UCLouvain

École polytechnique de Louvain

Effect of the subgrid scales on the training of a neural network based turbulent heat transfer model for liquid metals

Author: **Florentin PRUD'HOMME**
Supervisors: **Yann BARTOSIEWICZ, Lilla KOLOSZAR**
Readers: **Matilde FIORE, Matthieu DUPONCHEEL, Francesco CONTINO**
Academic year 2021–2022
Master [120] in Mechanical Engineering

Abstract

Two of the concepts of nuclear reactors selected by the generation IV international forum rely on a liquid metal coolant. Hence, a good prediction capability of the thermal hydraulics aspects of liquid metal flows is highly desirable in the design process of those reactors. However, liquid metals are characterized by low Prandtl numbers for which the widely used Reynolds analogy does not produce satisfactory results anymore. Consequently, a more sophisticated approach is required to model the turbulent heat flux. In that context, a neural network based thermal turbulence model aimed at modelling the turbulent heat flux for a wide range of Prandtl numbers was developed at the VKI in the framework of a Phd thesis. This data-driven approach uses DNS data to train an artificial neural network (ANN). The more data are provided to train the ANN, the more the resulting model is expected to generalize well in full scale industrial applications. Because DNS data are computationally very expensive to produce, there is an interest in exploring the possibility to use also LES data to train the model to increase the size of the training database at lower cost. In this work, LES of non-isothermal turbulent channel flows are performed for different Reynolds and Prandtl numbers and several grid resolutions. The statistics gathered are used to train the neural network to provide a preliminary assessment of the possibility to use LES data for the training. Several analyses tend to suggest that LES data can be used as training data at the cost of a prediction error dependent on the grid resolution. However, those analyses highlight also that using LES data of channel flows only is not sufficient to provide a strong conclusion and that investigations on more complex flows configurations are needed.

Acknowledgements

Foremost, I would like to express my gratitude to my supervisors, the Pr. Yann Bartosiewicz for his wise counsels and his guidance and Dr. Lilla Koloszar also for her guidance, for pushing me to improve myself and for being such a motivating person.

I would like to particularly thank Matilde Fiore, who developed the data-driven model, for sharing her expertise and for her patience, her answers to my numerous questions, her suggestions and her careful relectures.

I would like also to thank Lutgarde Schrijvers who has been the electrochoc which put me back to work and Edith Coune for generously accepting to freely lodge me near the VKI, sparing me a lot of hours in public transportation. Finally, I thank my family for the infallible emotional support they provided to me.

Contents

Nomenclature **ix**

1 Introduction **1**
  1.1 Computational fluid dynamics overview . . . 2
  1.2 Turbulent heat transfer modelling . . . 3

2 Data-driven thermal turbulence modelling **5**

3 Large eddy simulation : non-isothermal turbulent channel flow **9**
  3.1 Computational domain . . . 11
  3.2 Modified equations and wall boundary conditions . . . 11
  3.3 Numerical set-up . . . 13
    3.3.1 Mesh . . . 13
    3.3.2 Modified OpenFOAM codes . . . 14
    3.3.3 Main settings and running procedure . . . 15
  3.4 Results analysis . . . 16
    3.4.1 Normalisation and mean velocity profiles . . . 17
    3.4.2 Reynolds stress tensor and Turbulent kinetic energy . . . 19
    3.4.3 Dissipation rate of turbulent kinetic energy . . . 21
    3.4.4 Instantaneous velocity field . . . 23
    3.4.5 Temperature fields . . . 25
    3.4.6 Turbulent heat transfer . . . 28
    3.4.7 Temperature variance and dissipation rate of the temperature variance . . . 31
    3.4.8 Turbulent eddy-viscosity and eddy-diffusivity . . . 33

4 Artificial neural network training using LES **35**
  4.1 Direct neural network inputs . . . 35
  4.2 First training campaign . . . 36
  4.3 Integrated Gradient method analysis . . . 39
  4.4 Statistical analysis . . . 42
    4.4.1 Neural network re-dimensioning . . . 43
    4.4.2 Statistical outputs and LES resolution effect . . . 44
  4.5 Training with all data . . . 51

5 Conclusion and outcomes **57**

6 Appendix **59**
- 6.1 Channel flow profiles 59
  - 6.1.1 Momentum fields 59
  - 6.1.2 Thermal fields for $Pr = 0.71$ 61
  - 6.1.3 Thermal fields for $Pr = 0.025$ 62
  - 6.1.4 Thermal fields for $Pr = 0.01$ 63
- 6.2 Instantaneous fields for $Re_\tau = 395$ 65
- 6.3 Π profiles 66
- 6.4 Integrated Gradients 69
  - 6.4.1 First linear path : $\mathbf{x}'_{(Re_\tau=640,Pr=0.71)} \longrightarrow \mathbf{x}_{(Re_\tau=395,Pr=0.025)}$ . . . 69
  - 6.4.2 Second linear path : $\mathbf{x}'_{(Re_\tau=640,Pr=0.71)} \longrightarrow \mathbf{x}_{(Re_\tau=640,Pr=0.025)}$ . . 70
- 6.5 Complete statistical analysis 73
- 6.6 Training using all data (predictions for $Pr = 0.025$ and $Pr = 0.71$) . . 89

Nomenclature

Abbreviations

ANN	artificial neural network
CFD	Computational fluid dynamics
CFL	Courant-Friedrich-Lewy
DNS	Direct Numerical Simulation
EU	European Union
FV	finite volume
LES	Large Eddy Simulation
MYRTE	MYRRHA Research and Transmutation Endeavour
OpenFOAM	Open Field Operation and Manipulation
PISO	Pressure-Implicit with Splitting of Operators algorithm
RANS	Reynolds Averaged Navier-Stokes
ReLu	Rectified Linear unit
SESAME	thermal hydraulics Simulations and Experiments for the Safety Assessment of MEtal cooled reactors
SGS	subgrid-scale
SWAG	Gaussian Stochastic Weight Averaging algorithm
THF	turbulent heat flux
THINS	Thermal-Hydraulics of Innovative Nuclear Systems
TKE	turbulent kinetic energy
WALE	wall-adapting local Eddy-viscosity

Greek symbols

α	molecular heat diffusivity $[m^2/s]$
α_t	turbulent heat diffusivity $[m^2/s]$

α_{sgs}	SGS eddy-diffusivity $[m^2/s]$
$\Delta x, \Delta y, \Delta z,$	grid spacings in the x, y, z directions $[m]$
Δ	filter length, cube root of the cell volume $\Delta = (\Delta x \Delta y \Delta z)^{1/3}$ $[m]$
δ	half channel height $[m]$
ϵ	dissipation rate of the turbulent kinetic energy $[m^2/s^3]$
ϵ_θ	dissipation rate of the temperature variance $[K^2/s]$
η	Kolmogorov microscale $\eta = (\nu^3/\epsilon)^{1/4}$
η_θ	Obukhov-Corrsin microscale $\eta_T = \eta Pr^{-3/4}$
κ	Von Karman constant
λ	Taylor microscale
ν	molecular kinematic viscosity $[m^2/s]$
ν_t	turbulent kinematic viscosity $[m^2/s]$
ν_{sgs}	SGS eddy-viscosity $[m^2/s]$
ρ	density $[kg/m^3]$
τ_w	statistically averaged wall shear stress $[Pa]$
θ	transformed temperature $[K]$

Operators, subscripts and superscripts

$(\cdot)'$	fluctuating part
$(\cdot)^+$	wall normalized by δ, u_τ, ν and T_τ
$(\cdot)_{sgs}$	subgrid scale contribution
$\overline{(\cdot)}$	statistically averaged quantities
$\widetilde{(\cdot)}$	filtered quantities
$tr(\cdot)$	trace

Roman symbols

$\mathbf{u}$	velocity vector $[m/s]$
c_p	specific heat at constant pressure $[J/kg \cdot K]$
k	turbulent kinetic energy $[m^2/s^2]$
k_θ	temperature variance $[K^2]$
L_x, L_y, L_z	domain size in the x,y,z directions $[m]$
Nu	Nusselt number
P	reduced pressure $P = p/\rho$ $[m^2/s^2]$

Pr	Prandtl number
Pr_t	turbulent Prandtl number
Pr_{sgs}	SGS Prandtl number
q_w	wall heat flux $[W/m^2]$
Re_b	bulk Reynolds number $Re_b = u_b 2\delta/\nu$
Re_τ	friction Reynolds number $Re_\tau = u_\tau \delta/\nu$
T	temperature $[K]$
T_m	cup mixing temperature $T_m = [K]$
T_W	wall temperature $[K]$
T_τ	friction temperature $T_\tau = q_w/\rho c_p u_\tau$ $[K]$
u_b	averaged bulk velocity $[m/s]$
u_τ	friction velocity $u_\tau = \sqrt{\tau_w/\rho}$ $[m/s]$
x, y, z	Cartesian coordinates in the streamwise, wall-normal and spanwise directions $[m]$

Chapter 1

Introduction

Among the six concepts of nuclear reactors selected by the generation IV international forum (GIF, 2010), two designs are pool type reactors using a liquid metal coolant : the sodium-cooled fast reactor (SFR) and the lead-cooled fast reactor (LFR) along with its variant cooled by the lead-bismuth eutectic. Hence, good understanding and good predicting capabilities of the thermal hydraulics aspects of liquid metals flows are highly desirable in the design process of those reactors. More generally, liquid metal flows can be encountered in many other applications of interest like, by example, advanced concentrating solar power (CSP) plants [1], continuous casting of metals [2] or even characterization of Earth metal core [3].

In a design process, computational fluid dynamics (CFD) is nowadays an important tool to study thermal-hydraulics, particularly for liquid metal flows for which experiments are very challenging because liquid metals are opaque, chemically aggressive and generally operate at relatively high temperature. However, industrial applications are most often characterized by very complex geometries so that, given the computational power currently available, the industry has to rely on the RANS approach. While the CFD industrial codes generally offer a wide range of turbulence models to close the momentum equation, the closure of the energy equation is almost invariably done with the use of the Reynolds analogy.

This analogy relies on the similarity hypothesis between the momentum and the thermal fields. While this hypothesis is acceptable for fluids characterized by close to unity or higher than unity Prandtl numbers like air and water, it is no longer justified for liquid metals which are characterized by very low Prandtl numbers. This introduces the need for a thermal turbulence model to close the RANS energy equation in order to provide accurate predictions of the temperature field in liquid metal flows. In the frame of the THINS EU project, several models [4, 5, 6] were proposed to close the energy equation but this project highlighted the lack of validation data to assess the performance of those models. This is why in the frame of the SESAME and MYRTE EU projects, an extensive research effort was put into the generation of a DNS validation database [7]. The important amount of high fidelity data generated opened the path for a data-driven approach for the modelling of the turbulent heat flux (THF).

In that context, a neural-network based turbulent heat transfer model aimed at modelling the THF for a wide range of Prandtl numbers was developed at the VKI by Matilde Fiore in the frame of her Phd thesis. When a data-driven approach is chosen, the more data from diverse flow conditions are provided to train the model, the more the obtained model is likely to generalize well for full scale industrial applications. However, due to the high computational cost of DNS, the training database remains restricted. Hence, there is an interest in exploring the possibility to use LES, which are computationally cheaper and can still be of fair quality, beside DNS, to increase the size of the training database. This is what the present work intends to provide : a preliminary assessment of the LES option to train the neural network based turbulent heat transfer model. More precisely, an assessment of the option of using LES generated using an industrial CFD code based on the finite volume method.

To this aim, firstly the DNS, LES and RANS CFD approach are briefly exposed as well as the classical approach to model the THF. Then in a second chapter, the mathematical structure of the data-driven model is briefly reviewed. A third chapter exposes the generation process of an LES database of non-isothermal channel flows for different Reynolds and Prandtl numbers and different grid resolutions. This chapter also provides an LES grid convergence analysis. A fourth chapter provides several analyses aimed at evaluating the option of using the generated LES to train the neural network based turbulent heat transfer model. Finally, a fifth and last chapter summarizes the main outcomes of chapter 3 and 4 to conclude.

1.1 Computational fluid dynamics overview

There are three classical ways to perform CFD of turbulent flows (of which a much more in depth insight can be found in [8]). The first one is Direct Numerical Simulation (DNS) which is aimed at resolving the Navier-Stokes equations down to the smallest scales of turbulence characterized by the Kolmogorov microscale,

$$\eta = \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}}, \tag{1.1}$$

where turbulence is dissipated into heat. This provides a very accurate representation of real turbulent flows so that it is sometimes qualified as a "numerical experiment". However performing DNS is extremely computationally expensive. Indeed the number of grid points required to correctly capture the complete turbulence spectrum of a turbulent flow scales as $Re^{9/4}$. Hence DNS is generally limited to moderate Reynolds numbers and simple geometries.

The second one is Large Eddy Simulation (LES), the underlying philosophy of which is that the smallest scales of turbulence can be modeled rather than simulated because the smallest turbulent structures are more universal and independent of a particular geometry. This approach lowers the computational cost of turbulent flow simulations. It introduces the need for a subgrid scale model but it can provide a high fidelity representation of turbulent flows though of course not as faithful as

DNS [9]. However, up to now, this approach generally remains computationally unaffordable for complex full scale industrial applications like nuclear reactors.

Nowadays, the work-horse of the CFD industry is still the Reynolds Averaged Navier-Stokes (RANS) approach which models the complete spectrum of turbulence and provides only statistically averaged fields [10]. This approach is much more computationally affordable but the quality of the obtained flow predictions depends heavily on the quality of the RANS turbulence model which is used to close the averaged Navier-Stokes equations, and none of the existent turbulence models are flawless.

1.2 Turbulent heat transfer modelling

The turbulent heat transfer $\overline{\mathbf{u}'T'}$ is introduced in the averaged energy equation :

$$\frac{\partial \overline{T}}{\partial t} + \overline{u}_i \frac{\partial \overline{T}}{\partial x_i} = \alpha \frac{\partial^2 \overline{T}}{\partial^2 x_i} - \frac{\partial \overline{u_i' T'}}{\partial x_i}, \tag{1.2}$$

and it needs to be modelled to close the equation. Most often the Reynolds analogy is used. This model relies on the simple gradient diffusion hypothesis with a constant turbulent Prandtl number :

$$\overline{u_i' T'} = -\alpha_t \frac{\partial \overline{T}}{\partial x_i} = -\frac{\nu_t}{Pr_t} \frac{\partial \overline{T}}{\partial x_i}, \tag{1.3}$$

where ν_t is the turbulent eddy-viscosity. Figure 1.1 shows the comparison between the temperature profiles for DNS data and RANS data obtained with the Launder-Sharma $k - \epsilon$ model. A common value of 0.9 is chosen for the turbulent Prandtl number. One can clearly observe that the Reynolds analogy yields satisfactory results for a close to unity Prandtl number but that it significantly under-predicts the dimensionless temperature profile for a low Prandtl number so that a more sophisticated model is needed to accurately predict the temperature field.

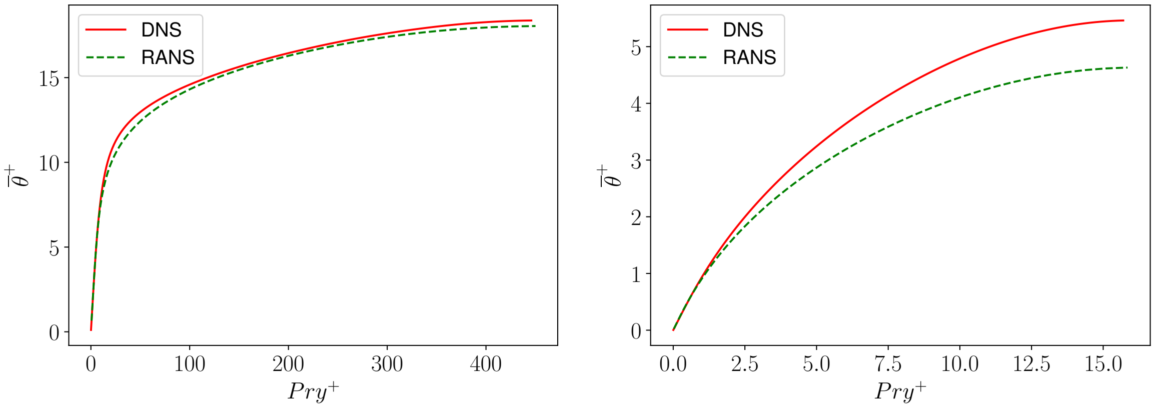


Figure 1.1: Temperature profiles obtained for $Pr = 0.71$ (left) and $Pr = 0.025$ (right) for $Re_\tau = 640$.

Another model based on the simple gradient diffusion hypothesis is the Kays correlation [11] for which the turbulent Prandtl is calculated as follows :

$$Pr_t = 0.85 + \frac{0.7}{Pr\left(\frac{\nu_t}{\nu}\right)} \tag{1.4}$$

This model is a better choice for RANS simulations and works quite well for channel flows [12]. However, being based on the simple gradient diffusion hypothesis, it can't provide a complete representation of the turbulent heat flux. Moreover, this model has intrinsic limitations when it comes to extend it to the mixed and natural convection flow regimes [7].

Shams et al. [13] provide a review for the many more advanced models which have been developed. Calibration of those advanced models is identified as a big challenge and this is one of the reason which motivates a data-driven approach to take advantage of the powerful high dimensional non linear regression capabilities that Machine Learning can provide.

Chapter 2

Data-driven thermal turbulence modelling

The global mathematical architecture and the main principles behind the artificial neural network based turbulent heat transfer model developed by M. Fiore need to be briefly reviewed for a good understanding of the next chapters. Moreover, the interested reader is invited to read her dedicated paper (Fiore et al. 2022 [14]) to get a much deeper understanding of the model.

This model is explicit and algebraic and computes the turbulent heat transfer as a product of the temperature gradient with a dispersion tensor:

$$\overline{\mathbf{u}'\theta'} = -\mathbf{D}\nabla T, \tag{2.1}$$

whose functional relationship is expressed as follows:

$$\mathbf{D} = f(\mathbf{b}, \mathbf{S}, \mathbf{\Omega}, |\nabla T|, k, \epsilon, k_\theta, \epsilon_\theta, Pr), \tag{2.2}$$

where $\mathbf{b}$, $\mathbf{S}$ and $\mathbf{\Omega}$ are tensors defined as :

$$b_{ij} = \frac{\overline{u_i' u_j'}}{k} - \frac{2}{3}\delta_{ij}, \tag{2.3}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right), \tag{2.4}$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} - \frac{\partial \overline{u}_j}{\partial x_i}\right). \tag{2.5}$$

It should be noted that the model is presently limited to forced convection flows. The definition of all the statistics present in expression 2.2 will be recalled throughout section 3.4. From the inputs in 2.2, a general dimensionless invariant basis and a dimensionalized tensor basis are derived under the assumption of 2D flow. The expressions constituting those bases are summarized in table 2.1. Polynomials of the base tensors with coefficients depending on the invariants can be used to express any high order products of the tensors $\mathbf{b}$, $\mathbf{S}$ and $\mathbf{\Omega}$. Hence the formulation remains mathematically general.

Invariant basis	Tensor basis
$\pi_1 = \frac{k^2}{\epsilon^2} tr(\mathbf{S}^2)$, $\pi_2 = \frac{k^2}{\epsilon^2} tr(\mathbf{\Omega}^2)$, $\pi_3 = \frac{1}{tr(\mathbf{b})}$, $\pi_4 = \frac{k}{\epsilon} tr(\mathbf{bS})$, $\pi_5 = \frac{1}{tr(\mathbf{b_2})}$, $\pi_6 = tr(\mathbf{bS}\Omega)\frac{k^2}{\epsilon^2}$, $\pi_7 = \frac{\lvert\nabla T\rvert}{\sqrt{k_\theta}}\frac{k^{3/2}}{\epsilon}$, $\pi_8 = \frac{k_\theta \epsilon}{k \epsilon_\theta}$, $Re_t = \frac{k^2}{\epsilon \nu}$, Pr	$\mathbf{T_1} = \frac{k}{\epsilon^{1/2}}\mathbf{I}$, $\mathbf{T_2} = \frac{k}{\epsilon^{1/2}}\mathbf{b}$, $\mathbf{T_3} = \frac{k^2}{\epsilon^{3/2}}\mathbf{S}$, $\mathbf{T_4} = \frac{k^2}{\epsilon^{3/2}}\mathbf{\Omega}$, $\mathbf{T_5} = \frac{k^2}{\epsilon^{3/2}}\mathbf{bS}$, $\mathbf{T_6} = \frac{k^2}{\epsilon^{3/2}}\mathbf{b\Omega}$, $\mathbf{T_7} = \frac{k^3}{\epsilon^{5/2}}\mathbf{S\Omega}$ $\mathbf{T_8} = \frac{k^3}{\epsilon^{5/2}}\mathbf{bS\Omega}$

Table 2.1: Invariant and tensor bases

Then the dispersion tensor is calculated as:

$$\mathbf{D} = \left[(\mathbf{A} + \mathbf{A}^T)(\mathbf{A}^T + \mathbf{A}) + \frac{k}{\epsilon^{0.5}}(\mathbf{W} - \mathbf{W}^T)\right], \tag{2.6}$$

where the tensors $\mathbf{A}$ and $\mathbf{W}$ are weighted sums of the tensors $\mathbf{T}^i$ of the tensor basis:

$$\mathbf{A} = \sum_{i=1}^{n} a_i \mathbf{T}^i, \qquad \mathbf{W} = \sum_{i=1}^{n} w_i \mathbf{T}^i. \tag{2.7}$$

The coefficients a_i and w_i are isotropic functions of the invariants:

$$a_i = f_i(\pi_j, Re_t, Pr), \qquad w_i = g_i(\pi_j, Re_t, Pr). \tag{2.8}$$

This mathematical structure was carefully built to enforce some fundamental properties to the model. Indeed, the formulation is rotational invariant and expression 2.6 make the model automatically satisfy the second law of thermodynamics.

An artificial neural network is used to model the coefficients a_i and w_i. The architecture of this network and its weights $\mathcal{W}$ define a general function:

$$\mathbf{Y} = \mathcal{M}(\mathbf{X}; \mathcal{W}), \tag{2.9}$$

where the input vector $\mathbf{X}$ is constituted by the invariants:

$$\mathbf{X} = [\pi_1, ..., \pi_8, Re_t, Pr], \tag{2.10}$$

and the output vector $\mathbf{Y}$ is constituted by the coefficients a_i and w_i:

$$\mathbf{Y} = [a_1, ..., a_8, w_1, ..., w_8]. \tag{2.11}$$

Figure 2.1 shows the structure of the ANN which was implemented using the Pytorch library. It is composed of two distinct branches. The first one takes the logarithmically transformed invariant basis as input layer. This layer is followed by seven hidden layers, all with rectified linear unit activation functions (ReLu), except for the last one which has hyperbolic tangent activation. This organization allows combination of the invariants through products thanks to the logarithm and hyperbolic tangent properties. The second branch takes the Prandtl number only as input layer and two hidden layers follow, the first and the second respectively with a ReLu and a hyperbolic tangent activation function. Finally, the two branches merge through a multiplication of their outputs, yielding the coefficients a_i and w_i. Every hidden layer of this network is constituted by 100 neurons.

The network weights $\mathcal{W}$ are randomly initialised and need to be adjusted through a training phase. This is where reference data are needed. During the training, a sufficiently large number of successive epochs are performed. An epoch applies the following sequence:

1. Several subpartitions of the training data are randomly selected to constitute a minibatch of which all the input data are passed through the ANN to make predictions of the THF.
2. The loss function computes in a particular way (exp. 2.12) the error related to the predictions when compared to the the reference THF.
3. The error is backpropagated and the Adaptive Moment Estimation (ADAM) optimizer is used to update the weights.

The following loss function is used:

$$\mathcal{L} = \frac{1}{N}\left(\sum_{i=1}^{N}\sum_{j=1}^{3}(\hat{q}_{i,j} - q_{i,j})^2\right) + \frac{10}{N}\left(\sum_{i=1}^{N}\sum_{j,k=1}^{3}\left|\frac{\partial \hat{q}_{i,j}}{\partial x_k} - \frac{\partial q_{i,j}}{\partial x_k}\right|\Delta x_k\right), \tag{2.12}$$

where $\hat{q}_{i,j}$ and $q_{i,j}$ are the j component of respectively the reference THF and the predicted THF for the i data point of the minibatch which contains N data points. The first term evaluates the error in the least square sense while the second involves derivatives of the THF to value the smoothness of the predictions. The smoothness of the THF obtained with the model is indeed important because the THF enters the averaged energy equation through its divergence.

