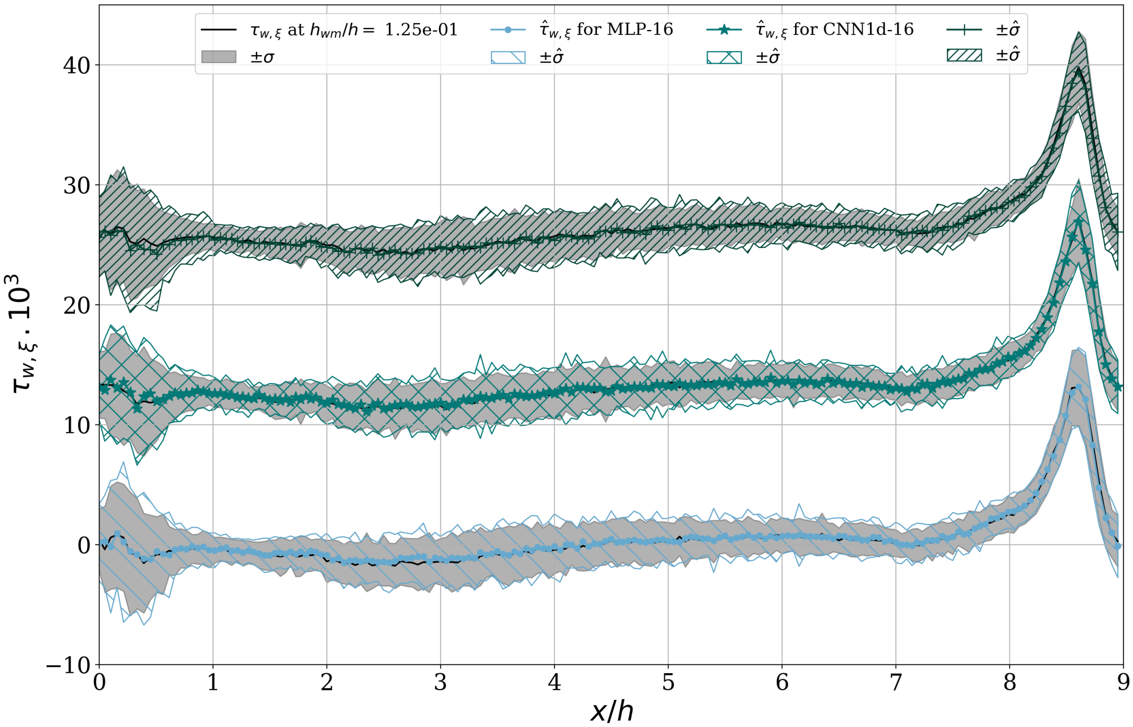
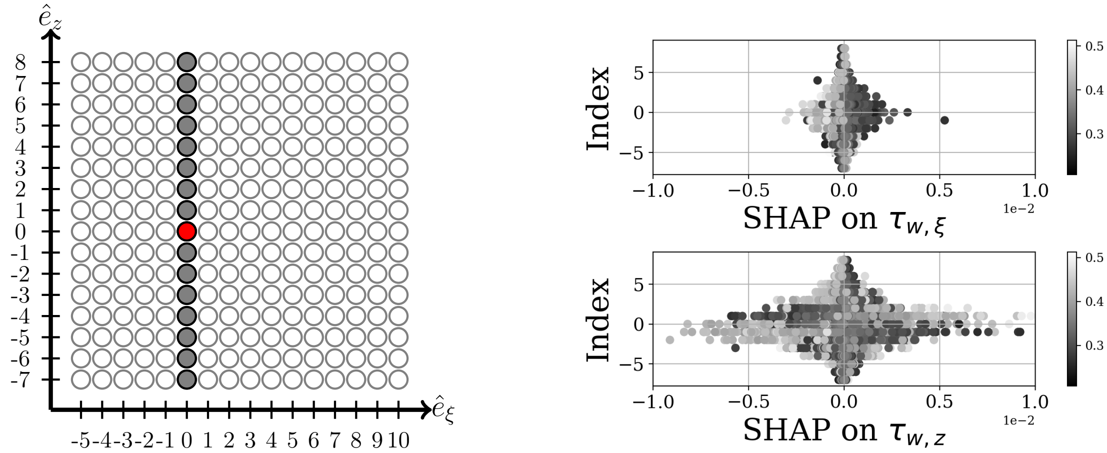


Figure 6.5: *A priori* test on PHL10595 to validate the architecture: similar legend as in Figure 6.4.

is therefore interesting to examine which points along the spanwise direction contribute more to the outcome after training the CNN2d-16x16 on a 16 × 16 input stencil. The averaged marginal contribution of the non-dimensionalized and scaled streamwise velocity component u^* (ranging from 0 to 1) on the outcomes $\tau_{w,\xi}$ and $\tau_{w,z}$ is shown on Figure 6.6b. The index indicates the spanwise positions, spanning from -7 to $+8$. The streamwise position in the 2D-stencil is fixed and corresponds to the current point, as illustrated in Figure 6.6a. In both graphs, the central points $\{-3, \ldots, 3\}$ contribute more to the outcome. This observation is consistent with Figure 4.16. The SHAP values point out the anti-correlation for $\delta z > 0$ and the correlation for $\delta z < 0$, as observed for the cross-correlation $U0T2$. Furthermore, the averaged marginal contribution of u^* on $\tau_{w,\xi}$ is weaker than the AMC of u^* on $\tau_{w,z}$ because the correlation $U0T0$ is weaker than $U0T2$ when no time delay was considered ($\delta t = 0$).



(a) Sketch of two-dimensional input stencil used to train CNN2d-16x16 (with 16 points along the streamwise and spanwise direction), where the red point corresponds to the *current* point.

(b) SHAP summary plot of Shapley values of the feature u^* on the outcome $\tau_{w,\xi}$ (top) and $\tau_{w,z}$ (bottom) evaluated for the network CNN2d-16x16 for PHU10595.

Figure 6.6: The gray circles on the left indicate the spanwise positions used to plot the SHAP summary on the right.

Conclusion The aim of these experiments was to determine the most suitable neural network architecture between an MLP and a CNN based on the number of input features. Although MLP-16 and CNN1d-16 produce comparable results, CNN1d-16 is preferred due to its smaller number of parameters and translation invariance. Additionally, the spanwise extension improves the prediction accuracy on the upper wall. Therefore, it is recommended to choose a CNN as the appropriate architecture for the targeted application. If two-dimensional inputs are considered, the number of points used in the spanwise direction could be reduced to 11, as revealed by the SHAP values.

6.4 Discussion about overfitting and underfitting

The present section discusses the bias-variance trade-off in machine learning. The goal is to obtain a model that balances underfitting and overfitting. *Overfitting* means that the network is too specialized on the training data, i.e., it is mainly trained to exactly fit the data. *Underfitting* means that the network is enable to capture the underlying relationship between the input and output variables. Ultimately, the goal is to develop a model that is rich enough to capture the underlying structure in the data but simple enough to generalize on unseen configurations and avoid fitting spurious patterns.

Although this bias-variance trade-off may seem inconsistent with the behavior of modern machine-learning methods, overfitting remains a risk in "*classical*" regime with under-parameterized neural networks that stay below the interpolation threshold (Belkin *et al.*, 2019). Because we have a limited number of test cases and our neural networks are relatively small compared to modern machine learning standards, overfitting remains a significant concern and can impact the model's ability to generalize.

Overfitting can be monitored by evaluating the currently trained network on an appropriated validation set (see Section 2.2.1.3). In the above training, the validation set was constructed as data from the same test case but at different time steps, and the training was stopped at the minimum of the validation loss curve, which is called *early stop*. However, because the data are extracted from the statistically converged state, they are not very different from each other. Therefore, it is more difficult to really evaluate the network capability to generalize. In the present section, we perform a leave-one-out approach where two datasets are employed to train the network, and the third one is left for validation purposes. Three combinations are possible.

- **TC1000-PHU10595**

  The generalizability of PHL10595 cannot be attributed to the neural network's ability to capture the underlying structure in the data. Rather, it is just because the database is not enough rich to extrapolate on the lower wall, which features separation and reattachment.

- **TC1000-PHL10595**

  There is potential for generalization on the upper wall, and it is important to ensure that the model does not overfit the two other databases.

- **PHU10595-PHL10595**

> Generalizability is possible if the training data includes a sufficient range of y^+ values to capture the logarithmic layer of the TC1000 test case.

A CNN1d-16 was trained on the turbulent channel flow at $Re_\tau = 1{,}000$ and the lower wall of the two-dimensional periodic hill. To monitor overfitting, the validation loss was evaluated on the upper wall of the two-dimensional periodic hill. Figure 6.7 shows the evolution of the two losses (training and validation). The training loss continues to decrease, but at epoch 82, the validation loss begins to increase. After epoch 82, the model starts to overfit the training data.

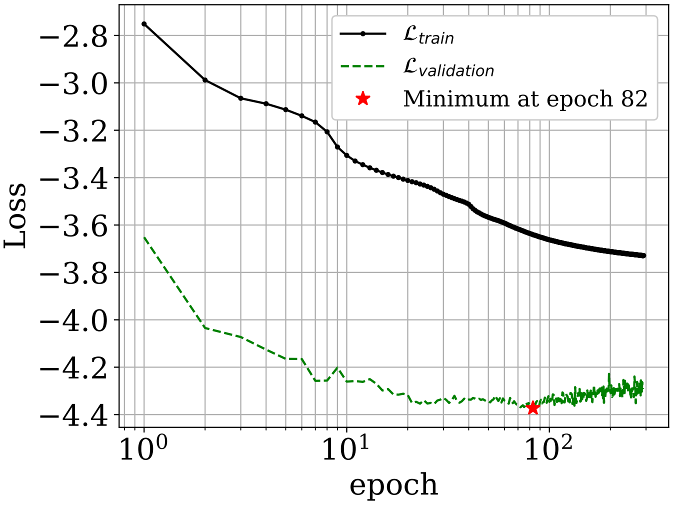


Figure 6.7: Evolution of the training and validation losses.

The mean and standard deviation of the prediction on PHU1059 are shown in Figure 6.8. Although the predictions are not perfect, they are already better than those produced by Reichardt's LOTW. The data-driven wall model predictions follow the trend of those produced by Reichardt's LOTW between $0 \leq x/h \leq 5$, which indicates a stronger relationship with the velocity field than with the pressure gradient. Therefore, the model imposed on the upper wall a similar relationship as on a turbulent channel. However, between $5 \leq x/h \leq 9$, the predictions align with the ground truth. The network is partially capable of transferring its knowledge from the TC1000 and PHL1059 to the upper wall of the periodic hill.

Conclusion A network trained on TC1000 and PHL10595 is partially capable of predicting the wall shear stress on the upper wall of the periodic hill. Although

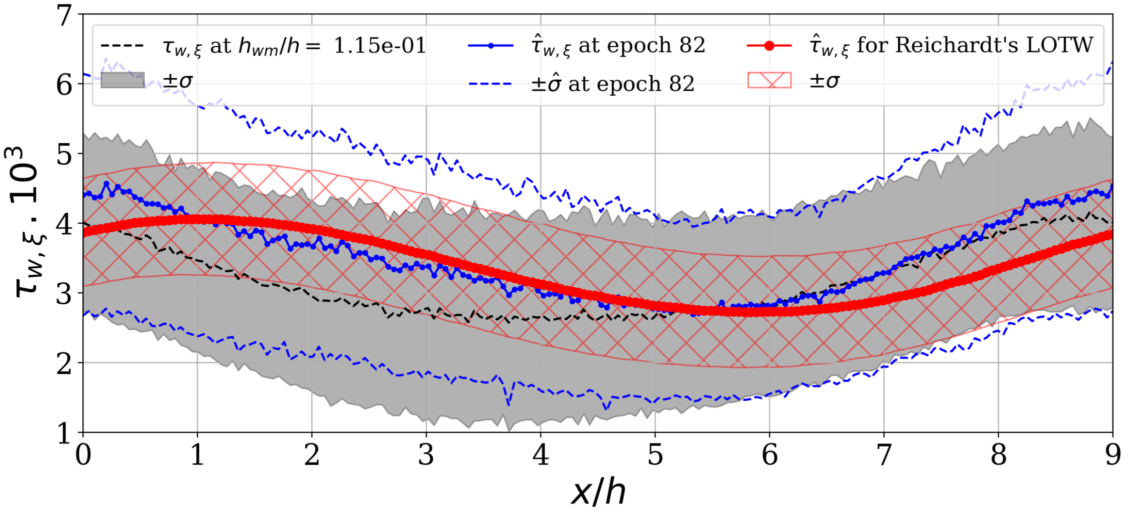


Figure 6.8: *A priori* predictions of the streamwise wall shear stress on the upper wall of the two-dimensional periodic hill generated by a CNN1d-16 trained on TC1000 and PHL10595.

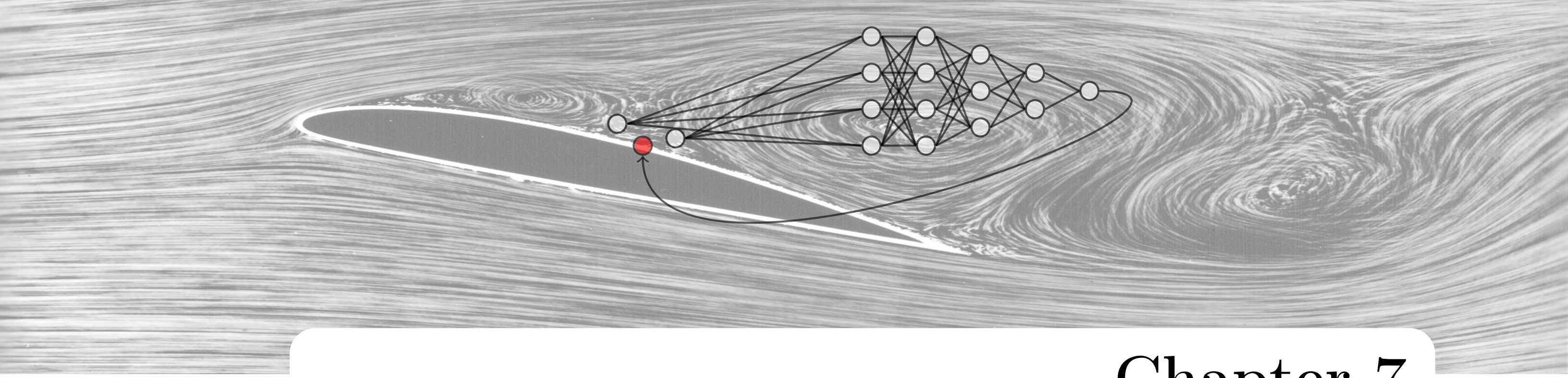
the predictions obtained by the data-driven wall model present some errors, there is an improvement compared to those produced by Reichardt's LOTW. The reduced capacity of the network to generalize may be attributed to the limited number of test cases. Better generalization is more likely when the database contains a wider range of physics.

6.5 Conclusion

This chapter provides a brief summary of the experiments conducted during the thesis that led to the definition of the final architecture used to develop new data-driven wall models presented in Chapters 7 and 8.

Experiments conducted in Section 6.1 aid in redefining the non-dimensionalization and adding two additional input features: mean pressure gradient and curvature. In Section 6.2, the *optimal* stencil size, deduced from the space-time correlations, has been confirmed. Knowing the number of inputs, two network architectures (MLP and CNN) are tested in Section 6.3. The CNN1d-16 produced similar results to the MLP-16 but with eight times fewer parameters. Moreover, the CNN performed even better when applied to two-dimensional inputs. Therefore, with a view of extending the input stencil in the spanwise direction, the CNN is preferred over the MLP. Finally, in Section 6.4, a CNN1d-16 was trained over the turbulent channel at $Re_\tau = 1{,}000$ and the lower wall of the two-dimensional periodic hill at $Re_b = 10{,}595$ and evaluated on the upper wall of the periodic hill. The training was stopped before the network overfitted the training data. The obtained network partially recovered the wall shear stress on the upper wall. This observation confirms that the database does not contain enough physics to generalize.

To summarize, the most successful network is a convolutional neural network trained with a stencil of size 16 where 5 points are taken upstream, 10 points are taken downstream, and the current point. The input features are the three components of the instantaneous velocity, the three components of the mean and instantaneous pressure gradient, and the curvature. If the stencil is extended in the spanwise direction, a two-dimensional convolutional neural network should be employed with 2D stencils of size 16 and 11 in steam and span, respectively. The spanwise stencil is symmetric about the current point (i.e., 5 points upstream and 5 points downstream).



Chapter 7 Fully developed turbulent flow in channel configuration

This chapter discusses the development of a data-driven WSS model applied to the turbulent channel flow only. Wall-modeled LES of the channel flow has been successful for a long time, and we do not aim to invent a new WSS model that will outperform other approaches. The main objective is to develop the methodology for training, validating, and testing the stochastic-based wall model on a canonical and well-known geometry. The training and validation parameters are presented in Section 7.1, while *a priori* tests are performed in Section 7.2 by analyzing the capabilities of the trained neural network to generalize.

To predict the distribution of the wall shear stress on a channel wall, a *simple* MLP combined with a GMH is chosen. As discussed in Section 5.1.1, the instantaneous predictions are sampled from a probability density function. This sampling is evaluated at each surface probe without considering time or space correlations. In a channel, the streamwise and spanwise directions are homogeneous and periodic. Therefore, the correlation length scale is similar at each (x, z) location, and the same is true for the correlation time scale. This observation simplifies the reconstruction of the space-time correlations for a turbulent channel flow. This reconstruction is treated in Section 7.3.

Once the neural network has been thoroughly verified *a priori*, it can be implemented in Argo-DG (i.e., in a production environment) to be tested *a posteriori*. The fact that the network performs well *a priori* does not mean that it will perform well *a posteriori*. Indeed, the production environment is different from the training environment. Firstly, the network will be confronted with the initial numerical transient not seen during the training. Secondly, because the input fields are not filtered according to the size of the wmLES mesh, the input fields will be different in terms of the structures carried in the fluid from those seen during the training. Thirdly, the prediction will affect the velocity, which in turn will affect the wall shear stress. There is a feedback loop between the velocity and wall shear stress that was not present during the training. All these points may affect the network behavior.

7.1 Training

Table 7.1 lists the trained networks with their corresponding non-dimensional input/output pairs, described in Section 5.2.1. The prediction of the wall shear stress distribution for turbulent channel flows is carried out by a *simple* MLP combined with one GMH for $\tau_{w,\xi}$ and one GMH for $\tau_{w,z}$. Because the solid wall is flat, the streamwise direction x corresponds to ξ.

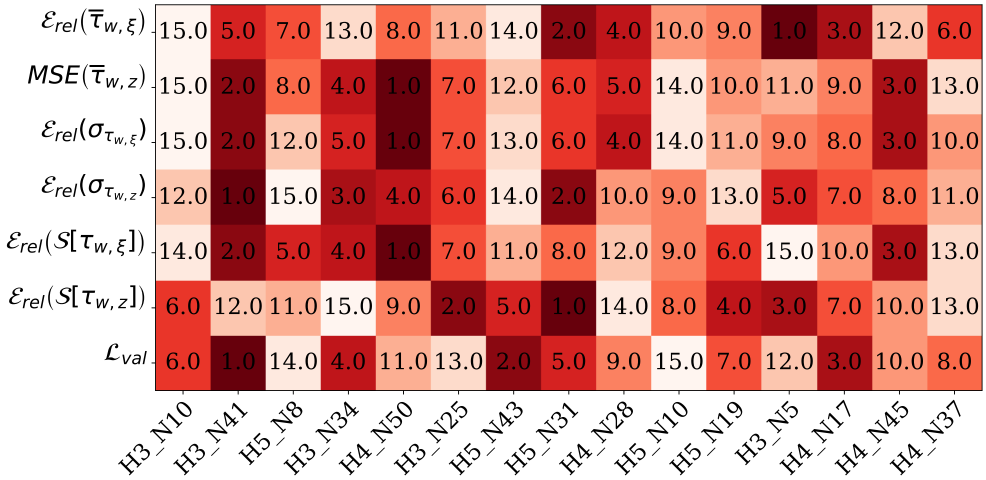
Table 7.1: List of the neural networks trained with their corresponding database, pairs of input/outputs, the size of the input stencil along the streamwise direction, and the number of wall model heights; each test case was described in Chapter 4.

Architecture	**MLP-GMH**	**CNN-skip-GMH**
Train database	TC5200	PHU10595, PHL10595, TC1000
Test database	TC950, TC1000, SYNTH.	TC950
Inputs	$u^*, v^*, w^*, \overline{u}^*$	$u^*, v^*, w^*, u_p^*, v_p^*, w_p^*, \overline{u}_p^*, \overline{v}_p^*, \overline{w}_p^*, \mathcal{K}^*$
Input size	$[0]$	$[-5:10]$
Nb. of $\boldsymbol{h_{wm}}$	95	5, 5, 35
Outputs	$\tau_{w,\xi}^*, \tau_{w,z}^*$	$\tau_{w,\xi}^*, \tau_{w,z}^*$

As a reminder, the MLP architecture consists of an input layer, several hidden layers, and an output layer. The number of neurons in the input layer is four: three corresponding to the components of the velocity field and one for the mean of the streamwise velocity. The output layer is replaced here by two GMHs composed of K Gaussians. A neural network has many hyperparameters to tune: the number of hidden layers, the number of neurons per layer, the type of initialization, the batch size, and the learning rate, to name a few. A common approach to determining the optimal set of hyperparameters for a dataset is through trial and error, which is the principle of hyperparameter optimization. Two popular techniques exist: *grid* *search* and *random search*. While grid search exhaustively explores every possible hyperparameter combination to determine the best model, random search only picks out and tests a random combination of hyperparameters. To reduce the training effort and to make the task a little easier, the number of neurons in each hidden layer is kept identical. To evaluate the optimal number of hidden layers H and neurons per layer N of the MLP-GMH, a random search is applied with H and N uniformly distributed on the interval $[h_0, h_1]$ and $[n_0, n_1]$, respectively,

$$H \sim \mathcal{U}\left(h_0, h_1\right) \quad \text{and} \quad N \sim \mathcal{U}\left(n_0, n_1\right) \ ,$$

where $h_0 = 3$, $h_1 = 5$ (a too deep network is not useful here), $n_0 = 5$ and $n_1 = 50$.



	H3_N10	H3_N41	H5_N8	H3_N34	H4_N50	H3_N25	H5_N43	H5_N31	H4_N28	H5_N10	H5_N19	H3_N5	H4_N17	H4_N45	H4_N37
$\mathcal{E}_{rel}(\overline{\tau}_{w,\xi})$	15.0	5.0	7.0	13.0	8.0	11.0	14.0	2.0	4.0	10.0	9.0	1.0	3.0	12.0	6.0
$MSE(\overline{\tau}_{w,z})$	15.0	2.0	8.0	4.0	1.0	7.0	12.0	6.0	5.0	14.0	10.0	11.0	9.0	3.0	13.0
$\mathcal{E}_{rel}(\sigma_{\tau_{w,\xi}})$	15.0	2.0	12.0	5.0	1.0	7.0	13.0	6.0	4.0	14.0	11.0	9.0	8.0	3.0	10.0
$\mathcal{E}_{rel}(\sigma_{\tau_{w,z}})$	12.0	1.0	15.0	3.0	4.0	6.0	14.0	2.0	10.0	9.0	13.0	5.0	7.0	8.0	11.0
$\mathcal{E}_{rel}(\mathcal{S}[\tau_{w,\xi}])$	14.0	2.0	5.0	4.0	1.0	7.0	11.0	8.0	12.0	9.0	6.0	15.0	10.0	3.0	13.0
$\mathcal{E}_{rel}(\mathcal{S}[\tau_{w,z}])$	6.0	12.0	11.0	15.0	9.0	2.0	5.0	1.0	14.0	8.0	4.0	3.0	7.0	10.0	13.0
$\mathcal{L}_{val}$	6.0	1.0	14.0	4.0	11.0	13.0	2.0	5.0	9.0	15.0	7.0	12.0	3.0	10.0	8.0

Figure 7.1: Random Search leading to 15 pairs of hidden layers (H) and neurons (N).

A total of fifteen networks were trained on the database TC5200. The best network is selected based on seven metrics: the averaged relative error on the predicted mean, variance, and skewness for both $\tau_{w,\xi}$ and $\tau_{w,z}$ and the loss function evaluated on the validation set. Since the mean wall shear stress in the spanwise direction is zero, the relative error is replaced by an MSE. Each network is evaluated using these metrics. For instance, the network with the lowest validation loss received 15 points, the network with the second-lowest loss received 14 points, and so on. Finally, the network with the highest score is considered the "*best*" network. Figure 7.1 shows the number of points received by each network for each metric. The best network is `H3_N10`. It has three hidden layers with ten neurons in each layer. The second best network is H5_N10, and the third best is H5_N8. The podium consists of networks with a small number of neurons per layer, i.e., networks with a few parameters.

From this Random Search, the network with three hidden layers and 10 neurons in each layer is retained for future training. Now that the *optimal* H and N have been evaluated, other hyperparameters: (i) the number K of Gaussians in each GMH, (ii) the learning rate, and (iii) the weight regularization are discussed. By doing multiple training with K from 1 to 5, we found that more Gaussians did not really help to capture the skewness better. Chapter 5.2 highlighted how important the skewness is in predicting $\tau_{w,\xi}$ and how it affects its mean. By reducing the skewness with the non-dimensionalization, it turned out that it was possible to obtain a good agreement with the ground truth by using a single Gaussian. The learning rate is also an important hyperparameter. It directly affects the search for the minimum. A too-high learning rate will generate oscillations in the loss, while a too-low learning rate will slow down the training and may lead to getting stuck in a local minimum. Since our batch size is 16 times larger than LeCun's recommended value of 32, the learning rate can be increased to compensate for the larger batch size. Lower learning rates have been tested, but the training was slowed down. Regarding the weight regularization, after several tests, the value of 10^{-5} seems to give proper results.

The hyperparameters used to train the MLP-GMH (Table 7.1) are summarized in Table 7.2 with the size of the training and validation datasets. The model is trained on the turbulent channel flow at $Re_\tau = 5{,}200$ and validated on the one at $Re_\tau = 1{,}000$. The training is stopped at the minimum of the validation loss. The loss is the sum of the negative log-likelihood on $\tau_{w,\xi}$ and $\tau_{w,z}$. No coefficients are used to amplify any of the terms in the loss because the outputs are also scaled between 0 and 1 using the min/max equation (Eq. 5.10).

Table 7.2: List of the hyperparameters used to train the MLP-GMH on turbulent channel flows composed of 336 parameters.

Hyperparameters	Values
Batch size	512
Learning rate	10^{-3}
Weight regularization	10^{-5}
Number of epochs	10^{3}
Training size	$\simeq 4 \cdot 10^{5}$
Validation size	$\simeq 9 \cdot 10^{4}$
Optimizer	`torch.optim.Adam()`
Nb. Gaussian/head	1
Activation fct.	Sigmoid

7.2 *A priori* testing

The model is first tested using the turbulent channel flows at $Re_\tau = 950$ and 1,000 (Section 7.2.1). A discussion on the model's capabilities to generalize is done in Section 7.2.2 on synthetic data extracted from the log-law (Eq. 4.5).

7.2.1 *A priori* testing on TC1000

The results for TC950 are similar to those for TC1000, and are, therefore, not presented here. One of the model constraints is to have the same behavior of the model at each wall model height. Figure 7.2 shows the predicted mean wall shear stress $\overline{\tau}_{w,\xi}$ for heights y^+ ranging from 30 to 200. At each height, the predicted mean and variance are in good agreement with the ground truth (i.e., mean and variance). The relative error is less than 1% at each height. The model constraint mentioned above is therefore satisfied. As Dupuy *et al.* (2023a), the instantaneous predictions are compared to the analytical profile of Reichardt (Eq. 3.6). The wall model predictions are reported in Figure 7.3 and compared against the ground truth in a (u^+, y^+) graph. The model accurately captures the LOTW and effectively estimates the variance caused by the velocity fluctuations.

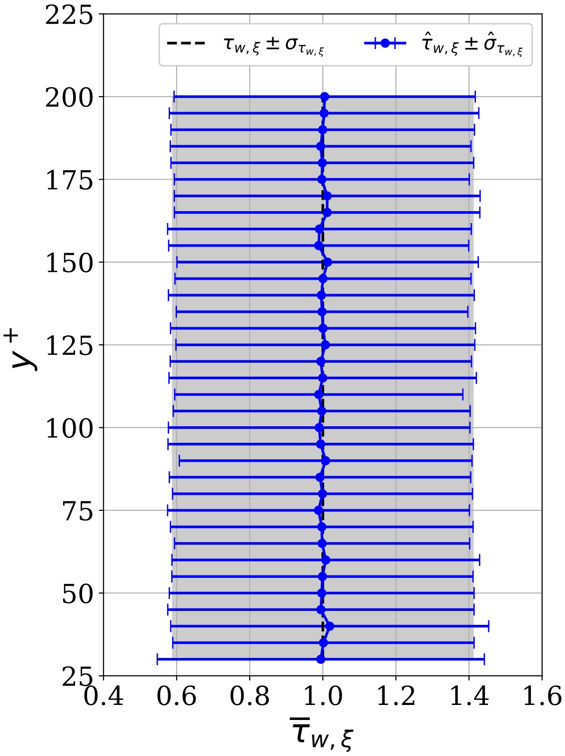


Figure 7.2: *A priori* test on TC1000: the black dashed line is the wrLES mean wall shear stress, while the gray area represents the wrLES standard deviation; the predictions are illustrated in blue, with the error bar indicating the predicted standard deviation.

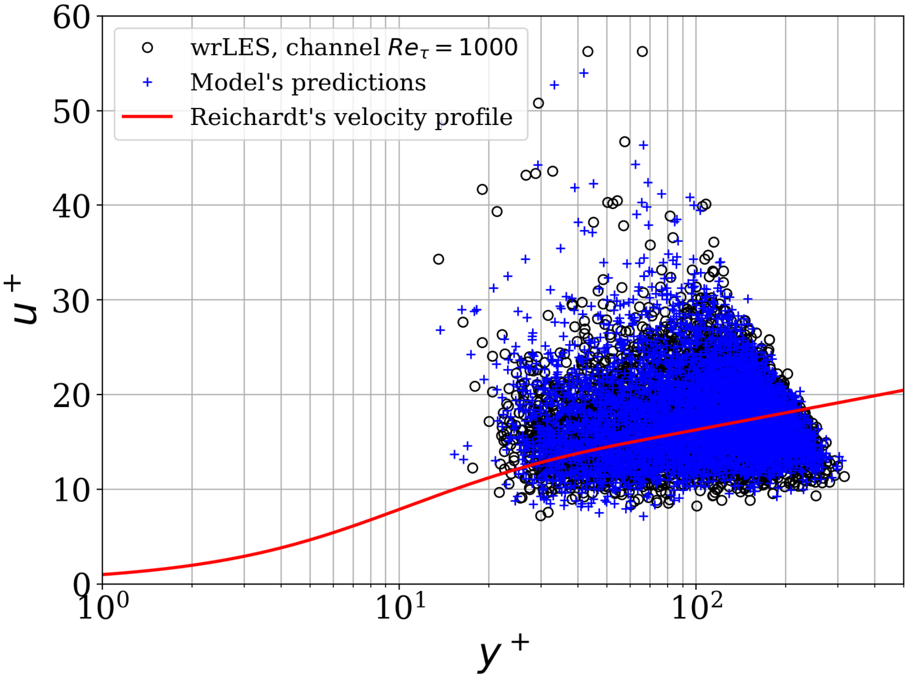


Figure 7.3: *A priori* test on TC1000: Distribution of the instantaneous scaled streamwise velocity u^+ as a function of the scaled distance to the wall y^+, using the target wall shear stress (black circle) and the predictions (blue plus symbol) to evaluate the wall unit scaling; Reichardt's profile is drawn in red.

7.2.2 *A priori* testing on synthetic data

The model is further tested on synthetic data. The benefit of such a database is that the model is assessed at higher friction Reynolds numbers, allowing for a better understanding of its generalizability. The generalization of the model is analyzed on the basis of three *functional levels*:

- the model can maintain its accuracy within the training limits (i.e., $30 \leq y^+ \leq 500$) when applied to higher friction Reynolds numbers (i.e., $Re_\tau > 5200$);
- the model is applied beyond its training limits (i.e., $y^+ > 500$) for the same friction Reynolds number (i.e., $Re_\tau = 5200$);
- the model is applied outside of its training boundaries (i.e., $y^+ > 500$) for higher Reynolds numbers (i.e., $Re_\tau > 5200$).

Figure 7.4 is the key to answering the above questions. Every Reynolds number overlaps each other and draws a unique curve that aligns with the LOTW for $90 \leq y^+ \leq 400$. For $250 < y^+ < 400$, the prediction slightly overestimates the LOTW (with a maximum relative error of 1.5%) due to an underestimation of the wall shear stress. For $y^+ > 400$, the curve quickly overestimates the LOTW, and no generalization above the upper training bounds is possible. To conclude, the model can generalize to higher Reynolds numbers as long as the input fields are extracted between the training bounds (i.e., $30 \leq y^+ \leq 500$).

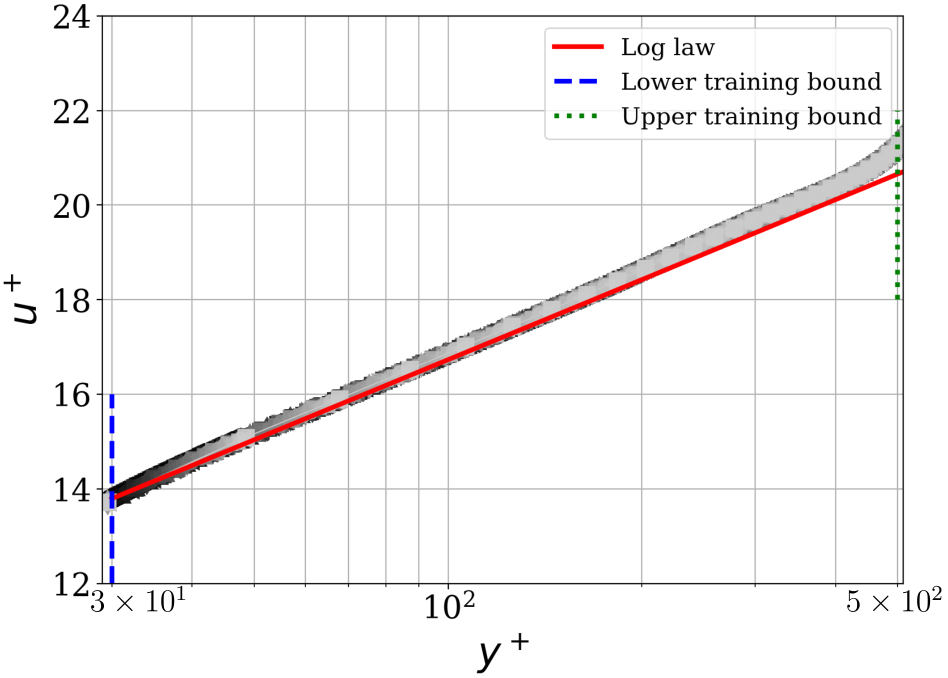


Figure 7.4: MLP-GMH averaged predictions on synthetic data extracted from the LOTW (Eq. 4.5); each gray-scale symbol represents a different friction Reynolds number from 180 to 10^4; the red line is the LOTW; the dash blue line and the green dotted line represents the lower ($y^+ = 30$) and upper ($y^+ = 500$) training bounds.

7.3 Reconstruction of the wall shear stress time-space correlations

The network is designed to predict a PDF rather than a pointwise estimate. This PDF is local and does not take into account any space or time correlations between neighboring surface probes. Although the predicted wall shear stress field at time t has the correct mean and variance, the time and space correlations are not recovered. As shown in Figure 7.5, the wall shear stress structures are incorrect, and too much energy is injected into the small scales, while for a channel the wall shear stress presents elongated structures in the streamwise direction due to the strong convection along that direction.

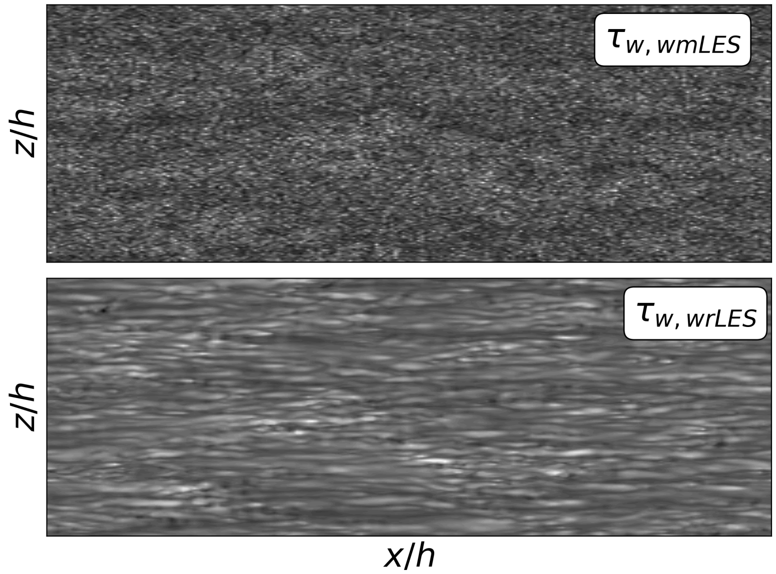


Figure 7.5: Instantaneous a priori predictions of the streamwise wall shear stress using the trained MLP-GMH for the channel flow at $Re_\tau = 950$, compared with the wrLES wall shear stress at a given time step t.

A statistical reconstruction of the space-time correlations is envisioned to improve the instantaneous prediction of the data-driven wall shear stress. The present section is thus dedicated to the *a priori* reconstruction (i.e., only theoretical considerations) of the space-time correlations of the predicted wall shear stress for a turbulent channel flow. The development presented has not been implemented in Argo-DG, except for the time reconstruction, because of two major drawbacks. Firstly, the reconstruction of the space-time correlation necessitates knowing *a priori* or being able to estimate at each (t, ξ, z) two integral length scales and the integral time scale. Therefore, it seems that reconstructing the instantaneous structures is a problem as complicated as the prediction of the wall shear stress on its own. Secondly, such a reconstruction can be computationally expensive and memory-intensive. Please note that the data-driven wall shear stress is implemented in a high-order flow solver (refer to Section 5.3), which already has many functionalities where memory can be the limiting factor. Wall-modeled LES has been introduced to reduce the computational time and make high Reynolds number geometry simulations feasible. Therefore, the wall model should be as efficient as possible to accelerate the computation.

You may be wondering why we predict the wall shear stress locally instead of predicting it on the entire solid wall. Predicting globally would have solved the issue with space and time correlations. However, doing a global prediction would have posed generalizability issues. Indeed, the periodic hill geometry is completely different from a channel wall. Therefore, it would be challenging to pass from one geometry to another. Additionally, the global prediction would require more communications in a highly scalable flow solver and therefore will not meet the design constraints stated in Section 3.5.

The reconstruction of the space-time correlation of the streamwise wall shear stress can be subdivided in different steps: (i) the reconstruction of the time correlation only, (ii) the reconstruction of the space correlation only, and (iii) ultimately the reconstruction of both space and time correlation.

7.3.1 Reconstruction of the time correlation

The reconstruction of the time correlation for the streamwise wall shear stress is local. This technique only requires the current and previous time steps to ensure correlation and communications are not necessary for its implementation. However, this method is a little more expensive in terms of storage because it requires the storage of the previous prediction at time step $t - \mathrm{d}t$ in an adequate container. The reconstruction of the time correlation uses conditional sampling, the theory behind Gaussian Processes (GP). The new prediction is conditionally sampled from the predicted distribution (i.e., the distribution predicted by the network as a linear combination of Gaussian distributions), knowing the previous wall shear stress and assuming a given kernel (e.g., exponential, square-exponential, or Matérn).

To use the formula of the conditional distribution of a multivariate normal distribution, we made the strong assumption that the marginal distribution is Gaussian. The MLP-GMH described in Table 7.2 uses one Gaussian distribution, and the assumption is thus true. However, a more general MDN predicts a mixture of K Gaussians, which renders the present reconstruction no longer possible. The conditional distribution of a multivariate normal distribution is defined here below.

Conditional distribution of a multivariate normal distribution

Let x follow a multivariate normal distribution

$$x \sim \mathcal{N}(\mu, \Sigma)$$

Then, the conditional distribution of any subset vector x_1, given the complement vector x_2, is also a multivariate normal distribution

$$x_1 | x_2 \sim \mathcal{N}(\mu_{1|2}, \Sigma_{1|2})$$

where the conditional mean and covariance are

$$\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \tag{7.1}$$

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{7.2}$$

with the block-wise mean and covariance defined as

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \tag{7.3}$$

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \tag{7.4}$$

Due to the periodicity and homogeneity of the streamwise and spanwise directions in a channel, the correlation time scale $\mathcal{T}$ is identical at each (x, z) position on the solid walls. A strong assumption is made about the correlation kernel, which ensures the smoothest of the reconstructed field. The γ-exponential kernel is a well-known kernel in GP, and it is defined as

$$f(r;l,\gamma) = \sigma_0^2 \exp\left\{-\left(\frac{r}{l}\right)^\gamma\right\} , \tag{7.5}$$

where l is the integral time (then noted as l_t) or length scale (then noted as l_ξ or l_z), γ an exponent to be fixed, and r a time or a distance. For the accurate reconstruction of the time correlation, l_t needs to be *a priori* set. In a new flow configuration, this time scale is not known *a priori*. One solution is to train a neural network to predict the integral time scale, l_t. This topic will be further explored in the thesis perspective.

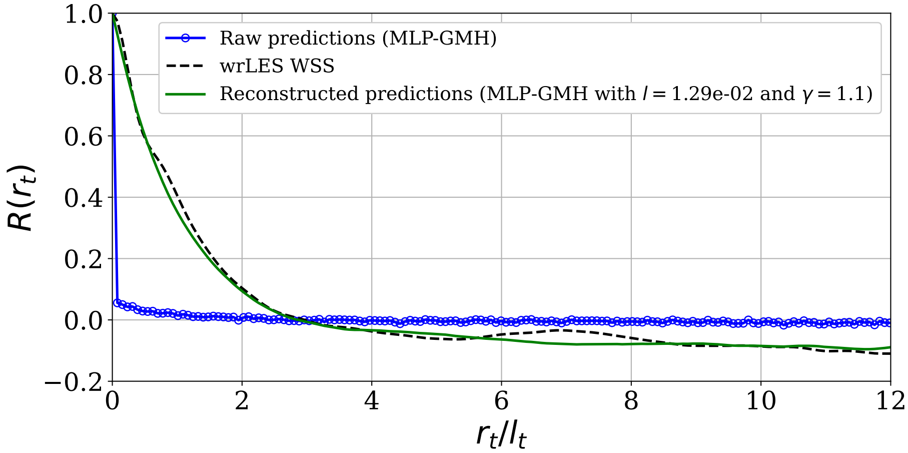


Figure 7.6: Local time correlation reconstruction: Average time correlations for the streamwise wall shear stress at $x/h = 0$ on the channel solid wall of the raw predictions in the blue circle line, the wrLES wall shear stress in the black dashed line, and the reconstructed predictions using the conditional sampling in the plain green line.

Assuming l_t and the kernel, the new predicted mean and variance are evaluated with Eq. 7.1 and 7.2 respectively with the block-wise covariance terms computed as,

$$\Sigma_{22} = \sigma_{t-1}^2 , \quad \Sigma_{11} = \sigma_t^2 , \quad \Sigma_{21} = \sigma_{t-1}\sigma_t \exp\left(\frac{-\mathrm{d}t^2}{2l_t^2}\right) ,$$

where $\mathrm{d}t$ is the time step, σ_t and σ_{t-1} are the standard deviation at time t and $t-\mathrm{d}t$ predicted by the neural network. Figure 7.6 shows the average time correlation $R(r_t)$

of the raw predictions, the reconstructed prediction using conditional sampling, and the wrLES $\tau_{w,x}$ at $x/h = 0$ on the channel solid wall. The average is computed along the spanwise direction. The raw predictions do not account for the previously predicted wall shear stress, and therefore, no time correlation is detected compared to the wrLES data, which shows a non-negligible time correlation. Since the time correlation is known *a priori*, an optimization procedure is set up to evaluate the parameter l of the γ-exponential kernel. The γ exponent is fixed at 1.1 because the slope of the time correlation is steep at the origin, but it can also be optimized. After about 100 iterations of the gradient descent, the optimal integral time scale $l(\nu/h^2)$ =1.3526×10^{-5} is obtained. As already mentioned, the present method is only applicable if the data-driven WSS model can predict the time scale itself or if such information is available *a priori*. The reconstructed time correlation is in good agreement with that of the wrLES, validating conditional sampling as a technique for reconstructing the time correlation of wall shear stress in turbulent channel flow.

7.3.2 Reconstruction of the space correlation

Whereas the reconstruction of time was relatively straightforward in case the marginal distributions are Gaussian, the reconstruction of the two-dimensional wall shear stress structures raises certain implementation complexities. There are two ways to reconstruct the space correlation of τ_w: global and local reconstruction.

Global reconstruction (Section 7.3.2.1) is based on sampling the wall shear stress from a multivariate normal distribution, which requires the evaluation of a huge covariance matrix. For instance, to generate 192 × 192 samples on the surface, a 192^4 container would be required to store the covariance matrix, corresponding to a memory storage of 5GB. A similar matrix is also evaluated for the spanwise wall shear stress. In a channel, the upper and lower walls are equivalent, but in other geometry, such as the two-dimensional periodic hill, the storage is double. Moreover, the serial implementation of the Cholesky decomposition has a cubic complexity. This approach is computationally expensive and requires high memory storage. Reducing memory storage is possible with compression methods (Humphreys *et al.*, 2015; Heavens *et al.*, 2017).

The second approach is local reconstruction (Section 7.3.2.2), where the current prediction is reconstruced using a small set of neighbours while considering the correlation length scales in wall-parallel directions (l_ξ and l_z). This approach is computationally more efficient for MPI communication and memory storage.

7.3.2.1 Global reconstruction approach

The method is based on the construction of the covariance matrix. This covariance matrix is not known *a priori*, therefore, its construction is also based on kernels, as in GP. The selected kernel is also the γ-exponential (Eq. 7.5) in both directions (i.e., the wall-parallel directions). This technique requires the access (or the knowledge) of the integral length scales: l_ξ and l_z. If direct access to the streamwise and spanwise correlations is possible, then the parameter l of Eq. 7.5 can be fitted by a quick optimization procedure as performed in Section 7.3.1. This optimization leads to $l_\xi/h = 0.15185$ with $\gamma = 1.1$ in the streamwise direction and $l_z/h = 0.026832$ with $\gamma = 1$ in the spanwise direction. The γ exponent is chosen to be close to one because of the steep slope of the space correlation at the origin. Since l_ξ and l_z are constant for each (x, z) position on the wall, the covariance matrix $\Sigma \in \mathbb{R}^{n_\xi n_z \times n_\xi n_z}$ is easily evaluated using blocks of size $n_\xi \times n_z$.

Although the memory storage for this matrix is high, it is symmetric, and therefore, the memory storage is divided by two. Furthermore, this matrix is usually sparse. In this example, with the optimal integral length scales obtained through the optimization procedure, only 4.7% of the matrix entries are non-zero, with a threshold defined at 10^{-12}. As a result, the memory storage is reduced from 5GB to only 250MB. Once the covariance matrix has been evaluated, its Cholesky decomposition is computed. Then, to generate the wall shear stress, the lower triangular Cholesky factor is multiplied by a white Gaussian noise of size $n_\xi n_z$ (i.e., samples extracted from a normal distribution with zero mean and unit variance). The resulting vector is subsequently resized to $n_\xi \times n_z$ and rescaled through the use of the predicted mean $\boldsymbol{\mu} \in \mathbb{R}^{n_\xi \times n_z}$ and standard deviation $\boldsymbol{\sigma} \in \mathbb{R}^{n_\xi \times n_z}$. Figure 7.7 shows the instantaneous reconstructed and the wrLES wall shear stress. Comparing the reconstructed field with the raw prediction in Figure 7.5 for the same time step, the long narrow streaks aligned with the streamwise direction are observed. However, in the spanwise direction, the structures are not as large as in the wrLES field.

Figure 7.8 shows the averaged space correlations for the streamwise wall shear stress, where the reconstructed correlations (in the solid green line) are in good agreement with the wrLES correlations (in the dashed black line). Notice that the proposed kernel in the spanwise direction does not allow negative correlation. This point

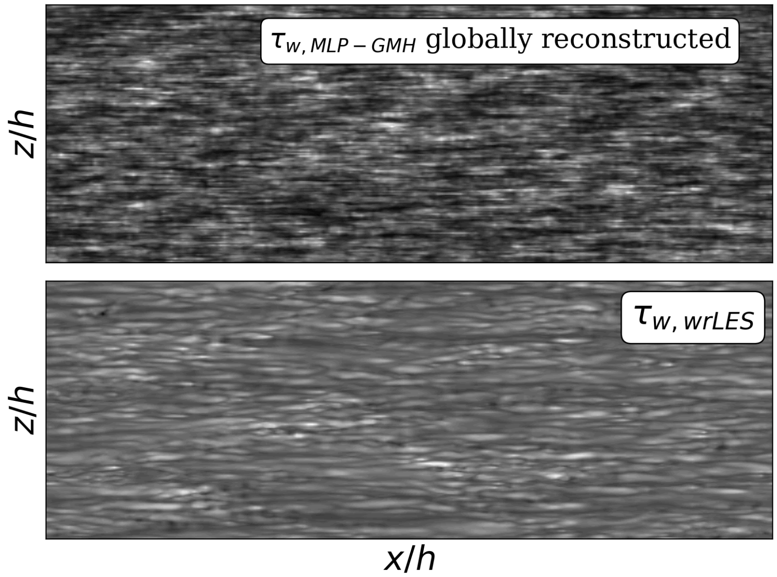


Figure 7.7: Global space correlation reconstruction: Instantaneous *a priori* predictions reconstructed in space of $\tau_{w,\xi}$ using a global method requiring the evaluation of the covariance matrix for the channel flow at $Re_\tau = 950$, compared to the wrLES wall shear stress.

should be improved. Figure 7.8 validates the global approach to reconstructing the space correlations.