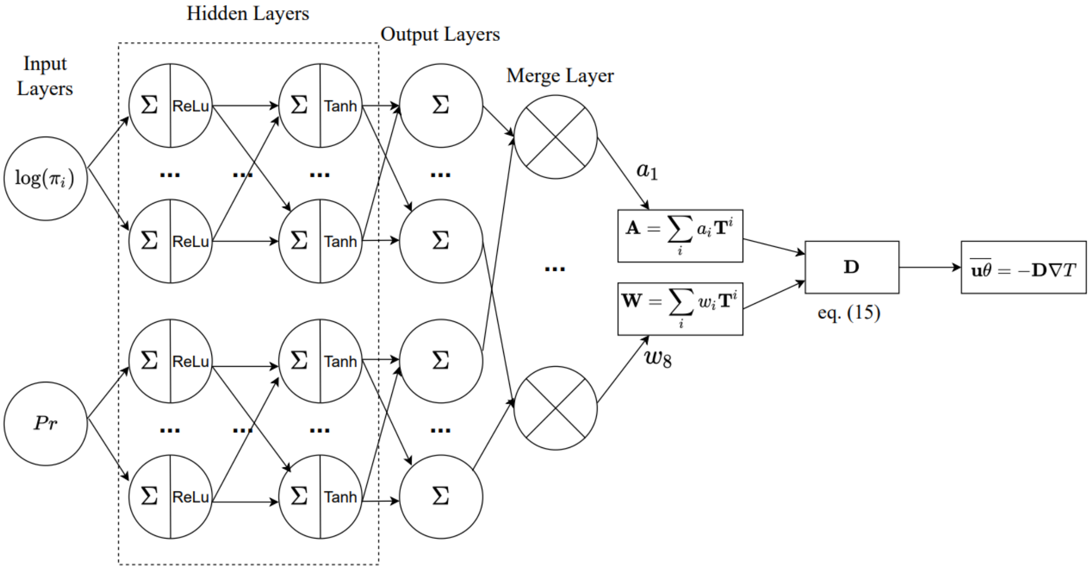


Figure 2.1: ANN architecture (from Fiore et al. [14])

Not all the reference data are used for the training. Around one quarter has to be left as validation data. At each epoch, those data are also passed through the ANN and related losses are also calculated but not backpropagated. This is done to compare the evolution of the losses related to the minibatch data and the losses related to the validation data. If the training losses tend to decrease while at the same time the validation losses tend to increase, it means that the ANN is overfitting the training data.

The Gaussian Stochastic Weight Averaging (SWAG) algorithm [15] is implemented to quantify the uncertainty related to the model prediction. For the last 2000 epochs, the mean μ and the variance Γ of each weight are computed. The weights are assumed to follow a Gaussian distribution :

$$\mathcal{W} \sim \mathcal{N}(\mu, \Gamma). \tag{2.13}$$

After the training, the mean prediction and the standard deviation of the network predictions are calculated by averaging the results obtained by sampling the parameters from the Gaussian distribution.

The ultimate goal of this model is of course to be used in conjunction with a RANS momentum model which has to provide the momentum statistics present in expression 2.2. However, all the RANS momentum turbulence models are imperfect and can't provide the same level of accuracy as a DNS for the momentum statistics. Because the ANN is trained using DNS data (and in this work, using LES data as well), the obtained THF model can perform well a priori when high quality data are used as input while performing poorly a posteriori with a RANS momentum turbulence model which is not accurate enough. Hence, the lack of robustness regarding the accuracy of the input momentum statistics is currently a weakness of this model.

It is evident that in order for a reference database to be used for the training of the ANN, it needs to provide the inputs of the model (expressions 2.2) as well as its outputs (the components of the THF). For the particular case of channel flows, which is the subject of the next chapter, the statistically averaged data are 1D (there is no evolution of the averaged quantities in the flow direction) so that the tensors $\mathbf{S}$ and $\mathbf{\Omega}$ are fully determined by the mean velocity gradient $\frac{d\overline{u}}{dy}$. Another consequence is that the two first invariants are strictly equal. Moreover, in that case, only the second column of the dispersion tensor (components D_{yy} and D_{xy}) is relevant for the calculation of the THF.

Chapter 3

Large eddy simulation : non-isothermal turbulent channel flow

This chapter presents the generation process and a convergence analysis of the large eddy simulation database which will be used throughout this work. The physical case considered here is a developed turbulent flow between two uniformly heated infinite plates. Given that incompressibility is assumed, that thermophysical property variations with temperature are neglected and that no buoyancy effects are considered, this type of flow is fully characterized by a Reynolds number, a Prandtl number and a thermal boundary condition.

The governing equations are the filtered Navier-Stokes equations:

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0, \tag{3.1}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial \widetilde{\tilde{u}_i \tilde{u}_j}}{\partial x_j} = -\frac{\partial \widetilde{P}}{\partial x_i} + \frac{\partial}{\partial x_j}(2\nu \widetilde{S}_{ij} + \tilde{\sigma}_{ij}^{sgs}), \tag{3.2}$$

$$\frac{\partial \widetilde{T}}{\partial t} + \frac{\partial \widetilde{\widetilde{T} \tilde{u}_j}}{\partial x_j} = \frac{\partial}{\partial x_j}(\alpha \frac{\partial \widetilde{T}}{\partial x_j} + \tilde{\tau}_T^{sgs}), \tag{3.3}$$

where $\widetilde{P} = \widetilde{p}/\rho$ is the filtered reduced pressure, S_{ij} is the strain rate tensor, $\tilde{\sigma}_{ij}^{sgs} = \widetilde{\tilde{u}_i \tilde{u}_j} - \widetilde{u_i u_j}$ is the subgrid stress tensor and $\tilde{\tau}_T^{sgs} = \widetilde{\widetilde{T} \tilde{u}_j} - \widetilde{T u_j}$ is the subgrid heat flux. Provided that an eddy-viscosity SGS model is chosen, the deviatoric part of the subgrid stress tensor $\tilde{\tau}_{ij}^{sgs} = -2\nu_{sgs}\widetilde{S}_{ij}$. For the sake of clarity, the filter operator "$\tilde{}$" won't be used anymore hereafter and filtering will be implicitly assumed whenever LES related quantities are mentioned.

The objective is to reproduce some of the non-isothermal channel flow DNS cases available, namely from the database maintained by Kawamura et al.[16] and that from Tiselj et al.[17]. The contents of those databases is summarized in the following table:

Author	Re_τ	Pr	Thermal BC
Kawamura	180,395,640	0.025-0.71	NF
Tiselj	180,395,590	0.01	F, NF, C

Table 3.1: reference non-isothermal DNS channel flow database (F : fluctuating, NF : non-fluctuating, C : conjugate heat transfer)

The Reynolds number $Re_\tau = \frac{u_\tau \delta}{\nu}$ is based on the friction velocity $u_\tau = \sqrt{\tau_w/\rho}$ and the half channel height δ. Among those Prandtl numbers, $Pr = 0.71$ is relevant for air while $Pr = 0.025$ and $Pr = 0.01$ can be considered representative of liquid metals such as sodium and the lead-bismuth eutectic.

Two target Reynolds numbers and four Prandtl numbers are chosen for the LES: $Re_\tau = 395$ and 640 each for $Pr = 1.0, 0.71, 0.025$ and 0.01 which make eight physical cases summarized in table 3.2. The Prandtl number $Pr = 1.0$ is considered to verify the similarity between momentum and thermal fields which should be mathematically satisfied for this Prandtl number, this as part of the convergence analysis. On the other hand, only the non-fluctuating thermal boundary condition is chosen because of its simplicity and its availability for both DNS data sets.

	$Pr = 1$	$Pr = 0.71$	$Pr = 0.025$	$Pr = 0.01$
$Re_\tau = 395$	/	K	K	T
$Re_\tau = 640$	/	K	K	/

Table 3.2: Chosen physical cases: "K" stands for a case for which there is a DNS validation available from Kawamura, "T" one from Tiselj and "/" none

Despite its wide availability among the DNS database and its low computational cost, the $Re_\tau = 180$ flow case is disregarded because performing a rigorous LES for this very low Reynolds number is actually challenging. Indeed, applying the general LES mesh requirements to this case lead to a very coarse mesh unable to properly capture any turbulent structure. Also, turbulence can be difficult to trigger and maintain for this barely turbulent flow. This is why LES for such a low Reynold number are scarcely found in the literature.

Finally, the LES are performed for three different resolutions to conduct a convergence analysis and assess the effect of the grid refinement on the training of the neural network (chapter 4). Because no buoyancy effects is assumed, the temperature acts as a passive scalar and the momentum field is independent on the temperature field so that multiple energy equations can be solved simultaneously for a unique momentum field. Hence, this results in six simulations in total (three for both targeted Reynold numbers) each resolving four energy equations. Those simulations are run with OpenFOAM (version 6). It is an open source `C++` toolbox providing an implementation of the finite volume method offering CFD capabilities for a wide range of applications. Moreover the user is free to modify the base solvers and utilities or to implement new ones, a feature which will be useful in this work (see section 3.3.2).

3.1 Computational domain

A finite sub-domain of the channel with periodic boundary conditions in the streamwise and in the spanwise directions is used to approximate the two infinite flat plates (Figure 3.1). This allows to numerically simulate this physical case. However, the sub-domain needs to be large enough to encompass the largest turbulent structures of the flow. A $L_x \times L_y \times L_z = 6.4\delta \times 2\delta \times 3.2\delta$ domain size is used (whith δ,the half channel height) just like the domain used by Kawamura et al.[16] did for their DNS. Generally a very similar domain size is used in the literature for turbulent channel flow configurations [18, 19, 12].

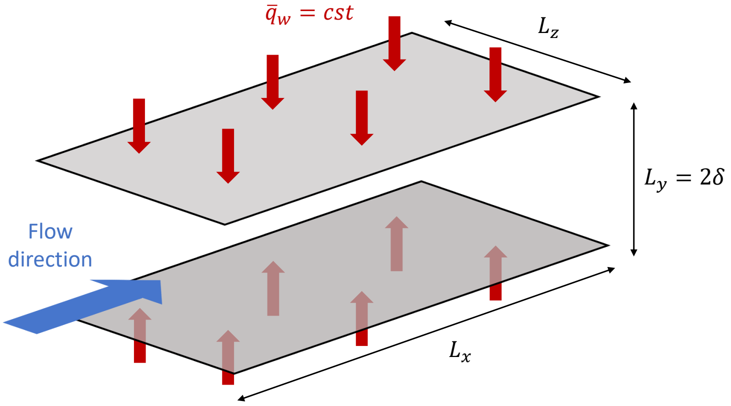


Figure 3.1: Computational domain sketch for the channel flow. Red arrows symbolize the wall heat flux.

Throughout this work the x, y and z directions will always refer respectively to the streamwise, the wall-normal and the spanwise directions.

3.2 Modified equations and wall boundary conditions

The filtered Navier-Stokes equations need to be slightly modified to handle the periodic boundary conditions. While the continuity equation remains unchanged, a source term in the form of a streamwise pressure gradient $\mathbf{F} = -\frac{dP_m}{dx}\hat{e}_x$ needs to be added to the momentum equation to drive the flow:

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = F_i - \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}(2\nu_{eff} S_{ij}), \tag{3.4}$$

where $\nu_{eff} = \nu + \nu_{sgs}$. This term is adjusted in time so that the volume averaged velocity (over the whole computational domain) reaches a targeted bulk velocity u_b. In practice this is done using the `meanVelocityForce` `fvoptions` utility.

For a constant mean wall heat flux case, the time averaged temperature evolves linearly in the x direction [20] so that in order to reach a fully developed non-isothermal flow in such condition, a change of variable is introduced:

$$T = x\frac{dT_m}{dx} - \theta, \tag{3.5}$$

where θ is the modified temperature, statistically independent of x, and where the cup mixing temperature is defined as:

$$T_m = \int_0^{2\delta} \overline{u}\overline{T} dy / \int_0^{2\delta} \overline{u} dy.$$

Applying an energy balance on a control volume $dx \times 2\delta \times dz$ yields:

$$\rho u_b c T_m 2\delta dz + 2\overline{q}_w dx dz = \rho u_b c (T_m + dT_m) 2\delta dz \Leftrightarrow \frac{dT_m}{dx} = \frac{\overline{q}_w}{\rho c u_b \delta} = cst.$$

By injecting expression 3.5 into the energy equation 3.3 one obtains the modified energy equation :

$$\frac{\partial \theta}{\partial t} + \frac{\partial \theta u_j}{\partial x_j} = \frac{\partial}{\partial x_j}(\alpha_{eff}\frac{\partial \theta}{\partial x_j}) + u_1 \frac{\overline{q}_w}{\rho c u_b \delta}, \tag{3.6}$$

where a source term compensating the mean energetic flux at the walls appears and where $\alpha_{eff} = \alpha + \alpha_{sgs}$. The subgrid scale heat diffusivity is defined as $\alpha_{sgs} = \nu_{sgs}/Pr_{sgs}$ following the Reynolds analogy. In the present work, the subgrid scale Prandtl number Pr_{sgs} is set to a constant with a value of 0.4 in agreement with [10]. However, Grötzbach et al.[10] showed that for low Prandtl number such as $Pr = 0.025$ and $Pr = 0.01$, a grid sufficiently refined to perform a well resolved LES should be able to capture the smallest scales of the temperature field, those scales being considerably larger then the smallest scales of the velocity field. Hence in that case the energy equation can be solved in a DNS way, thus without a subgrid scale model, by setting $\alpha_{sgs} = 0$.

Regarding the wall boundary conditions, the following ones are applied:

$$\mathbf{u}(y = 0) = \mathbf{u}(y = 2\delta) = 0, \tag{3.7}$$

$$\nu_{sgs}(y = 0) = \nu_{sgs}(y = 2\delta) = 0, \tag{3.8}$$

$$\frac{\partial p}{\partial y}(y = 0) = \frac{\partial p}{\partial y}(y = 2\delta) = 0, \tag{3.9}$$

$$\theta(y = 0) = \theta(y = 2\delta) = 0. \tag{3.10}$$

The last expression (3.10) is the non-fluctuating thermal boundary condition. As explained much more completely in [8], this simplified thermal BC is a good approximation of reality for, by example, an air/steel combination of fluid nature and wall material but for a liquid metal/steel combination it is less valid. Indeed, in that case, the fluctuations in the temperature field significantly penetrate into the solid walls so that a conjugate heat transfer approach (simulating also the conduction within the walls) is required to obtain the details of those fluctuations. However, the study of Tiselj et al. [17] shows that the type of thermal BC applied has an almost negligible impact on the mean temperature profile. Consequently, using the non-fluctuating BC can still be considered as an acceptable simplification.

3.3 Numerical set-up

3.3.1 Mesh

As mentioned earlier, three resolutions are considered for the meshes. Resolution **M2** follows the guidelines for a wall-resolved LES expressed in wall units provided by Menter(2015) [21] while resolutions **M1** and **M3** are respectively a coarsened and a refined version of resolution **M2**. Six orthogonal meshes targeting those resolutions are constructed with a constant stretching ratio applied vertically such that $\Delta x \approx \Delta y$ at the center of the channel. This was preferred to the use of a hyperbolic tangent stretching law because S. Buckingham shows in her thesis [18] that using such a law or applying a constant stretching ratio produces very similar results. The different resolutions and their corresponding pair of mesh sizes are summarized in table 3.3.

Resolution	Δx^+, Δz^+	y^+ (first cell center)	Mesh size	
			$Re_\tau = 395$	$Re_\tau = 640$
M1	60,30	1.5	$42 \times 42 \times 42$	$68 \times 66 \times 68$
M2	40,20	0.95	$64 \times 64 \times 64$	$102 \times 100 \times 102$
M3	30,15	0.95	$85 \times 80 \times 85$	$137 \times 126 \times 137$
DNS [16]	10,5	0.15	$256 \times 192 \times 256$	$512 \times 256 \times 512$

Table 3.3: Grid resolutions and mesh sizes. Note: Kawamura actually performed the $Re_\tau = 640$ DNS on a different computational domain and the mesh size presented in this table was adapted to correspond to a mesh of equivalent resolution but on a $6.4\delta \times 2\delta \times 3.2\delta$ computational domain.

A quality check of the meshes can be made thanks to the DNS data. Indeed, one can compare the grid filter length $\Delta = \sqrt[3]{\Delta x \Delta y \Delta z}$ (cube root of the cell volume) and the Δy spacing to the Taylor microscale. This particular length scale is defined as follows :

$$\lambda = \sqrt{\frac{10\nu k}{\epsilon}}. \tag{3.11}$$

While the Kolmogorov microscale characterizes the smallest scales of turbulence, the Taylor microscale characterizes the scales at which the viscous effects start to significantly impact the turbulent eddy dynamics. As such it is generally interpreted as corresponding to the end of the inertial range and the beginning of the dissipation range. Hence, if a grid has a filter length Δ and a Δy spacing of the same order of magnitude as the Taylor microscale λ, it should be able to cover all the scales of the inertial range. Figure 3.2 shows that, even for resolution **M1**, Δ and Δy are of the same order of magnitude as λ throughout the channel.

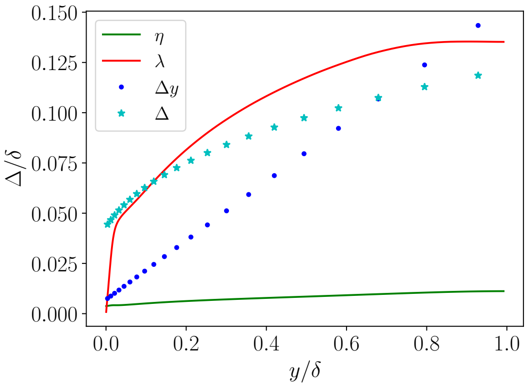


Figure 3.2: Comparison of the Kolmogorov microscale η and the Taylor microscale λ with the mesh filter length Δ and the Δy grid spacing for resolution **M1** and $Re_\tau = 395$.

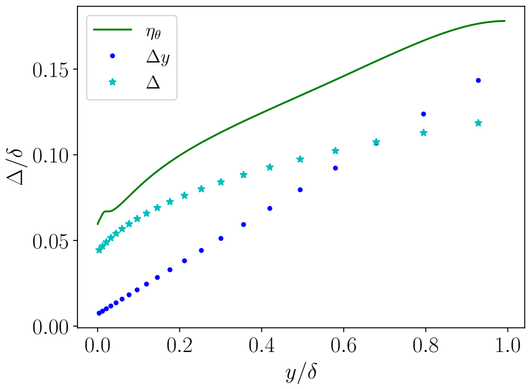


Figure 3.3: Comparison of the Obukhov-Corrsin length scale η_θ with the mesh filter length Δ and the Δy grid spacing for resolution **M1** and $Re_\tau = 395$.

A similar analysis can be performed to check wether the grid is refined enough for the energy equation to be resolved in a DNS way. The smallest turbulent scales in the temperature field are characterized by the Obukhov-Corrsin scale for low Pr flows [22]:

$$\eta_\theta = \frac{\eta}{Pr^{3/4}}, \tag{3.12}$$

where η is the Kolmogorov microscale. Figure 3.3 shows that, considering resolution **M1** for $Pr = 0.025$, the grid is clearly refined enough to capture the smallest scales of the temperature field and this confirms that the energy equation does not require a subgrid scale model in that case. Obviously, this is even truer for higher resolutions or lower Prandtl numbers.

3.3.2 Modified OpenFOAM codes

Some OpenFoam codes need to be slightly modified to comply with the specificities of the studied case (see section 3.2) and to gather the required statistics:

- `pimpleFoam` $\Rightarrow$ `uPimpleFoam` : the chosen unsteady solver for incompressible flows is modified to store the field resulting from the double inner product $\frac{\partial u_i}{\partial x_j}\frac{\partial u_i}{\partial x_j}$ whose mean values are needed for the calculation of the dissipation rate of the turbulent kinetic energy ϵ.
- `scalarTransport` $\Rightarrow$ `Temperature` : the passive scalar solver is modified to include the source term appearing in the energy equation. It also stores two additional fields resulting from the following operations : $u_i\theta$ and $\frac{\partial \theta}{\partial x_i} \cdot \frac{\partial \theta}{\partial x_i}$. The

mean values of those fields are needed respectively for the calculation of $\overline{u_i'\theta'}$ and ϵ_θ. Moreover, the code is adapted so that the specification or not of a parameter in the `controlDict` file trigger on or off the turbulence model for the energy equation.

- `PostSpanwiseAverage` : this user-provided utility allows to average some commonly gathered statistics along a specified direction for structured grids. It is modified to apply the spatial averaging on every statistic gathered in this work.

3.3.3 Main settings and running procedure

Given that OpenFOAM offers a lot of different settings to play with, this section will not be exhaustive. Rather, some particular settings of interest are discussed:

- **Subgrid scale model** : The WALE subgrid scale model [23] is chosen. It is an algebraic eddy-viscosity model so it does not require the resolution of additional transport equations. This is a positive feature regarding the computational cost. Moreover, contrary to the Smagorinsky model, this model does not require wall damping functions because it has been designed to behave correctly in the near wall region. The default OpenFOAM value 0.325 is kept for C_W, the only parameter in this model. Regarding the filter length calculation, the classical cube root of the cell volume is used.
- **Reynolds number** : The kinematic viscosity and the half channel height are kept constant. Hence only the bulk velocity u_b (needed to calculate the source term in the momentum equation) is changed to adapt the case to a different bulk Reynolds number Re_b. Kawawura et al. [16] obtained $Re_b = 14147$ and $Re_b = 24428$ respectively for $Re_\tau = 395$ and $Re_\tau = 640$ for their DNS and those values were used to calculate the bulk velocities to be applied.
- **Schemes** : Second-order Gauss linear schemes and linear interpolation are used for spatial discretization while a second order Crank-Nicolson implicit time scheme is chosen for time integration. An off-centering coefficient $\psi = 0.9$ is applied to this time scheme to ensure numerical stability as advised in the OpenFOAM user guide ($\psi = 0$ corresponds to Euler implicit and $\psi = 1.0$ to pure Crank-Nicolson). Overall, second-order accuracy should be preserved.
- **Solver** : The modified `pimpleFOAM` solver is used to solve the momentum equation and it is parameterized such that it applies the PISO algorithm. This pressure-velocity coupling algorithm first computes a momentum prediction and then successively apply two corrections involving a pressure equation such that the continuity equation is satisfied.
- **Time step** : The time step is fixed so that the maximum CFL number stays lower than 0.6.

The following procedure is used to run the simulations:

1. **Initialisation** : For the lowest resolution, the case is initialised by a RANS simulation to which a 10% random perturbation is applied using the `Randomize` function to trigger turbulence. However, for a higher resolution, it is initialized by mapping the fields obtained with a lower resolution on the finer mesh.
2. **Numerical transient** : The case is decomposed so that the simulation is run in parallel on sixteen cores. There is a transient before fully developed turbulence is achieved. This is monitored by looking at the evolution of the field values at some points in the domain.
3. **Time averaging** : Once the statistical steady state is reached, the `fieldAverage` function is used to time average the fields of interest for at least a 20 flow-through time.
4. **Spanwise averaging** : The time averaged fields are spatially averaged in the z direction using the `postSpanwiseAverage` utility. Then the `singleGraph` function can be used to sample the fields along a line for further post-processing.
5. **Python post-processing** : The data obtained are post-processed in Python with three codes. The first one loads the data and organises them into a database. The second one performs operations on the data, as for example normalisation, and store the results in the database. Finally the third one uses the database to visualize the results.

The most computationally expensive part of this procedure is by far the numerical transient. Since a mean velocity forcing is used, the velocity field relatively quickly reaches a statistical steady state. The same can be said for the temperature fields for low Prandtl numbers because the excess or the lack of heat is very efficiently removed. What make the transient computationaly so expensive are the temperature fields for higher Prandtl numbers. Indeed, from one resolution to another the mean temperature varies significantly for $Pr = 0.71$ and $Pr = 1.0$ so that a lot of excess heat has to be removed at a rather slow rate until statistical equilibrium is reached.

3.4 Results analysis

The objective of the present analysis is threefold : analysing LES grid convergence in OpenFOAM for wall-bounded flows, defining and visualizing the base inputs and ideal outputs of the ANN based turbulent heat transfer model and make some observations about turbulence in channel flows.

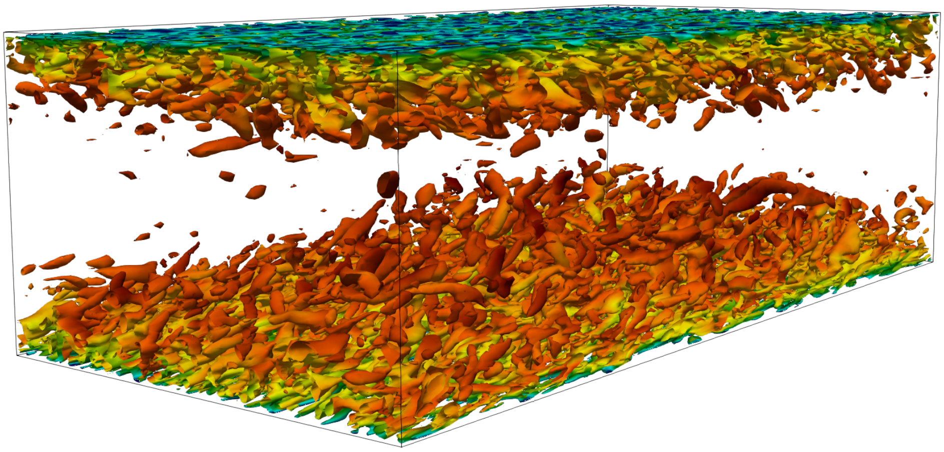


Figure 3.4: Turbulent structures visualisation by the Q criterion for $Re_\tau = 640$ (**M3**) , colorized by velocity magnitude.

Naturally, not all the results constituting the LES database are shown throughout this section but the interested reader can find the profiles not presented in this chapter in appendix 6.1.

3.4.1 Normalisation and mean velocity profiles

Given that the velocity evolves linearly in the laminar sublayer, the friction velocity can be calculated using the first cell values for mean velocity and wall-normal spacing:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} = \sqrt{\nu \left.\frac{d\overline{u}}{dy}\right|_{y=0}} = \sqrt{\nu \frac{\overline{u}[0]}{y[0]}}. \tag{3.13}$$

Based on this friction velocity, a new Reynolds number $Re_\tau = u_\tau \delta / \nu$ is calculated. As can be seen on table 3.4, it does not exactly reach the targeted Re_τ numbers and it is heavily grid resolution dependent. However a monotonic convergence is observed.

Resolution		**M1**	**M2**	**M3**
$Re_\tau = 395$	Calculated Re_τ	346.4	369.6	379.4
	Relative error	12%	6.4%	3.9%
$Re_\tau = 640$	Calculated Re_τ	568.8	604.9	618.4
	Relative error	11%	5.5%	3.4%

Table 3.4: Calculated friction velocity based Reynold numbers.

The results presented in this analysis are wall-normalized (inner scaling), based on the previously calculated friction velocity u_τ. When it is deemed pertinent, results normalized in outer scaling (based on the bulk velocity u_b) or even non-normalized results are also provided. The spatial evolution is either given in wall units or simply normalized by the half channel height δ. A consequence of having a grid dependent friction velocity is that a particular y/δ value for two different resolutions doesn't correspond to a same $y^+ = yu_\tau/\nu$ value.

Since a mean velocity forcing is used, the outer-scale normalized mean velocity profiles obtained for the different resolutions are quite close to each other (Fig. 3.5 and 3.7). However the velocity gradients differ throughout the channel.

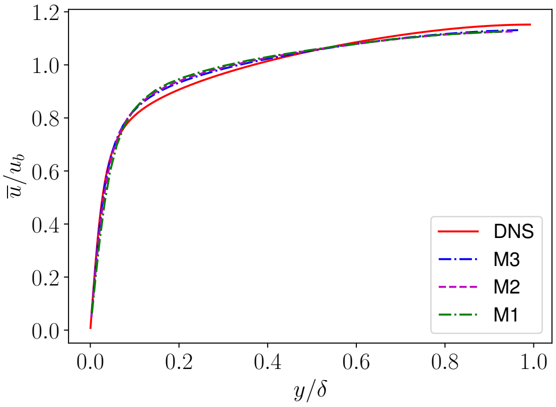


Figure 3.5: Outer-scale normalized mean velocity profiles for $Re_\tau = 395$.

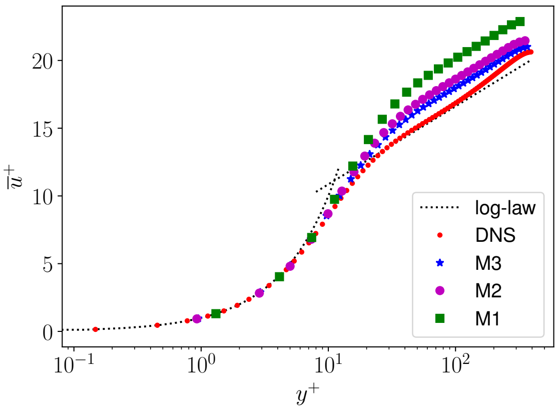


Figure 3.6: Inner-scale normalized mean velocity profiles for $Re_\tau = 395$.

In inner scaling it is interesting to compare the velocity profiles with the well known law of the wall :

$$u^+ = y^+ \text{ for } y^+ < 5, \tag{3.14}$$

$$u^+ = \frac{1}{\kappa} ln(y^+) + B \text{ for } y^+ > 30, \tag{3.15}$$

where $\kappa \approx 0.41$ is the von Karman constant and $B \approx 5.0$. This comparison reveals a significant mean velocity overshoot in the log region whose amplitude diminishes with grid refinement (Fig. 3.6 and 3.8). However, the $1/\kappa$ slope in that region is well captured by every resolution. This suggests that the buffer layer is where the grid resolution matters particularly.

Those results can be compared with the literature. Duponcheel et al.[24] performed LES in channel flows for $Re_\tau = 590$, $Re_\tau = 640$ and $Re_\tau = 2000$. They obtained significantly better results, very close to DNS, despite using a grid resolution close to resolution **M2**. However, they performed their simulations with a code using a fourth order finite difference scheme for the spatial discretization with which OpenFOAM can't compete regarding accuracy. Indeed, as is generally the case for finite volume method implementations, OpenFOAM is only second order accurate at best. Hence the numerical errors dominate the subgrid-scale modelling error here [25]. In fact, the velocity overprediction in the log region is commonly observed when using a finite volume based solver [18, 26, 19].

Moreover, extensive studies [27, 28] show that the LES grid convergence using a FV solver is not a trivial matter. Rezaeiravesh et al. [28] concluded that good results can be obtained overall using $\Delta x^+ \leq 18$ and $\Delta z^+ \leq 12$ for the grid resolution. This is not that far from resolution **M3** but is getting close to DNS grid requirements and is computationally expensive. It seems however that better results can be obtained at a more reasonable computational cost using isotropic mesh refinement near the walls to better capture the small energetic eddies there [29, 30, 31] or by modifying the solver to minimize numerical dissipation and using an advanced anisotropic subgrid scale model [32]. So there is definitely room for improvements, but this is outside the scope of this work.

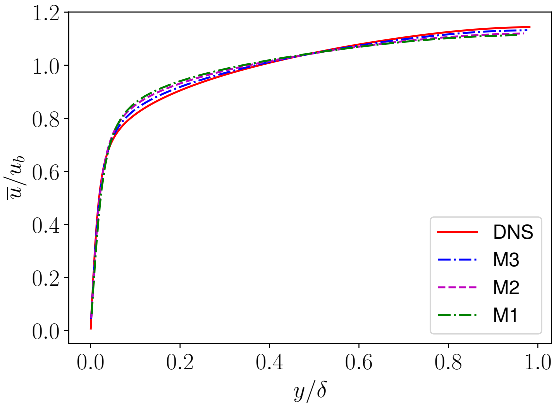


Figure 3.7: Outer-scale normalized mean velocity profiles for $Re_\tau = 640$.

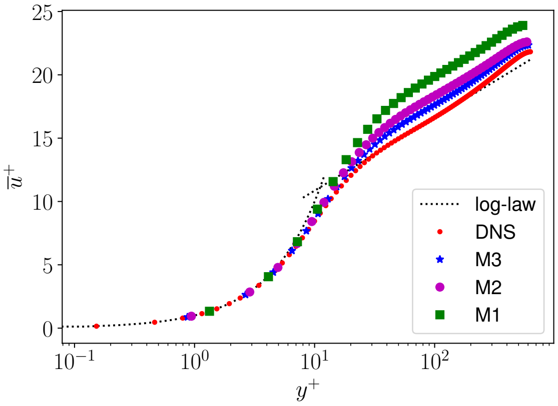


Figure 3.8: Inner-scale normalized mean velocity profiles for $Re_\tau = 640$.

3.4.2 Reynolds stress tensor and Turbulent kinetic energy

The streamwise velocity fluctuations (Fig. 3.9) are overpredicted in the near wall region and underpredicted in the outer region whereas the spanwise (Fig 3.11) and wall-normal (Fig 3.10) velocity fluctuations are underpredicted throughout all the channel. Just like the underprediction of skin friction and the velocity overshoot, this behaviour is commonly observed in the literature. The slope difference of the $\overline{u'v'}^+$ profile (Fig 3.12) in the outer region is directly linked to the skin friction underprediction.

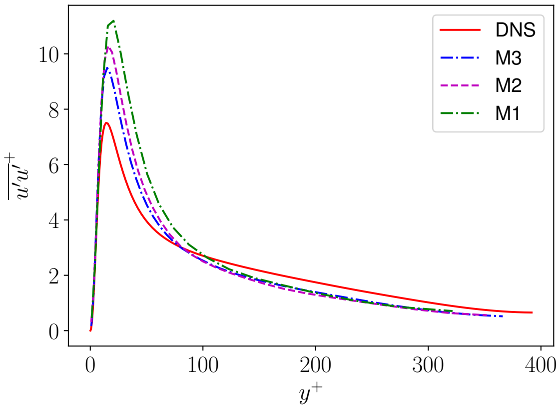


Figure 3.9: Streamwise velocity fluctuations ($Re_\tau = 640$).

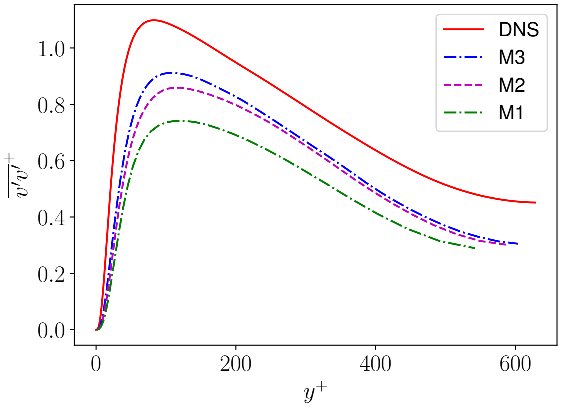


Figure 3.10: Wall-normal velocity fluctuations ($Re_\tau = 640$).

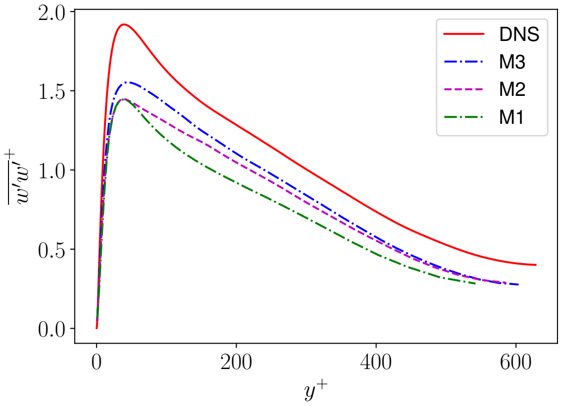


Figure 3.11: Spanwise velocity fluctuations ($Re_\tau = 640$).

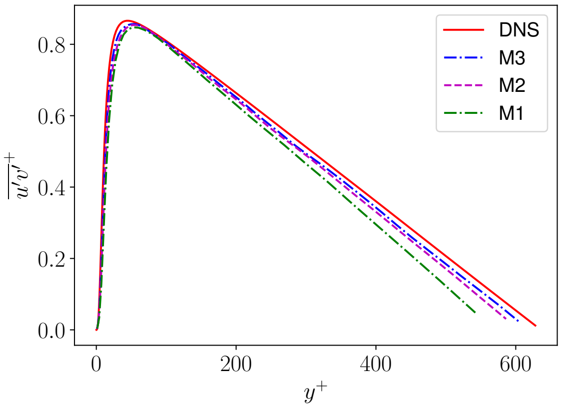


Figure 3.12: Streamwise and spanwise velocity covariance ($Re_\tau = 640$).

Based on the velocity fluctuations, the resolved turbulent kinetic energy can be calculated :

$$k_r = \frac{\overline{u'u'} + \overline{v'v'} + \overline{w'w'}}{2}, \tag{3.16}$$

while, according to the OpenFOAM implementation, the subgrid scales kinetic energy is calculated as:

$$k_{sgs} = (\frac{\nu_{sgs}}{C_k \Delta})^2, \tag{3.17}$$

where $C_k = 0.094$ is a constant. The total TKE is simply $k = k_r + k_{sgs}$ (fig. 3.13). One can observe that while the peak TKE overprediction is greatly reduced by increasing resolution, the convergence in the outer region is rather slow. Figure 3.14 shows the stress decomposition with $\tau_m^+ = \frac{d\overline{u}^+}{dy^+}$, the molecular stress dominant at the wall, $\tau_t^+ = \overline{u'v'}^+$, the turbulent stress dominant in most of the channel and $\tau_{sgs}^+ = \nu_{sgs}^+ \frac{d\overline{u}^+}{dy^+}$, the subgrid scales contribution. The total stress $\tau_{tot}^+ = \tau_m^+ + \tau_t^+ + \tau_{sgs}^+$ should be linear.

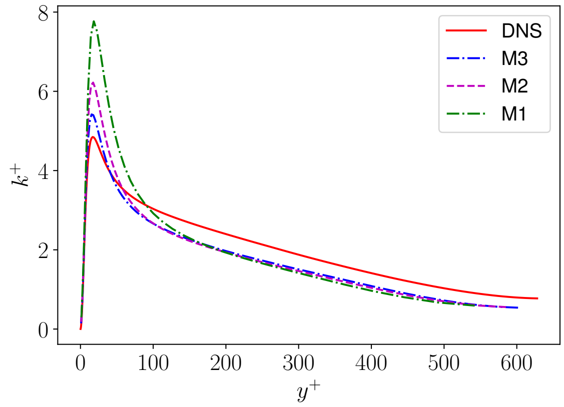


Figure 3.13: Turbulent kinetic energy ($Re_\tau = 640$).

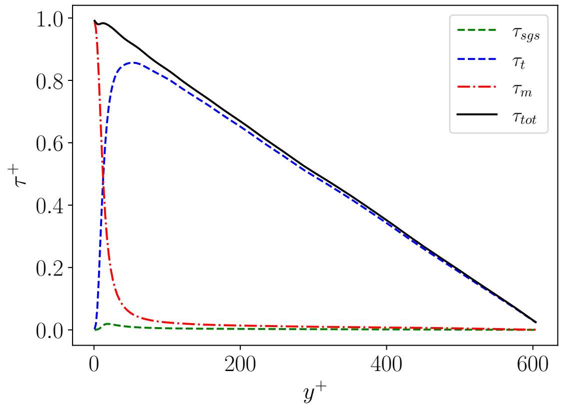


Figure 3.14: Stress decomposition ($Re_\tau = 640$, **M3**).

3.4.3 Dissipation rate of turbulent kinetic energy

The following simplified definitions are used for the calculation of the resolved and subgrid contribution of the dissipation rate of the TKE:

$$\epsilon_r = \nu \overline{\frac{\partial u'_i}{\partial x_j} \cdot \frac{\partial u'_i}{\partial x_j}} = \nu\left(\overline{\frac{\partial u_i}{\partial x_j} \cdot \frac{\partial u_i}{\partial x_j}} - \frac{\partial \overline{u_i}}{\partial x_j} \cdot \frac{\partial \overline{u_i}}{\partial x_j}\right), \tag{3.18}$$

$$\epsilon_{sgs} = \overline{\nu_{sgs}} \overline{\frac{\partial u'_i}{\partial x_j} \cdot \frac{\partial u'_i}{\partial x_j}} = \overline{\nu_{sgs}}\left(\overline{\frac{\partial u_i}{\partial x_j} \cdot \frac{\partial u_i}{\partial x_j}} - \frac{\partial \overline{u_i}}{\partial x_j} \cdot \frac{\partial \overline{u_i}}{\partial x_j}\right). \tag{3.19}$$

The wall normalized dissipation rate profiles ($\epsilon^+ = \nu\epsilon/u_\tau^4$) are presented on figures 3.15 and 3.16. The overall shapes of the profiles are satisfactory but the relative error between DNS and LES is high. This has major consequences on the ANN based turbulent heat transfer model because ϵ is intensively used to normalize the invariant basis (table 2.1). Furthermore, the grid convergence seems to be rather slow. The turbulent dissipation rate is a matter of small scales because those are the scales that dissipate the TKE. By definition, the smallest scales are modelled for an LES so that the subgrid contributions ϵ_{sgs} should not be negligible. Hence a possible explanation for the high relative error observed is insufficiently accurate subgrid scales modelling . On the other hand, let us not forget that the high numerical dissipation in those simulations contributes notably to dissipate TKE, so that the profiles presented here do not give a complete picture of the true dissipation rate.

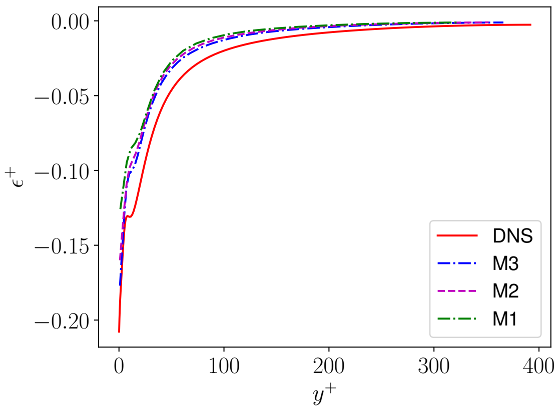


Figure 3.15: Dissipation rate of the TKE ($Re_\tau = 395$).

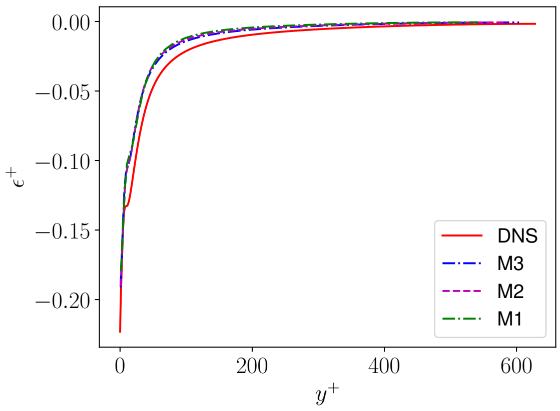


Figure 3.16: Dissipation rate of the TKE ($Re_\tau = 640$).

The rigorous definitions for the resolved and subgrid scales part of the turbulent dissipation rate are actually:

$$\epsilon_r = 2\nu\overline{S'_{ij}\cdot S'_{ij}} = 2\nu(\overline{S_{ij}\cdot S_{ij}} - \overline{S_{ij}},\cdot\overline{S_{ij}}), \tag{3.20}$$

$$\epsilon_{sgs} = 2\overline{\nu_{sgs}S'_{ij}\cdot S'_{ij}} = 2(\overline{\nu_{sgs}S_{ij}\cdot S_{ij}} - \overline{\nu_{sgs}S_{ij}}\cdot\overline{S_{ij}}). \tag{3.21}$$

Expressions 3.18 and 3.20 are strictly equivalent for homogeneous isotropic turbulence, which is not the case here. Figure 3.17 compares the turbulent dissipation rate profiles obtained with the different sets of definitions. This comparison is made for the case for which using one set of definitions or the other should have the biggest overall impact : using the coarsest resolution for which the subgrid scales contribution is the most important and for $Re_\tau = 395$ because turbulence is more anisotropic for lower Reynolds numbers. One can observe that the results obtained with the two different definitions for the resolved part are indistinguishable while the total turbulent dissipation rate is little affected by the variation of the subgrid scales contribution. Hence the simplified expressions 3.18 and 3.19 are considered acceptable for the present work.

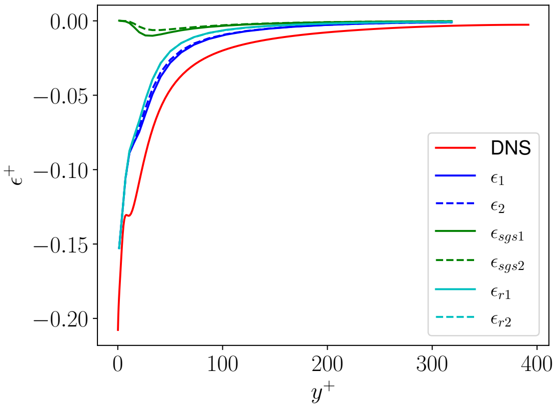


Figure 3.17: Influence of the dissipation rate definition, subscript "1" and "2" denote respectively the simplified and rigorous set of definitions

3.4.4 Instantaneous velocity field

Figure 3.18 gives a picture of the turbulent structures inside the channel. It should be noted that a choice was made to present those results in a non-interpolated way, so as to show how well the grid truly captures the eddies. Hence the fields look pixelated.

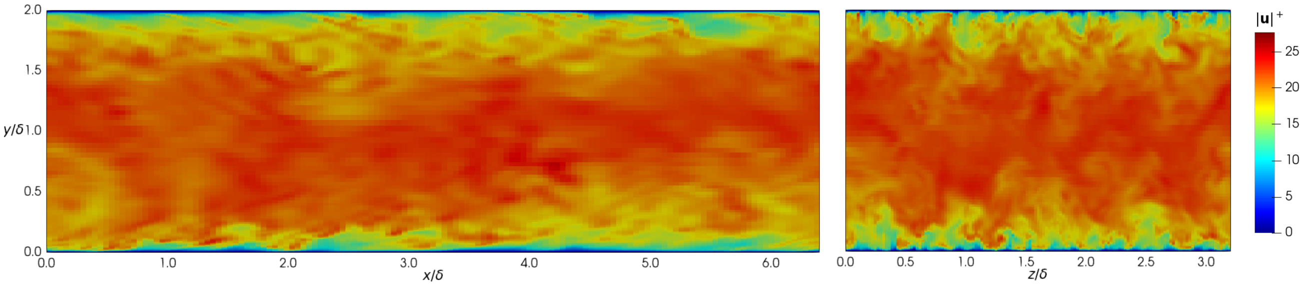


Figure 3.18: Streamwise (left) and spanwise (right) instantaneous velocity field magnitude ($Re_\tau = 640$, **M3**)

Figures 3.19 and 3.20 show the velocity field magnitude normalized by the local mean velocity $\overline{u}$ for a wall-normal plane situated respectively at the center of the channel and at a distance corresponding to the TKE peak value (those particular pictures are however shown interpolated). Comparing those two figures highlights how different the turbulent structures are in the near wall region and in the outer region. At the channel center, the flow is characterized by close to isotropic patterns and large length scales whereas in the near wall region, the eddies are highly anisotropic with very small length scales especially in the spanwise direction.

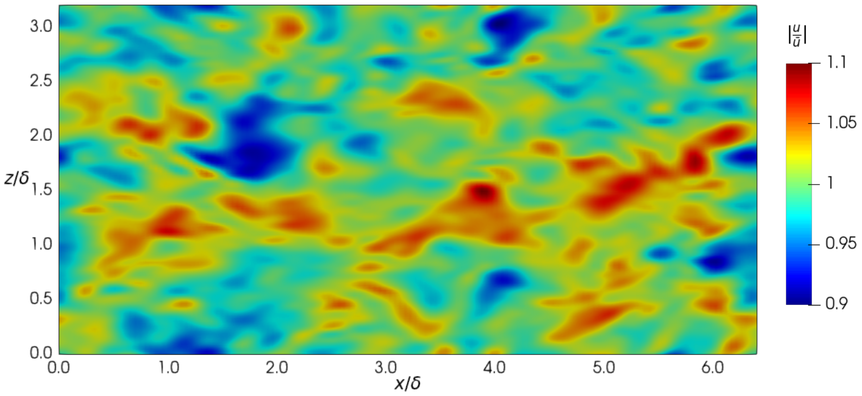


Figure 3.19: Velocity field magnitude normalized by local mean velocity for $y/\delta = 1$ ($Re_\tau = 640$, **M3**)

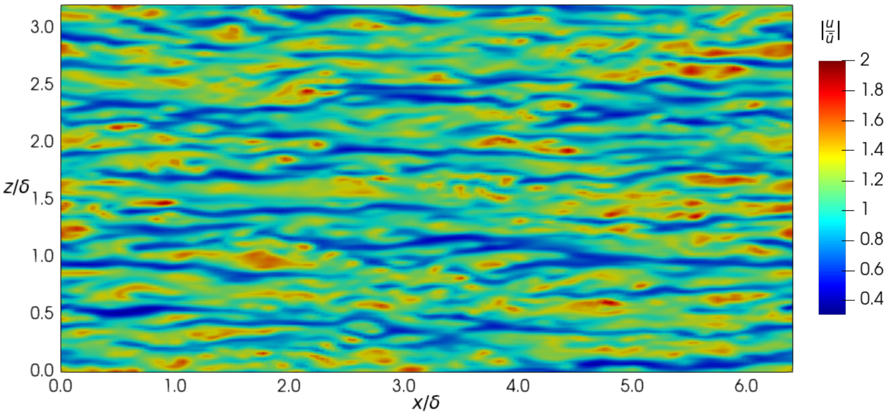


Figure 3.20: Velocity field magnitude normalized by local mean velocity for $y/\delta = 0.05$ ($Re_\tau = 640$, **M3**)

The Kolmogorov microscale (fig. 3.21) seems to give a qualitatively good view of the variation of length scale across the channel. One can note that the LES can predict the correct order of magnitude for this length scale.