7.3.2.2 Local reconstruction approach

For the local reconstruction of the two-dimensional wall shear stress structures, three alternative methods are considered: (1) two-dimensional autoregressive (AR) processes, (2) conditional sampling, and (3) linear reconstruction using blending coefficients. The methods (1) and (2) require communications between MPI ranks to pass information from one side of the surface to the other side. The prediction is thus propagated over the surface because the evaluation of the new prediction requires knowledge of its neighbors or a subset of its neighbors. In a brute-force algorithm, this evaluation is purely sequential. However, part of the evaluation can be parallelized, but this depends on the shape of the neighbors needed to reconstruct the space correlation. One might be interested in a purely local method where the prediction can be made simultaneously at each position with a limited

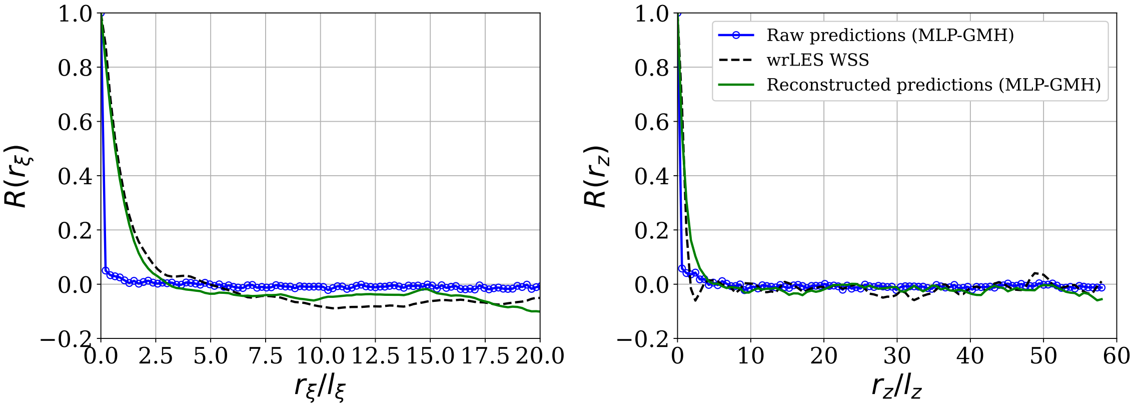


Figure 7.8: Global space correlation reconstruction: Averaged space correlations of $\tau_{w,\xi}$ at a given time step t on the channel solid wall of the raw predictions in the blue circle line, the wrLES wall shear stress in the black dashed line, and the reconstructed predictions using the conditional sampling in the plain green line; on the left the streamwise space correlation averaged in the spanwise direction; on the right the spanwise space correlation averaged in the streamwise direction.

amount of communication. Method (3), which uses blending coefficients, answers the requirement for simultaneous predictions. Unlike methods (1) and (2), which have some difficulties at the boundary, method (3) could probably deal better with boundary conditions. Methods (1) and (3) are discussed below, while conditional sampling is not presented because it has already been presented in Section 7.3.1 for the reconstruction of time correlations. The method has proven successful in the reconstruction of time correlations, but it encounters problems with the boundary conditions when applied to the reconstruction of two-dimensional space structures.

Two-dimensional autoregressive model for space correlation reconstruction 2D-AR models are used for texture reconstruction and analysis (Vaishali *et al.*, 2014; Kashyap and Eomm, 1988), and image filtering (Britos and Ojeda, 2018; Bustos *et al.*, 2009). Different real-world scenarios can be described by 2D-AR models with only a few parameters. For example, the first-order 2D-AR model can represent a wide range of textures with only two parameters. Considering a two-dimensional field τ_w of size $n_\xi \times n_z$, each position in the field is characterized by its location i, j and can be represented as $\tau_w[i, j] = \tau_{w,i,j}$, where $0 \leq i \leq n_\xi - 1$ and $0 \leq j \leq n_z - 1$.

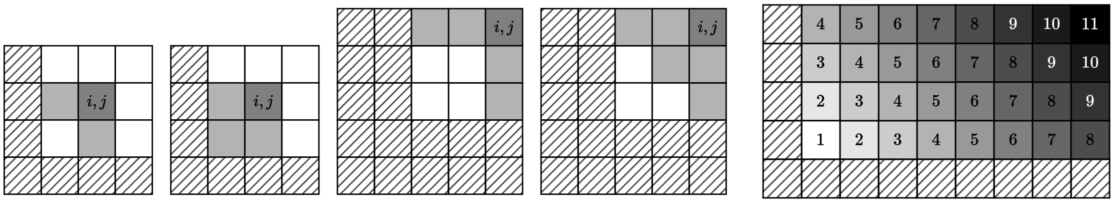
A two-dimensional (2D) autoregressive (AR) model is defined as,

$$\tau_{w,i,j} = \sum_{m=0}^{p_1-1} \sum_{n=0}^{p2-1} \alpha_{m,n} \tau_{w,i-m,j-n} + w_{i,j} \, , \tag{7.6}$$

where $[m,n] \neq [0,0]$, $\alpha_{m,n}$ is the AR coefficient, $w_{i,j}$ is a 2D white noise process with standard deviation $\sigma_w > 0$, and $p_1 \times p_2$ is the order of the model. The AR coefficient model $\alpha_{0,0}$ is assumed to be 1 for scaling purposes, therefore $[p_1 \times p_2 - 1]$ unknown coefficients have to be found. The conventional Yule-Walker equations are given by,

$$\sum_{m=0}^{p_1-1} \sum_{n=0}^{p_2-1} \alpha_{m,n} r_{m-k,n-l} = r_{-k,-l} \, , \tag{7.7}$$

for $k = 0, \ldots, p_1 - 1$ and $l = 0, \ldots, p_2 - 1$, where $[k,l] \neq [0,0]$, $[m,n] \neq [0,0]$, and, $r_{m,n} = \mathbb{E}[\tau_{w,i,j} \tau_{w,i+m,j+n}]$. An example of region of support (ROS) is depicted in Figure 7.9a.Note that the ROS could have been chosen symmetrically in the spanwise direction due to the homogeneity and periodicity of this direction. However, the present method is developed for an arbitrary flow that may not be extruded in the spanwise direction. If a transverse pressure gradient is imposed in the channel, a symmetrical reconstruction is no longer possible. Therefore, the regions of Figure 7.9a allow to get a causal autoregressive process.



(a) Region of Support (ROS) of a two-dimensional AR model; the two left are first order with $p_1 = p_2 = 2$ and the two right are second order with $p_1 = p_2 = 3$; the hatch areas are the ghost points initialized to zero; the ROS can takes multiples shapes: cross, quarter, or diamond.

(b) For a surface of size $n_\xi \times n_z$, $n_\xi + n_z + 1$ ghost terms are initialized to zero and each number corresponds to one of the $n_\xi + n_z - 1$ sequential steps.

Figure 7.9: ROSs of 2D-AR models and the parallel implementation of the first-order 2D-AR model with a cross ROS.

The simplest 2D-AR model is the first order AR model based on a cross ROS (see Figure 7.9a, 1st from the left). There are only two coefficients to find. Using the Yule-Walker system of equations, a system of two linear equations is obtained:

$$\begin{bmatrix} r_{0,0} & r_{1,-1} \\ r_{-1,1} & r_{0,0} \end{bmatrix} \begin{bmatrix} \alpha_{0,1} \\ \alpha_{1,0} \end{bmatrix} = \begin{bmatrix} r_{0,-1} \\ r_{-1,0} \end{bmatrix}. \tag{7.8}$$

The correlations are also approximated by a kernel. The optimization of l_ξ and l_z are already done in Section 7.3.2.1 and the same optimal parameters are used to evaluate the terms $r_{-k,-l}$ using the γ-exponential kernel. The variance of the WGN is adjusted to obtain a field with a zero mean and a unit variance. The predicted field is then rescaled with its local mean and variance. This instantaneous prediction is compared with the wrLES wall shear stress in Figure 7.10. Although the reconstructed field presents elongated structures aligned with the streamwise direction, these structures appear slightly skewed, probably due to the selected ROS.

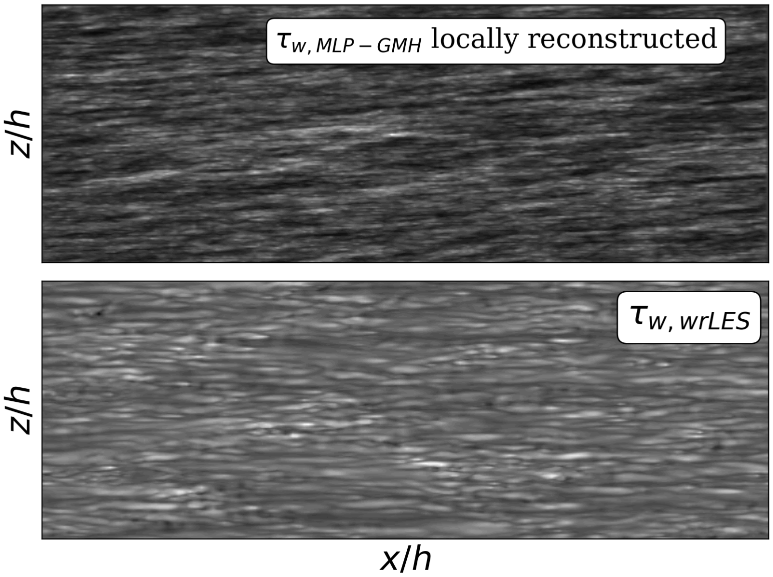


Figure 7.10: Local space correlation reconstruction: Instantaneous *a priori* predictions reconstructed in space of $\tau_{w,\xi}$ using a local method based on a first-order 2D-AR model (with $\alpha_{0,1} = 0.211$ and $\alpha_{1,0} = 0.736$) for the channel flow at $Re_\tau = 950$, compared to the wrLES wall shear stress.

Figure 7.11 shows the averaged space correlations for the streamwise wall shear stress, where the reconstructed correlations (in the solid green line) are also in good agreement with the wrLES correlations (in the dashed black line). The predictions are as good as those obtained with the global method, and they are obtained with only two parameters. However, this reconstruction involves more communication and introduces a latency between the evaluation of the lower left corner and the upper left corner. This latency can be reduced by grouping positions that can be evaluated simultaneously rather than sequentially, as in a brute-force algorithm. Figure 7.9b illustrates the partial parallelization that can be implemented to speed up the predictions. The example counts $(n_\xi \times n_z) = (8 \times 4)$ positions. In a sequential algorithm, the evaluation requires 32 steps, while the improved approach counts only 11 sequential steps. The number of sequential steps is evaluated as the number of diagonals in the matrix, which is $n_\xi + n_z - 1$. Regarding the example used in Section 7.3.2.1 with the 192 × 192 positions, the clever implementation of the reconstruction using 2D-AR models will require only 383 steps instead of 36864 in the brute-force implementation, leading to a reduction of about 100.

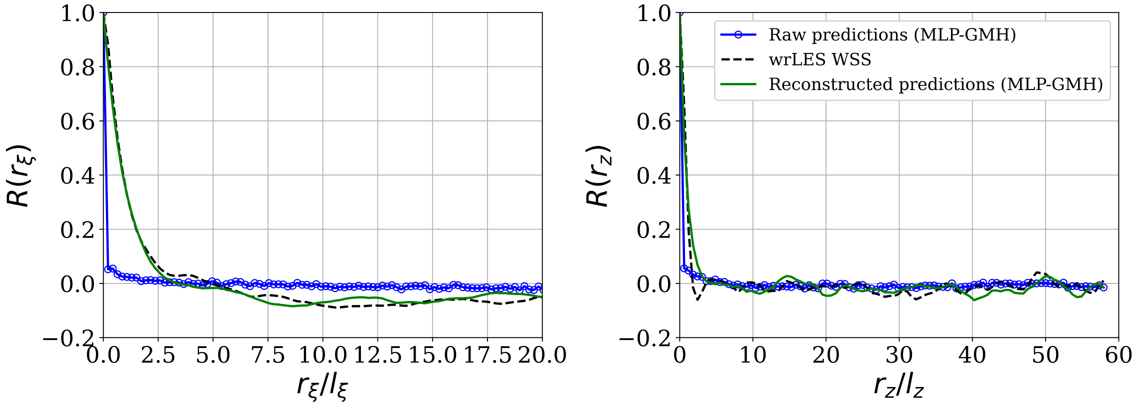


Figure 7.11: Local space correlation reconstruction: Averaged space correlations of $\tau_{w,\xi}$ at a given time step t obtained with a first-order 2D-AR model (with $\alpha_{0,1} = 0.211$ and $\alpha_{1,0} = 0.736$) on the channel solid wall: the raw predictions are drawn in the blue circle line, the wrLES wall shear stress in the black dashed line, and the reconstructed predictions in the plain green line; on the left the streamwise space correlation averaged in the spanwise direction; on the right the spanwise space correlation averaged in the streamwise direction.

Linear reconstruction using blending coefficients This technique is based on the modeling of the new predictions as a linear combination of their nearest neighbours for a simultaneous evaluation of the entire surface. For the streamwise and spanwise reconstruction, the linear combination is written as

$$\tau_{w,i,j} = \sum_{q,r} \beta_{q,r} \epsilon_{i+q,j+r} + w_{i,j} \, , \tag{7.9}$$

where q, r can be both positive and negative (i.e., up and downstream information and symmetrical stencil in the spanwise direction if needed), $\epsilon_{i,j}$ are the "*raw predictions*" sampled from the predicted distribution and $w_{i,j}$ is a white Gaussian noise of variance σ_w^2 as defined in the AR process. The unknowns are the $\beta_{q,r}$ coefficients as well as the variance of the WGN. To write the system of equations, we first have to ensure the mean and the variance of the new predictions. The second step is to ensure the space correlations with the neighbours implied in the linear combinations. To evaluate the correlations, kernels and *a priori* estimates of l_ξ and l_z are again used. To simplify the nonlinear system of equations, a hypothesis is drawn. The $\epsilon_{i,j}$ are independent: $\mathrm{Cov}(\epsilon_{i,j}, \epsilon_{k,l}) = \delta_{i,j,k,l}$. In the end, the system is overdetermined. If the equation of the variance is removed, the new predicted field has the wrong variance. If the mean equation is removed, the system has no real solution. To conclude, blending random predictions inevitably reduces the variance of the reconstructed field. Although the method was promising, the reduction of the variance is not a viable solution.

Conclusion Two methods for reconstructing the two-dimensional structures of the streamwise wall shear stress have been proposed: a global and a local reconstruction with their respective advantages and disadvantages. The global method involves a computationally expensive Cholesky decomposition of the covariance matrix, but its storage can be reduced due to symmetry and sparsity. The local method uses a first-order 2D-AR with a compact stencil and potential for parallelization, but it can not handle boundary conditions. The slight skew of the reconstructed field can be corrected by using a different ROS.

7.3.3 Reconstruction of the time and space correlation

Ultimately, an effective reconstruction should account for both time and space correlations. The time reconstruction is always based on a local method while the space reconstruction can be global or local as discussed in Section 7.3.2. Based on results obtained in the previous section, the global method, based on the computation of the covariance matrix, is selected to reconstruct the two-dimensional structures of the wall shear stress and this method is combined to a first-order AR model in time. This AR(1) model is written as,

$$\tau_w^t = \alpha \tau_w^{t-1} + w^t \,, \tag{7.10}$$

where $|\alpha| < 1$ and $w^t \sim \mathcal{N}(0, \sigma_w)$.The α coefficient is computed using the Yule-Walker equations, where the time correlation is approximated by a γ-exponential with $l_t^* = 0.01185$ and $\gamma = 1.1$ as in Section 7.3.1. To ensure a unit variance of τ_w^t, the variance of the WGN σ_w^2 is set to 0.3514, knowing that the variance of a first-order AR model is given by $\mathbb{V}\text{ar}\left(\tau_w^t\right) = \sigma_w^2 / \left(1 - \alpha^2\right)$. At each position (i, j) on the surface, the AR model (Eq. 7.10) generates a time sequence with a zero-mean and unit-variance. At each time step t, the surface of size $n_\xi \times n_z$ is extracted from the time sequence and multiplied by the lower-triangular Cholesky factor in Section 7.3.2.1. For a statistically stationary flow, the covariance matrix is assumed to remain unchanged from one-time step to the next, allowing the evaluation of the covariance matrix and its Cholesky decomposition only once at the beginning of the simulation. This assumption significantly reduces the computational time. The result of the matrix-vector multiplication is rescaled using the predicted mean and variance at each (i, j) position on the surface. The wall shear stress to be imposed as a boundary condition is finally obtained by dimensionalizing the reconstructed predictions according to Eq. 5.9.

Figure 7.12 shows the averaged time and space correlations. The time correlation is evaluated over 400 time steps and averaged along the streamwise and spanwise directions. The streamwise (spanwise) correlation is averaged in time and along the spanwise (streamwise) direction. The time and space correlations are accurately reconstructed, except for the streamwise correlation, which appears to have a smaller integral length scale when compared to the wrLES. Nevertheless, the method is *a priori* easily implemented, but a clever implementation in the Argo-DG flow solver is required to avoid a severe deterioration of the computational time. This implementation was not addressed during the thesis since no method to acquire *a posteriori* the integral time and length scales (l_t, l_ξ, l_z) at each (i, j)-position has been developed.

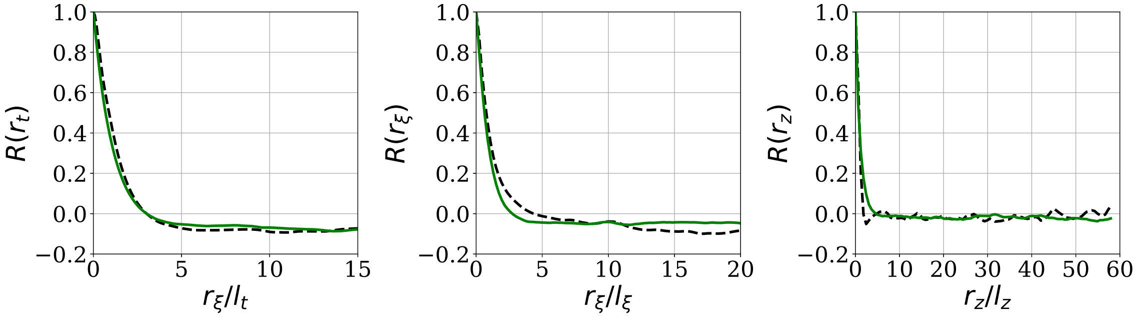


Figure 7.12: Local time and local space reconstruction: Averaged time and space correlations of $\tau_{w,\xi}$ on the channel solid wall at a given time step t obtained with an AR(1) model in time and a global reconstruction: the legend is similar as Figure 7.11; from left to right the time $R(r_t)$, streamwise $R(r_\xi)$ and spanwise $R(r_z)$ correlations.

7.4 *A posteriori* testing

The novel Mixture Density Network wall model is assessed *a posteriori* on two turbulent channel flows: $Re_\tau = 950$ (Section 7.4.1), and $Re_\tau = 2{,}000$ (Section 7.4.2). Both simulations are performed at a polynomial order $p = 4$, as suggested in the thesis of Frère (2018). The influence of the polynomial order has not been investigated. Both simulations are initialized by a randomly perturbed Reichardt profile and driven by a constant pressure gradient. Starting from this disturbed profile, the simulation experiences a numerical transient, which is case-dependent, mesh-dependent, and code-dependent, before reaching a statistically converged state.

Remember that the data used to train the neural network was acquired solely from the statistical converged state because data extracted from the numerical transient is unsuitable for the reasons mentioned. In the early stage of the *a posteriori* test on the channel flow, the data-driven WSS model was directly turned on even in the initial transient. Unfortunately, because the network was not trained on this very specific state, the flow behavior was incorrect. The flow was able to reach a statistically converged state but with a too-low bulk velocity.

To overcome this situation, the wmLES is initiated with a WSS model based on Reichardt's velocity profile (i.e., the analytical WSS model implemented by Frère (2018)) to evacuate this initial transient (after $\sim 50t_c$). A constant pressure gradient drives the flow for this part of the wmLES. Once a statistically converged state is obtained, the WSS model is switched to our novel data-driven WSS, and the source term is switched to a constant mass flow rate. Due to a switch in WSS models, a small transient is observed but does not affect the overall flow behavior. Once the flow has converged, the statistics are accumulated for about $50t_c$.

Note the change from a constant pressure gradient to a constant mass flow rate. Multiple tests were conducted with a constant pressure gradient or mass flow rate. It was noticed that the data-driven WSS model was quite sensitive to the constant pressure gradient. In the case of a constant pressure gradient, the predicted wall shear stress must exactly match the source term to stabilize the simulation. Although the neural network was trained with an acceptable error of 1%, the simulations did not converge: either the model under-predicted the wall shear stress (i.e., $\tau_w = 0.99$) and the flow constantly accelerated or the model over-predicted the wall shear stress (i.e., $\tau_w = 1.01$) and the flow constantly decelerated.

7.4.1 Turbulent channel flow at $Re_\tau = 950$

The effectiveness of the data-driven wall model is initially evaluated on the channel flow at $Re_\tau = 950$. This case had been used before to *a priori* evaluate the wall model and is now adopted for the *a posteriori* test to determine if the model behaves similarly in the simulation environment. The results (i.e., mean and Reynolds stress profiles) are compared against the DNS of Hoyas and Jimenez (2008), the wrLES of the same test case with Argo-DG, and the wmLES of Dupuy *et al.* (2023b).

The domain size is $2\pi h \times 2h \times \pi h$, where h is the half-height of the channel. Following the recommendation of Frère (2018), the simulation is performed at a fourth polynomial order ($p = 4$). In contrast to the wrLES mesh, no geometric progression is applied in the near-wall region. Therefore, the first mesh cell has a wall unit size of 95. The wmLES mesh is regular and uniform and composed of $(N_x, N_y, N_z) = (20, 20, 10)$ cells which corresponds to an effective mesh resolution of $(\Delta x^+, \Delta y^+, \Delta z^+) \simeq (75, 25, 75)$. The instantaneous data are extracted at $y^+ = 100$, which approximately corresponds to the first rake of interpolation points in the second grid cell.

The time integration is performed with an implicit method. The time step is fixed at $\mathrm{d}t(u_\tau/h) = 1 \times 10^{-3}$ to match a convective CFL number of 0.25. The simulation

Table 7.3: wmLES parameters for the turbulent channel flow at $Re_\tau = 950$.

Parameters	**Values**
Time step	$\mathrm{d}t(u_\tau/h)$ =1×10^{-3}
Polynomial order	$p = 4$
Initial condition	Randomly perturbed Reichardt's velocity profile
Source term	Mixed: $([\frac{\nu}{\rho}\frac{\partial p}{\partial x}]^{1/3})^+$ =1.0172×10^{-1} and $u_b^+ = 19.775$
Mesh size	$(N_x, N_y, N_z) = (20, 20, 10)$
Efficive mesh size	$(\Delta x^+, \Delta y^+, \Delta z^+) = (75, 24, 75)$
Wall-modeled height	$h_{wm}^+ = 100$
Wall models	AWSSR (initial transient) and MLP-GMH (statistics)

is run and equilibrated using prescribed bulk velocity u_b. The wmLES is performed with the MLP-GMH network, which has been trained on the TC5200 test case (see Table 7.1). After evacuating the initial transient, the statistics are accumulated over approximately 60 flow through time (i.e., $t_c = L_x/u_b$, where L_x is the streamwise domain length). Figure 7.13 shows the mean velocity profiles where a good agreement with the references is observed. The wmLES profile overlaps with the DNS one (Hoyas and Jimenez, 2008) and our wrLES. The mean profile obtained by Dupuy *et al.* (2023b) with the Graph Neural Network (GNN) using far input to correct for the log-layer mismatch (Kawai and Larsson, 2012) overestimates the DNS for every wall-normal coordinates. In their case, the input velocity is extracted at three cells away from the wall, which corresponds approximately to $h_{wm}^+ \simeq 110$ for their original mesh. As expected, the mean profile matches Reichardt's velocity profile at that specific point (i.e., h_{wm}). Our wmLES (Argo-DG and MLP-GMH) aligns well with the LOTW within the logarithmic range from 93 to 140, which is very small at this friction Reynolds number. The mean velocity profile obtained with our novel WSS model oscillates in the first cell (i.e., the gray area in Figure 7.13) because the solution is not a physical solution of the flow in this first DG cell.