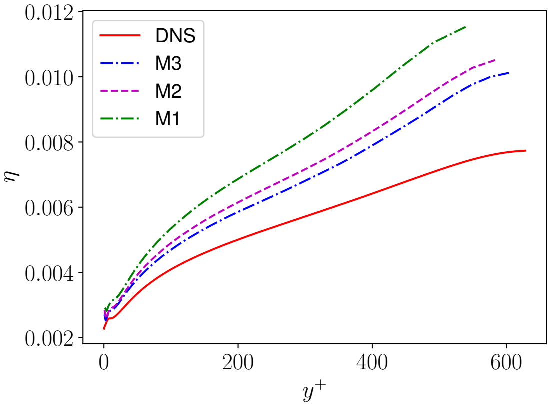


Figure 3.21: Kolmogorov microscale ($Re_\tau = 640$)

3.4.5 Temperature fields

The temperature fields are wall normalized by the friction temperature $T_\tau = \frac{q_w}{\rho c u_\tau}$. Moreover, the thermal profiles are presented as functions of $Pr \cdot y^+$. Figures 3.22 and 3.24 present the mean temperature profiles for $Pr = 0.025$ and $Pr = 0.01$. The results follow DNS for the first half of the graphs and then diverge more or less depending on the resolution. However, the normalisation in wall units might be a bit misleading when assessing the general quality of those profiles. Indeed, the non-normalized temperature profiles for $Pr = 0.01$ (fig. 3.23) match the DNS really well, nearly regardless of the resolution, so it should be considered as close to fully converged. This is not as true for $Pr = 0.025$ but the non-normalized profiles are still very close to DNS (fig. 3.25).

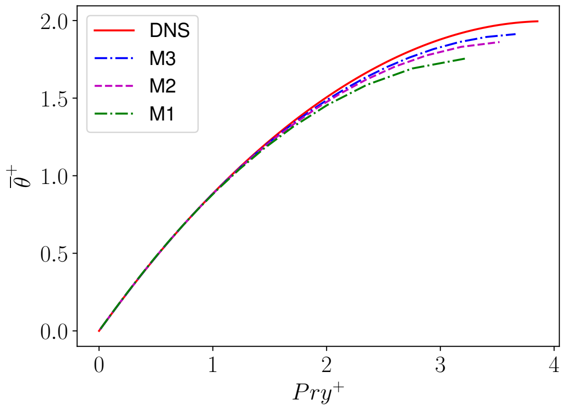


Figure 3.22: Mean temperature profiles for $Re_\tau = 395$ and $Pr = 0.01$.

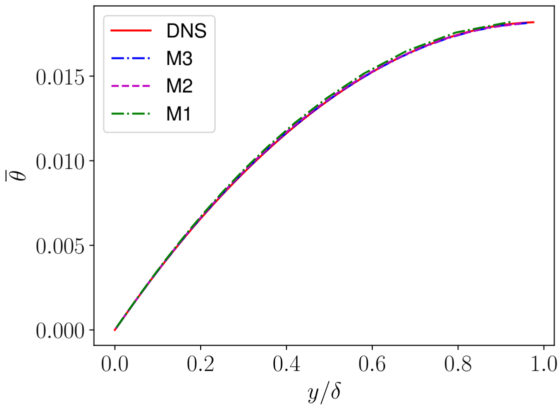


Figure 3.23: Dimensional mean temperature profiles for $Re_\tau = 395$ and $Pr = 0.01$.

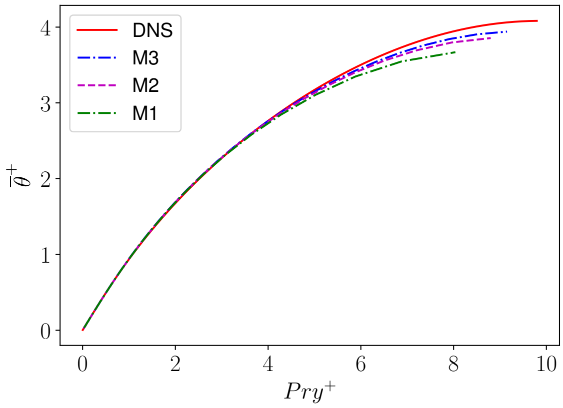


Figure 3.24: Mean temperature profiles for $Re_\tau = 395$ and $Pr = 0.025$.

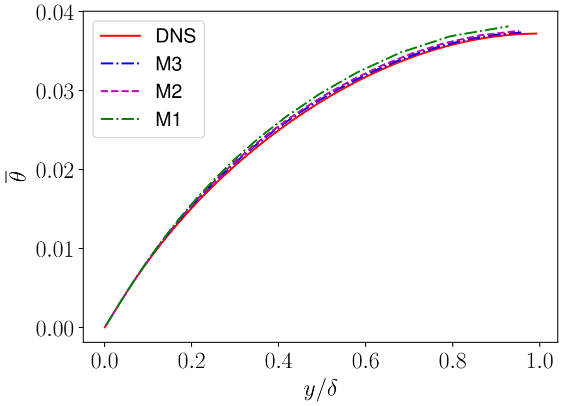


Figure 3.25: Dimensional mean temperature profiles for $Re_\tau = 395$ and $Pr = 0.025$.

For $Pr = 0.71$, the way the results converge is very Reynolds number dependent as can be seen on figures 3.26, 3.27, 3.28 and 3.29. For $Re_\tau = 640$ the wall normalized results behave in a similar way to the velocity profiles but the non-normalized profiles exhibits a highly resolution dependent bulk temperature overshoot. This suggests that a very well resolved velocity field is needed to correctly predict the bulk temperature. For $Re_\tau = 395$, the bulk temperature overshoot is less resolution dependent and the wall-normalized profiles exhibit a peculiar convergence.

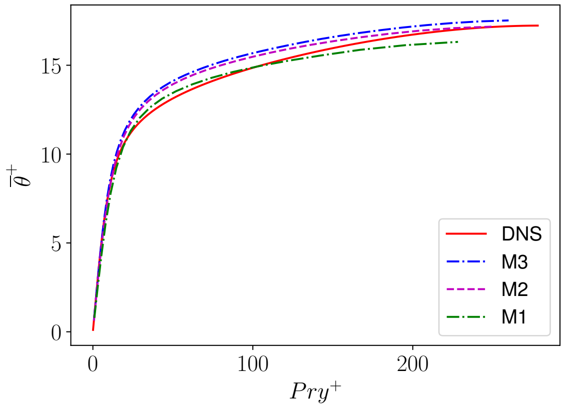


Figure 3.26: Mean temperature profiles for $Re_\tau = 395$ and $Pr = 0.71$.

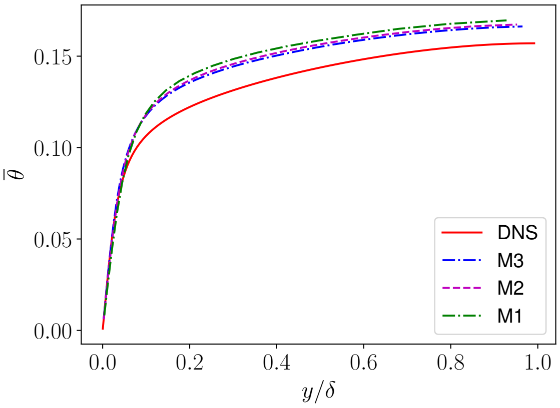


Figure 3.27: Dimensional mean temperature profiles for $Re_\tau = 395$ and $Pr = 0.71$.

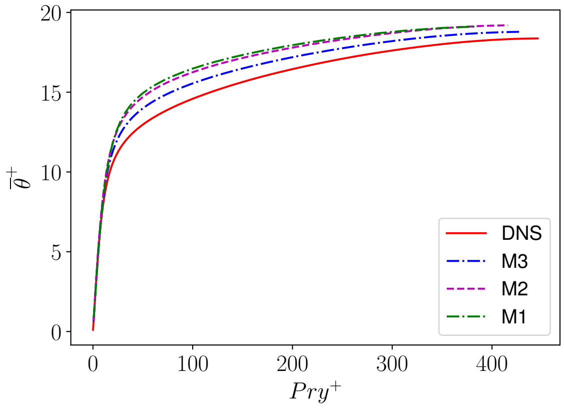


Figure 3.28: Mean temperature profiles for $Re_\tau = 640$ and $Pr = 0.71$.

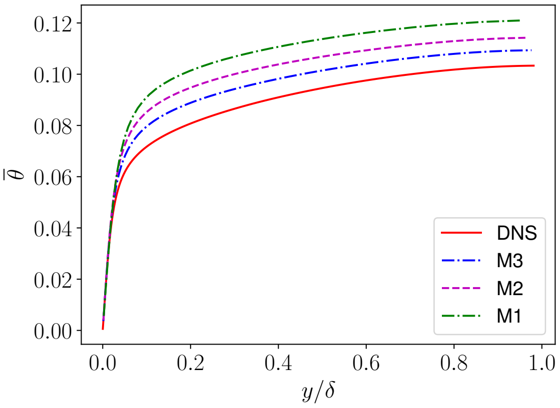


Figure 3.29: Dimensional mean temperature profiles for $Re_\tau = 640$ and $Pr = 0.71$.

Overall, one can conclude that the results look better for low Prandtl numbers when looking at the dimensional profiles whereas for close to unity Prandtl numbers, the results look better when examining the wall-normalized profiles.

No DNS validation data are available for $Pr = 1.0$ but an interesting analysis can still be performed. For this particular Prandtl number, the wall-normalized mean velocity and temperature profiles should match. Examining figures 3.30-

3.35 reveals that the extent to which $\overline{u}^+$ and $\overline{\theta}^+$ match is resolution and Reynolds number dependent. Indeed, an **M3** resolution seems to be needed to obtain an acceptable matching for $Re_\tau = 395$ whereas an **M2** resolution seems to be sufficient for $Re_\tau = 640$. Part of the poor matching for the lowest resolution may be attributed to the poor prediction of the friction velocity for this resolution because the velocity is normalized by u_τ and the temperature by T_τ which is inversely proportional to u_τ.

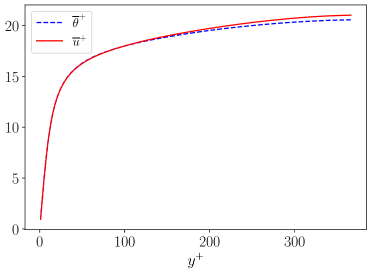


Figure 3.30: $Re_\tau = 395$, **M3**

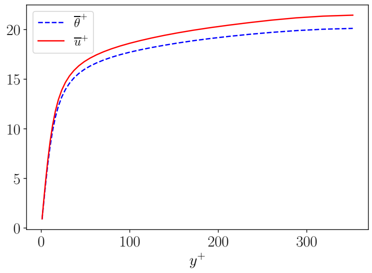


Figure 3.31: $Re_\tau = 395$, **M2**

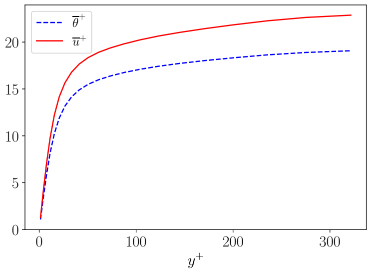


Figure 3.32: $Re_\tau = 395$, **M1**

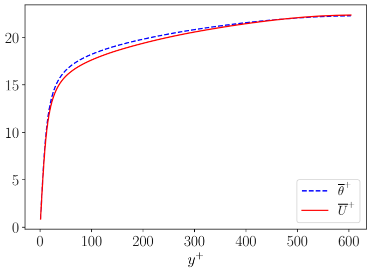


Figure 3.33: $Re_\tau = 640$, **M3**

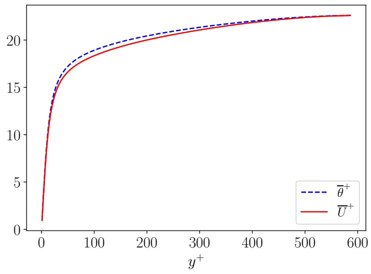


Figure 3.34: $Re_\tau = 640$, **M2**

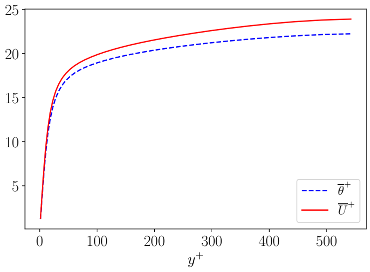


Figure 3.35: $Re_\tau = 640$, **M1**

The instantaneous temperature field for $Pr = 0.71$ exhibits a behaviour very similar to the velocity field as can be seen on figure 3.36. This is the kind of Prandtl number for which the Reynolds analogy holds pretty well. However, the instantaneous temperature field for $Pr = 0.025$ (Figure 3.37) shows a significantly different behaviour. Indeed, it is characterized by much larger length scales and a much smoother field. This illustrates well why, for low Prandtl numbers, a grid can be insufficiently refined to resolve all the scales of the velocity field while still being sufficiently refined at the same time to resolve the complete turbulence spectrum for the temperature field.

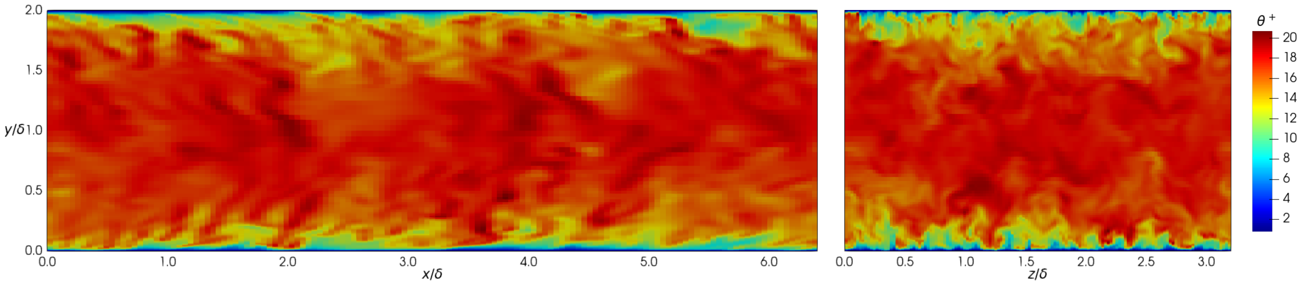


Figure 3.36: Streamwise (left) and spanwise (right) instantaneous temperature field for $Pr = 0.71$ ($Re_\tau = 640$, **M3**)

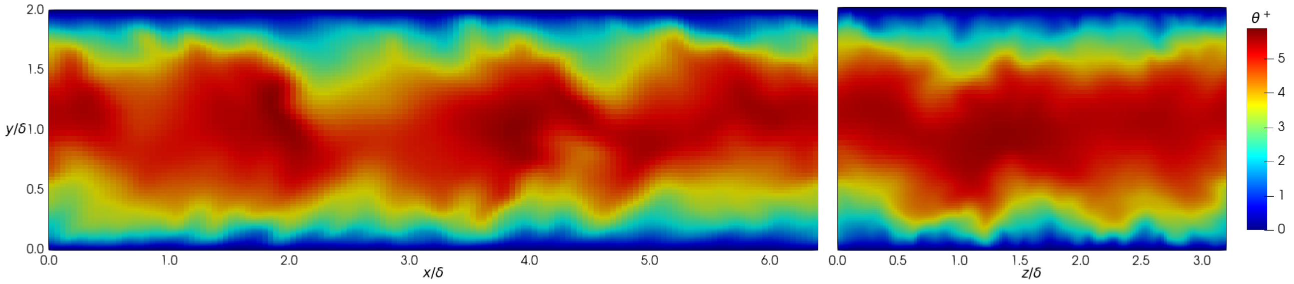


Figure 3.37: Streamwise (left) and spanwise (right) instantaneous temperature field for $Pr = 0.025$ ($Re_\tau = 640$, **M3**)

3.4.6 Turbulent heat transfer

The turbulent heat flux is the main focus of this work and deserves a particular attention. It is calculated as follows :

$$\overline{\mathbf{u}'\theta'} = \overline{\mathbf{u}\theta} - \overline{\mathbf{u}}\overline{\theta}. \tag{3.22}$$

The wall normalized THF, $\overline{\mathbf{u}'\theta'}^+ = \overline{\mathbf{u}'\theta'}/u_\tau T_\tau$, is actually independent of the friction velocity. The subgrid scale contribution to the wall normal component, $\overline{v'\theta'}$, is:

$$q^+_{sgs} = \alpha^+_{sgs}\frac{d\overline{\theta}^+}{dy^+}. \tag{3.23}$$

Let's remember that for low Prandtl numbers, this contribution is equal to zero. For $Pr = 0.71$, the components $\overline{u'\theta'}$ (fig. 3.38 and fig. 3.40) and $\overline{v'\theta'}$ (fig. 3.39 and fig. 3.41) behave similarly respectively to the components $\overline{u'u'}$ and $\overline{u'v'}$ of the Reynolds stress tensor. For $Pr = 0.025$ (fig. 3.43, 3.42,3.44,3.45), the peaks of the profiles are much less sharp and are shifted away from the wall. The way the THF converges is very Reynolds number dependent. The $\overline{v'\theta'}^+$ profiles indeed behave significantly better for $Re_\tau = 640$ than for $Re_\tau = 395$. The streamwise THF $\overline{u'\theta'}^+$ for $Pr = 0.025$ is systematically underpredicted and while it seems to monotically converge toward DNS values for $Re_\tau = 395$, it seems to converge to lower values for $Re_\tau = 640$. All of this holds true as well for $Pr = 0.01$ (see appendix). The grids are refined enough to cover the whole thermal turbulence spectrum for low Prandtl numbers, so one can only assume the non negligible errors observed here are mainly due to the errors from the velocity field.

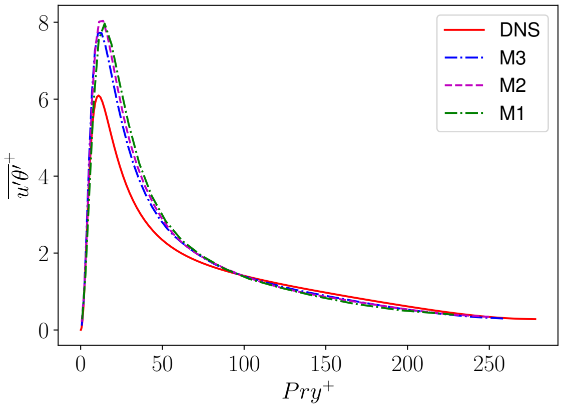


Figure 3.38: Streamwise THF for $Pr = 0.71$ and $Re_\tau = 395$

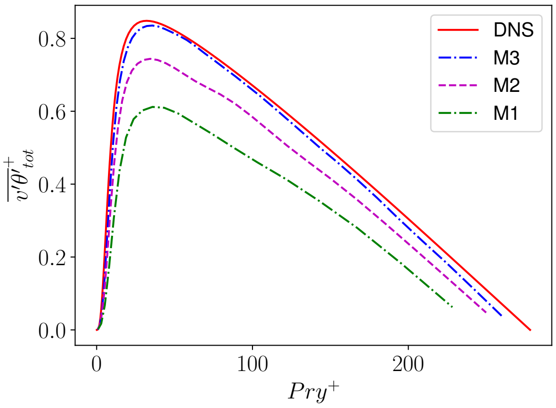


Figure 3.39: Wall-normal THF for $Pr = 0.71$ and $Re_\tau = 395$

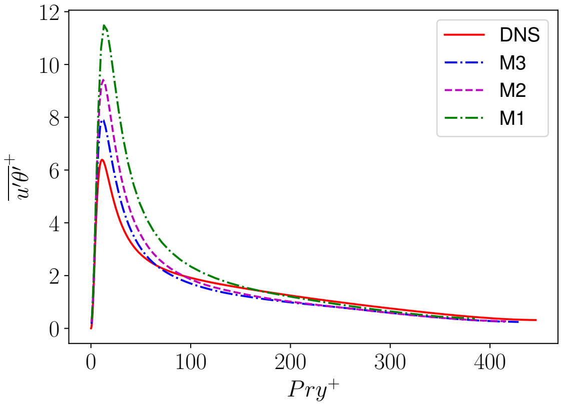


Figure 3.40: Streamwise THF for $Pr = 0.71$ and $Re_\tau = 640$

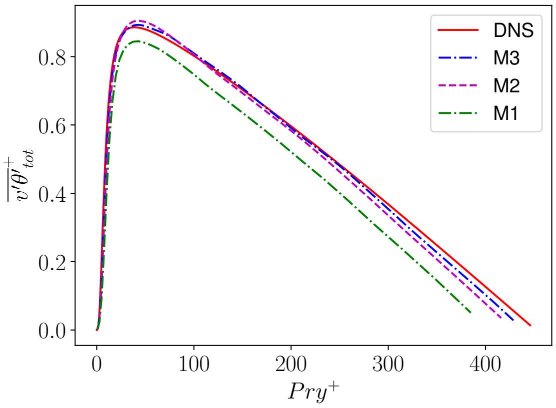


Figure 3.41: Wall-normal THF for $Pr = 0.71$ and $Re_\tau = 640$

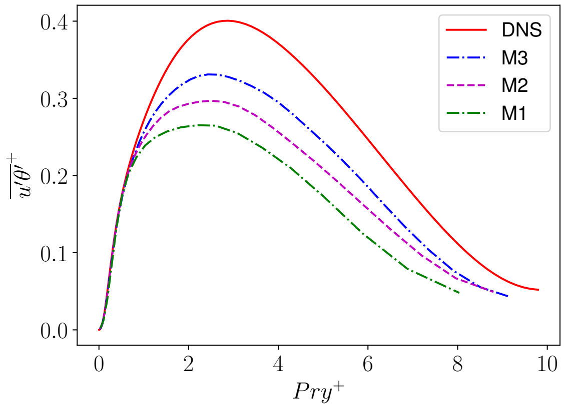


Figure 3.42: Streamwise THF for $Pr = 0.025$ and $Re_\tau = 395$

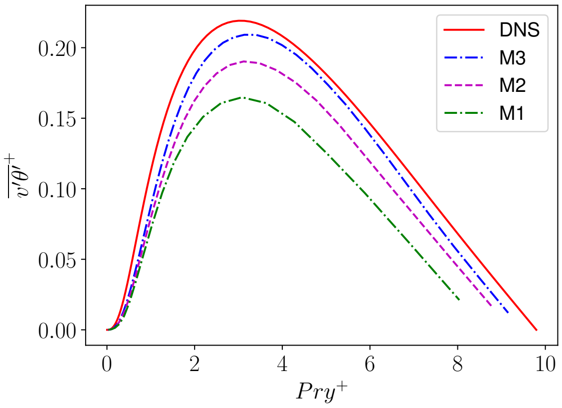


Figure 3.43: Wall-normal THF for $Pr = 0.025$ and $Re_\tau = 395$

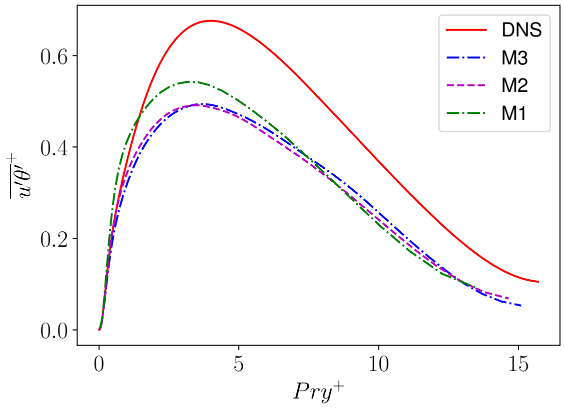


Figure 3.44: Streamwise THF for $Pr = 0.025$ and $Re_\tau = 640$

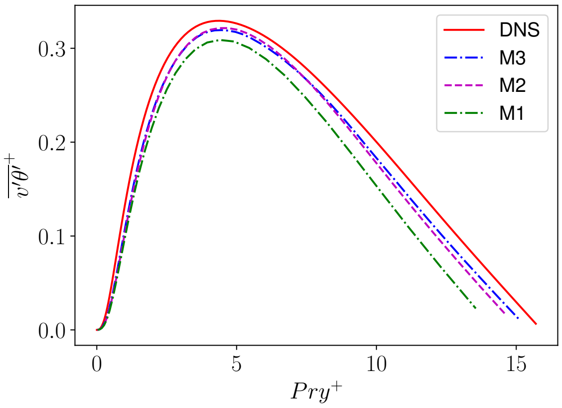


Figure 3.45: Wall-normal THF for $Pr = 0.025$ and $Re_\tau = 640$

Comparing the heat flux decomposition on figures 3.46 and 3.47 illustrates well the peculiarity of heat transfer in liquid metals. The molecular heat flux is calculated as $q_m^+ = \frac{1}{Pr}\frac{d\overline{\theta}^+}{dy^+}$, the turbulent heat flux is $q_t = \overline{v'\theta'}^+$, the subgrid scale contribution q_{sgs}^+ is defined above and the sum yields the total wall-normal heat flux : $q_{tot}^+ = q_m^+ + q_t^+ + q_{sgs}^+$. For $Pr = 0.71$, the heat flux decomposition is actually very similar to the stress decomposition : the turbulent heat transfer mostly dominates, just like the turbulent stress mostly dominates in the stress decomposition. For $Pr = 0.025$ however, the molecular conduction dominates or is of the same order of magnitude as the turbulent heat transfer. For higher Reynolds numbers, the turbulent heat flux would take more importance but the molecular conduction would still play a much more important role than if the Prandtl number was close to unity. On the other hand, it might seems a bit surprising that for $Re_\tau = 395$ and $Pr = 0.01$ or $Pr = 0.025$, the dimensional mean temperature is rather well predicted for every resolution (fig. 3.25) despite a significant underprediction of the THF (fig 3.43). This is at least partly explained by the fact that for such low Reynolds and Prandtl numbers, the turbulent heat flux is actually a small part of the total heat flux, so that the temperature profile is weakly sensitive to THF. This is less and less true as the Reynolds number increases but at the same time it seems that the THF is also better predicted for higher Reynolds numbers.

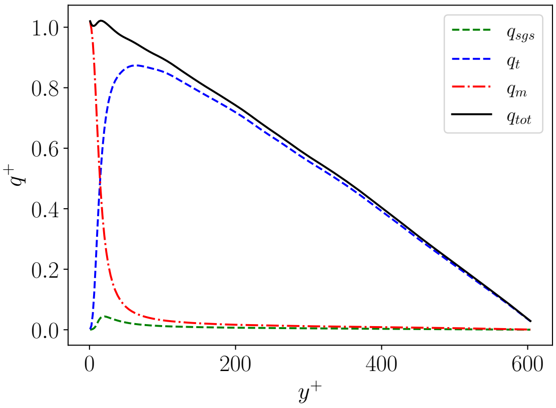


Figure 3.46: Heat flux decomposition for $Pr = 0.71$, ($Re_\tau = 640$, **M3**)

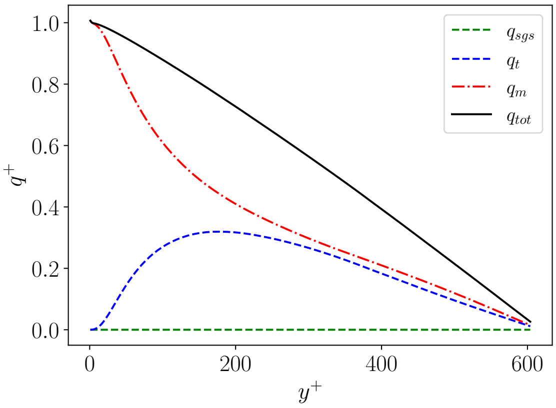


Figure 3.47: Heat flux decomposition for $Pr = 0.025$, ($Re_\tau = 640$, **M3**)

3.4.7 Temperature variance and dissipation rate of the temperature variance

The temperature variance,

$$k_\theta = \overline{\theta'\theta'}/2, \tag{3.24}$$

is shown on figures 3.48 and 3.49 for different Prandtl numbers. The resolved part of the dissipation rate of the temperature variance is calculated as follows :

$$\epsilon_{\theta_r} = \frac{\nu}{Pr}\overline{\frac{\partial\theta}{\partial x_i}' \cdot \frac{\partial\theta}{\partial x_i}'} = \frac{\nu}{Pr}(\overline{\frac{\partial\theta}{\partial x_i} \cdot \frac{\partial\theta}{\partial x_i}} - \overline{\frac{\partial\theta}{\partial x_i}} \cdot \overline{\frac{\partial\theta}{\partial x_i}}), \tag{3.25}$$

while the subgrid scale contribution is calculated using a simplification similar to the one used for the dissipation rate of the TKE:

$$\epsilon_{\theta_{sgs}} = \frac{\nu_{sgs}}{Pr}\overline{\frac{\partial\theta}{\partial x_i}' \cdot \frac{\partial\theta}{\partial x_i}'} = \frac{\nu_{sgs}}{Pr}(\overline{\frac{\partial\theta}{\partial x_i} \cdot \frac{\partial\theta}{\partial x_i}} - \overline{\frac{\partial\theta}{\partial x_i}} \cdot \overline{\frac{\partial\theta}{\partial x_i}}). \tag{3.26}$$

The wall normalized total dissipation rate of the temperature variance, $\epsilon_\theta^+ = \nu\epsilon_\theta/T_\tau^2 u_\tau^2$ is also shown for different Prandtl numbers on figure 3.51 and 3.50.

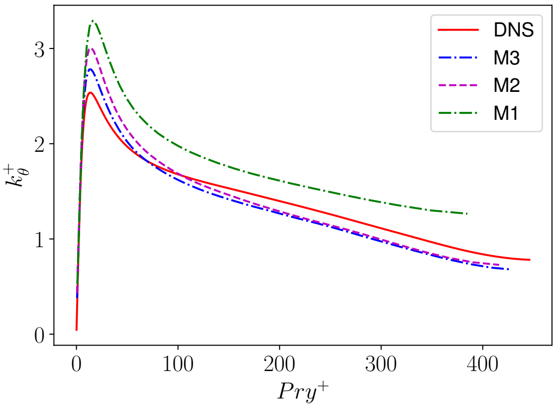


Figure 3.48: Temperature variance for $Pr = 0.71$ and $Re_\tau = 640$

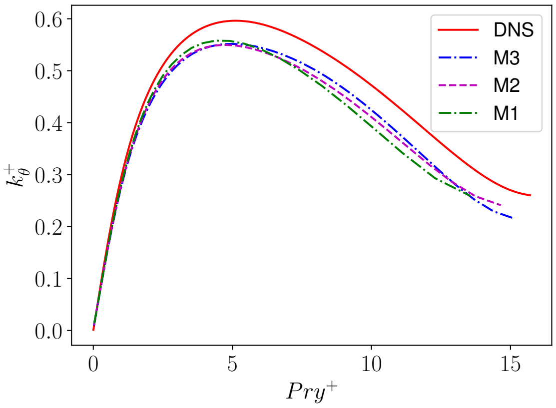


Figure 3.49: Temperature variance for $Pr = 0.025$ and $Re_\tau = 640$

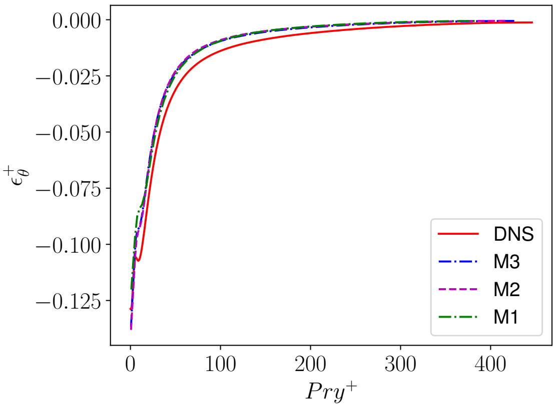


Figure 3.50: Dissipation rate of the temperature variance for $Pr = 0.71$ and $Re_\tau = 640$

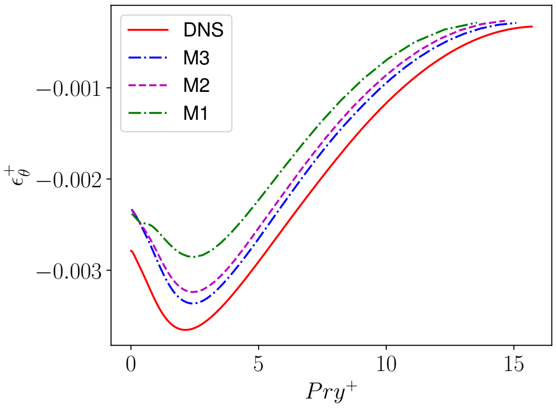


Figure 3.51: Dissipation rate of the temperature variance for $Pr = 0.025$ and $Re_\tau = 640$

3.4.8 Turbulent eddy-viscosity and eddy-diffusivity

The turbulent eddy-viscosity and eddy-diffusivity can respectively be recovered with the gathered statistics as follows :

$$\nu_t^+ = \frac{\overline{u'v'}^+}{\frac{d\overline{u}^+}{dy^+}} + \nu_{sgs}^+, \tag{3.27}$$

$$\alpha_t^+ = \frac{\overline{v'\theta'}^+}{\frac{d\overline{\theta}^+}{dy^+}} + \alpha_{sgs}^+. \tag{3.28}$$

It is more difficult to make those particular statistics converge. Indeed, the profiles obtained (fig. 3.52 to 3.57) tend to not be very smooth except for the turbulent eddy-diffusivity for $Pr = 0.025$. Moreover, one does not observe a monotonic convergence here.

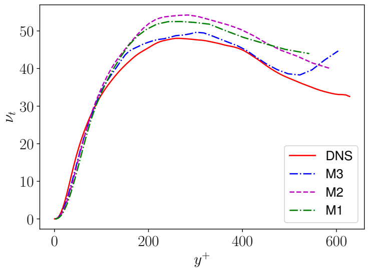


Figure 3.52: ν_t^+ for $Re_\tau = 640$

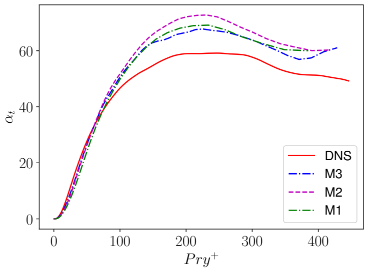


Figure 3.53: α_t^+ for $Re_\tau = 640$ and $Pr = 0.71$.

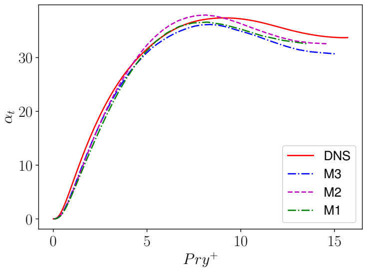


Figure 3.54: α_t^+ for $Re_\tau = 640$ and $Pr = 0.025$.

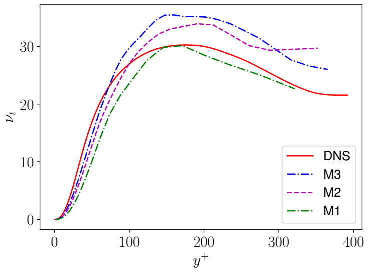


Figure 3.55: ν_t^+ for $Re_\tau = 395$.

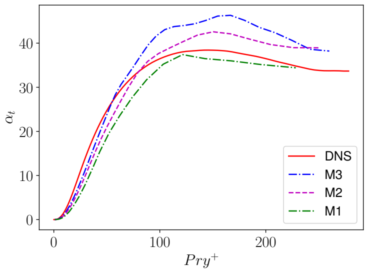


Figure 3.56: α_t^+ for $Re_\tau = 395$ and $Pr = 0.71$.

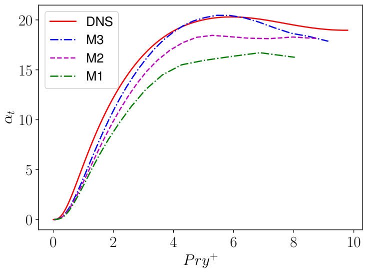


Figure 3.57: α_t^+ for $Re_\tau = 395$ and $Pr = 0.025$.

Chapter 4

Artificial neural network training using LES

This chapter aims at exploring the option of using LES to train the neural network which is part of the THF model reviewed in chapter 2 using the LES data of channel flows generated through the process described in chapter 3. This is done through several levels of analysis which will also give an insight of the model robustness toward its inputs variations/imperfections. At first, the quality of the direct inputs (the invariant and tensor bases) of the model calculated with LES data is briefly discussed. Then a first training campaign is performed to get a preliminary idea of the ANN capability to deal with LES data as well as to introduce some important concepts and make some structuring choices for the rest of the analysis. This is followed by a reflection based on the Integrated Gradients method to identify if a model trained with LES reacts similarly to a model trained with DNS to variations of its inputs. Those first investigations will motivate to perform a statistical analysis of the model outputs for different training sessions to assess the effect of the LES resolution on the quality of the model obtained. Finally, training the ANN using DNS and LES (for all the resolutions) data of channel flows together is also tested and investigated.

All the results presented in this chapter are obtained with training sessions performing 8000 epochs each and using minibatches composed of four physical cases except when it is explicitly stated otherwise. For the sake of simplicity, in this chapter, the wall-normalized THF is simply noted $\overline{\mathbf{u}\theta}$ instead of $\overline{\mathbf{u}'\theta'}^{+}$.

4.1 Direct neural network inputs

In order to be used to train the ANN, the LES data are interpolated on a thousand points and the invariant and tensor bases are computed using the expressions in table 2.1. A first level of analysis consists in examining the invariants profiles. All invariants depend only on momentum quantities except for the seventh and eighth invariants and the Prandtl number. Figures 4.1 and 4.2 present the logarithmically transformed first and eighth invariants profiles for the different resolutions. All the other invariants profiles can be found in appendix 6.3. The general trends are satisfactory for every invariant and every resolution but although a logarithmic transformation is applied, important shifts between the profiles of different resolutions can still be observed so that the amplitude errors are actually quite high. As stated before, the dissipation rate ϵ is a major driver of those errors. However, a consistent monotonic convergence is observed from low to high resolution.

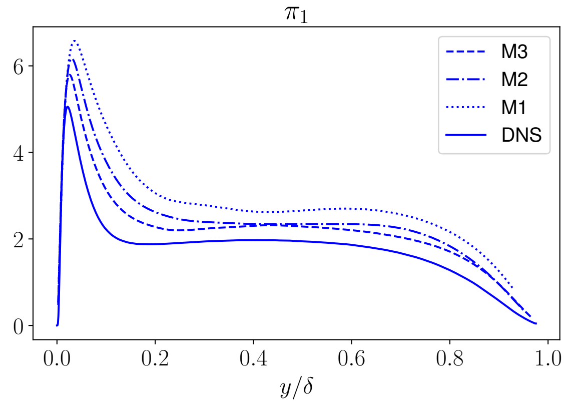


Figure 4.1: First invariant profiles comparison for $Re_\tau = 395$.

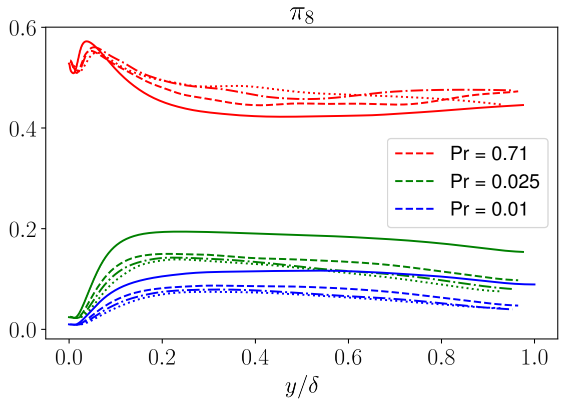


Figure 4.2: Eighth invariant profiles comparison for $Re_\tau = 395$.

4.2 First training campaign

Initially, training the ANN using only LES data for a particular resolution is tested. One quarter of the data is not used to train the network and is left as validation data. When a large quantity of data is available, those data can be randomly chosen but that is not the case here : there are only eight LES physical cases available for each resolution, all reduced to 1D data. Hence the choice of the training data requires a bit of care to hope to get good results from the ANN. Table 4.1 indicates which data are left as validation data. Those are chosen to be well in the range of Prandtl numbers considered instead of extremums so that the prediction of the ANN for the validation data is an interpolation between different training data rather than an extrapolation.

$Pr =$	1.0	0.71	0.025	0.01
$Re_\tau = 395$	T	T	V	T
$Re_\tau = 640$	T	V	T	T

Table 4.1: Choice of training (T) and validation (V) data for training with LES only.

The ANN trained using the resolution **M2** is able to fit rather well the LES THF components using the input training data (fig. 4.3, left). There is a slight underprediction of the streamwise component but this is generally also observed when training with DNS data. However, significant errors are observed when using the input validation data. Moreover, examining the evolution of the losses (fig 4.4, left) shows that the training and validation losses are decoupled after a few epochs. This is a consequence of the fact that the training database is very small. The SWAG functions indicate a higher uncertainty related to the streamwise component of the THF. Fortunately, it is the wall-normal component which matters to obtain a good temperature profile for channel flows. Indeed, the THF, which is independent of x, enters the averaged energy equation through its divergence so that the streamwise component does not play any role in it :

$$\nabla \cdot \overline{\mathbf{u}\theta} = \frac{\partial \overline{u\theta}}{\partial x} + \frac{\partial \overline{v\theta}}{\partial y} = \frac{\partial \overline{v\theta}}{\partial y}. \tag{4.1}$$

In order to increase the size of the training database, the DNS of Kawamura et al. [16] for $Re_\tau = 180$ are added to it. This increases from eight to fifteen the number of combinations of Reynolds and Prandtl numbers present in the database. The reasoning behind this addition is that channel flows DNS for such a low Reynolds number are relatively cheap to produce (around one million cells for the mesh) so that the overall computational cost of the resulting database is still appreciably lower than a full DNS channel flow database. However, the database considered remains small and the validation data must still be chosen with care. Table 4.2 indicates which physical cases were chosen as validation data. Except for section 4.5, the same choice of validation data is used for all the analyses performed in this chapter. This introduces a potential loss of generality to the observations made throughout this section and section 4.4. In what follows, whenever an **Mx** database is mentioned, it will refer to a database composed of all LES of resolution **Mx** and all DNS of Kawamura for $Re_\tau = 180$.

$Pr =$	1.0	0.71	0.6	0.4	0.2	0.1	0.05	0.025	0.01
$Re_\tau = 180$	/	T	T	V	T	V	T	T	/
$Re_\tau = 395$	T	V	/	/	/	/	/	T	T
$Re_\tau = 640$	T	T	/	/	/	/	/	V	T

Table 4.2: Choice of training (T) and validation (V) data for training with LES and low Reynolds DNS

Figure 4.3 (right) shows that, when using LES and low Reynolds DNS instead of LES only, the ANN fits just as well the LES using the training data and that it performs a bit better using the validation data. Moreover the validation losses and training losses are less decoupled. Hence, adding low Reynolds DNS to the LES training database seems to improve the predicting capabilities of the ANN.

It is interesting to investigate the performance of a model trained with LES data when DNS data are provided as inputs to it. Indeed, DNS data provide the ideal inputs to the network and a model trained with lower quality data should preferably be able to remain consistent when higher quality data are used as inputs otherwise the model has arguably little value. Figure 4.5 gives an insight of the consistency of two models trained with the same database when DNS inputs are provided. The general trends of the THF are preserved when DNS data are used as inputs. However one can clearly observe a significant divergence between the predictions of the first and second models. While it seems that the SWAG functions are able to capture the variability of the training output associated with training data, it is clearly not the case when DNS are used as inputs. This high variance of the obtained model for equivalent training conditions tends to complexify any assessment regarding the ability of the ANN to use LES data for the training or regarding attempts to improve its performances.

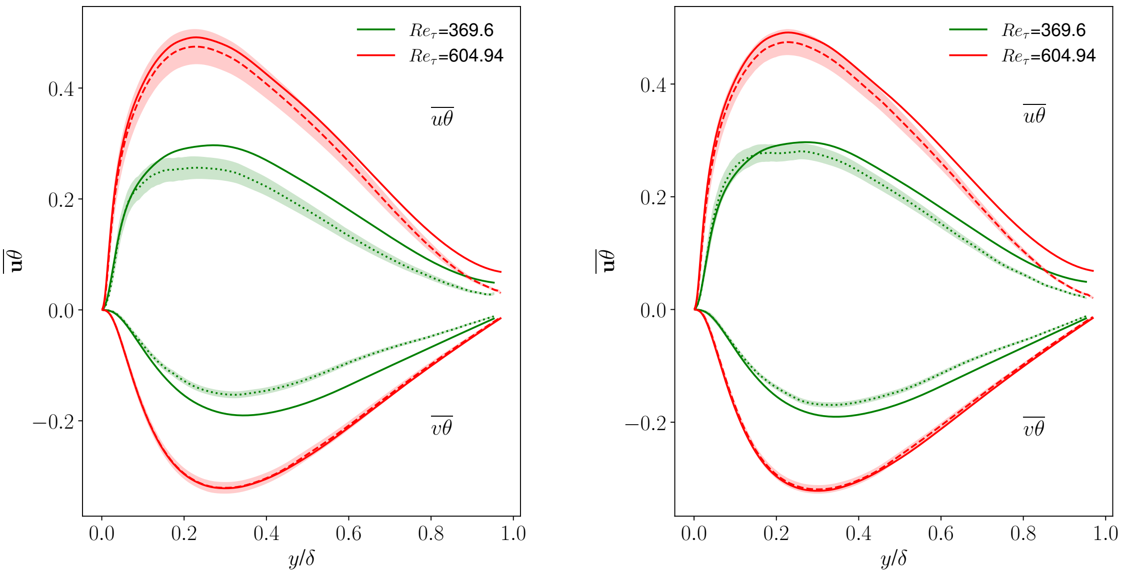


Figure 4.3: THF for $Pr = 0.025$ (right) training with **M2** only (left) or **M2** and low Reynolds DNS (right), $M2$ LES data (solid), model outputs using training data (dash) or validation data (dots) with uncertainty range calculated with the SWAG functions (shaded areas).

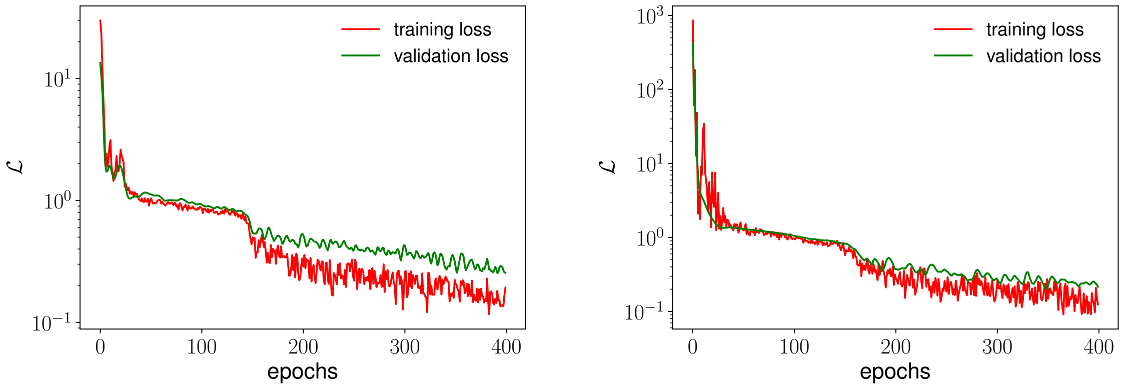


Figure 4.4: Training and validation losses evolution for training with LES only (left) or with LES and low Reynolds DNS (right).