Examining the Reynolds stresses profiles in Figures 7.14a, 7.14b, and 7.14c, small discrepancies with the references are noticeable for u' and w', only. The profile of u' is already underestimated in the wrLES of the same test case. The wmLES is

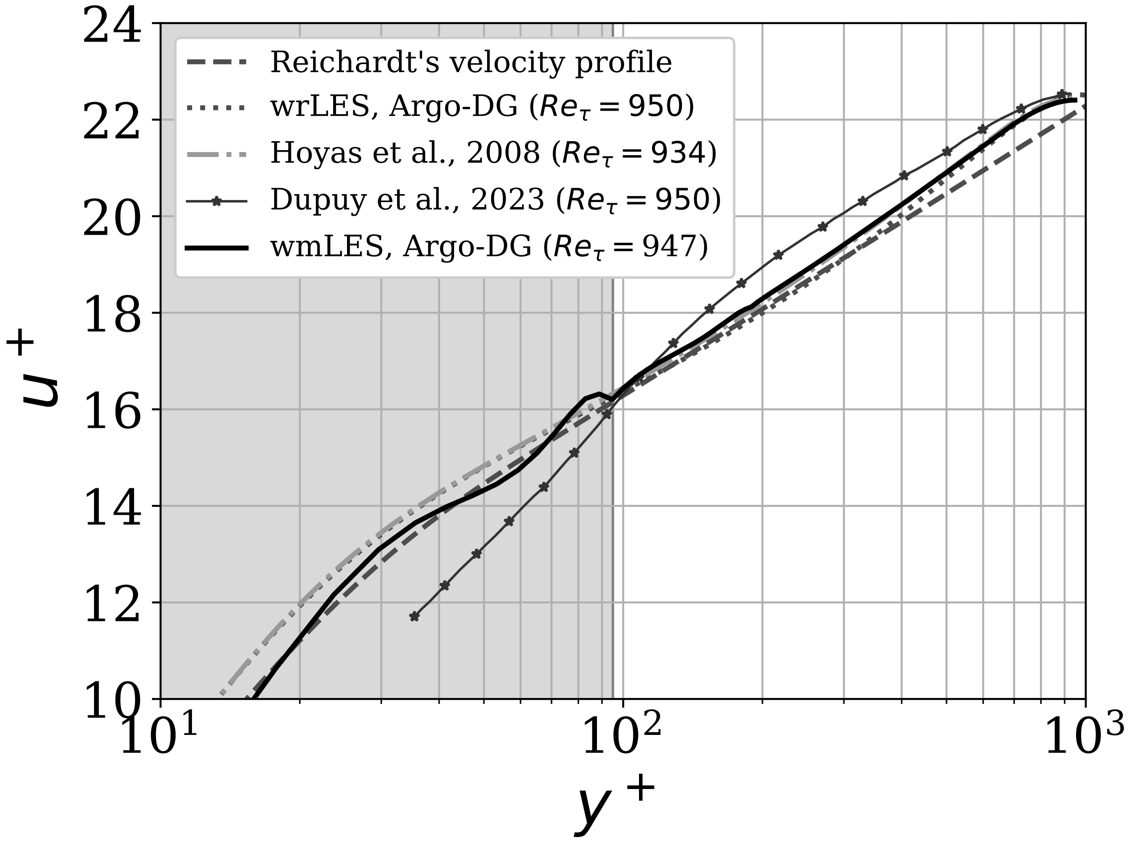
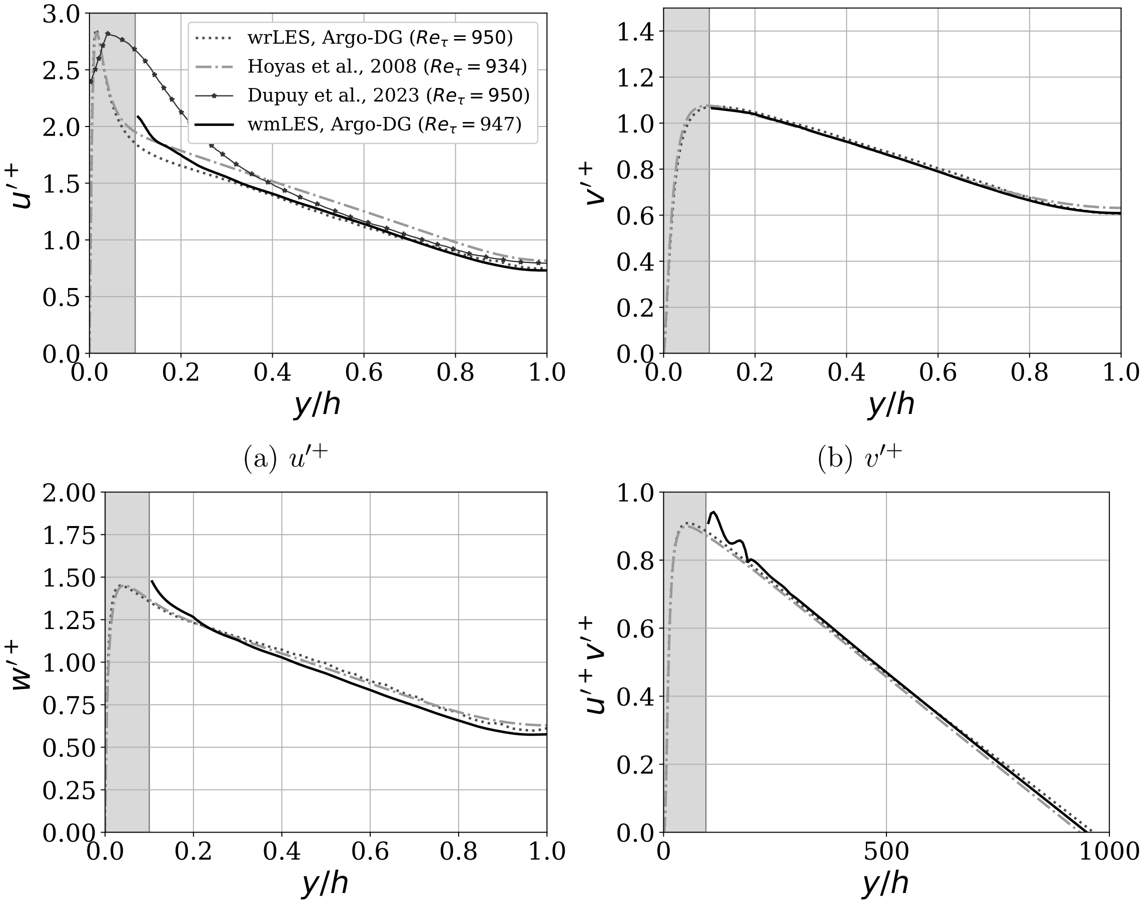


Figure 7.13: Mean streamwise velocity profile in wall units for a channel at $Re_\tau = 950$ simulated with a data-driven wall model; all profiles are compared with the DNS of Hoyas and Jimenez (2008) and with the wrLES of the same test case using Argo-DG; the gray area indicates the size of the first grid cell.

in good agreement with the wrLES curve (dotted gray), except that a mismatch is observed at the interface between the first and second cells. This discrepancy is also visible in the covariance between u and v. Oscillations occur in the second cell. The RMS profile u'^+ obtained in Dupuy *et al.* (2023b) presents a larger discrepancy with the DNS between $y/h = 0.1$ and $y/h = 0.3$ (i.e., close to the wall). This observation suggests that such a discrepancy is common in wmLES. Most of the wmLES of the channel flows are performed on uniform meshes. The size of the second grid cell may be too large to accurately capture the second statistical moment of the velocity. Increasing the resolution of this cell could potentially resolve this issue.



(a) u'^+

(b) v'^+

(c) w'^+

(d) Covariance of u and v in wall units

Figure 7.14: Reynolds stresses profils in wall units for a channel at $Re_\tau = 950$ simulated with a data-driven wall model; all profiles are compared with the DNS of Hoyas and Jimenez (2008); the gray area indicates the size of the first grid cell.

The Reynolds stress profile v'^+ is in good agreement with the DNS in the second grid cell. At the channel center, v'^+ and w' are slightly underestimated. Nonetheless, these discrepancies with the DNS were already observed by Frère (2018). Therefore, they are not specific to the data-driven wall model but to the DG method for wrLES and wmLES.

7.4.2 Turbulent channel flow at $Re_\tau = 2{,}000$

This friction Reynolds number is selected because it has a logarithmic layer included in the training y^+ values. According to Marusic *et al.* (2013), the extension of the logarithmic region varies with the friction Reynolds number as follows,

$$3\sqrt{\frac{\delta u_\tau}{\nu}} \leq y^+ \leq 0.15\frac{\delta u_\tau}{\nu}.$$

For a friction Reynolds number of $Re_\tau = 2{,}000$, the logarithmic region extends from $y^+ = 135$ to $y^+ = 300$, which is included in the training bounds. The domain size is the same in Section 7.4.1. The mesh is also regular and uniform and composed of $(N_x, N_y, N_z) = (20, 16, 16)$ cells which corresponds to an effective resolution of $(\Delta x^+, \Delta y^+, \Delta z^+) \simeq (160, 62, 100)$. The instantaneous data are extracted at $y^+ = 280$ which is located in the second grid cell. The time step is fixed at $\mathrm{d}t(u_\tau/h) = 10^{-3}$ to match a CFL number of 0.28. All the parameters of the wmLES are summarized in Table 7.4.

Table 7.4: wmLES parameters for the turbulent channel flow at $Re_\tau = 2{,}000$.

Parameters	Values
Time step	$\mathrm{d}t(u_\tau/h) = 10^{-3}$
Polynomial order	$p = 4$
Initial condition	Randomly perturbed Reichardt's velocity profile
Source term	Mixed: $([\frac{\nu}{\rho}\frac{\partial p}{\partial x}]^{1/3})^+ = 7.937 \times 10^{-2}$ and $u_b^+ = 21.66$
Mesh size	$(N_x, N_y, N_z) = (20, 16, 16)$
Efficive mesh size	$(\Delta x^+, \Delta y^+, \Delta z^+) = (160, 62, 100)$
Wall-modeled height	$h_{wm}^+ = 280$
Wall model	AWSSR (initial transient) and MLP-GMH (statistics)

After evacuating the initial transient, statistical data is collected over approximately $\sim 50t_c$. The mean and Reynolds stress profiles are shown in Figures 7.15 and 7.16, respectively. They are compared with the DNS data of Hoyas and Jimenez (2008) and to the wmLES of Dupuy *et al.* (2023b) and of Radhakrishnan *et al.* (2021). The former used a Graph Neural Network to predict the wall shear stress, and the latter used Gradient Boosted Decision Tree. Dupuy *et al.* (2023b) used the same wall unit mesh refinement as for $Re_\tau = 950$. Therefore, the input velocity is extracted at $h^+_{wm} \simeq 110$. Our results are compared with the finer mesh of Radhakrishnan *et al.* (2021) with a wall-normal height of $h^+_{wm} = 31$. In both cases, this matching location is visible in the mean velocity profile.

The mean profile agrees well with the DNS results, although a relative error of 1% is detected in the wake region. The mean profile of Radhakrishnan *et al.* (2021) is overestimated near the wall and underestimates the wake. The mean profile of Dupuy *et al.* (2023b) overestimates the DNS at every wall-normal coordinate. Our novel WSS model outperforms the existing data-driven WSS model in the literature. The oscillations of the mean velocity profile in the first cell (i.e., the under-resolved region) are directly linked to the DG method, as discussed in the thesis of Frère (2018).

The Reynolds stress profiles show a bigger offset in comparison to the turbulent channel at lower Re_τ. This offset is constant over the entire domain. Although the u'^+ profile of Dupuy *et al.* (2023b) underestimates the DNS, their offset is smaller. Firstly, their wmLES grid is finer than ours. Secondly, their input velocity is extracted closer to the wall. Therefore, their boundary layer is better resolved, resulting in a better fit of u'^+ close to the wall.

The Reynolds stresses v'^+ and w'^+ are underestimated at the center of the channel. Small oscillations are still observed in the second grid cell. Resolution issues are highlighted in Figure 7.16d where the covariance between u and v in wall units is shown. The curve presents oscillations between $y^+ = 250$ and $y^+ = 500$, corresponding to the second grid cell. Nevertheless, for $y^+ > 500$, there is a good agreement with the DNS. The small discrepancy between the two curves is due to the slightly different friction Reynolds numbers.

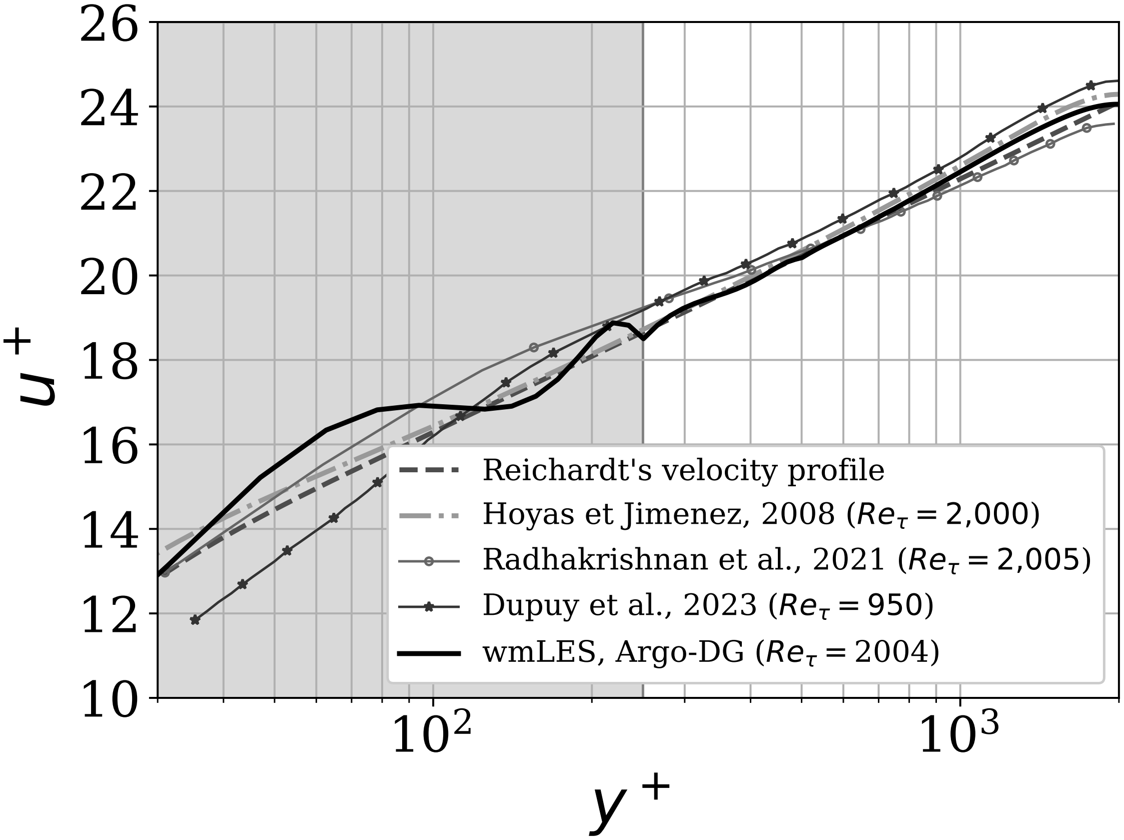
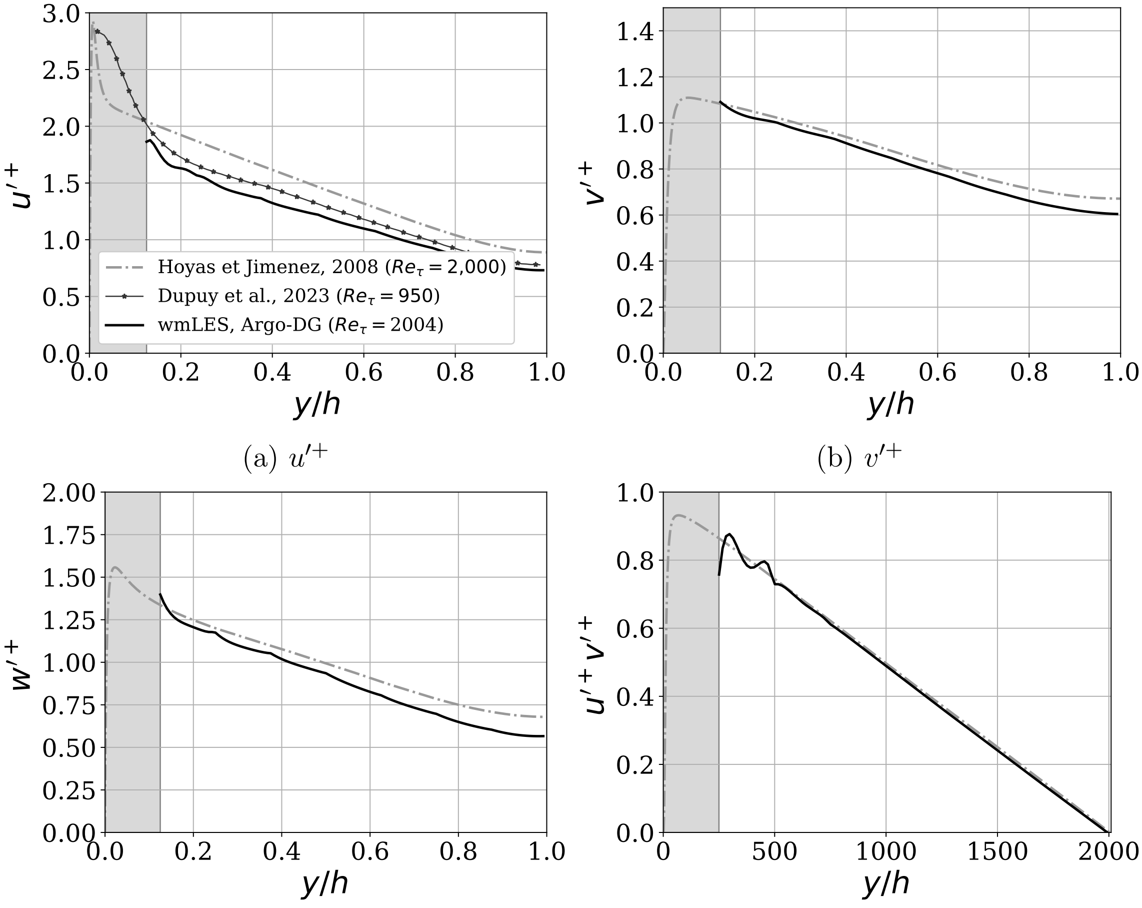


Figure 7.15: Mean streamwise velocity profile in wall units for a channel at $Re_\tau =$ 2,000 simulated with a data-driven wall model; all profiles are compared with the DNS of Hoyas and Jimenez (2008) and with the wrLES of the same test case using Argo-DG; the gray area indicates the size of the first grid cell.



(a) u'^+

(b) v'^+

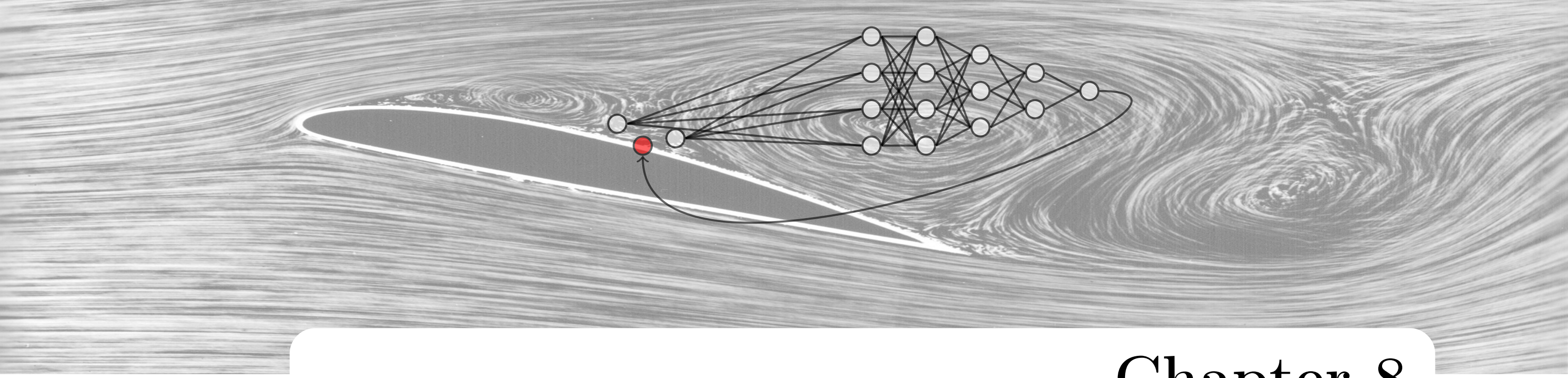
(c) w'^+

(d) Covariance of u and v in wall units

Figure 7.16: Reynolds stresses profils in wall units for a channel at $Re_\tau = 2{,}000$ simulated with a data-driven wall model; all profiles are compared with the DNS of Hoyas and Jimenez (2008); the gray area indicates the size of the first grid cell.

7.5 Conclusion

This chapter focuses on the training, *a priori* and *a posteriori* tests of our novel data-driven WSS model (MLP-GMH) on turbulent channel flows. The MLP-GMH was trained on the Channel5200 test case extracted from the JHTDB website. The data were collected at 95 wall-normal locations, which allowed, with the non-dimensionalization $.^*$, to extrapolate to higher friction Reynolds numbers *if and only if* the wall-modeled height remained within the training bounds (i.e., $30 \leq h^+_{wm} \leq 500$). The *a priori* test was satisfactory in terms of the mean and the standard deviation of the predicted wall shear stress compared to the analytical WSS model based on Reichardt's velocity profile. The *a posteriori* tests were conducted at two friction Reynolds numbers: $Re_\tau = 950$ and $Re_\tau = 2{,}000$. The turbulent statistics were in good agreement with the DNS of Hoyas and Jimenez (2008) for both friction Reynolds numbers. Moreover, our novel WSS model outperforms the existing data-driven WSS models in the literature when comparing the mean velocity profile in wall units. Concerning the Reynolds stresses, our predictions are quite good as compared to the DNS data, and are similar to those obtained when using algebraic wall models as in Frère (2018).



Chapter 8
Separated flow in the two-dimensional periodic hill configuration

The two-dimensional periodic hill, presented in Chapter 4, is a benchmark to validate new wall models (Krank *et al.*, 2019; Zhou *et al.*, 2021, 2022). This test case is simple regarding the simulation setup and the boundary conditions. However, it is sufficiently complex in terms of the turbulence phenomena (i.e., separation and reattachment). This test case has been selected to provide insight into the performance of our novel data-driven WSS model in separated turbulent boundary layers. Therefore, this chapter discusses the development of a data-driven WSS model applied to the two-dimensional periodic hill. A similar methodology as in Chapter 7 is used to train, validate, and test the novel WSS model assisted by neural networks. The training and validation parameters are presented in Section 8.1, while the *a priori* tests are performed in Section 8.2.

Based on the recommendations of Section 5.1.2, the input stencil is increased in the streamwise direction to better capture the correlation maps. A CNN is now considered to model the non-linear relationship between volume fields and the wall shear stress in separated turbulent boundary layers, as discussed in Chapter 6. This CNN is also connected to two GMHs: one for the prediction of $\tau_{w,\xi}$ and one for $\tau_{w,z}$. In this chapter, the reconstruction of the wall shear stress structures is not discussed. It appears that the instantaneous prediction of the wall shear stress on the lower wall shows better space correlations, except that too much energy is still injected into the small structures. This better reconstruction is probably due to the variation of the mean along the streamwise direction, which was not the case for the turbulent channel.

Upon successful *a priori* examination of the neural network, it can subsequently be implemented in Argo-DG and tested *a posteriori*. As previously discussed in

Chapter 7, a neural network that successfully performs when assessed *a priori* is not guaranteed to perform well when evaluated *a posteriori*. The production environment differs from the training environment, with several factors affecting the prediction, such as the feedback loop between the prediction and volume fields and the SGS model's impact. Additionally, the accurate resolution of the free shear layer is vital in predicting the reattachment location for the two-dimensional periodic hill. It is as essential as accurately capturing the separation location. These points are further treated in Section 8.3.

8.1 Training

To predict the distribution of wall shear stress (i.e., $\tau_{w,\xi}$ and $\tau_{w,z}$) for the two-dimensional periodic hill flow, a combination of a CNN and GMHs. The CNN architecture is not limited to convolution layers alone but consists of a series of convolutional and residual blocks. The interest for skip connections were discussed in Section 5.1.2.2. The design of the architecture is similar to a ResNet, but to avoid confusion, it is labeled as CNN-skip-GMH (see Table 7.1). The model is made up of 10,212 parameters due to the increase in the depth of the network and the size of the input stencil. The hyperparameters chosen for training are summarized in Table 8.1 and discussed in Section 8.1.1.

Figure 8.1 shows the training and validation losses for the CNN-skip-GMH network. The loss is negative because the network is trained on the negative log-likelihood, unlike networks that are trained on minimizing the mean square error. During the first 100 epochs, the loss undergoes a rapid decay. The training and validation losses mostly follow a straight line, indicating an appropriate learning rate. No fluctuations are detected in the losses, which is probably due to the smooth loss landscape provided by the skip connections. At around 90 epochs, the loss experiences a small drop. From this epoch, the losses continue to decrease but at a slower rate. It suggests that the training has reached a plateau, and reducing the learning rate might have been beneficial. Usually, the training is stopped before the validation loss increases again to avoid overfitting. Nonetheless, such behavior is not observed in Figure 8.1. The validation loss follows the training loss closely but exhibits small oscillations. The model is therefore saved at the epoch where the validation loss is minimal. One possible explanation is that the model is not enough trained, implying that a better minimum has not yet been reached. Another hypothesis is that the validation set does not differ sufficiently from the training set. It would have required a larger database with more test cases, including separation. This perspective will be addressed within the conclusion of the thesis.