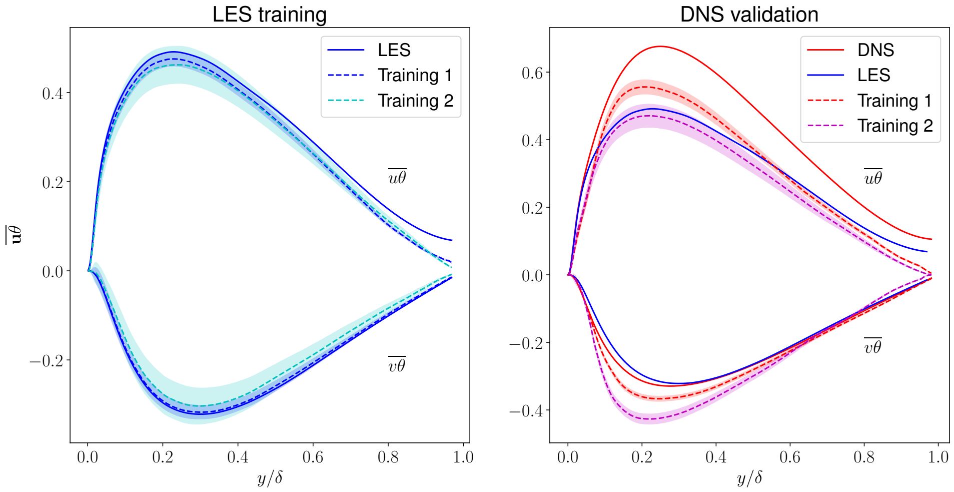


Figure 4.5: THF predictions of two models trained with resolution **M2** data for $Re_\tau = 640$ and $Pr = 0.025$ either when **M2** data are provided in input (left) or DNS data (right)

4.3 Integrated Gradient method analysis

The Integrated Gradient method [33] can be used to interpret a model obtained after a particular training. Following a linear path from a baseline input set $\mathbf{x}'$ to another input set $\mathbf{x}$, this method attributes the variation of the model predictions to each element of the input set. A sensitivity analysis code exploiting this method using the Captum library was already developed by M. Fiore [14]. It has only been slightly extended for the present work to allow visual comparison of different models. It works in two stages because there are two input injection steps in the mathematical structure of the model (see fig. 2.1, from left to right). The first occurs at the input layer of the ANN, where the invariant basis is injected and the second when the tensor basis is injected to calculate the tensors $\mathbf{A}$ and $\mathbf{W}$. At first, the Integrated Gradient method is applied to a portion of the model to attribute the variation of the output $\mathbf{D}$ tensor components to the tensors T_i and the coefficients a_i and w_i. Then it is applied to attribute the variation of each coefficient a_i and w_i to each of the invariants.

The following linear path is considered :

$$\mathbf{x}'_{(Re_\tau=640,Pr=0.71)} \longrightarrow \mathbf{x}_{(Re_\tau=395,Pr=0.025)} \tag{4.2}$$

The input sets $\mathbf{x}'$ and $\mathbf{x}$ are taken from the DNS database. Figure 4.6 shows that two models both initialised with random weights and trained with the same database can behave quite differently despite both fitting the DNS data well. This suggests that the optimizer can find a different local minimum of the loss function depending on the neural network weights initialisation. So different models can emerge from the training phase, there are multiple possible "strategies" to model the THF. However, one should note that the two models on figure 4.6 nonetheless have a lot in common

: typically the coefficient a_1 and the tensor $\mathbf{T}_1$ remain major contributors for both models. On the other hand, figure 4.7 shows that once a model has been obtained after a large number of training epochs, continuing to train this model for an equivalently large number of epochs doesn't fundamentally change the way the model behaves, it is still essentially the same model. Actually, the small differences of attribution observable in figure 4.7 reflect what the SWAG functions are able to capture : the remanent variations of a mostly established model. This means that the SWAG approach is not able to capture the total model uncertainty, as it only gives a measure of the variability around the local minimum found in each training. Nonetheless, it would probably better approximate the total model uncertainty if more data were added to the database as the training would be more constrained.

The previous developments open another angle of analysis to assess the option of using LES to train the ANN in the form of the following question : is a local minimum of the loss function for DNS data approximately also a local minimum for LES data or are DNS and LES data too different for this to be the case? Examining figure 4.8 suggests that the first affirmation holds true. It compares the behaviour of a first model trained with DNS data to a second model trained with an **M2** database but initialised with the weights of the first model. Clearly, the two models have a very similar behaviour. Furthermore, ignoring the vicinity of the wall, it can be observed in figure 6.51 that the variation of the coefficient a_1 (which is the most important one according to figure 4.8) is mostly attributed to the invariants π_7, π_8, Re_t and Pr in both cases. So the model is indeed tuned to **M2** data rather than completely changed. All of this holds true as well for resolutions **M3** and **M1** (appendix 6.4).

Investigations on other linear paths for different models tend to show that the variation of the D_{yy} component when the Reynolds number changes, is generally mostly attributed to the invariant Re_t while, when the Prandtl number changes, it is attributed to the invariants π_7, π_8 and Pr (which is logical because those are the thermally dependent invariants).

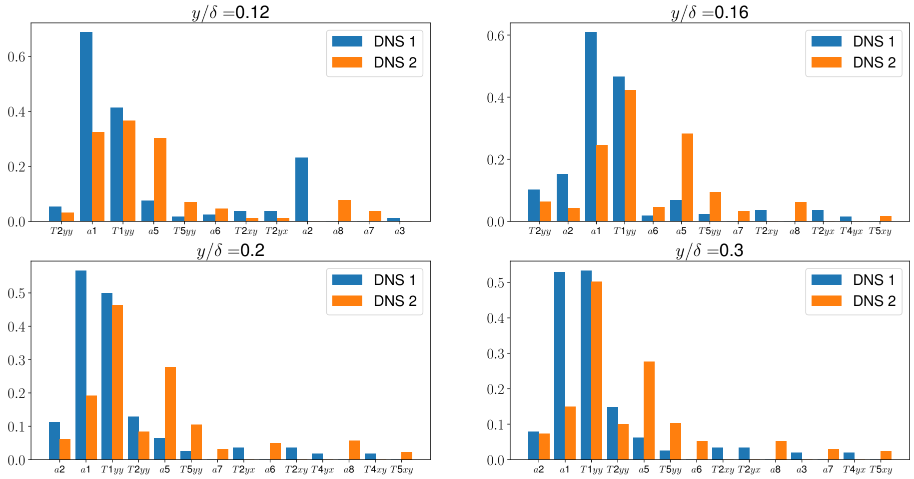


Figure 4.6: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between the base model trained with a DNS database and another model trained with the same database and initialised with random weights.

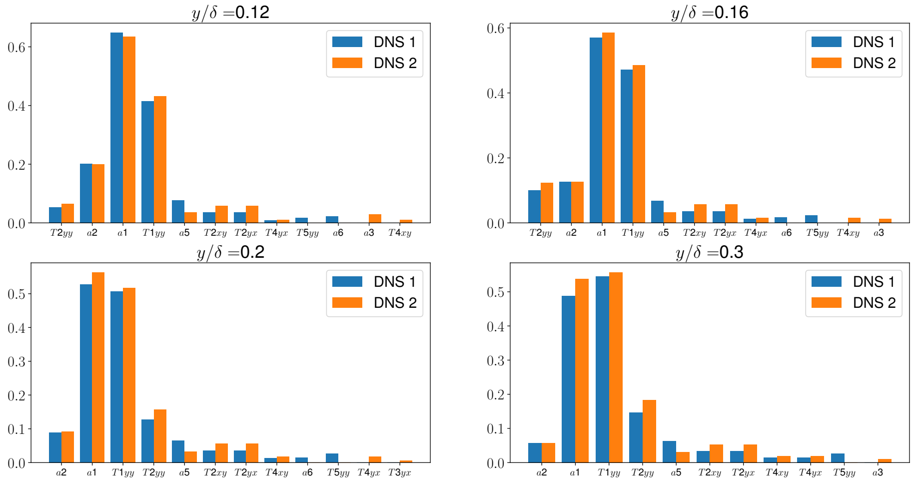


Figure 4.7: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between the base model trained with a DNS database and another model trained with the same database and initialised with the weights of the base model.

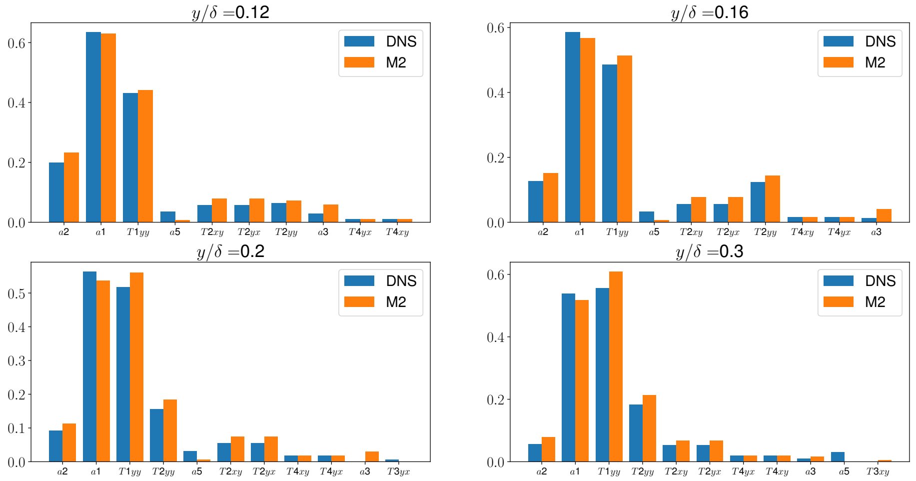


Figure 4.8: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between a first model trained with a DNS database and a second model trained with an **M2** database and initialised with the weights of the first model.

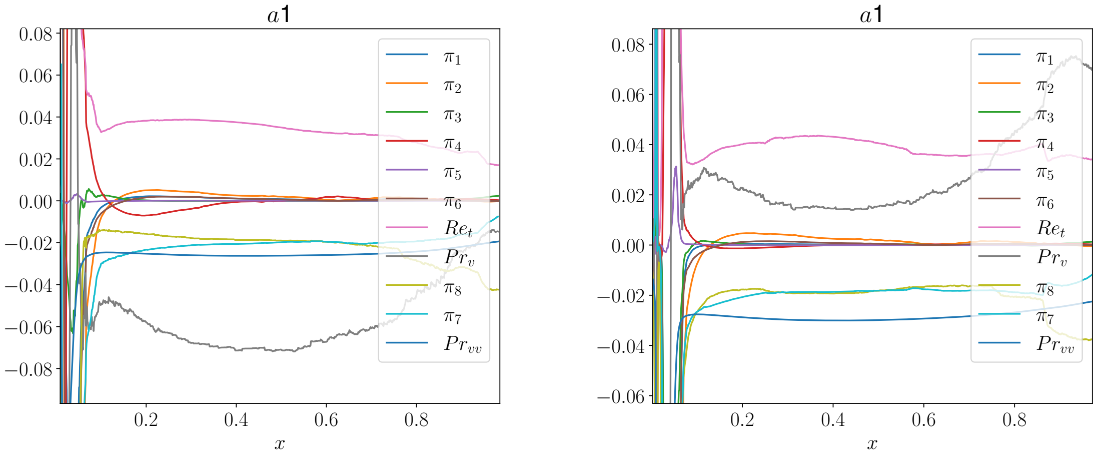


Figure 4.9: Attributions of the variation of a_1 to the invariant basis. First model trained with a DNS database (left) and second model trained with an **M2** database and initialised with the weights of the first model (right). Pr_v and Pr_{vv} designate the Prandtl number injected respectively in the first and second branch of the ANN.

4.4 Statistical analysis

The previous sections highlighted the fact that significantly different models can emerge from the training phase so that one particular training with an **Mx** database does not constitute a complete assessment of the performances one can expect when training with an **Mx** resolution. This motivates to make statistical analyses of the outputs given by an important number of models randomly initialised and trained with the same database in order to obtain an envelop of the possible predictions. Because the databases considered for the training in the present work are small, this is computationally affordable. It is nonetheless still expensive so that only 50 training sessions were performed for each statistical analysis. 50 training sessions is a limited statistical sample but a comparison between two equivalent statistical analyses suggested that it is enough to provide the major trends.

4.4.1 Neural network re-dimensioning

The ANN was developed and dimensioned to be trained with 2D flows data while in the present work it is only trained with 1D data. Hence, 100 units per hidden layers might be excessive when training with 1D data only. This could be a source of the undesirable model variance from one training to another. However, statistical investigations of the THF predictions for different numbers of units per hidden layers do not corroborate this intuition. Indeed, there is not a consistent trend which can be identified by the examination of the variance of THF predictions (fig. 4.11). Furthermore, 100 units per hidden layer clearly remains the best choice to minimize both the training and validation losses (fig. 4.10) and consequently this parameter is left unchanged for the rest of the chapter.

The previous observation indicates that the variance of the obtained model is much more likely to be attributed to the fact that the mathematical structure of the model was developed for 2D flows rather then to the fact that the ANN was dimensioned for 2D flows. The model is supposed to provide the four components of the dispersion tensor. However, when using 1D data only (as is the case in this work), the training is under-constrained because the components D_{xx} and D_{yx} of the dispersion tensor can take any value without affecting the losses. Moreover, the invariant and tensor bases were derived under the assumption of 2D flows and offer too many degrees of freedom when training with 1D data only.

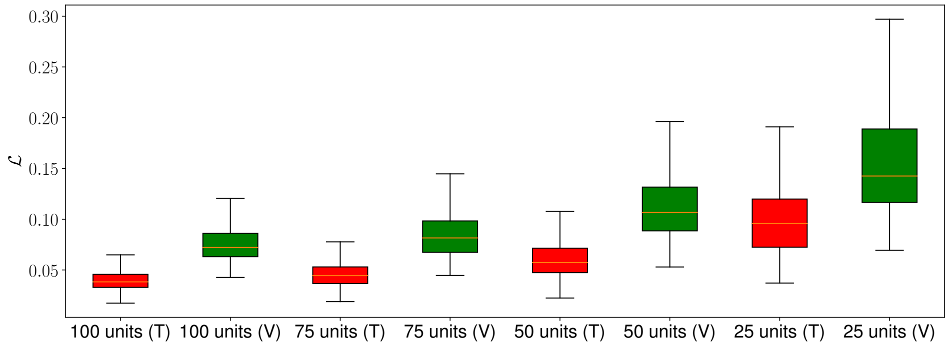


Figure 4.10: Statistics of the training(T) and validation(V) losses for the last 2000 epochs of 50 training sessions for different numbers of units per hidden layers (**M3** database). Outliers are not represented.

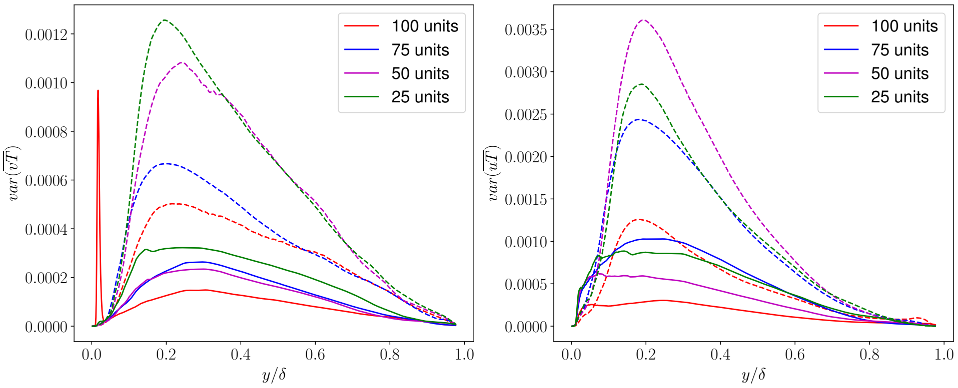


Figure 4.11: Variance of the wall-normal (left) and spanwise (right) prediction of the THF for $Re_\tau = 640$ and $Pr = 0.025$, **M3** training (solid) and DNS validation (dash).

4.4.2 Statistical outputs and LES resolution effect

Figure 4.12 shows that the ability of the neural network to fit LES data is statistically on par for resolutions **M3** and **M2** but that it encounters more difficulties to fit the data for resolution **M1**.

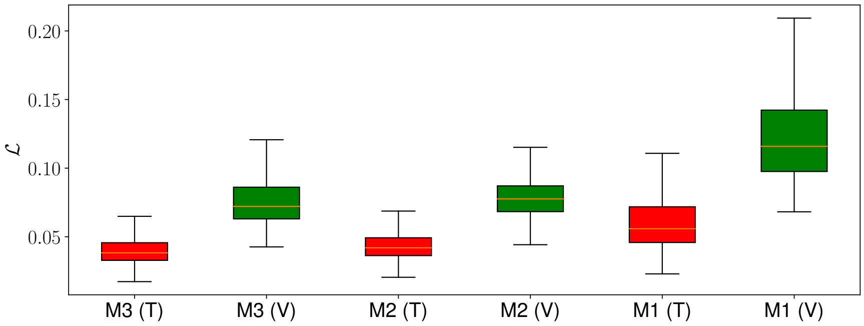


Figure 4.12: Statistics of the mean training(T) and validation(V) losses for the last 2000 epochs of 50 training sessions for different resolutions. Outliers are not represented.

The obtained statistics for the predictions of the models all trained with a database related to a particular LES resolution are examined through several layers of analysis. Beside the THF, the relevant components of the dispersion tensor (D_{yy} and D_{xy}) and the turbulent Prandtl number are also analyzed. The turbulent Prandtl number is defined as follows :

$$Pr_t = \frac{\nu_t}{\alpha_t} = \frac{\nu_t}{D_{yy}}, \tag{4.3}$$

where the turbulent eddy-viscosity ν_t is either the one from LES or DNS data depending if LES training or DNS validation is presented. Actually, it is the inverse of the turbulent Prandtl number which is presented in this work like it is sometimes done in the literature [10].

The point wise median as well the areas between the first and third quartiles and between the 5th and 95th percentiles are provided for both LES training (training data fitting) and DNS validation (DNS data used as inputs). The results obtained are compared with the Reynolds analogy for $Pr_t = 0.9$ (a common value for close to unity Prandtl number fluids, denoted as "RA_1") and $Pr_t = 2$ (denoted as "RA_2") and with the Kays correlation [11].

Furthermore, for channel flows, the averaged modified wall normalized energy equation is reduced to an ordinary differential equation :

$$\frac{1}{Re_\tau Pr}\frac{d^2\theta^+}{d^2y} + \frac{d}{dy}(\alpha_t^+\frac{d\theta^+}{dy}) + \frac{u^+}{u_b^+} = 0. \tag{4.4}$$

This ODE can be integrated using as turbulent eddy-diffusivity the component D_{yy} obtained with a data-driven model using DNS data as inputs. Doing so gives a picture of the performance of a thermal turbulence model independently of errors introduced by a momentum turbulence model. It is nonetheless not perfectly rigorous because there are thermally dependent invariants in the ANN inputs (π_7 and π_8) which introduce a dependency of the turbulent eddy-diffusivity α_t on the temperature θ.

Finally, the obtained temperature profile can be used to calculate a Nusselt number :

$$Nu = \frac{q_w}{\frac{k}{2\delta}|T_W - T_m|} = \frac{2Pr \cdot Re_\tau}{\theta_m^+}, \tag{4.5}$$

where $\theta_m^+ = \int u^+\theta^+ d(y/\delta)/u_b^+\delta$. A comparison of the Nusselt numbers obtained for $Re_\tau = 640$ and $Pr = 0.025$ is provided in table 4.3. It is the most pertinent physical case to compare them among the low Prandtl number cases because the turbulent heat flux is the closest in magnitude to the molecular heat flux in that case, so that it influences the temperature profile more importantly.

	DNS	**M3**	**M2**	**M1**	Kays	Ra_1	Ra_2
Data	7.89	7.77	7.83	7.63	7.79	9.59	7.39
Median	/	8.05	8.22	8.26	/	/	/
5th percentile	/	7.8	7.73	7.63	/	/	/
95th percentile	/	8.35	8.57	8.68	/	/	/

Table 4.3: Comparison of the Nusselt numbers obtained for $Re_\tau = 640$ and $Pr = 0.025$. "Data" refers to a Nusselt number directly calculated with the related simulation data (for the Kays correlation and the Reynolds analogy, DNS data are used). The median, the 5th percentile and the 95th percentile of the Nusselt number calculated using the ANN predictions for ideal inputs is provided for each LES resolution.

As this statistical investigation is performed for every resolution and most of the physical cases, it ends up producing a lot of plots only a few of which are presented in this section to make some observations of particular interest. However, appendix

6.5 might be worth the detour to fully grasp how the data-driven model performs for different training sessions depending on the database quality. To begin, a focus is made on the results obtained for $Re_\tau = 640$ and $Pr = 0.025$ using an **M3** database to illustrate observations which generally hold true. Figure 4.13 confirms what was expected : the dispersion of the results is more important for DNS validation than for LES training. This is simply because, contrary to **M3** data, DNS data are out of training conditions. While the LES training nearly systematically under-predicts the wall-normal component of the THF in the same way as LES do, the DNS validation tends to over-predict it instead. Hence, training with LES does not necessarily imply that the predictions of the ANN behave in the same way as LES when DNS data are used as inputs. The relevant components of the dispersion tensor are shown in figure 4.14 from the wall to $y/\delta = 0.8$ only because in the center of the channel the temperature gradient is vanishing and the dispersion tensor components can take very high or very low values which bias the eye perception of the results. This particular matter is out of the scope of this work and is discussed in [14]. Figure 4.15 illustrates well that the turbulent Prandtl number varies throughout the channel, explaining why the Reynolds analogy fails for both turbulent Prandtl numbers considered. On the other hand, the under-prediction of Pr_t for LES seems to be partially compensated when DNS data are used.

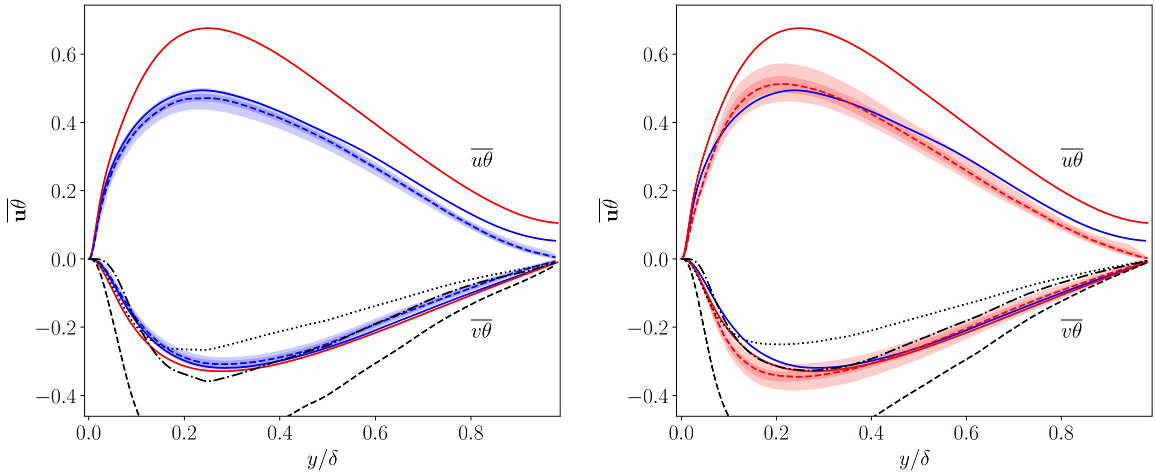


Figure 4.13: Statistical analysis of the prediction of the THF predicted for $Re_\tau = 640$ and $Pr = 0.025$, **M3** resolution is used for the training. DNS (solid red), **M3** data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median **M3** training(dash blue) and DNS validation (dash red), shaded areas encompass 90% of the predictions while darker shaded areas encompass 50% of the predictions.

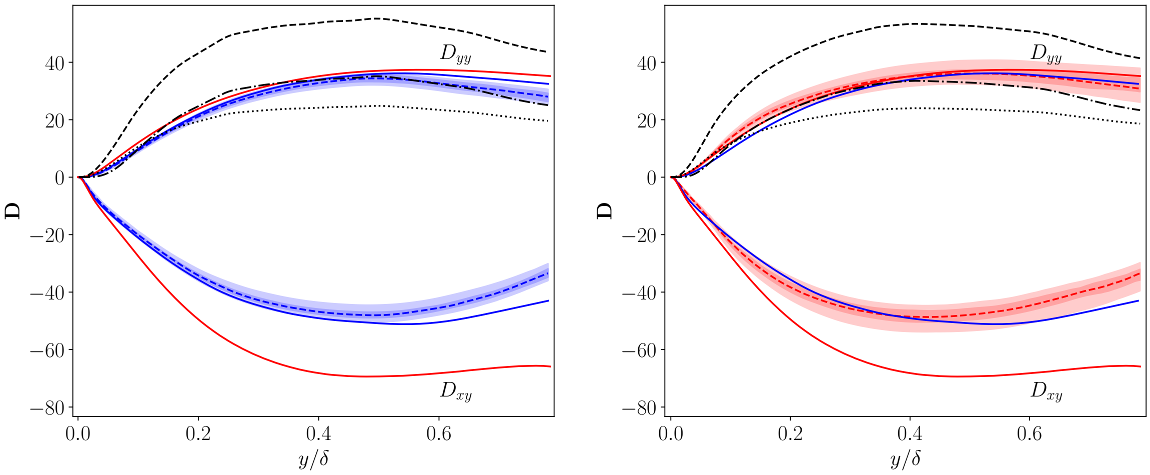


Figure 4.14: Statistical analysis of the relevant components of the $\mathbf{D}$ tensor predicted for $Re_\tau = 640$ and $Pr = 0.025$, **M3** resolution is used for the training. DNS (solid red), **M3** data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median **M3** training(dash blue) and DNS validation (dash red), shaded areas encompass 90% of the predictions while darker shaded areas encompass 50% of the predictions.

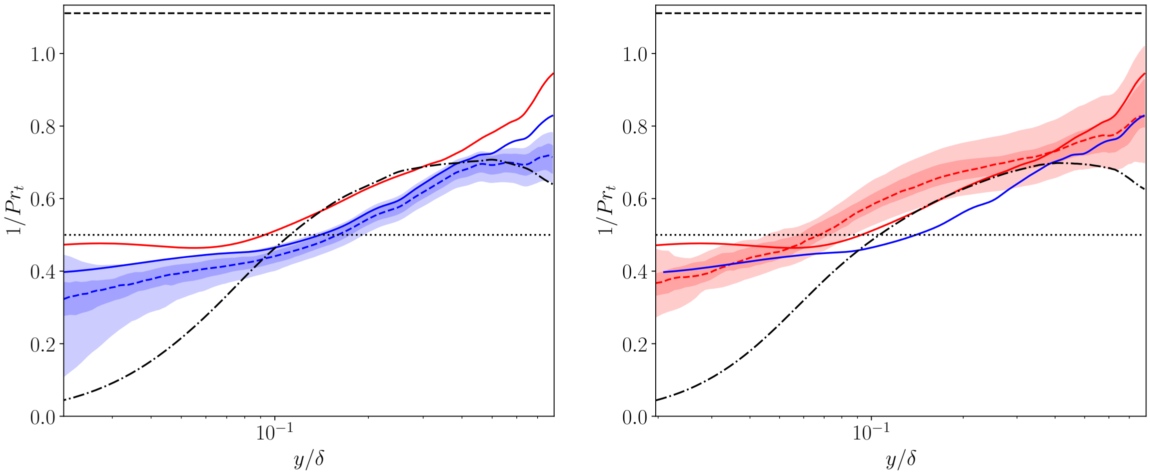


Figure 4.15: Statistical analysis of the inverse of Pr_t predicted for $Re_\tau = 640$ and $Pr = 0.025$, **M3** resolution is used for the training. DNS (solid red), **M3** data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median **M3** training(dash blue) and DNS validation (dash red), shaded areas encompass 90% of the predictions while darker shaded areas encompass 50% of the predictions.

Figure 4.16 illustrates an unexpected but welcome behaviour of the ANN which is always observed for close to unity Prandtl numbers : it is able to compensate the high over-prediction for the peak of the streamwise THF when DNS data are used, even for the lowest LES resolution.

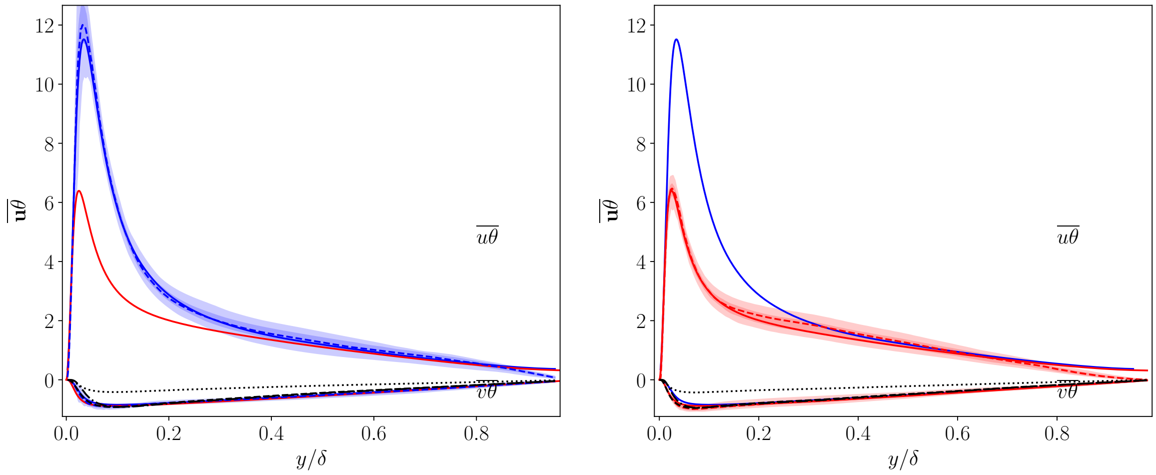
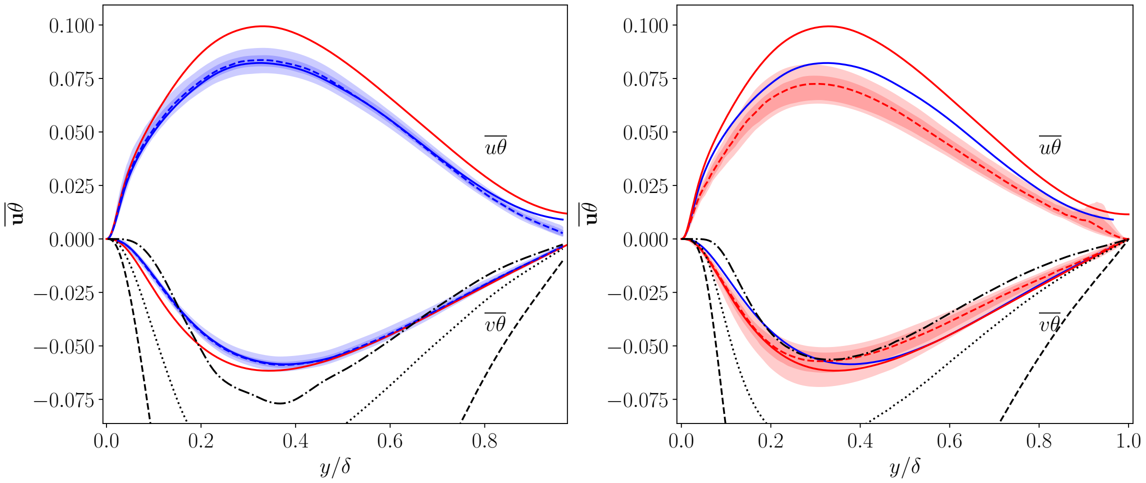


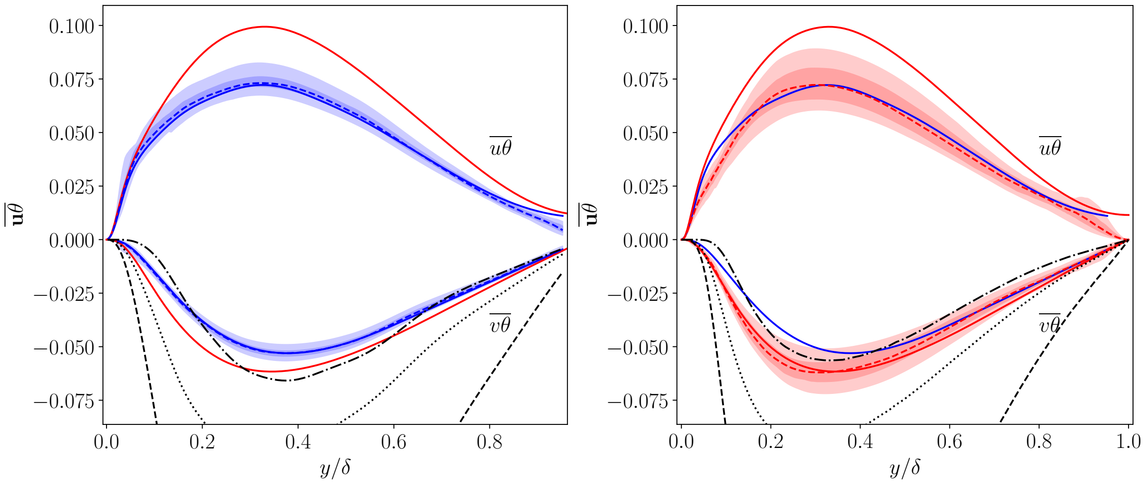
Figure 4.16: Statistical analysis of the THF predicted for $Re_\tau = 640$ and $Pr = 0.71$, **M1** resolution is used for the training. DNS (solid red), **M1** data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median **M1** training(dash blue) and DNS validation (dash red), shaded areas encompass 90% of the predictions while darker shaded areas encompass 50% of the predictions.

The effect of the LES resolution on the results is particularly noticeable for $Pr = 0.01$ and $Re_\tau = 395$ (figure 4.17). One can clearly see that the dispersion of the predictions for the DNS validation increases as the resolution decreases. This is because, when training with a lower resolution, the DNS data are further from training conditions so that the difference between the models appears more clearly as they extrapolate in their own way the THF when using DNS data as inputs instead of training data. Furthermore, the median prediction for DNS validation for the wall-normal THF tend to me more accurate as the resolution increases. So a globally consistent convergence toward DNS wall-normal THF is observed as the resolution increases. However, the same can not be said for the streamwise THF. On the other hand, for this Prandtl number, except for the resolution **M1**, the predictions of the ANN are generally superior to the Kays correlation, especially in the central region of the channel.

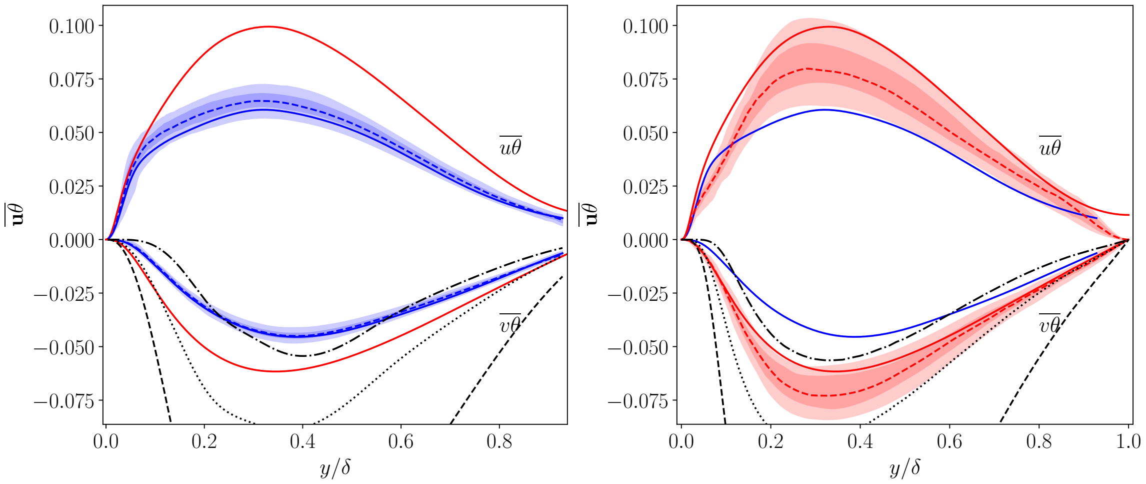
Figure 4.18 shows that for $Pr = 0.025$ and $Re_\tau = 640$, the error for the prediction of the maximum temperature is lower or comparable to the one obtained with the Kays correlation for the median of the ANN predictions depending on the LES resolution. However, in that case, the Kays correlation generally performs better than the ANN predictions near the wall. For $Pr = 0.71$, the predictions of the ANN are mostly comparable to RA and the Kays correlation. Nonetheless, the dispersion of the predictions implies that the Reynolds analogy is still the most robust option in that case.



(a) LES training (**M3**). (b) DNS validation (**M3**).

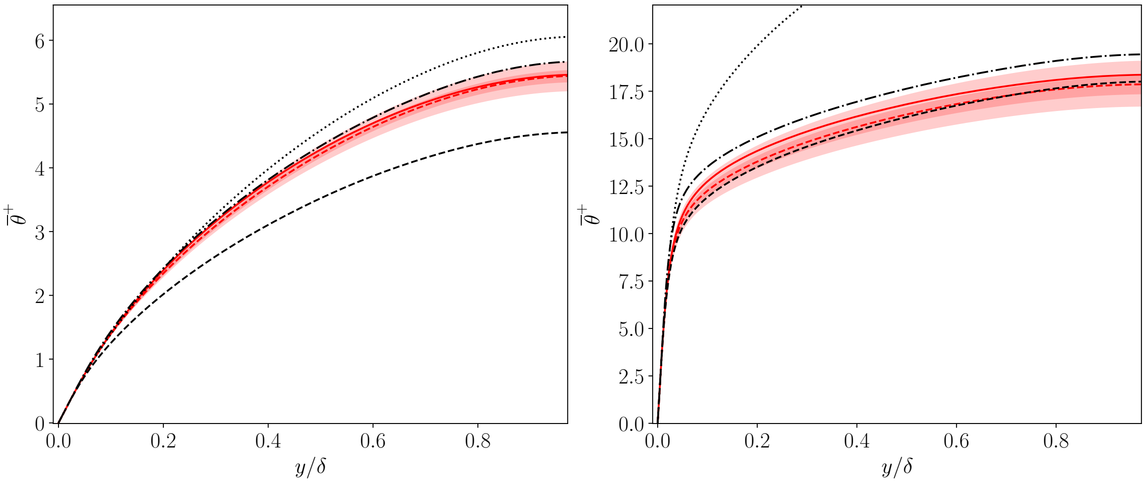


(c) LES training (**M2**). (d) DNS validation (**M2**).



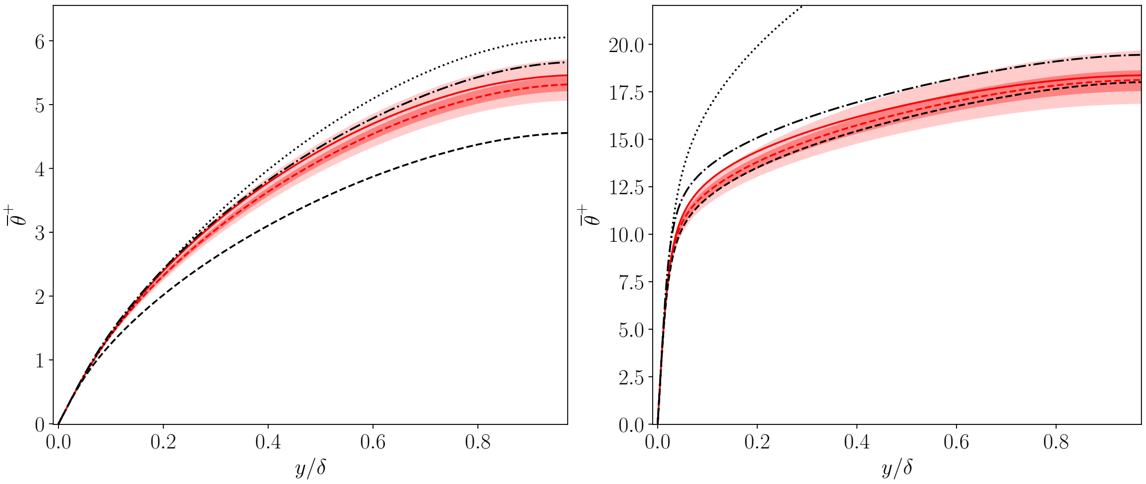
(e) LES training (**M1**). (f) DNS validation (**M1**).

Figure 4.17: Statistical analysis of the THF predicted for $Re_\tau = 395$ and $Pr = 0.01$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompass 90% of the predictions while darker shaded areas encompass 50% of the predictions.



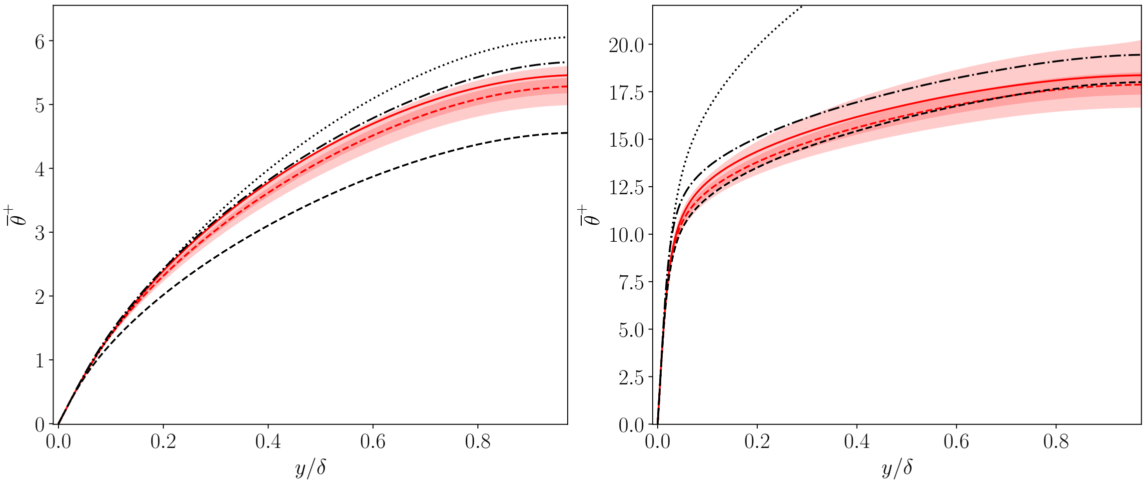
(a) $Pr = 0.025$, **M3**.

(b) $Pr = 0.71$, **M3**.



(c) $Pr = 0.025$, **M2**.

(d) $Pr = 0.71$, **M2**.



(e) $Pr = 0.025$, **M1**.

(f) $Pr = 0.71$, **M1**.

Figure 4.18: Statistical analysis of the temperature profile obtained by integration of exp. 4.4 for $Re_\tau = 640$. DNS (solid red), RA(dash black), RA_2(dot black), Kays(dot-dash black), median DNS validation (dash red), shaded areas encompass 90% of the predictions while darker shaded areas encompass 50% of the predictions.

4.5 Training with all data

It is interesting to investigate if the training of the ANN based model can converge when all the available channel flows data are used to constitute a single training database. That is, using the DNS of Kawamura and Tiselj beside the LES data for resolution **M3**, **M2** and **M1**. This brings to 41 the total number of cases so that 10 cases can be randomly chosen as validation data without significantly risking extrapolation. The size of the minibatches are increased from 4 to 8 as it has proved to be decisive to obtain good models with the complete database. Figure 4.19 shows that the training can converge using all data, and figure 4.23 shows that the obtained model can make predictions which fit well the trends of the THF for all the resolutions. The predictions of the ANN for other Prandtl numbers can be found in appendix 6.6.

The data constituting the complete database clearly does not have the same quality. DNS provides the ideal inputs and outputs of the model while LES provides lower and lower fidelity data as the resolution is decreased. Hence, ideally, fitting the THF data of higher resolutions should be prioritized over fitting the THF data of lower resolutions. Two strategies are explored to prioritize the higher resolutions. The first consists in adding noise with a normal distribution and an amplitude depending on the data resolution to the invariants entering the first branch of the ANN. This has proved to produce poor models even for low amplitude noise. The second consists in weighting the losses. A weight β_i depending on the resolution of the related training data is attributed to the losses $\mathcal{L}_i$ calculated for each points of a minibatch using expression 2.12. So the total loss related to a minibatch is now calculated as follows :

$$\mathcal{L}_{minibacth} = \sum_{i=1}^{N} \beta_i \mathcal{L}_i / \sum_{i=1}^{N} \beta_i. \tag{4.6}$$

At first, the arbitrary weights $\beta_i = 1, 0.5, 0.25, 0.125$ are chosen respectively for DNS, **M3**, **M2** and **M1** data. Figure 4.21 shows the losses obtained for a training with this modification and figure 6.81 compares the predictions obtained for different resolutions. The wall-normal THF is very well predicted for DNS data and the agreement with LES is satisfactory as well. Attributing arbitrary weights lacks generalization capability. Hence, a calculation of the weights based on the modeled proportion of the turbulent kinetic energy is proposed :

$$\beta_i = \frac{1}{1 + \gamma \frac{k_{sgs}}{k_{tot}}}, \tag{4.7}$$

which equals 1 for DNS because in that case $k_{sgs} = 0$. The parameter γ is empirically fixed to 100. Figure 4.20 provides a comparison of $\frac{k_{sgs}}{k_{tot}}$ for the different LES resolutions. Figures 4.22 and 4.25 suggest that this is a valid method. However, it is difficult to assess if weighting the losses really has the expected impact on the training because the neural network actually tends to fit well the THF for every resolution.

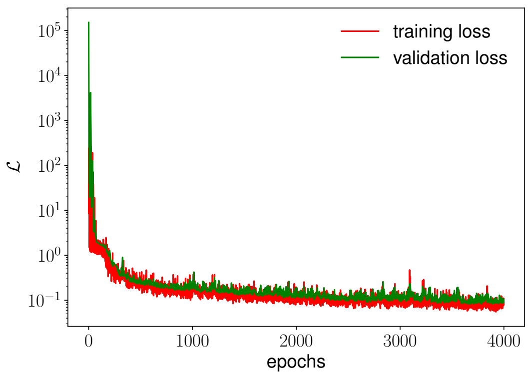


Figure 4.19: Training and validation losses for the training without losses weighting

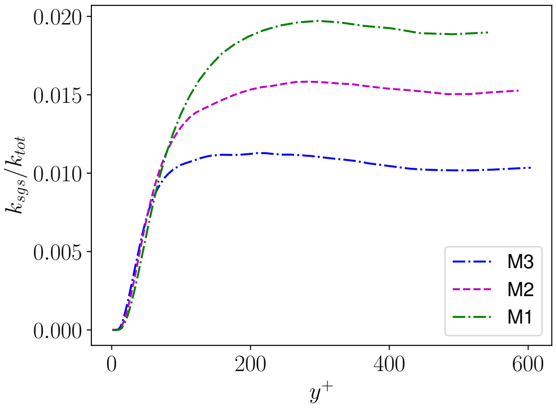


Figure 4.20: Modeled proportion of the TKE for Re_τ.