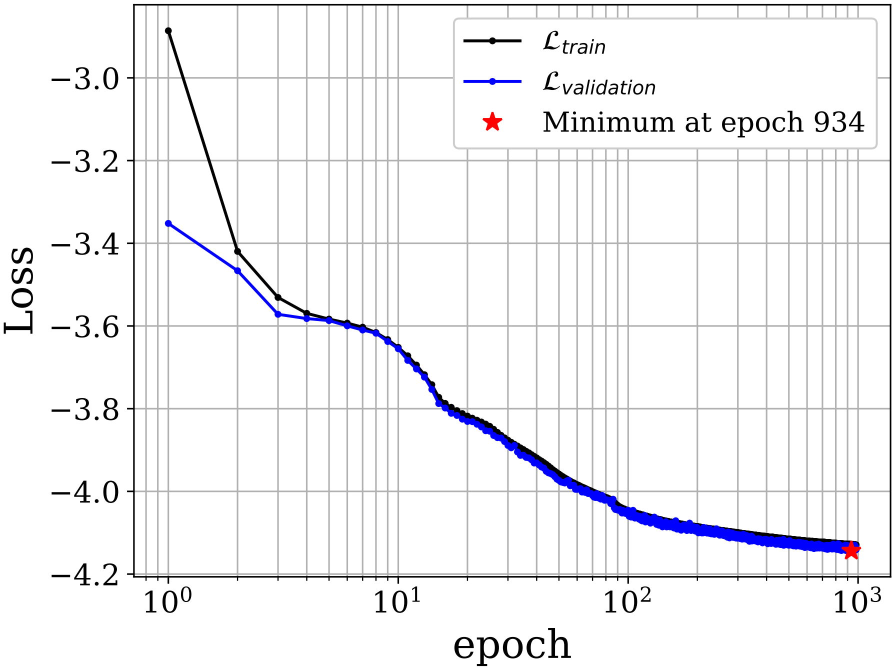


Figure 8.1: Evolution of the training and validation losses during the training of the network CNN-skip-GMH

8.1.1 Discussion on the hyperparameters

In Chapter 7, a random search was used to select the best number of hidden layers and neurons per layer. However, in a CNN, there are many more hyperparameters to fit. A convolutional layer consists of a kernel size, a padding, a stride, a dilation, and the output channel size. The dilation, the stride, and the padding are set to their default values (i.e., 1, 1 and 0, respectively), except in the residual block, where the padding is set to 1. A common value for the kernel size in image recognition is 3. The output channel is set at twice the size of the input channel. The aim is to maintain the number of parameters at a *reasonable* level to keep the training time around one day.

Table 8.1: List of the hyperparameters used to train the CNN-skip-GMH composed of 10,212 parameters.

Hyperparameters	**Values**
Batch size	512
Learning rate	10^{-3}
Weight regularization	10^{-5}
Number of epochs	10^{3}
Training size	$\simeq 3.7 \cdot 10^{6}$
Validation size	$\simeq 3.7 \cdot 10^{5}$
Optimizer	`torch.optim.Adam()`
Nb. Gaussian/head	2
Activation fct.	Sigmoid

These five hyperparameters are not the only ones: one may cite the learning rate, the weight regularization, the number of Gaussians K, the activation functions, and the batch size. As already mentioned in Section 7.1, the optimal batch size is 32, as stated by LeCun. A smaller batch size helps to better investigate the loss landscape and avoid getting stuck in local minima. However, the batch size is increased to 512, and to compensate for this higher batch size, a higher learning rate is used. The weight regularization is kept at 10^{-5}, as for the training of MLP-GMH. Two activation functions are evaluated: sigmoid and ReLU. Even if the Sigmoid is prone to the vanishing gradient problem, this activation function gives better predictions of our wall shear stress components. The number of Gaussians K was modified from 1 to 5. No significant improvement in the predictions is noticed by increasing the number of Gaussian. Moreover, the non-dimensionalization proposed in Section 5.2.1 reduces the skewness of the target distributions. Therefore, these distributions are more Gaussian-like, and fewer Gaussians are required in the Mixture. Finally, two Gaussians appear to be a good trade-off between the prediction accuracy and the number of parameters to fit.

8.2 *A priori* testing

The trained model is tested *a priori* on the periodic hill upper and lower walls. Note that the network is trained on these test cases, but the data used to evaluate the performance of the network are taken at different time steps. The model is also assessed on the turbulent channel flow at $Re_\tau = 950$ to see if a *more advanced* model can still predict the dynamic of a zero-pressure gradient attached turbulent flow. In Chapter 7, the model MLP-GMH was assessed on synthetic data. However, due to the increase in the stencil size and the addition of the pressure gradient, the extension of these synthetic data in this case is no longer very representative and is unlikely to produce reliable results.

8.2.1 *A priori* test on PHL10595

Although the lower wall has the most complex physics due to the separation from a curved wall and the reattachment of the free shear on the flat lower surface, the neural network predictions are in good agreement with the wrLES wall shear stress. Increasing the size of the input really helps the network to discriminate among the various flow physics. This behavior was also mentioned by Dupuy *et al.* (2023a,b). Moreover, the addition of the curvature further assists the network in this discrimination task.

Figure 8.2 shows the mean and the standard deviation of the predictions $\hat{\tau}_{w,\xi}$ obtained from fields extracted at $y/h = 0.115$ and compared to the ground truth $\tau_{w,\xi}$. The mean wrLES (black dashed line) and the predicted $\tau_{w,\xi}$ (blue circle line) has a Pearson correlation coefficient of 0.997. The high variance in the separation vicinity highlights how the separation point moves instantaneously over a large portion of the hill crest. The variance is mostly independent of the position on the flat surface ($2.0 < x/h < 7.0$). It reduces drastically as the flow accelerates on the windward of the next hill ($x/h \simeq 8.0$). This graph indicates that the friction from $x/h \simeq 3$ to $x/h \simeq 5$ has a very small slope. Therefore, even a small mistake in the prediction of the wall shear stress can cause a significant error in the reattachment point's position. Bear in mind that the reattachment point is highly sensitive to mispredictions, and any errors will impact the entire physics on the lower wall.

The distribution of the instantaneous scaled streamwise velocity u^+ as a function of the scaled distance to the wall y^+ was already shown in Section 4.3.4 for compari-

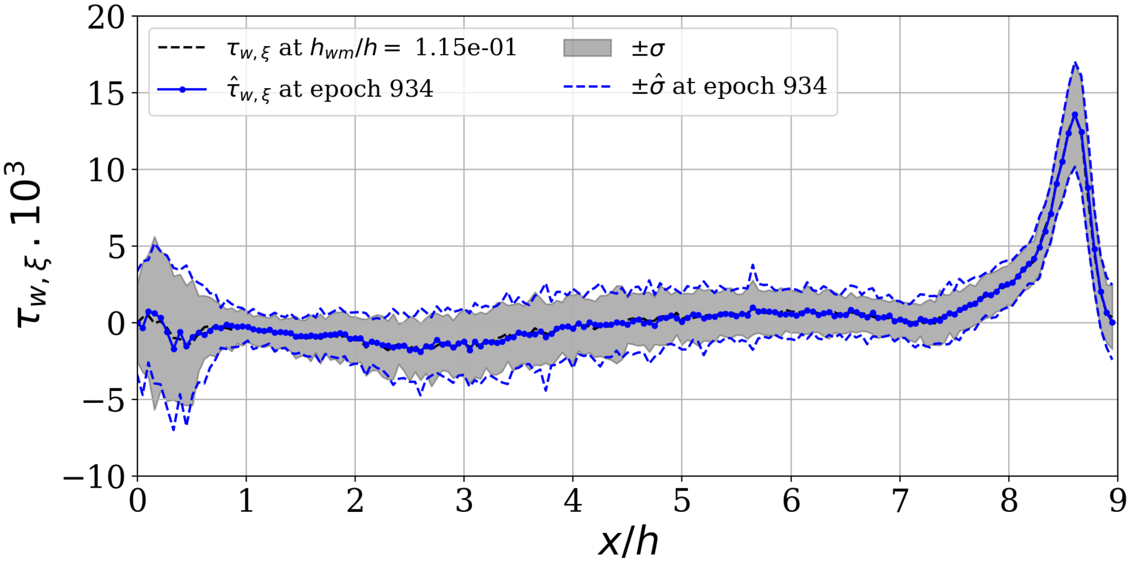


Figure 8.2: *A priori* test on PHL10595: the black dashed line is the wrLES mean wall shear stress, while the gray area represents the wrLES standard deviation; the CNN-skip-GMH predictions are drawn in blue: the blue circle line is the mean while the blue dashed line is the standard deviation.

son with the prediction of a WSS model based on Reichardt's velocity profile. As anticipated, the quasi-analytical WSS model was not able to accurately predict the separation, and the predictions had significantly less variance than the wrLES wall shear stress. However, the CNN-skip-GMH is able to overcome this limitation by correctly predicting the distribution of the wall shear stress at each $(\xi, z)/h$-position on the surface. Figure 8.4a confirms this observation, as the model predictions show the same variability as the wrLES data.

8.2.2 *A priori* test on PHU10595

At first sight, predicting the wall shear stress on the upper wall of the two-dimensional periodic hill is of the same complexity as predicting the wall shear stress on the channel wall. However, this is not the case. Although a streamwise correlation was found between $\tau_{w,\xi}$ and the velocity, the mean streamwise velocity is anti-correlated with the pressure gradient: where the mean pressure gradient has a peak, the friction presents a valley at the exact same location. As the derivative of the pressure gradient presents oscillations, the network has a hard time detecting such a correlation. Therefore, a window average of the pressure gradient is also used as an entry to help the network to detect the existing correlation between the wall shear stress and the pressure gradient, otherwise the network will only base its prediction on the velocity field as an AWSSR will do, leading to an incorrect wall shear stress.

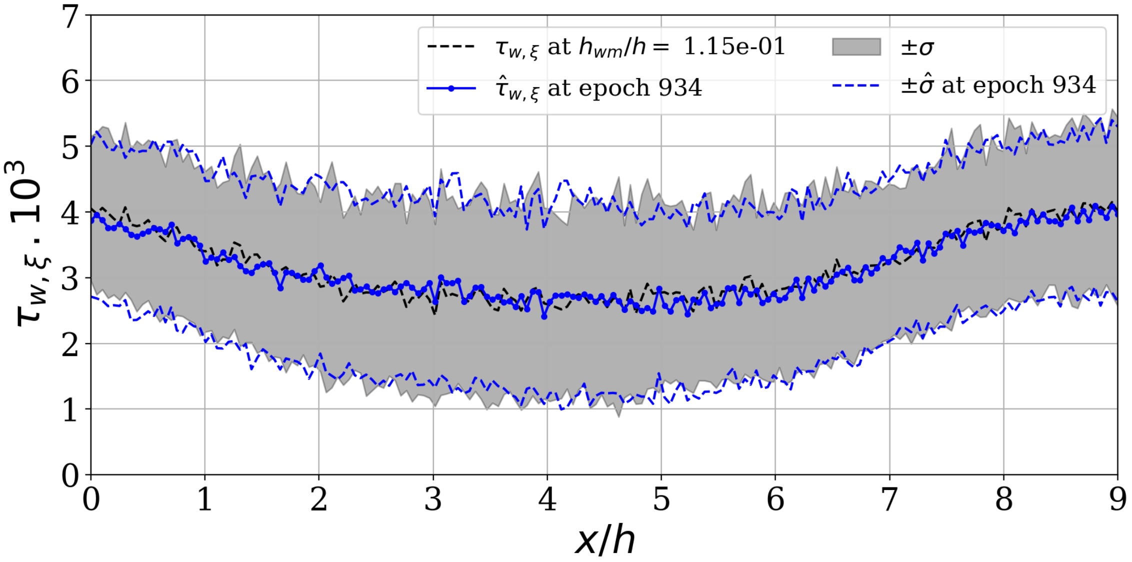
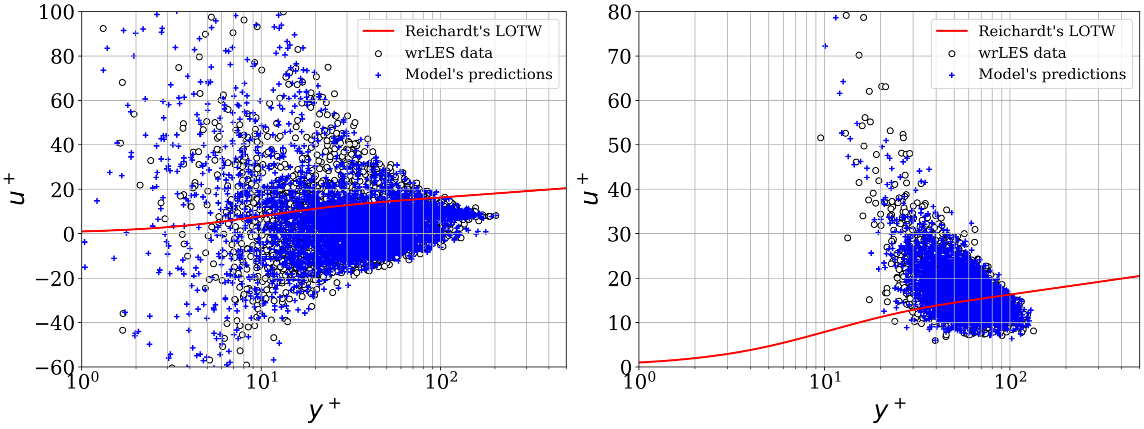


Figure 8.3: *A priori* test on PHU10595: the black dashed line is the wrLES mean wall shear stress, while the gray area represents the wrLES standard deviation; the CNN-skip-GMH predictions are drawn in blue: the blue circle line is the mean while the blue dashed line is the standard deviation.

Figure 8.3 shows the predicted streamwise wall shear stress on the upper wall. The mean prediction (blue circle line) follows the wrLES (black dashed line) well. The Pearson correlation coefficient between the mean prediction and the ground truth is 0.954. The standard deviation is also well predicted by the network. Unlike the lower wall, the variance is independent of position.

In Figure 8.4b, the predictions are compared to the ground truth on a (u^+, y^+) graph. The model predictions and wrLES data overlap, but the overall shape is different from that observed in a channel. In turbulent channel flow, the effects of the Reynolds shear stress are less significant at lower y^+ values, resulting in a smaller variance and a more condensed shape around Reichardt's profile. However, for similar y^+ values on the upper wall, the pressure gradient affects the overall shape by elongating it vertically.



(a) On the lower wall

(b) On the upper wall

Figure 8.4: A priori test on two-dimensional periodic hill: Distribution of the instantaneous scaled streamwise velocity u^+ as a function of the scaled distance to the wall y^+, using the target wall shear stress (black circle) and the predictions (blue plus symbol) to evaluate the wall unit scaling; Reichardt's profile is drawn in red.

8.2.3 *A priori* test on TC950

The model was trained on the test case TC1000 extracted from the JHTDB website and is now evaluated on TC950 obtained with Argo-DG at three heights: 100, 150, and 200 in wall units. The channel is not a separated flow, but it is important to evaluate the capability of the network to discriminate between different flow physics: separated, moderate pressure gradient, and attached flows. Figure 8.5 shows the predicted distributions of the two components of the wall shear stress. These predicted distributions (blue circle line) are compared with the wrLES distributions (black histogram). Examining the distributions, the model performs well for both the streamwise and spanwise wall shear stress. The mean and standard deviation appear to be properly predicted.

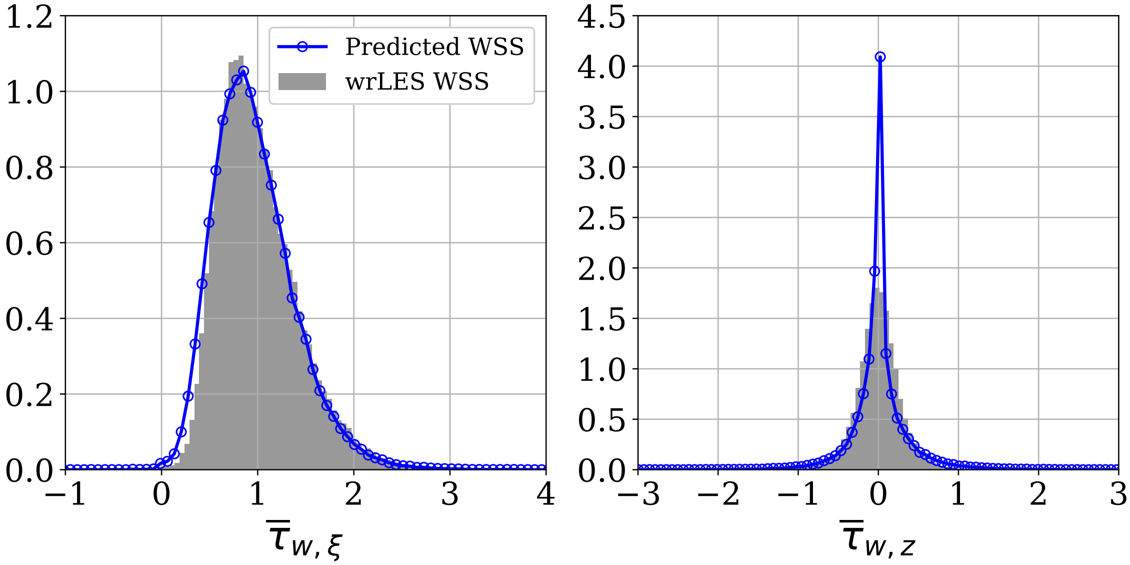


Figure 8.5: wrLES and predicted distributions of $\tau_{w,\xi}$ (left) and $\tau_{w,z}$ (right) on the turbulent channel flow at $Re_\tau = 950$; the predictions are obtained from the model CNN-skip-GMH.

Examining the first three statistical moments in Table 8.2, the predicted streamwise wall shear stress is in good agreement with the wrLES one, except for the mean where a relative error of 3% is detected. Even the skewness is recovered although the model is not explicitly trained to match this third moment of the distribution. The network seems to have more difficulty in predicting the wall shear stress in the spanwise direction. As observed in Figure 8.2, the standard deviation and skewness are slightly overestimated.

Table 8.2: First three statistical moments of the wrLES and predicted wall shear stress (denoted with a hat symbol).

Statistics	$\boldsymbol{\tau}_{w,\xi}$	$\hat{\boldsymbol{\tau}}_{w,\xi}$	$\boldsymbol{\tau}_{w,z}$	$\hat{\boldsymbol{\tau}}_{w,z}$
$\boldsymbol{\mu}$	1.008	0.965	-0.002	0.013
$\boldsymbol{\sigma}$	0.424	0.423	0.284	0.352
$\boldsymbol{\mathcal{S}}$	1.008	0.936	-0.067	0.285

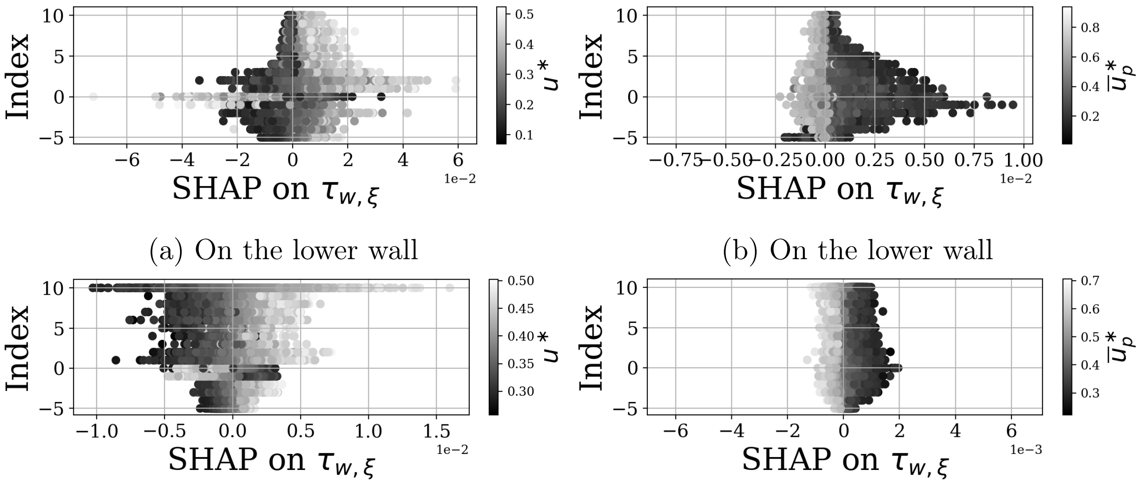
8.2.4 SHAP values

SHAP (SHapley Additive exPlanations), introduced in Section 2.4, are evaluated for several entries of the data-driven wall model. To track the average marginal contribution of a feature to the outcome, the outcome is not sampled from the predicted distribution, but is evaluated as the linear combination of the mean as $\sum_k \pi_k \mu_k$.

Figure 8.6 shows the SHAP values for two features (i.e., inputs) of the CNN-skip-GMH on the periodic hill upper and lower wall. The graph combines feature location (vertical axis) with feature importance (color map) and feature effect (horizontal axis). The vertical axis corresponds to the streamwise coordinates, defined as an index in the figure. A positive index indicates downstream position $\xi/h > 0$ while a negative index indicates upstream position $\xi/h < 0$. The horizontal axis corresponds to the SHAP values. The color map corresponds to the value of the feature from low (in black) to high (in white).

The figures 8.6a and 8.6c are images of the effect of u^* on $\tau_{w,\xi}$. In both cases the velocity is the feature with the most important effect on the wall shear stress. For the lower wall, the indices $-2, -1, 0, 1, 2, 3$ are those that most affect the wall shear stress, with a visible anti-correlation at index 0. For downstream positions, a higher velocity affects the wall shear stress, while a lower velocity seems to have no effect (i.e. the SHAP value is mostly zero). On the upper wall, downstream points affect the wall shear stress more than upstream points. This observation is consistent with the analysis of the space-time correlation in Chapter 4. The positive correlation between velocity and wall shear stress is clear, except at index -1 and -0 where an anticorrelation is found. The enlargement of the input stencil is justified on these graphs.

Figures 8.6b and 8.6d show the SHAP values obtained for the mean pressure gradients $\overline{(\nabla p)}$ where ∇p is independent of the forcing term. On the lower wall, the pressure gradient is highly position dependent. Downstream points indicate an anti-correlation with the pressure gradients, while upstream points indicate a positive correlation. Indices around 0 are those that contribute the most information to the network. The observation of anti-correlation is consistent with the space-time correlation found in chapter 4. On the upper wall the SHAP values are less affected by the positions. An anticorrelation is observed at every position. Although the SHAP values are about two times smaller than those for velocity, the addition of the mean pressure gradient helps the network to improve the prediction of the wall shear stress on the upper wall.



(a) On the lower wall

(b) On the lower wall

(c) On the upper wall

(d) On the upper wall

Figure 8.6: SHAP summary plot on the two-dimensional periodic walls; on the left, the SHAP values related to the streamwise velocity and on the right, those related to the mean pressure gradients.

The SHAP values are a tool for interpreting the network outputs, and this interpretation partially reflects the analysis of space-time correlations carried out in chapter 4.

8.3 *A posteriori* testing

The network has been *a priori* evaluated in Section 8.2, and the model predictions were in good agreement with the wrLES wall shear stress. Therefore, the model is *a posteriori* assessed on the two-dimensional periodic hill at the same bulk Reynolds number (Re_b = 10,595). At first sight, this task is *simple* because the model has been trained on this specific test case. However, the production environment is different from the training environment because the wall model will interact with the resolved volume data. Moreover, the reattachment location is very sensitive to small errors, as observed in Figure 8.2. Accurate resolution of the free shear layer is of high importance. The non-equilibrium boundary layer within the recirculation bubble necessitates the accurate resolution of all terms in the Navier-Stokes equations to yield correct results. For these reasons, the wmLES mesh is very similar to the wrLES one, except that the cells within $0 \leq \eta/h \leq 0.1$ are removed on both the upper and lower walls. The mesh is shown in Figure 8.7. The lower half of the mesh is refined to accurately capture the free shear layer. The objective is to accurately resolve the fine structures resulting from the separation and the free shear layer which is a region of high gradients.

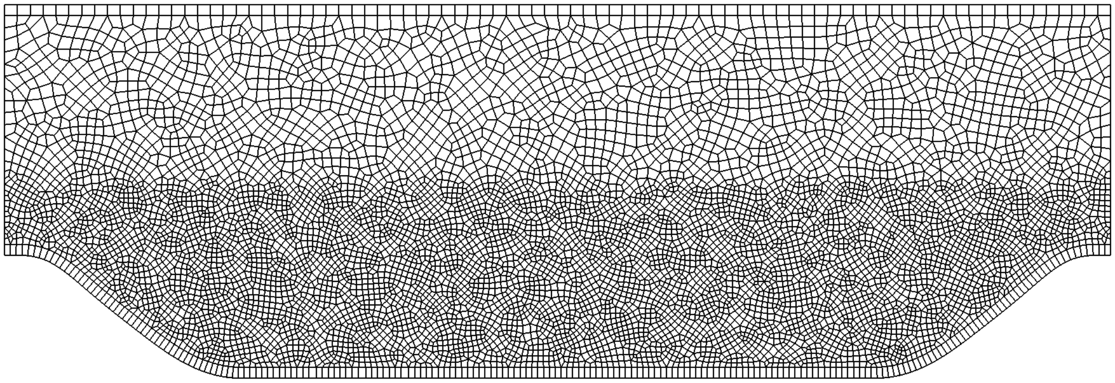


Figure 8.7: wmLES mesh for the two-dimensional periodic hill at $Re_b = 10{,}595$ with the first cell height of size $\Delta\eta/h = 0.1$

Several experiments have been conducted and are summarized in Table 8.3. The parameters changed from one experiment to another are:

- WSS model

  The data-driven wall model is compared to the analytical WSS model based on Reichardt's velocity profile (AWSSR).

- the alignment of the predicted wall shear stress to the velocity

  The AWSSR predicts the magnitude of the wall shear stress and aligns it with the velocity extracted at h_{wm}. In our approach, the neural networt predicts the wall-parallel components of the wall shear stress (i.e., $\tau_{w,\xi}$

and $\tau_{w,z}$). Therefore, there is no explicit alignment of the predicted wall shear stress. For the test `DD-A-512-p4`, $\tau_{w,\xi}$ and $\tau_{w,z}$ are sampled from their respective distributions, the magnitude is evaluated, and then the wall shear stress is aligned with the velocity projected in the local frame of reference made by the solid wall.

- the polynomial order p

  According to Frère (2018), the WSS model in Argo-DG gives better results at an even order polynomial degree. Because the model was trained on data aquired at $p = 3$, the two polynomial orders are tested. Therefore, the data-driven wmLES is run at $p = 4$, not on the mesh presented in Figure 8.7, but on a slightly coarser mesh allowing an equivalent number of degrees of freedom (see Table 8.3).

- the wall model height (h_{wm}).

As the model is not trained to capture the numerical transient, the simulation is restarted from a coarse wrLES of the same test case. This point needs to be improved and will be discussed in the perspectives (Chapter 9). The simulation `Rcht-A-512-p4` is also restarted from the same coarse wrLES for a fair comparison. For each wmLES, the time step is fixed at $\mathrm{d}t(u_b/h) = 5 \times 10^{-3}$. The Mach number is fixed at the low value of 0.1, and a pressure gradient is imposed to ensure the bulk Reynolds number as already mentioned in Section 4.3.

Table 8.3: Summary of the numerical experiments conducted on the two-dimensional periodic hill at Re_b = 10,595, where **NA**, and **A** stands for **N**o **A**lignment, and **A**lignment; t_c corresponds to the number of flow through time.

	WSS model	**Align.**	$\boldsymbol{p}$	**DOF**	$\boldsymbol{h_{wm}/h}$	**Accum.**
`DD-NA-512-p3`	CNN-skip-GMH	✗	3	25,473,600	0.1	$\sim 36\,t_c$
`DD-NA-512-p4`	CNN-skip-GMH	✗	4	28,788,750	0.08	$\sim 25\,t_c$
`DD-A-512-p4`	CNN-skip-GMH	✓	4	28,788,750	0.1	$\sim 19\,t_c$
`Rcht-A-512-p4`	AWSSR	✓	4	28,788,750	0.1	$\sim 15\,t_c$

Mean friction coefficient The four wmLES are compared to one reference: the wrLES used to train the data-driven wall model. The accuracy of the wrLES was assessed in Section 4.3. Figure 8.9 shows the wall shear stress obtained on the lower wall for the different wmLES. The data-driven wmLES without alignment (i.e., `DD-NA-512-p3` and `DD-NA-512-p4`) correctly predict the mean separation location (see Table 8.4). The friction is overestimated in the separation vicinity, as also observed in the work of Krank *et al.* (2019) and Zhou *et al.* (2022). The tiny separation before the windward foot of the next hill is well captured, as is the friction peak. These two simulations show a good agreement with the reference, except at the reattachment. The two wmLES that align the wall shear stress to the velocity (i.e., `DD-A-512-p4` and `Rcht-A-512-p4`) miss the mean separation line. Indeed, in the separation vicinity, the wall shear stress does not align with the velocity.

Figure 8.8 illustrates this *misalignment* phenomenon. On the left, the mean velocity profile at the top of the hill is plotted, while on the right, the mean wall shear stress predicted by the AWSSR model is compared to the same references as in Figure 4.11. At the matching location, the velocity is highly positive, and the AWSSR model predicts a positive wall shear stress while the flow is already separated near the wall. The AWSSR model only recognizes flow separation when the velocity field at the corresponding location becomes negative downstream. Therefore, the separation is delayed and appears at $x/h = 0.5$. `DD-A-512-p4` correctly predicts the friction peak while `Rcht-A-512-p4` completely misses the strong acceleration on the hill and underpredicts the peak.

None of the four wmLES matches the mean reattachment location. Such a reattachment issue was also observed by Dupuy *et al.* (2023a,b) on the backward-facing step. All wmLES predict a faster reattachment (see Table 8.4), probably due to energy loss in the free shear layer. This faster reattachment allows the flow to recover over a larger part of the flat bottom surface, creating higher friction between $x/h = 4$ and $x/h = 7$, at least for `DD-NA-512-p3` and `DD-NA-512-p4`. This overestimation of the friction on the flat bottom surface is reduced for `DD-A-512-p4`, which has the closest reattachment point compared to the reference. According to the best practices section of URF 3-30[1], the flow physics is very sensitive to the grid resolution and quality, as well as the numerical scheme, which should not be too dissipative to ensure appropriate resolution of the scales and not to damp them.

[1] `https://kbwiki.ercoftac.org/w/index.php/UFR_3-30_Best_Practice_Advice`

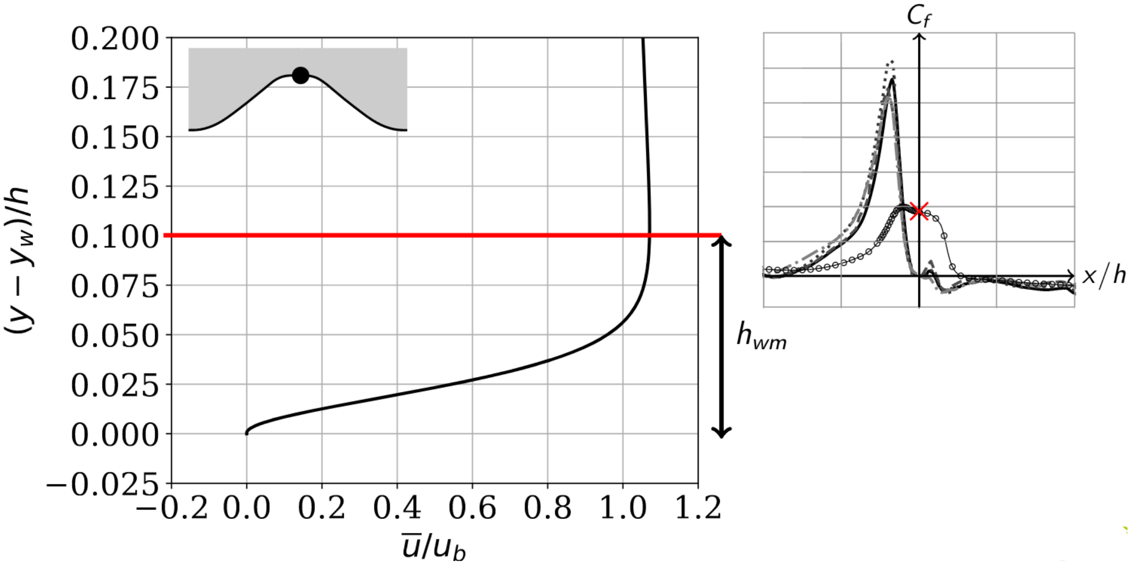


Figure 8.8: Illustration of the mean velocity profile at $x/h \simeq 0$ in the two-dimensional periodic hill geometry with its corresponding predicted wall shear stress using the WSS model based on Reichardt's velocity profile.

Table 8.4: Mean position of the separation and reattachment lines

	DD-NA-512-p3	DD-NA-512-p4	DD-A-512-p4	Rcht-A-512-p4	Expe.
x_{sep}/h	0.217	0.216	0.573	0.538	0.19
x_{reatt}/h	3.690	3.589	3.935	3.652	4.21

In Figure 8.10, the friction on the upper wall of the periodic hill is shown. It can be observed that the friction on this wall is more correlated with the pressure gradient than with the velocity, which explains why the friction imposed by `Rcht-A-512-p4` does not match the reference. On the other hand, the predicted friction of `DD-NA-512-p3` and `DD-NA-512-p4` agrees well with the reference. However, `DD-A-512-p4` overestimates the friction on the upper wall. The curve is shifted by a constant value of approximately 10^{-3}. It is noteworthy that the mean friction value from `DD-A-512-p4` matches the mean friction value predicted by `Rcht-A-512-p4`.

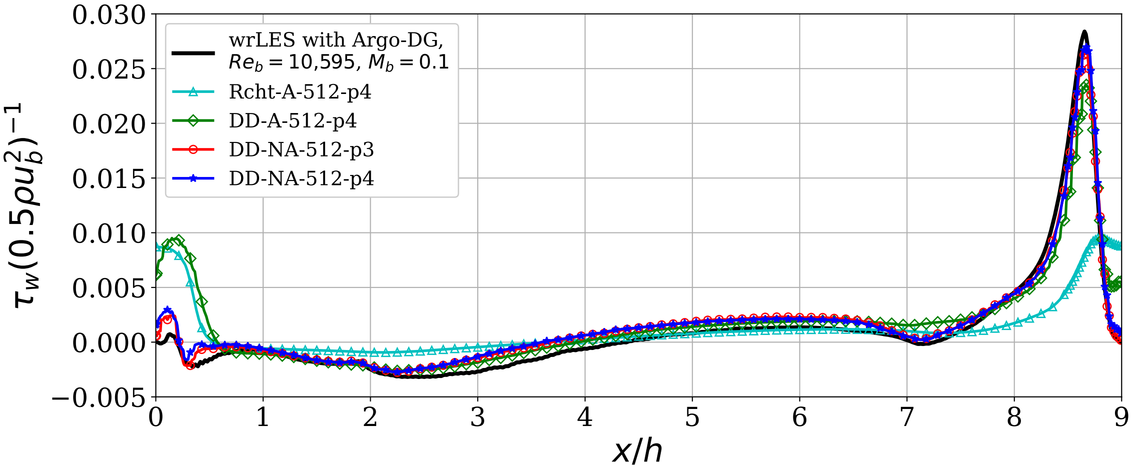


Figure 8.9: Mean streamwise wall shear stress measured on the lower solid wall of the periodic hill at $Re_b = 10{,}595$.

This overestimation is likely due to the alignment of the wall shear stress.

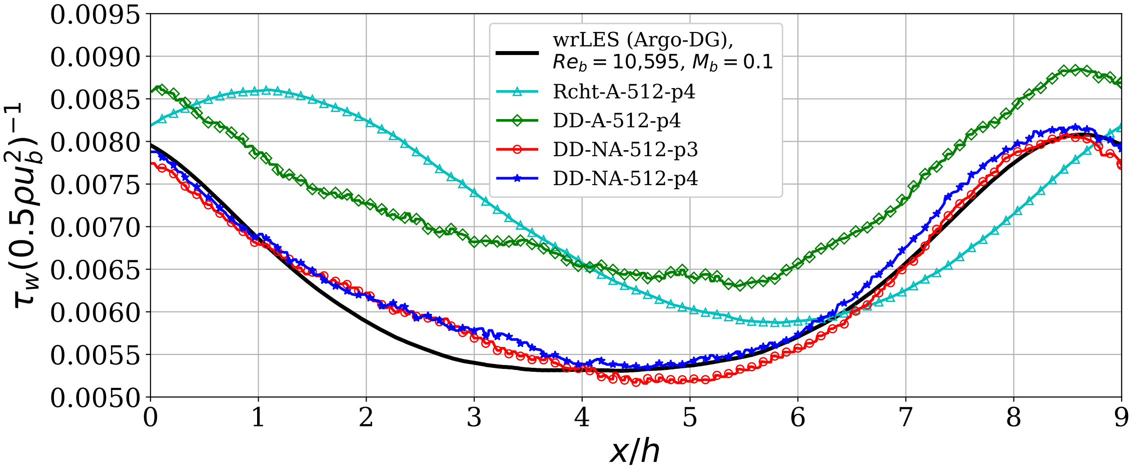
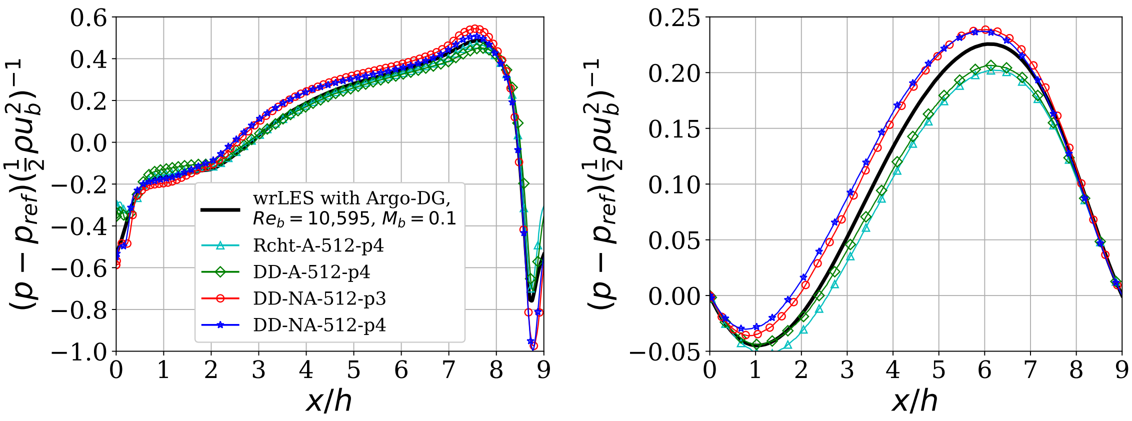


Figure 8.10: Mean streamwise wall shear stress measured on the lower solid wall of the periodic hill at $Re_b = 10{,}595$, compared to our wrLES of the same test case.

Mean pressure coefficient The mean pressure coefficient C_p on the upper and lower walls is shown in Figure 8.11. This coefficient is defined as $C_p = (p_w -$

$p_{ref})/(\rho u_b^2/2)$, where p_w is the mean pressure on the wall and p_{ref} is the reference pressure extracted at $x/h = 0$ on the top wall. The wmLES captures the qualitative trend of the mean C_p on the lower wall, including the adverse pressure gradient (APG) and favorable pressure gradient (FPG) regimes. A deviation is visible at the top of the hill, where a sudden change in pressure from strong FPG to strong APG is observed, corresponding to the separation emergence. The models that align the wall shear stress are in good agreement with the reference except in the separation vicinity $0 \leq x/h \leq 2$, while the models that do not align observe a deviation in the C_p curve on the flat bottom surface $3 \leq x/h \leq 6$. Although the friction peak is correctly captured by `DD-NA-512-p3` and `DD-NA-512-p4`, these two wmLES overestimate the APG on the top of the hill. On the upper wall (Figure 8.11b), although none of the wmLES exactly matches the C_p curve, the qualitative shape is recovered. The maximum relative error measured is 5%.



(a) On the lower solid wall

(b) On the upper solid wall

Figure 8.11: Pressure coefficient on both walls of the two-dimensional periodic hill at Re_b = 10,595 evaluated with the wmLES and compared with the wrLES using Argo-DG.

Mean velocity profiles Figure 8.12 shows the mean velocity profiles at ten stations (x/h = 0.05, 0.5, 1, 2, 3, 4, 5, 6, 7, 8). Overall, good agreement with the reference is observed. `DD-NA-512-p3` and `DD-NA-512-p4` accurately predict the mean velocity profile at the separation, even in the first cell (the grey area in Figure 8.12), compared to `DD-A-512-p4` and `Rcht-A-512-p4` which underestimate the strong acceleration at the top of the hill. There is a discrepancy due to the early reattachment of the free shear layer for DD-NA-512-p3 and DD-NA-512-p4. The faster reattachment allows the flow to recover over a larger part of the flat bottom surface, as shown by the slight overestimation of the velocity for $3 \leq x/h \leq 7$. Conversely, DD-A-512-p4 and Rcht-A-512-p4 better capture the velocity profile near the reattachment but overestimate $\overline{u}$ on the upper wall, which is consistent with the overestimation of the friction coefficient in Figure 8.10.

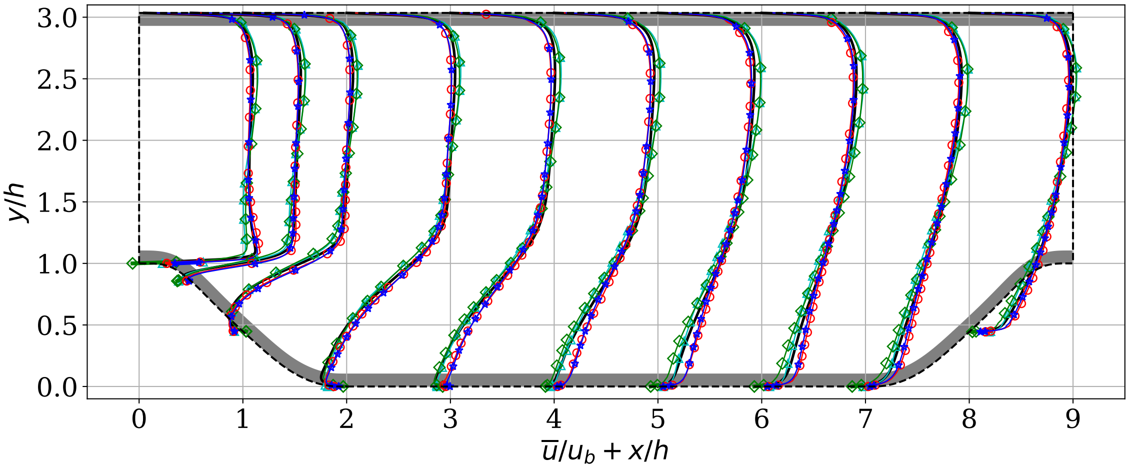
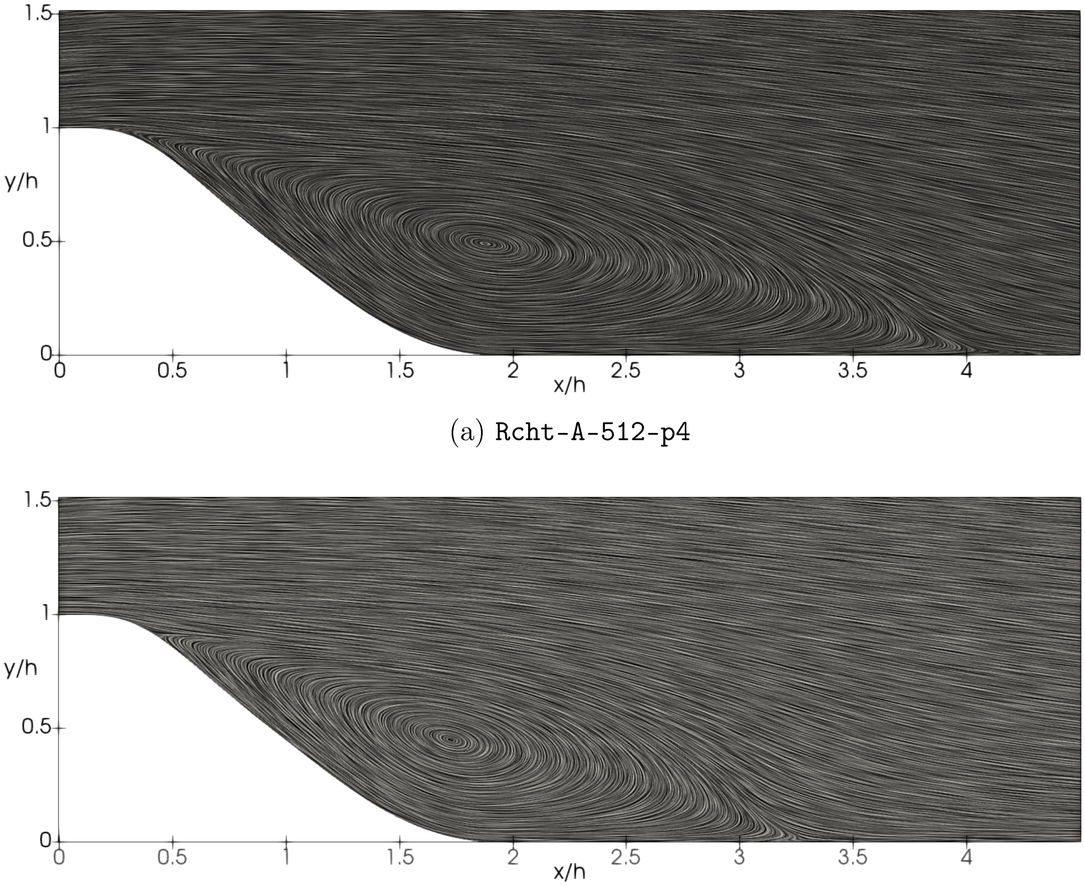


Figure 8.12: Mean velocity profiles (see legend of Figure 8.9).

Streamlines Figure 8.13 shows the streamlines for Rcht-A-512-p4 and DD-NA-512-p4. Although DD-NA-512-p4 predicts the separation better, this is not reflected in the starting point (i.e., the averaged location of the separation $\tau_w = 0$) of the recirculation bubble compared to Rcht-A-512-p4, for which the bubble rises much higher up the hill. The size of the recirculation bubble, not only its length but also its height, affects the channel obstruction (i.e., the bubble somehow locally modifies the geometry), and, therefore, the imposed forcing term. The source term must be modified to ensure the same mass flow at each time step despite the change in the separation and reattachment lines. The recirculation bubble predicted by Rcht-A-512-p4 is very similar to the DNS bubble, and, therefore, the source term is unchanged. However, due to the reduction in height of the recirculation bubble predicted by DD-NA-512-p4, the flow is subjected to less constriction, and, therefore, the source term is reduced compared to the DNS one.



(a) `Rcht-A-512-p4`

(b) `DD-NA-512-p4`

Figure 8.13: Streamlines obtained from two wmLES on the two-dimensional periodic hill at $Re_b = 10{,}595$.

Mean Reynolds stresses The most important discrepancy in the Reynolds stress $\overline{u'u'}$ profiles (Figure 8.14) is inside the recirculation bubble ($1 \leq x/h \leq 4$). `DD-NA-512-p3` and `DD-NA-512-p4` overestimate $\overline{u'u'}$ while `DD-A-512-p4` underestimates this component of the Reynold stress. Although `Rcht-A-512-p4` mispredicts the friction on both the lower and upper walls, it matches very well $\overline{u'u'}$ throughout the domain except in the near wall region on the lower wall where it tends to overpredict this component. Similar observations are made on $\overline{v'v'}$ (see Figure 8.15), except that the deviation is smaller. Regarding the covariance $\overline{u'v'}$ in Figure 8.16, all simulations show a good agreement with the reference, except inside the recirculation bubble where a slight deviation is observed.

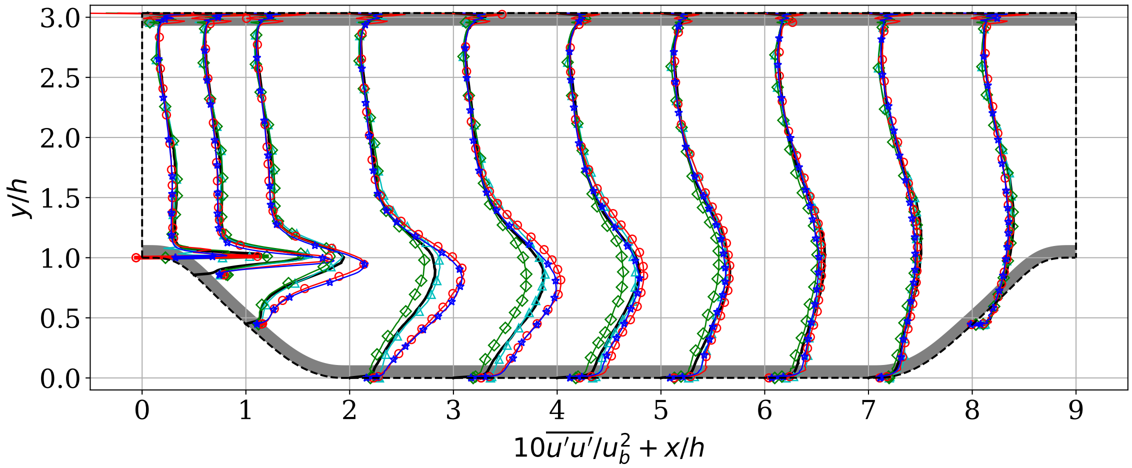


Figure 8.14: Reynolds stress $\overline{u'u'}$ profiles (see legend of Figure 8.9).

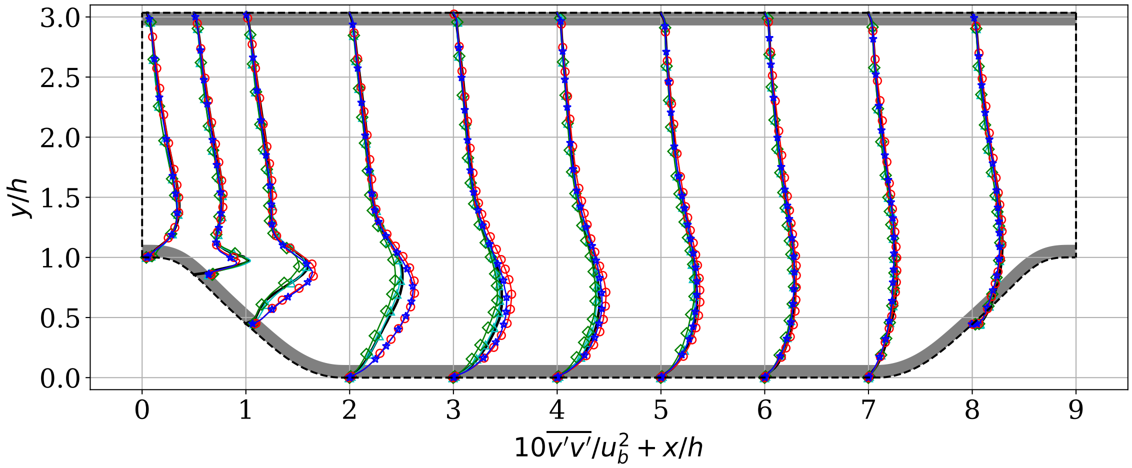


Figure 8.15: Reynolds stress $\overline{v'v'}$ profiles (see legend of Figure 8.9).

Summary It is difficult to determine the source of the misprediction of the reattachment location in this particular configuration. It is also unclear why `Rcht-A-512-p4`

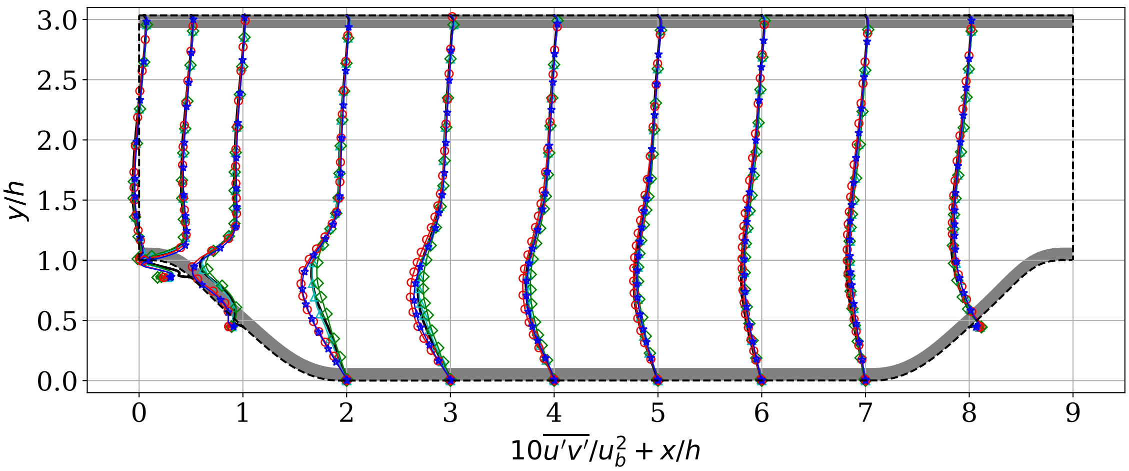


Figure 8.16: Covariance between u and v (see legend of Figure 8.9).

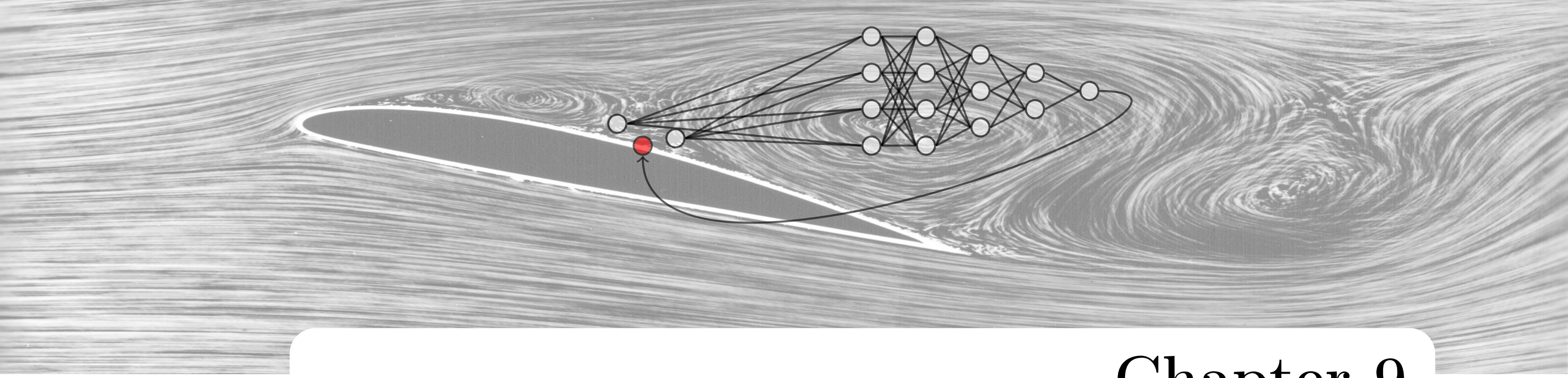
produces better results for the mean velocity profiles and Reynolds stresses, although this analytical WSS model mispredicts the wall shear stress. In addition, its inherent modeling assumptions do not hold for separated flows. There may be compensation errors between different terms that allow for such results, but we have not quantified them. Nevertheless, the main difference between (`DD-NA-512-p3`, `DD-NA-512-p4`) and (`DD-A-512-p4`, `Rcht-A-512-p4`) is the imposition of the alignment with the velocity. Many standard WSS models in the literature naturally align the wall shear stress with the velocity extracted at the matching location. At the separation, there is a strong misalignment between τ_w and u as already mentioned and illustrated in Figure 8.8. We hypothesize that the data-driven WSS model is too dissipative at the separation due to this misalignment. Indeed, the velocity in the volume is highly positive due to the strong acceleration on the windward foot of the hill. The data-driven WSS model forces the wall shear stress to be mostly opposite to the velocity. The flow is perturbed by the rapid change in direction, leading to high fluctuations in the turbulent kinetic energy and a reduction in the total kinetic energy. The model creates an energy sink in the separation vicinity. Moreover, the sampling from the predicted distribution without accounting for space-time correlations creates small structures at the wall that are quickly dissipated creating another energy sink. The lack of energy forces the free shear layer to reattach to the flat bottom surface early. Therefore, the amplitude of the wall shear stress is not the only important element in a wall model and misalignment issues should be treated differently.

8.4 Conclusion

A convolutional neural network was trained on three databases: the lower and upper walls of the two-dimensional periodic hill at $Re_b = 10{,}595$, and the channel flow at $Re_\tau = 950$ to predict the distribution of the wall-parallel components of the wall shear stress based on the theory of Gaussian Mixture Network. *A priori*, the model accurately predicts the mean and the variance of $\tau_{w,\xi}$ and $\tau_{w,z}$ on both the upper and lower walls. On the lower wall, the curvature input has a great effect on the prediction of the peak of friction. On the upper wall, the introduction of the mean pressure gradient as a model input helps to recover the proper friction.

A posteriori, four wmLES were run on the same configuration as the training configuration: the two-dimensional periodic hill at $Re_b = 10{,}595$. As already mentioned, the production environment is different from the training one. In the production environment, the model will actually make predictions, which in turn will interact with the volume fields in a feedback loop. Such a feedback is not part of the training and can affect the accuracy of the model. Different parameters are tested in the four wmLES, such as the polynomial order, the wall model height, the WSS model, and the alignment of the predicted wall shear stress. Firstly, the numerical experiments show that the data-driven WSS model is independent of the matching location and that the polynomial order does not affect the network. Secondly, the alignment of the wall shear stress helps improving the prediction of the mean reattachment line by reducing the energy sink introduced via misalignment in the separation vicinity. Nonetheless, the alignment delays the mean separation line.

The new data-driven WSS model improves the prediction of the wall shear stress on both the lower and upper walls compared to the analytical WSS model based on Reichardt's velocity profile. Nonetheless, the new model is too dissipative in the separation vicinity, forcing the free shear layer to reattach earlier. The analytical WSS model, on the other hand, is energy conservative (i.e., there is neither injection nor extraction of energy in the near wall region), which allows to obtain the correct recirculation bubble size, although the prediction of the mean wall shear stress is wrong. This observation leads to the question of whether *a WSS model is the best approach to tackle separated flows?*



Chapter 9
Conclusions and perspectives

The main objective of this PhD thesis was to exploit the approximation capabilities of deep neural networks to establish a more general wall shear stress model to capture the complex relationship between instantaneous flow quantities, geometric parameters, and wall shear stress using DNS or wrLES databases. This new WSS model has overcome the weaknesses of the standard models to be applicable to the problem of separate flows in that no *prior* knowledge about the law-of-the-wall is imposed on the network. This main objective is subdivided into four parts: (i) to define the appropriate test cases to address the improvement of WSS models for the separation phenomenon, (ii) to select the relevant features for the training of the data-driven model, (iii) to identify the most suitable neural network architecture that will embed the appropriate invariant and satisfy to the implementation constraints, and (iv) finally to validate the novel data-driven WSS model on a test case that features separation and reattachment.

9.1 Conclusion

The path to a novel data-driven wall shear stress model started with the definition of suitable test cases that are computationally affordable on modern clusters and easy to set up. For this purpose, the two-dimensional periodic hill was selected for the present study. This test case is used to validate new wall models, as the geometry was carefully designed to provide a massive separation from the hill crest and a reattachment, followed by a recovery of the flow on the flat lower surface. Furthermore, this test case offers two walls with different physics. The lower solid wall experiences, as mentioned, a separation and reattachment, while the upper wall has an attached fully turbulent boundary layer subjected to a moderate pressure gradient. Even though standard wall models can correctly predict the wall shear stress on the upper wall, it is of interest to include such physics in the neural network. A wrLES of this test case was successfully performed with the high-order flow solver Argo-DG, which produced mean velocity and Reynolds stress profiles that were in good agreement with the existing literature, reinforcing confidence in the use of this test case as a reliable labeled database.

Directly feeding the raw databases to the network is not ideal if a universal model is sought. The preprocessing of the databases and their in-depth analysis was a crucial step towards a new data-driven WSS model. The periodic hill flow was deeply analyzed in Chapter 4 using space-time correlations and preprocessed in Chapter 5. The analysis of space-time correlations acted as a feature selection for the neural network and provided insight into the underlying physics. The correlation of the wall shear stress with several flow quantities (i.e., velocity and pressure gradient) is evaluated in the streamwise and spanwise directions. Such an analysis was also carried out on the turbulent channel flow for comparison with a fully turbulent, attached, and equilibrium boundary layer. The correlations observed in the channel and on the upper wall were quite similar, which was not surprising as the physics is very similar. The correlations took the form of an inclined ellipse, aligned with the local mean velocity. The domain of high correlation indicated a time delay, and to compensate for the delay the correlation was also shifted downstream. This observation suggested that there was a strong relation between the local mean velocity, time delay, and spatial displacement at the height at which the correlation is extracted. The inclination of the correlation is supported by other studies indicating that convection of the near-wall structures across the domain promotes the inclination of these structures.

The periodic hill upper wall differs from the channel in that correlations with the pressure gradient are detected, suggesting that the pressure gradient is an important entry to the neural network. The most interesting correlations were observed in the separation vicinity. At this location, the domain of high correlations of the detected correlations was split into two lobes. One lobe was shifted downstream, indicating that the free-shear layer and the WSS at the separation were highly correlated. The second lobe indicated an anti-correlation between the velocity developing on the convex windward wall of the next hill and the WSS at the separation point. In other words, the more the velocity is accelerated on the hill, the more the flow gains in inertia and will separate at the hilltop. As the model must be instantaneous, the input stencil was enlarged in the streamwise direction to encompass the domain of high correlation. The main conclusion of the space-time correlation analysis was to consider the pressure gradient as a potential input to model the wall shear stress of a boundary layer subjected to moderate and high-pressure gradients and to increase the size of the input stencil. The discussion of the input stencil was found in most studies on data-driven wall models but is rarely based on the use of space-time correlation. Only two articles (Dupuy *et al.*, 2023a,b), to the authors' knowledge, discussed the size of the input data, but in relation to the predictions of the neural network. They found that the model was able to better discriminate between the different physics. This observation was consistent with our findings on correlations.

After generating the databases and identifying the need to incorporate spatial information into the neural network, a network that satisfies the design constraints stated in Section 3.5 was selected. The appropriate network for handling one or two-dimensional inputs is the convolutional neural network. However, this network on its own was not sufficient. As mentioned by Zhou *et al.* (2022), predicting the mean wall shear stress is not adequate for separated flows. Traditional neural networks, trained with (for example) the MSE loss to perform regression, predict the conditional average of the ground truth, conditioned on the input. In the reviewed data-driven wall models, the turbulent wall shear stress statistics are often ignored, and most wall shear stress models are trained using the MSE. Consequently, many authors have found that the predicted values have less variance than the actual or filtered DNS values. In our work, we accounted for the statistics of the wall shear stress by predicting the probability distribution instead of a point estimate. For this purpose, the CNN was connected to a Mixture Density Network to predict the WSS distribution as a linear combination of K Gaussians, where the output of the overall model is a specific realization based on the predicted distribution at a given time

t. To the authors' knowledge, this work is the first attempt to create a statistically based wall shear stress model. Such a model was trained on the channel flow and the two-dimensional periodic hill. The model is able to generalize to higher Reynolds numbers thanks to the physics-based non-dimensionalization process if and only if the wall model height stays within the training bounds.

After selecting the neural network and training it on the pre-processed database, the model was implemented in the Argo-DG flow solver and used to perform wmLES on several test cases. The *a posteriori* testing of the model is the most important step to evaluate the performance and robustness of the model in the production environment. Once the model is in production, it will produce values that interact with the simulation environment. The wall model was applied to two turbulent channel flows, one at the low Reynolds number of 950 and one at the high Reynolds number of 2,000. The model outperforms the existing data-driven wall models in predicting accurately the mean velocity profile. However, the Reynolds stress $\overline{u'u'}$ was underestimated in both cases. This observation was already made by Frère (2018) in her PhD thesis. The covariance $\overline{u'v'}$ showed oscillations in the second grid cell, which can be damped by reducing this cell size. The results for the two-dimensional periodic hill are promising. There is a clear improvement over the analytical WSS model based on Reichardt's velocity profile in the prediction of the wall shear stress on the upper and lower walls. However, an underestimation of the recirculation bubble size is observed, which affects the physics in the whole domain. Although the separation point was accurately predicted, the reattachment location is shifted upward compared to the DNS predictions. This observation suggests that the free shear layer is not correctly captured and that the structures that compose it dissipate more quickly. The main hypothesis is that the data-driven WSS model is too dissipative in the separation vicinity, forcing the free shear layer to reattach earlier. This underestimation is also visible in other studies applying data-driven wall models to separated flow (Zhou *et al.*, 2021; Lozano-Durán and Bae, 2021; Dupuy *et al.*, 2023a,b).

9.2 Perspectives

Three major perspectives have been identified: short, medium, and long-term. The short-term perspective involves operational tasks that can be quickly implemented and tested using the tools developed during the thesis. The medium-term perspective focuses on extending the study to other non-equilibrium configurations. Finally, the long-term perspective looks at the future of data-driven wall models and their development within the scientific community.

Short-term perspective The first operational perspective concerns the coupling between the streamwise and spanwise wall shear stress. The neural network architectures used in the present work have two independent heads, one dedicated to predicting the distribution of $\tau_{w,\xi}$ and another for $\tau_{w,z}$. Such a configuration assumes that the two wall-parallel components of the wall shear stress are independent. This assumption is no longer valid for real three-dimensional geometries. A possible solution is to predict the joint distribution using a mixture of multivariate Gaussian distributions. The network will then predict the mean vector $[\mu_\xi, \mu_z]^T$ and the covariance matrix:

$$\begin{bmatrix} \mathbb{V}\mathrm{ar}[\tau_{w,\xi}] & \mathrm{Cov}[\tau_{w,\xi}, \tau_{w,z}] \\ \mathrm{Cov}[\tau_{w,z}, \tau_{w,\xi}] & \mathbb{V}\mathrm{ar}[\tau_{w,z}] \end{bmatrix}$$

Moreover, this covariance matrix is the transformation that projects the two wall-parallel components of the wall shear stress into a space where they are uncorrelated.

The current wall model utilizes volume probes to interpolate the input fields and surface probes to predict the wall shear stress. The second operational perspective is to eliminate these probes to remove intermediate steps and speed up the overall procedure. A possible solution is to encode the input with Graph Neural Networks (GNNs) by using the mesh connectivity directly. In addition to removing the interpolation steps, GNNs are better than traditional convolutional neural networks for working on non-Euclidean manifolds and for dealing with unstructured data. They are more appropriate for encoding stencils on curved walls. Such work has already been done by Dupuy *et al.* (2023b), but the idea is to connect the GNN to a Mixture Density Network.

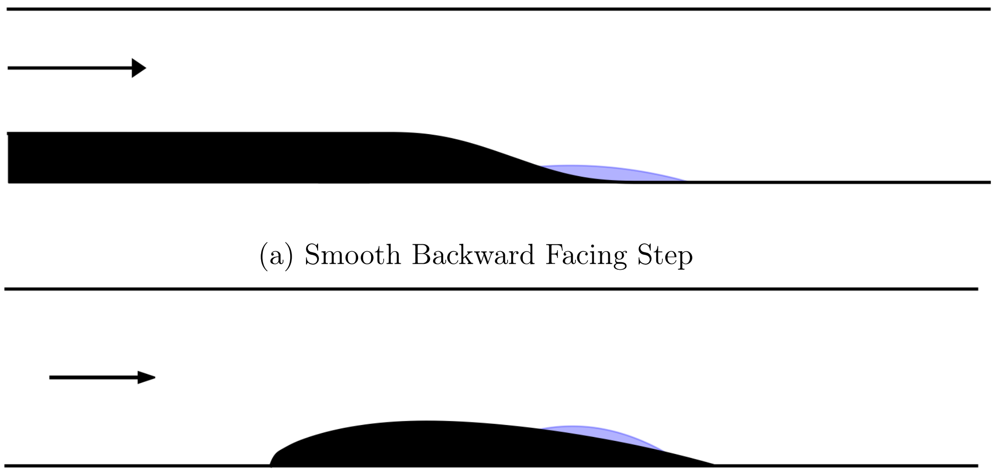
The Mixture Density Network is not the only network that performs density estimation. Normalizing flows, introduced by Rezende and Mohamed (2015), are generative models that provide a mechanism for transforming simple distributions into more complex ones. Therefore, they can produce tractable distributions where both sampling and density evaluation are exact and efficient (Kobyzev *et al.*, 2020). Although the primary use of normalizing flows is to perform density estimation, modifications to the original architecture are necessary to condition the outcome on a set of input features. A migration to conditional Normalizing Flows is attractive, but will require a deeper understanding of their capabilities and applicability to wall modeling.

A third operational perspective is the treatment of the numerical transient. This transient is not physical and is case and mesh-dependent. Therefore, the initial transient cannot be added to the database for training. The first approach to treat this numerical transient is to use the analytical WSS model based on Reichardt's velocity profile. This technique has already been used for the channel flow, but not for the periodic hill. The analytical WSS model has been proved to perform well on the two-dimensional periodic hill when restarting from a coarse wrLES of the same test case. The question is: *does this approach work on more complex test cases?*

Medium-term perspective The first midterm perspective is the implementation of the reconstruction of the space-time correlations as discussed in Chapter 7 in Argo-DG. The implementation could be firstly validated on several turbulent channel flows where the integral time and length scales are known *a priori*. For more complex cases, these scales should be predicted by the neural network directly. The reconstruction of wall shear stress structures from distribution is an innovative approach, as it involves statistics and image reconstruction. According to Balasubramanian *et al.* (2021); Güemes *et al.* (2021); Guastoni *et al.* (2021), the wall shear stress structures are difficult to reconstruct, which makes this perspective even more challenging.

Generating high-fidelity databases is an heavy procedure. It is not just about running test cases, but also setting up appropriate test cases that add valuable information to the database. For these reasons, the present work does not explore many different test cases. Therefore, the database can be extended, as a second midterm perspective, to other non-equilibrium configurations such as the Backward Facing Step, where the flow separates due to a discontinuity in the geometry and moreover, to more complex three-dimensional configurations (with or without skewed boundary layers) and wall-parallel directions that do not necessarily coincide with the streamwise and spanwise directions. In this context, the Smooth Backward Facing Step (SBFS), sketched in Figure 9.1a, is an interesting test case featuring a turbulent separation induced by an adverse pressure gradient. It is in line with the recent EuroHPC submission, where three variations of this configuration are considered: an incipient, a moderate, and a strong separation.

A potential test case, which is less or not documented, is an *half-blade* connected to a flat plate. Figure 9.1 sketches a half NACA 4412 blade with flat plate extension. As for the SBFS, a turbulent boundary layer (TBL) can grow in front of the blade, the boundary layer will ramp up and depending on the set of conditions, the TBL may separate from the rear part of the blade, mainly due to an adverse pressure gradient. The curvature and the pressure gradient could be adjusted to be representative of compressor blades, but with a well-controlled turbulent inflow. This test case can also be used to the study of turbulent separation at higher Reynolds numbers.



(a) Smooth Backward Facing Step

(b) NACA 4412 airfoil with flat plate extending from its trailing and leading edges

Figure 9.1: Potential configurations to study the turbulent separation at higher Reynolds numbers.

In line with the generation of more complex configurations, the application of the new data-driven WSS model to a turbomachinery test case would be of great interest. The low-pressure compressor blade cascade SOLIDITY is targeted because it features a turbulent separation near the trailing edge. The laminar boundary layer developing on the suction side becomes turbulent near the leading edge via a separation-induced transition. The turbulent boundary layer then develops on the suction side until it separates near the trailing edge, depending on the operating conditions. This test case is more advanced because it involves dealing with a laminar boundary layer on the pressure side and a laminar separation bubble on the suction side near the leading edge.

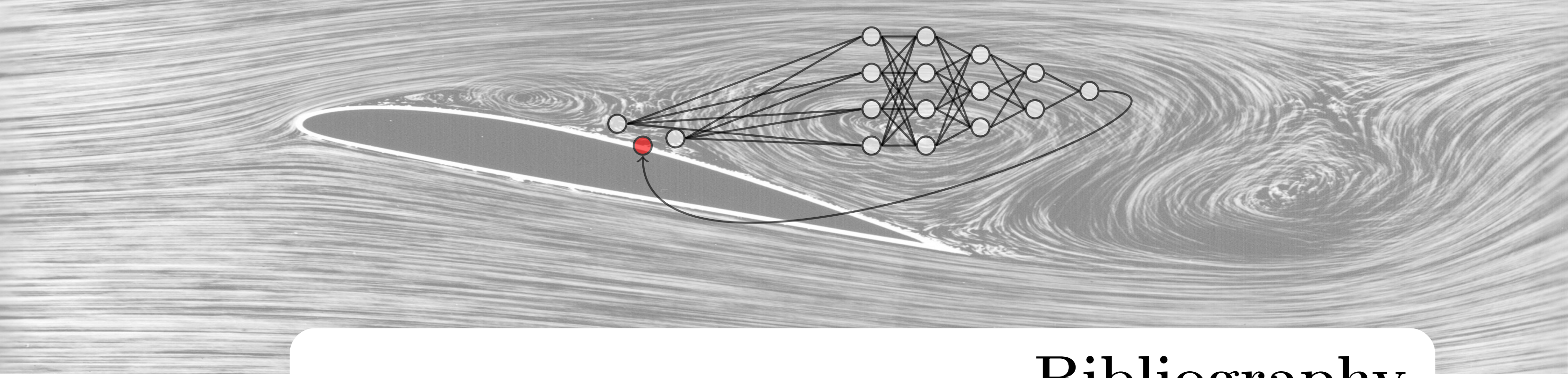
Long-term perspective Over the last decade, we have seen a important emergence of Machine Learning and Deep Learning techniques to model physical phenomena. Specifically, in the field of wmLES, researchers aim to develop a *universal* wall model by using deep neural networks that have been trained on a large and diverse database. However, this requires high-quality, well-labeled databases. Although the computational resources have improved, allowing the DNS of high-Reynolds numbers that were previously unfeasible, generating a huge database is not feasible for a *single researcher group* because the physics depends on many different parameters.

The work of Lozano-Durán and Bae (2021) is a promising approach to simplifying the physics by assuming that a complex flow is nothing more than a non-linear combination of simpler flows, called building-blocks. This approach focuses on well-defined test cases that represent only one type of physics. In contrast, other authors focus on a specific physics, such as the separation phenomenon (Dupuy *et al.*, 2023a; Zhou *et al.*, 2022) or the rotating turbulent channels (Kunz, 2019). However, to develop a universal wall model, the network needs access to a considerable amount of *meaningful* data. Advances in image segmentation, reconstruction, and classification are the result of large labeled databases that are now shared globally. To build universal databases, the scientific community must collaborate to make their databases available. Data sharing has already started to improve RANS turbulence models by using averaged data, which requires less memory. However, the development of new wall models requires access to instantaneous data, which is more challenging to share due to storage limitations.

At some point, it may be necessary to consider a different perspective when attempting to solve the storage issue. One approach exploited to model RANS is to identify regions within a computational domain where one model performs better than another and use ML and DL techniques to create a unified model that can switch between models during a simulation, and adjust predictions at the region interfaces. A similar approach could be considered for building new wall models based on existing ones. This method would require segmentation to detect and classify the different regions within a computational domain (e.g., separation, transition, laminar or turbulent boundary layer, corner flows, and recirculation bubble, to cite a few). The unified model could be constructed using standard wall models and possibly new ones trained on specific configurations that standard models cannot accurately predict. The challenge lies in switching between models at the region interfaces without introducing discontinuities that could lead to instabilities and robustness issues. A balance must be found between data-driven and standard wall models, without disregarding years of hard work in wall modeling. While neural networks are not a universal solution, they can be used judiciously to achieve the desired outcome.

One aspect not yet addressed in this perspective is the impact of feedback loops in the *a posteriori* tests and how to integrate them into the training loop to create more robust data-driven wall models. In the *a posteriori* tests, feedback loops may lead the neural network to make incorrect prediction. Specifically, in the flow solver, the predicted wall shear stress can influence the velocity field and other flow quantities, which in turn can affect the prediction of the wall shear stress at the next time step. This retroaction was not considered during the offline training. A potential approach to incorporate this feedback into the training loop is to train the neural network online while simultaneously backpropagating the gradient through the flow solver. This approach represents a long-term perspective as it includes a significant engineering challenge: evaluating the gradient of the flow solver. Furthermore, this raises questions about generalizability: how can the network be trained on various test cases? Is transfer learning feasible?

Undoubtedly, the introduction of machine learning and deep learning techniques into the field of fluid dynamics opens up a whole new world of network models (e.g., Graph Neural Networks, Normalizing Flows, Generative Adversarial Neural Networks, Variational Auto-encoders, Physics Informed Neural Networks, Neural Operator, to only cite a few). This thesis represents a first step towards integrating data-driven wall models into high-order flow solvers. However, this is just the beginning, and the scientific community is creative enough to develop new models to overcome the current limitations.



Bibliography

Abel, M., Stojkovic, D., and Breuer, M. (2006). Nonlinear stochastic estimation of wall models for LES. *Heat and Fluid Flow*, **27**(2), 267–278.

Aggarwal, C. C. and Yu, S. P. (2005). An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB Journal*, **14**, 211–221.

Alam, J. M. and Fitzpatrick, L. P. J. (2017). Large eddy simulation of urban boundary layer flows using a canopy stress method. *arXiv: Fluid Dynamics*.

Arnold, D., Brezzi, F., Cockburn, B., and Marini, L. (2002). Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, **39**, 1749–1779.

Bae, H. J. and Koumoutsakos, P. (2022). Scientific multi-agent reinforcement learning for wall-models of turbulent flows. *Nature Communications*, **13**(1), 1443.

Bakar, Z., Mohemad, R., Ahmad, A., and Deris, M. (2006). A comparative study for outlier detection techniques in data mining. In *2006 IEEE Conference on Cybernetics and Intelligent Systems*, pages 1–6. IEEE.

Balaras, E., Benocci, C., and Piomelli, U. (1996). Two-layer approximate boundary conditions for large-eddy simulations. *AIAA J.*, **34**(6), 1111–1119.

Balasubramanian, A. G., Guastoni, L., Discetti, S., Schlatter, P., Azizpour, H., and Vinuesa, R. (2021). Predicting rge near-wall region of turbulence through convolutional neural networks. In *Proc. 13th ERCOFTAC Symp. on Engineering Turbulence Modeling and Measurements (ETMM13), Rhodes, Greece, September 15-17, 2021*, ETMM13.

Beck, A. D. (2015). *High Order Discontinuous Galerkin Methods for the Simulation of Multiscale Problems*. Ph.D. thesis, University of Stuttgart, Institute of Aerodynamics and Gas Dynamics.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *PNAS*, **116**(32), 15849–15854.

Benocci, C. and Pinelli, A. (1990). The role of the forcing term in the large-eddy simulation of equilibrium channel flow. *In Engineering Turbulence Modeling and Experiments. Elsevier*, pages 287–296.

Bose, S. T. and Moin, P. (2014). A dynamic slip boundary condition for wall-modeled large-eddy simulation. *Physics of Fluids*, **26**, 015104.

Bose, S. T. and Park, G. I. (2018). Wall-modeled large-eddy simulation for complex turbulent flows. *Annual Review of Fluid Mechanics*, **50**, 535–561.

Boxho, M., Rasquin, M., Toulorge, T., Dergham, G., Winckelmans, G., and Hillewaert, K. (2022). Analysis of space-time correlations to support the development of wall-modeled LES. *Flow, Turbulence and Combustion*, **109**(4), 1081–1109.

Breuer, M., Kniazev, B., and Abel, M. (2007). Development of wall models for LES of separated flows using statistical evaluations. *Computers & Fluids*, **36**(5), 817–837.

Breuer, M., Peller, N., Rapp, C., and Manhart, M. (2009). Flow over periodic hills - numerical and experimental study in a wide range of reynolds numbers. *Computers & Fluids.*, **38**(2), 433–457.

Britos, G. M. and Ojeda, S. M. (2018). Robust estimation for two-dimensional autoregressive processes based on bounded innovation propagation representations.

Brunton, S. L., Noack, B. R., and Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, **52**, 477–508.

Bustos, O., Ojeda, S., and Vallejos, R. (2009). Spatial arma models and its applications to image filtering. *Brazilian Journal of Probability and Statistics*, **23**(2), 141–165.

Cadieux, F., Sadique, J., Yang, X. I., Meneveau, C., and Mittal, R. (2016). Wall-modeled large eddy simulation of laminar and turbulent separation bubble flows. *In 46th AIAA fluid dynamics conference. AIAA*, pages 2016–3189.

Carlton, J. S. (2019). Chapter 7 - theoretical methods—basic concepts. In J. S. Carlton, editor, *Marine Propellers and Propulsion (Fourth Edition)*, pages 141–175. Butterworth-Heinemann.

Carton de Wiart, C. (2014). *Towards a discontinuous Galerkin solver for scaleresolving simulations of moderate Reynolds number flows, and application to industrial cases*. Ph.D. thesis, UCL - Université Catholique de Louvain/iMMC.

Carton de Wiart, C. and Hillewaert, K. (2012). DNS and ILES of transitional flows around a sd7003 airfoil using a high order discontinuous galerkin method (paper iccfd7-2012-3604). *In Seventh International Conference on Computational Fluid Dynamics, Hawai, USA*.

Carton de Wiart, C., Hillewaert, K., Duponcheel, M., and Winckelmans, G. (2013a). Assessment of a discontinuous galerkin method for the simulation of vortical flows at high reynolds number. *Int. J. Numer. Meth. Fluids*, **74**, 469–493.

Carton de Wiart, C., Hillewaert, K., Bricteux, L., and Winckelmans, G. (2013b). LES using a discontinuous galerkin method: isotropic turbulence, channel flow and periodic flow. *In Proceedings of the ERCOFTAC Workshop "Direct and Large-Eddy Simulation 9", Dresden, Germany, April 3-5 2013. ERCOFTAC*.

Carton de Wiart, C., Hillewaert, K., Bricteux, L., and Winckelmans, G. (2014). Implicit LES of free and wall bounded turbulent flows based on the discontinuous galerkin/symmetric interior penalty method. *International Journal of Numerical Methods in Fluids*, **78**(6), 335–354.

Carton de Wiart, C., Hillewaert, K., Bricteux, L., and Winckelmans, G. (2015). LES using a discontinuous galerkin method: Isotropic turbulence, channel flow and periodic hill flow. *In: Fröhlich, J., Kuerten, H., Geurts, B., Armenio, V. (eds) Direct and Large-Eddy Simulation IX. ERCOFTAC Series, vol 20. Springer, Cham.*

Chatterjee, I., Zhou, M., Abusorrah, A., Sedraoui, K., and Alabdulwahab, A. (2021). Statistics-based outlier detection and correction method for amazon customer reviews. *Entropy*, **23**(12), 1645.

Chaudhuri, A. and Hu, W. (2019). A fast algorithm for computing distance correlation. *Computational Statistics & Data Analysis*, **135**, 15–24.

Cheng, W., Pullin, D., and Samtaney, R. (2015). Large-eddy simulation of separation and reattachment of a flat plate turbulent boundary layer. *J. Fluid Mech.*, **785**(2), 78–108.

Choi, H. and Moin, P. (2012). Grid-point requirements for large eddy simulation: Chapman's estimates revisited. *Physics of Fluids*, **24**(1), 011702.

Choi, J.-I., Edwards, J. R., and Baurle, R. A. (2009). Compressible boundary-layer predictions at high reynolds number using hybrid LES/RANS methods. *AIAA J.*, **47**(9), 2179–2193.

Colella, K. J. and Keith, W. L. (2003). Measurements and scaling of wall shear stress fluctuations. *Experiments in Fluids*, **34**, 253–260.

Davidson, L. and Dahlström, S. (2005). Hybrid LES-RANS: an approach to make LES applicable at high reynolds number. *Int. J. CFD*, **19**(6), 415–427.

De Graaff, D. B. and Eaton, J. K. (2000). Reynolds-number scaling of the flat-plate turbulent boundary layer. *Journal of Fluid Mechanics*, **422**, 319–346.

Deardoff, J. W. (1970). A numerical study of three-dimensional turbulence channel flow at large reynolds numbers. *Journal of Fluid Mechanics*, **41**(2), 453–480.

Doutrelant, V. and Wong, K. (2021). Free webinar september 21: Alternatives to standard RANS models in automotive aerodynamics.

Duprat, C., Balarac, G., Métais, O., Congedo, P. M., and Brugière, O. (2011). A wall-layer model for large-eddy simulations of turbulent flows with/out pressure gradient. *Physics of Fluids*, **23**(1), 015101.

Dupuy, D., Odier, N., and Lapeyre, C. (2023a). Data-driven wall modeling for turbulent separated flows. *Journal of Computational Physics*, **487**, 112173.

Dupuy, D., Odier, N., Lapeyre, C., and Papadogiannis, D. (2023b). Modeling the wall shear stress in large-eddy simulation using graph neural networks. *Data-Centric Engineering*, **4**, e7.

Duraisamy, K., Iaccarino, G., and Xiao, H. (2019). Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, **51**, 357–377.

Edelmann, D., Mori, T. F., and Székely, G. J. (2021). On relationships between the pearson and the distance correlation coefficients. *Statistics & Probability Letters*, **169**, 108960.

European Parliament (2019). Self-driving cars in the eu: from science fiction to reality. *News European Parliament*.

Frère, A. (2018). *Towards wall-modeled Large-Eddy Simulations of high Reynolds number airfoils using a discontinuous Galerkin method*. Ph.D. thesis, UCL - Université Catholique de Louvain.

Frère, A., Carton de Wiart, C., Hillewaert, K., Chatelain, P., and Winckelmans, G. (2017). Application of wall-models to discontinuous galerkin LES channel flow. *Physics of Fluids*, **9**(8), 085111.

Frère, A., Hillewaert, K., Chatelain, P., and Winckelmans, G. (2018). High reynolds number airfoil: from wall-resolved to wall-modeled LES. *Flow, Turbulence and Combustion*, **101**, 457–476.

Fröhlich, J., Mellen, C. P., Rodi, W., Temmerman, L., and Leschziner, M. A. (2005). Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions. *Journal of Fluid Mechanics*, **526**, 19–66.

Garnier, P., Viquerat, J., Rabault, J., Larcher, A., Kuhnle, A., and Hachem, E. (2021). A review on deep reinforcement learning for fluid mechanics. *Computers & Fluids*, **225**, 104973.

Germano, M. (1986). Differential filters of elliptic type. *The Physics of Fluids*, **29**(6), 1757.

Gier, J., Franke, M., Hübner, N., and Schröder, T. (2010). Designing low pressure turbines for optimized airfoil lift. *ASME J. of Turbomachinery*, **132**(3), 1–11.

Gleick, J. (1988). *CHAOS Making a New Science*. Pinguin Books.

Gloerfelt, X. and Cinnella, P. (2015). Investigation of the flow dynamics in a channel constricted by periodic hills. *45th AIAA Fluid Dynamics Conference. Dallas. United States.*, page 2015.

Graham, J., Kanov, K., Yang, X., Lee, M., Malaya, N., Lalescu, C., Burns, R., Eyink, G., Szalay, A., Moser, R., and Meneveau, C. (2016). A web services-accessible database of turbulent channel flow and its use for testing a new integral wall model for LES. *Journal of Turbulence*, **17**(2), 181–215.

Grinstein, F. F., Margolin, L. G., and Rider, W. J. (2007). *Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics*. Cambridge University Press.

Grötzbach, G. (1987). in encyclopedia of fluid mechanics. *edited by N. P. Cheremisinoff (Gulf, West Orange, NJ, 1987), Vol. 6.*

Guastoni, L., Güemes, A., Ianiro, A., Discetti, S., Schlatter, P., Azizpour, H., and Vinuesa, R. (2021). Convolutional-network models to predict wall-bounded turbulence from wall quantities. *Journal of Fluid Mechanics*, **928**, A27. Publisher: Cambridge University Press.

Güemes, A., Discetti, S., Ianiro, A., Sirmacek, B., Azizpour, H., and Vinuesa, R. (2021). From coarse wall measurements to turbulent velocity fields through deep learning. *Physics of Fluids*, **33**(7), 075121.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Heavens, A. F., Sellentin, E., de Mijolla, D., and Vianello, A. (2017). Massive data compression for parameter-dependent covariance matrices. *arXiv:1707.06529v2*.

Heinz, S. (2020). A review of hybrid RANS-LES methods for turbulent flows: Concepts and applications. *Progress in Aerospace Sciences*, **114**, 100597.

Hillewaert, K. (2013). *Development of the Discontinuous Galerkin Method for high-resolution, large scale CFD and acoustics in industrial geometries*. Ph.D. thesis, Ecole polytechnique de Louvain/iMMC.

Hillewaert, K., Carton de Wiart, C., Verheylewegen, G., and T., A. (2014). Assessment of a high-order discontinuous galerkin method for the direct numerical simulation of transition at low reynolds number in the t106c high-lift low pressure turbine cascade. *In Proceedings of the ASME Turbine Technical Conference and Exhibition, number GT2014-26739. ASME Turbo Expo 2014*.

Hoffmann, G. and Benocci, C. (1995). Approximate wall boundary conditions for large eddy simulations. *In: Benzi, R. (eds) Advances in Turbulence V. Fluid Mechanics and Its Applications, vol 24. Springer, Dordrecht*.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, **4**(2), 251–257.

Horton, H. (1968). *Laminar Separation Bubbles in Two and Three Dimensional Incompressible Flow*. Ph.D. thesis, University of London, Department of Aeronautical Engineering, Queen Mary College.

Hoyas, S. and Jimenez, J. (2008). Reynolds number effects on the reynolds-stress budgets in turbulent channels. *Physics of Fluids*, **20**, 101511.

Humphreys, D., Harris, P., Rodríguez-Higuero, M., Mubarak, F., Zhao, D., and Ojasalo, K. (2015). Principal Component Compression Method for Covariance Matrices Used for Uncertainty Propagation. *IEEE Transactions on Instrumentation and Measurement*, **64**, 356.

Ilyas, I. F. and Chu, X. (2019). *Data Cleaning*. Association for Computing Machinery.

Jamaat, G. T. and Hattori, Y. (2023). *A priori* assessment of nonlocal data-driven wall modeling in large eddy simulation. *Physics of Fluids*, **35**(5), 055117.

Kashyap, R. and Eomm, K. (1988). Robust images techniques with an image restoration application. *IEEE Trans. Acoust. Speech Signal Process*, **36**(8), 1313–1325.

Kawai, S. and Larsson, J. (2012). Wall-modeling in large-eddy simulation: Length-scales, grid resolution, and accuracy. *Physics of Fluids*, **24**(1), 015105.

Kawai, S. and Larsson, J. (2013). Dynamic non-equilibrium wall-modeling for large eddy simulation at high reynolds numbers. *Physics of Fluids*, **25**, 015105.

Kent, C. (2020). Robotic surgery: a race to the top. *Medical Device Network Endoscopy*.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2020). Normalizing Flows: An Introduction and Review of Current Methods. *arXiv:1908.09257v4 [stat.ML]*.

Krank, B., Kronbichler, M., and Wolfgang, A. W. (2019). A multiscale approach to hybrid RANS/LES wall modeling within a high-order discontinuous Galerkin scheme using function enrichment. *International Journal for Numerical Methods in Fluids*, **90**(2), 81–113.

Kunz, R. F. (2019). Wall-modeled large-eddy simulations of spanwise rotating turbulent channels—comparing a physics-based approach and a data-based approach. *Physics of Fluids*, **31**(12), 125105.

Larsson, J., Kawai, S., Bodart, J., and Bermejo-Moreno, I. (2016). Large eddy simulation with modeled wall-stress: recent progress and future directions. *Mechanical Engineering Reviews*, **3**(1), 15–00418.

Lee, Y. M., Lee, J. H., and Lee, J. (2023). Artificial neural network-based wall-modeled large-eddy simulations of turbulent channel and separated boundary layer flows. *Aerospace Science and Technology*, **132**, 108014.

Lino, M., Fotiadis, S., Bharath, A. A., and Cantwell, C. D. (2023). Current and emerging deep-learning methods for the simulation of fluid dynamics. *Proc. R. Soc. A*, **479**, 20230058.

Lozano-Durán, A. and Bae, H. J. (2021). Self-critical machine-learning wall-modeled LES for external aerodynamics. *arXiv:2012.10005 [physics]*.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.*, NeurIPS.

Marusic, I., Monty, J. P., Hultmark, M., and Smits, A. J. (2013). On the logarithmic region in wall turbulence. *Journal of Fluid Mechanics*, **716**(R3).

Mason, P. J. and Callen, N. S. (1986). On the magnitude of the subgrid-scale eddy coefficient in large eddy simulation of turbulent channel flow. *Journal of Fluid Mechanics*, **162**, 439–462.

Morkovin, M. (1969). On the many faces of transition. *Viscous Drag Reduction. ed. C.S. Wells, Plenum Press, New York*, pages 1–31.

Moser, R. D. and Lee, M. (2015). Direct numerical simulation of turbulent channel flow up to Re$_\tau$=5200. *Journal of Fluid Mechanics*, **774**, 395–415.

Nicoud, F., Baggett, J. S., Moin, P., and Cabot, W. (2001). Large eddy simulation wall-modeling based on suboptimal control theory and linear stochastic estimation. *Physics of Fluids*, **13**, 2968.

Nikitin, N. V., Nicoud, F., Wasistho, B., Squires, K. D., and Spalart, P. R. (2000). An approach to wall modeling in large-eddy simulation. *Physics of Fluids*, **12**(7), 1629–1632.

Park, G. I. and Moin, P. (2016). Numerical aspects and implementation of a two-layer zonal wall model for LES of compressible turbulent flows on unstructured meshes. *J. Comput. Phys.*, **305**, 589–603.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peterson, D. M. and Candler, G. V. (2011). Simulations of mixing for normal and low-angled injection into a supersonic crossflow. *AIAA Journal*, **49**(12), 2792–2804.

Piomelli, U. (2008). Wall-layer models for large-eddy simulations. *Progress in Aerospace Sciences*, **44**(6), 437–446.

Piomelli, U. and Balaras, E. (2002). Wall-layer models for large eddy simulations. *Annual Review of Fluid Mechanics*, **34**, 349–374.

Piomelli, U., Moin, P., Ferziger, J. H., and Kim, J. (1989). New approximate boundary conditions for large-eddy simulations of wall-bounded flows. *Physics of Fluids A: Fluid Dynamics 1*, page 1061.

Piomelli, U., Balaras, E., Pasinato, H., Squires, K. D., and Spalart, P. R. (2003). he inner-outer layer interface in large-eddy simulations with wall-layer models. *Int. J. Heat Fluid Flow*, **24**, 538–550.

Pope, S. B. (2000). *Turbulent Flows*. Cambridge University Press.

Radhakrishnan, S., Gyamfi, L. A., Miró, A., Font, B., Calafell, J., and Lehmkuhl, O. (2021). A data-driven wall-shear stress model for LES using gradient boosted decision trees. In H. Jagode, H. Anzt, H. Ltaief, and P. Luszczek, editors, *High Performance Computing*, volume 12761, pages 105–121. Springer International Publishing. Series Title: Lecture Notes in Computer Science.

Rajagopalan, S. and Antonia, R. A. (1979). Some properties of the large structure in a fully developed turbulent duct flow. *The Physics of Fluids*, **22**, 614.

Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record*, **29**(2), 427–438.

Rezende, D. J. and Mohamed, S. (2015). Variational Inference with Normalizing Flows. *arXiv:1505.05770*.

Richardson, L. F. (2007). *Weather prediction by numerical process*. Cambridge University Press.

Roscher, R., Bohn, B., Duarte, M. F., and Garcke (2020). Explainable machine learning for scientific insights and discoveries. *IEEE Access*, **8**, 42200.

Sagaut, P. (2006). *Large Eddy Simulation for Incompressible Flows*. Springer Berlin, Heidelberg.

Schlichting, H. and Gersten, K. (2017). Extensions to the Prandtl Boundary–Layer Theory. In H. Schlichting and K. Gersten, editors, *Boundary-Layer Theory*, pages 377–411. Springer, Berlin, Heidelberg.

Schrooyen, P. (2015). *Numerical Simulation of aerothermal flows through ablative thermal protection systems*. Ph.D. thesis, UCL - Université Catholique de Louvain/iMMC.

Schumann, U. (1975). Subgrid-scale model for finite-difference simulations of turbulent flows in plane channels and annuli. *Journal of Computational Physics*, **18**(4), 376–404.

Sharma, P., Chung, W. T., Akoush, B., and Ihme, M. (2023). A Review of Physics-Informed Machine Learning in Fluid Mechanics. *Energies*, **16**(5), 2343.

Shih, T. H., Povinelli, L. A., Liu, N.-S., and Lumley, J. L. (1999). A generalized wall function. *Technical report, National Aeronautics and Space Administration, Glenn Research Center,.*

Shur, M. L., Spalart, P. R., Strelets, M. K., and Travin, A. K. (2008). A hybrid RANS-LES approach with delayed-DES and wall-modeled LES capabilities. *Heat Fluid Flow*, **29**(6), 1638–1649.

Song, S. and Eaton, J. (2004). Flow structures of a separating, reattaching, and recovering boundary layer for a large range of reynolds number. *Experiments in Fluids*, **36**, 642–653.

Spalart, P. R. (2009). Detached-eddy simulation. *Annual Review of Fluid Mechanics*, **41**, 181–202.

Spalart, P. R., H., J. W., M., S., and R., A. S. (1997a). Comments on the feasibility of LES for wings and on a hybrid RANS/LES approach. *Advances in DNS/LES: Direct numerical simulation and large eddy simulation*, pages 137–148.

Spalart, P. R., Jou, W.-H., Stretlets, M., and Allmaras, S. R. (1997b). Comments on the feasibility of LES for wings and on the hybrid RANS/LES approach. In *Proceedings of the First AFOSR International Conference on DNS/LES*.

Spalart, P. R., Deck, S., Shur, M. L., Squires, K. D., Strelets, M. K., and Travin, A. (2006). Approximate wall boundary conditions for large eddy simulations. *Theor. Comput. Fluid Dyn.*, **20**(3), 181–195.

Spalding, D. B. (1961). A single formula for the law of the wall. *Journal of Applied Mechanics*, **28**(3), 455–458.

Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, **35**(6), 2769–2794.

Temmerman, L., Leschziner, M., Mellen, C., and Fröhlich, J. (2003). Investigation of subgrid-scale models and wall-function approximations in large eddy simulation of separated flow in a channel with streamwise periodic constrictions. *International Journal of Heat and Fluid Flow*, **24**(2), 157–180.

Thiry, O. (2017). Investigation of wall shear stress models for large eddy simulation.

Upadhyaya, D. S. and Singh, K. (2012). Nearest neighbour based outlier detection techniques. *International Journal of Computer Trends and Technology*.

Vaishali, D., Ramesh, R., and Christaline, J. A. (2014). 2d autoregressive model for texture analysis and synthesis. *International Conference on Communications and Signal Processing (ICCSP)*, pages 1135–1139.

Wang, M. (1999). LES with wall models for trailing-edge aeroacoustics. *In Annu Res. Briefs. Center Turbul. Res. Stanford Univ. Calif.*, pages 355–364.

Werner, H. and Wengle, H. (1993). Large-eddy simulation of turbulent flow around a cube in a plane channel. *In Selected Papers from the 8th Symposium on Turbulent Shear Flows, ed. F. Durst, F., Friedrich, R., Launder, B. E., Schumann, U., Whitelaw, J. H., New York: Springer.*, pages 155–168.

Yang, X. I. A., Sadique, J., Mittal, R., and Meneveau, C. (2015). Integral wall model for large eddy simulations of wall-bounded turbulent flows. *Physics of Fluids*, **27**(2), 025112.

Yang, X. I. A., S., Z., J.-X., W., and H., X. (2019). Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Physical Review Fluids*, **4**, 034602.

Zhideng, Z., Xiang, X. I. A., Fengshun, Z., and Yang, X. (2023). A wall model learned from the periodic hill data and the law of the wall. *Physics of Fluids*, **35**(5), 055108.

Zhou, D., Whitmore, M. P., Griffin, K. P., and Bae, H. J. (2022). Multi-agent reinforcement learning for wall modeling in LES of flow over periodic hills.

Zhou, Z., He, G., and X., Y. (2021). Wall model based on neural networks for LES of turbulent flows over periodic hills. *Physical Review Fluids*, **6**, 054610.

Österlund, J., Johansson, A. V., Nagib, H. M., and Hites, M. H. (2000). A note on the overlap region in turbulent boundary layers. *Physics of Fluids*, **12**(1), 1–4.