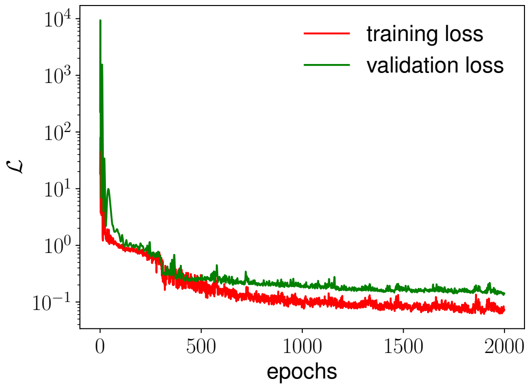


Figure 4.21: Training and validation losses for the training with arbitrary weights

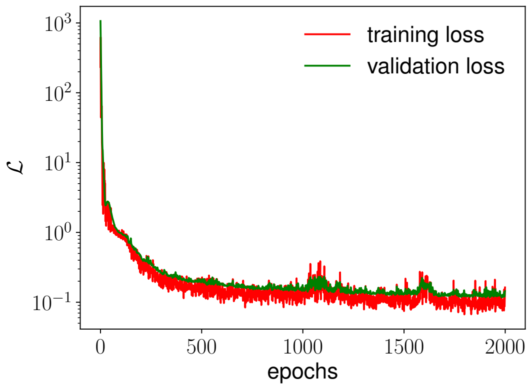
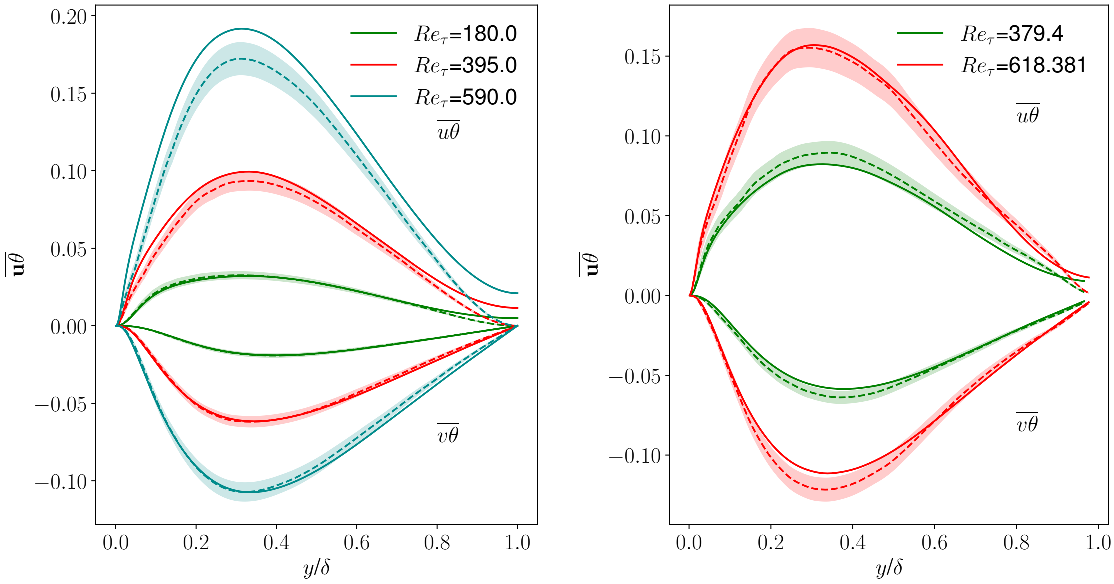
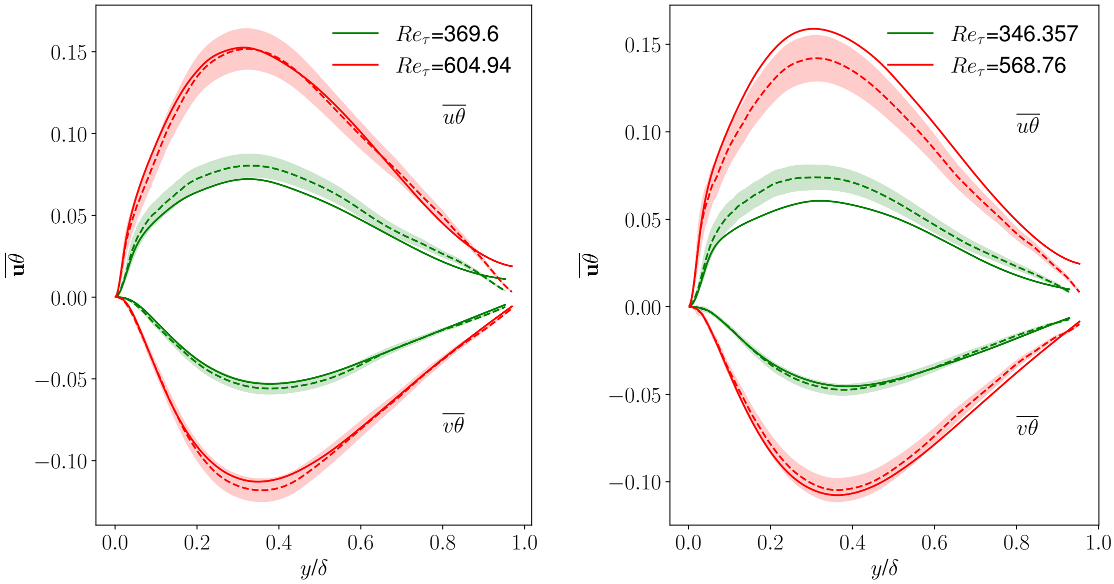


Figure 4.22: Training and validation losses for the training with weights dependent on the modeled proportion of the TKE.

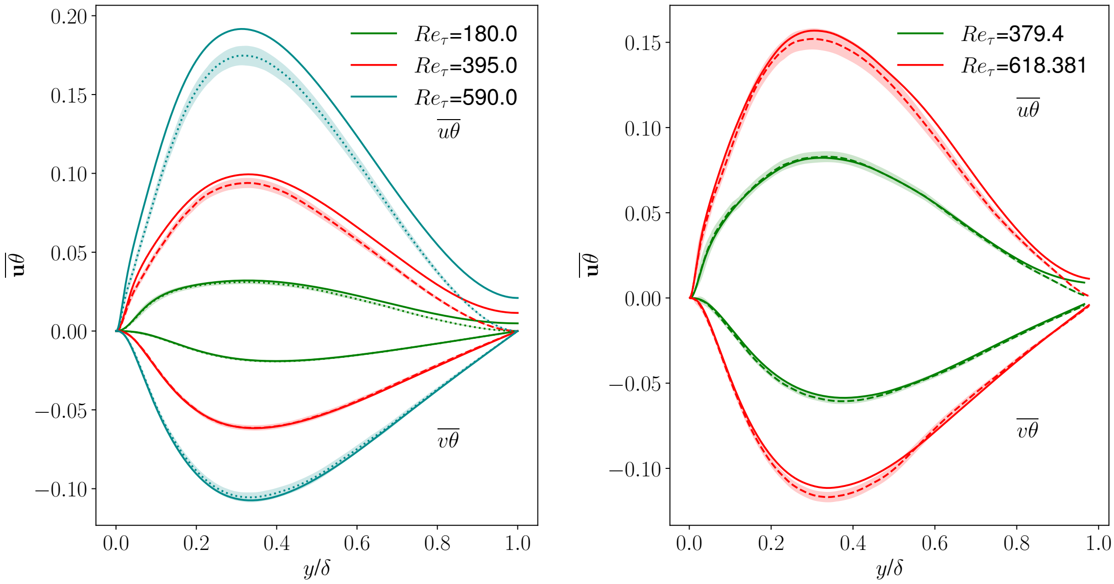


(a) DNS (b) **M3**



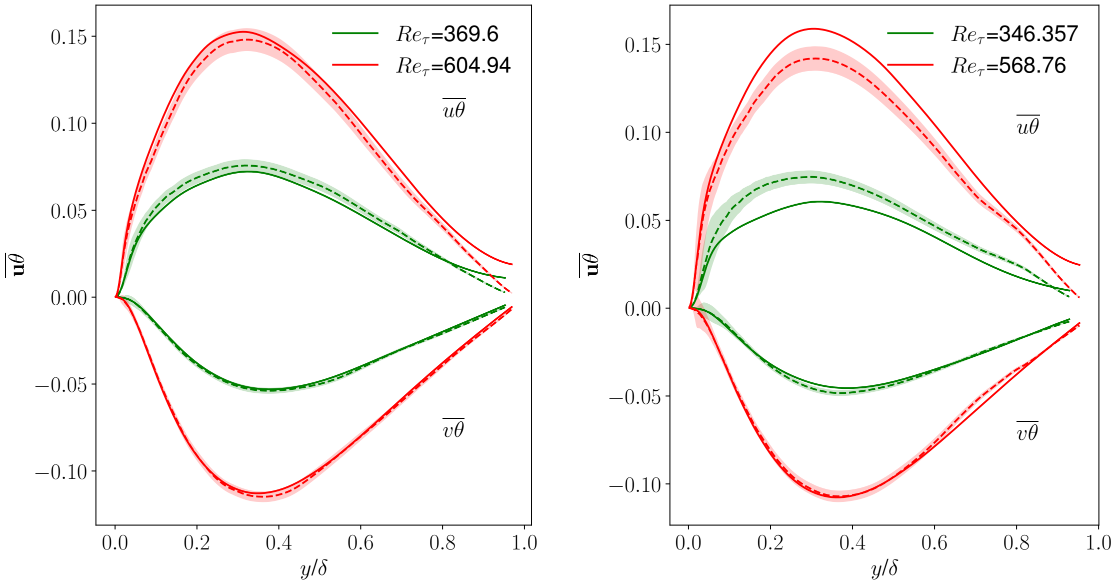
(c) **M2** (d) **M1**

Figure 4.23: Predictions of the THF for $Pr = 0.01$ by a model trained with all data and without losses weighting



(a) DNS

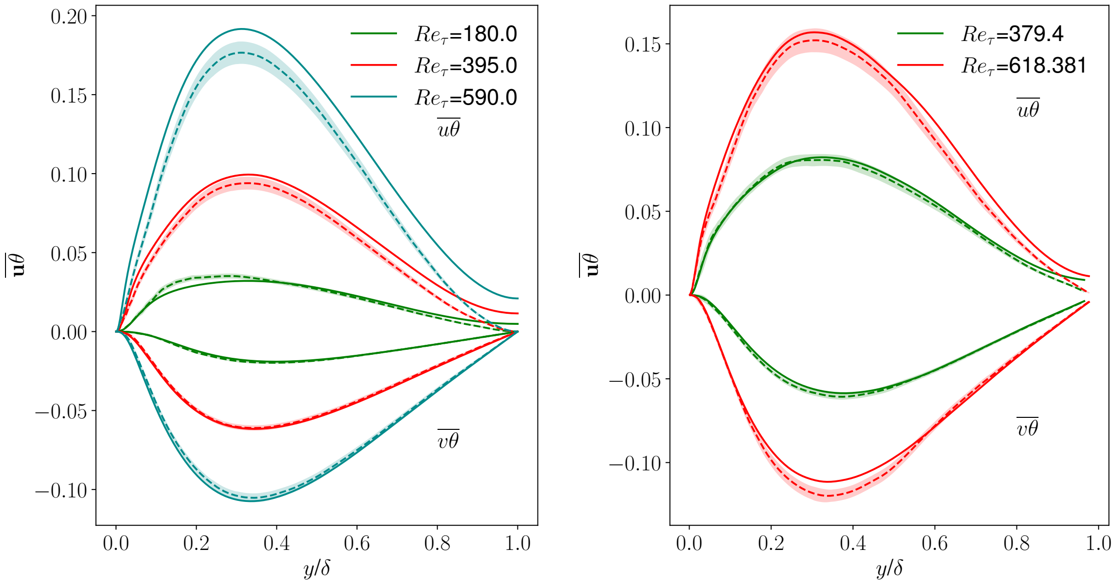
(b) **M3**



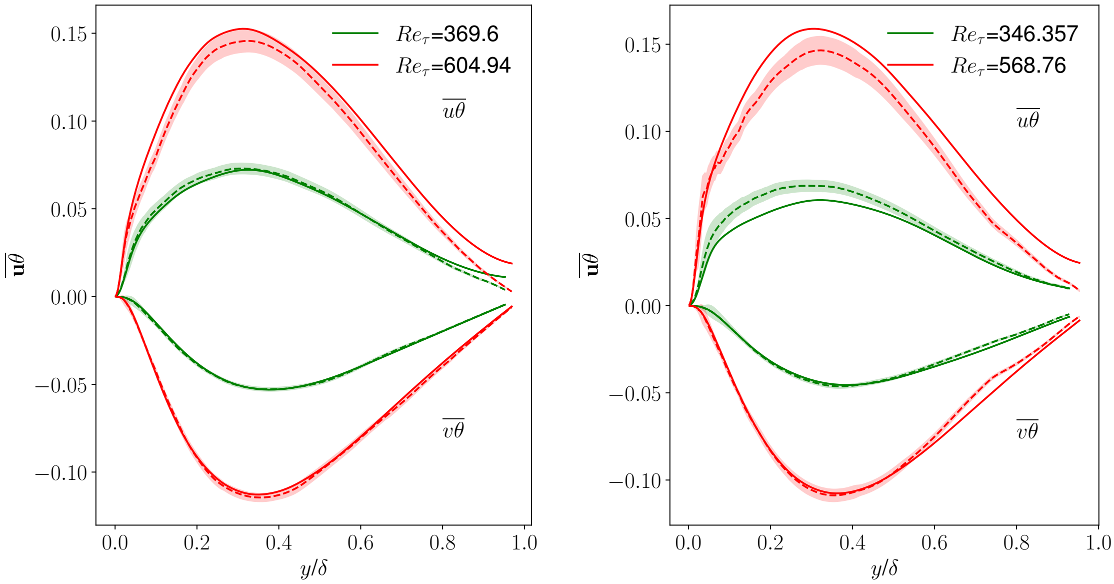
(c) **M2**

(d) **M1**

Figure 4.24: Predictions of the THF for $Pr = 0.01$ by a model trained with all data and with losses weighting using arbitrary weights.



(a) DNS (b) **M3**



(c) **M2** (d) **M1**

Figure 4.25: Predictions of the THF for $Pr = 0.01$ by a model trained with all data and with losses weighting using weights dependent on the modeled proportion of the TKE.

Chapter 5

Conclusion and outcomes

This work explores the possibility of using LES data generated through the use of the finite volume method instead of or beside DNS data, in order to train the neural network based turbulent heat transfer model developed in M.Fiore's thesis. At first, the mathematical structure of the neural network based turbulence model is briefly reviewed. Secondly, non-isothermal turbulent channel flow simulations are performed for different Reynolds and Prandtl numbers and for three different wall units based resolutions in order to constitute an LES database applicable to be used as a training database. This is done using OpenFOAM. Finally, several analyses are conducted to evaluate the behaviour and performance of the data-driven model when the obtained LES data are used to train it.

The LES grid convergence analysis conducted in chapter 3 highlights how heavily the quality of the gathered flow statistics is dependent on the grid resolution when using the finite volume method. In particular, an accurate prediction of the friction velocity seems to require a very refined grid. This is in agreement with what has been observed in many other works. Nonetheless, a monotonic convergence is generally observed. Furthermore, an overall better agreement with DNS data is observed when the Reynolds number is increased. This is a positive feature regarding the possibility to use LES to train the model. Indeed, it is when the Reynolds number increases that DNS may become unaffordable so that an LES substitute becomes particularly interesting.

Through a first training campaign and through the application of the Integrated Gradients method to several models obtained with equivalent training conditions, it is revealed that significantly different models can emerge from the training phase. This is probably mostly due to the fact that the databases used in the framework of this paper are very restricted in size. Hence, the Gaussian Stochastic Weight Averaging algorithm can't provide the full variability extent of the predictions made by different models obtained with randomly initialised training. Consequently, a statistical analysis has to be conducted to obtain the envelop of the possible model predictions in order to rightly assess the effect of using one LES resolution rather than another for the training. This analysis provides the following outcomes :

- The neural network is able to fit the LES data for every LES resolution considered though, on average, it encounters slightly more difficulties to fit the lowest resolution.
- When ideal inputs (DNS data) are injected in models trained with LES data and when the LES resolution is increased, the statistically expected prediction for the wall-normal component of the turbulent heat flux generally exhibits a monotonic convergence towards DNS data, though it does not converge in the same way than LES data do. However, the same can not be concluded for the streamwise component.
- The dispersion of the predictions when ideal inputs are injected, increases as the LES resolution decreases.
- When $Pr = 0.01$ the predictions obtained with models trained with LES data are comparable or superior to the Kays correlation, except for the lowest LES resolution. This is especially the case far from the wall.
- For $Pr = 0.71$, the model predictions are most often satisfactory. However the high dispersion of the predictions implies that the Reynolds analogy remains the robust choice for this close to unity Prandtl number.

The training of the neural network using LES data of different resolutions beside DNS data is also investigated. It is possible to obtain models which satisfactorily predict the good trends for both DNS data and LES data (of any of the resolutions considered). An approach to prioritize fitting high fidelity data over lower fidelity data is proposed. It consists in weighting the losses according to the quality of the training data so that fitting higher fidelity data has "more value" than fitting lower fidelity data. At first arbitrary weights are assigned, and afterwards weighting the losses according to the modeled proportion of the turbulent kinetic energy is tested. However, more investigations would be needed to identify if this approach indeed has the desired effect on the training.

The previous observations tend to suggest that, except maybe for the lowest LES resolution considered, LES generated with the finite volume method can be used to train the neural network and still produce satisfactory models. However, a severe limitation of the present work is that only 1D data are considered for the training while the mathematical structure of the data-driven model was derived for 2D flows. Hence the training is under-constrained and further investigations using more complex flow configurations like backward-facing steps or impinging jets would be needed to confirm or not that LES data can be used to increase the training database size while still allowing to produce satisfactory models.

Chapter 6

Appendix

6.1 Channel flow profiles

This section provides the channel flow profiles which were not presented in chapter 3.

6.1.1 Momentum fields

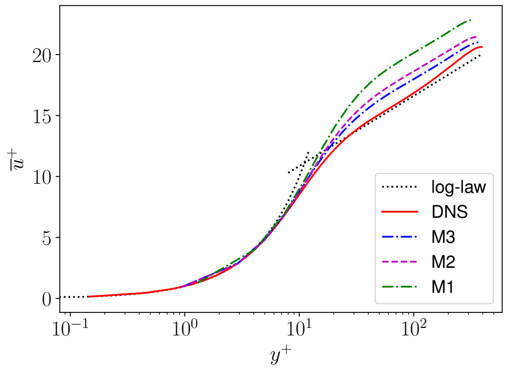


Figure 6.1: Inner-scale normalized mean velocity profiles for $Re_\tau = 395$.

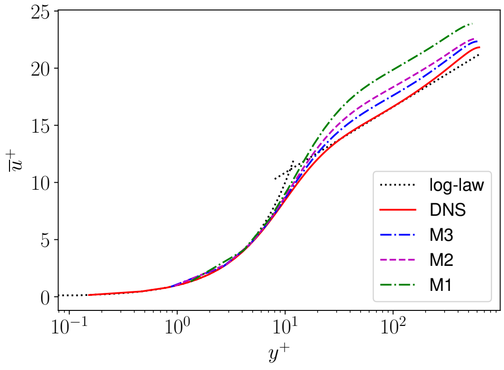


Figure 6.2: Inner-scale normalized mean velocity profiles for $Re_\tau = 640$.

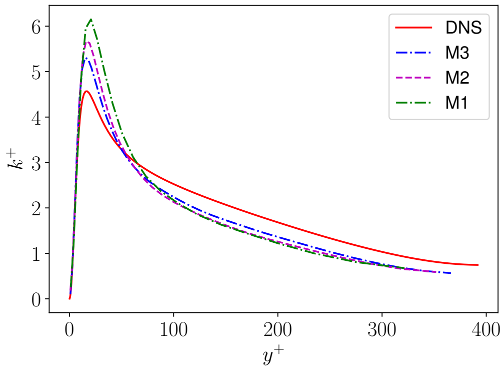


Figure 6.3: Total turbulent kinetic energy ($Re_\tau = 395$).

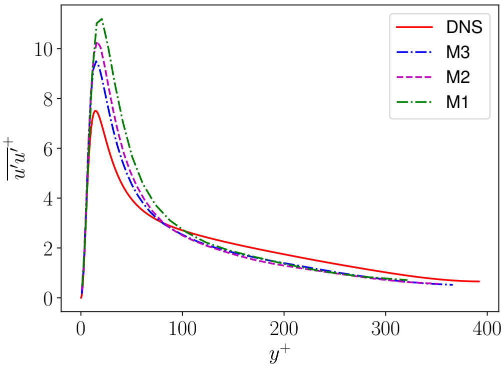


Figure 6.4: Streamwise velocity fluctuations ($Re_\tau = 395$).

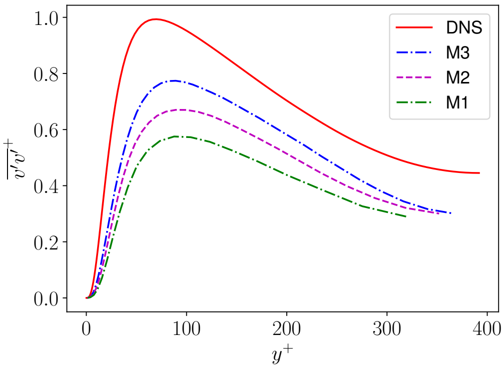


Figure 6.5: Wall-normal velocity fluctuations ($Re_\tau = 395$).

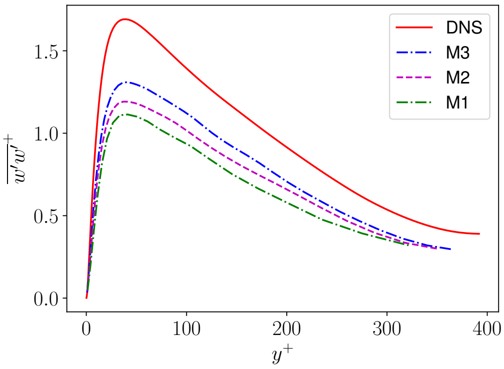


Figure 6.6: Spanwise velocity fluctuations ($Re_\tau = 395$).

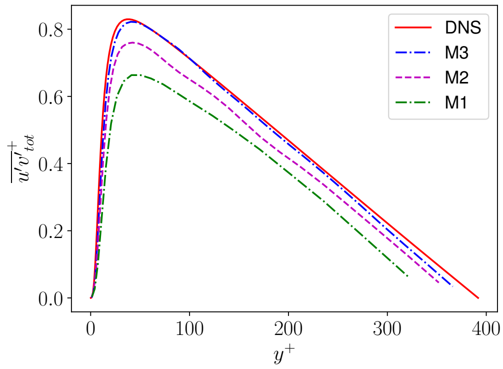


Figure 6.7: Streamwise and spanwise ve- locity covariance ($Re_\tau = 395$).

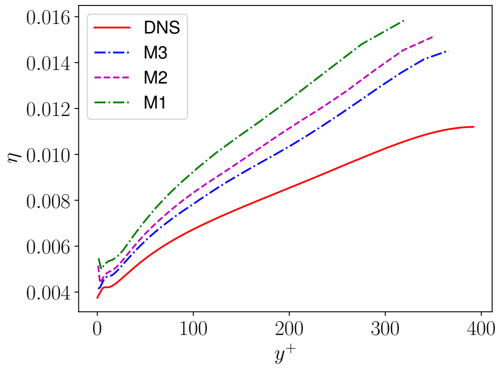


Figure 6.8: Kolmogorov microscale ($Re_\tau = 395$).

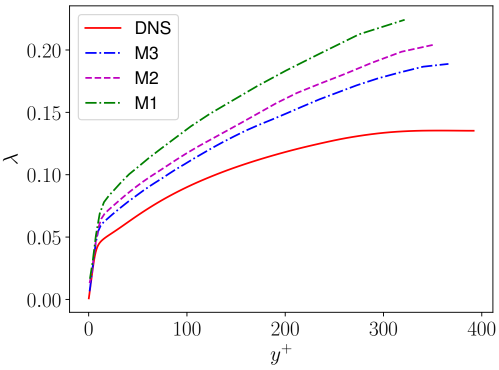


Figure 6.9: Taylor microscale ($Re_\tau = 395$).

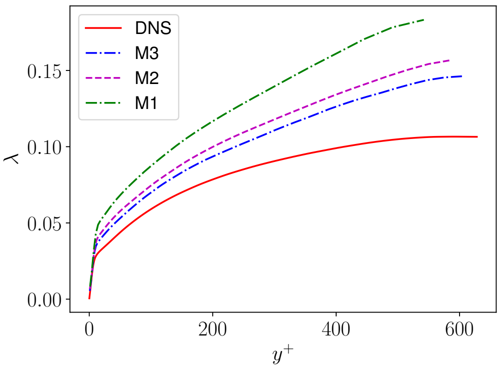


Figure 6.10: Taylor microscale ($Re_\tau = 640$).

6.1.2 Thermal fields for $Pr = 0.71$

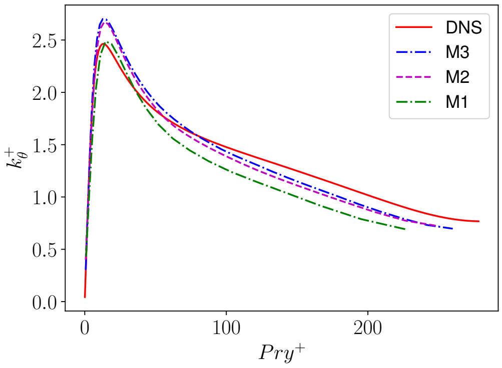


Figure 6.11: Temperature variance ($Re_\tau = 395$).

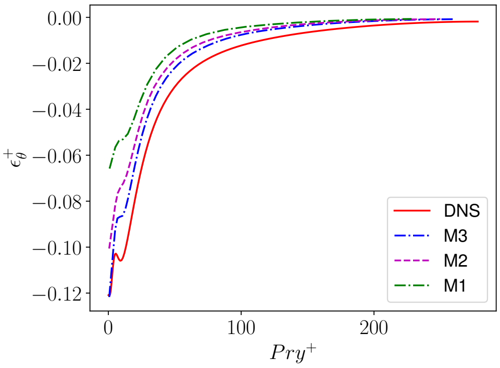


Figure 6.12: Dissipation rate of the temperature variance ($Re_\tau = 395$).

6.1.3 Thermal fields for $Pr = 0.025$

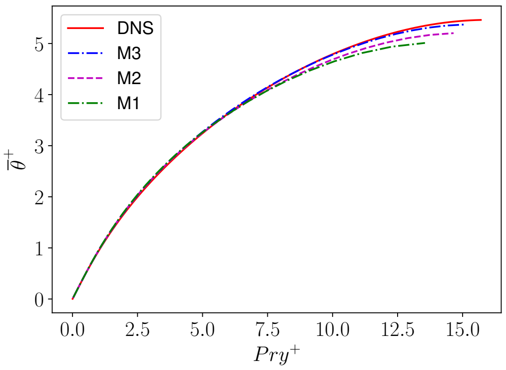


Figure 6.13: Mean temperature profiles ($Re_\tau = 640$).

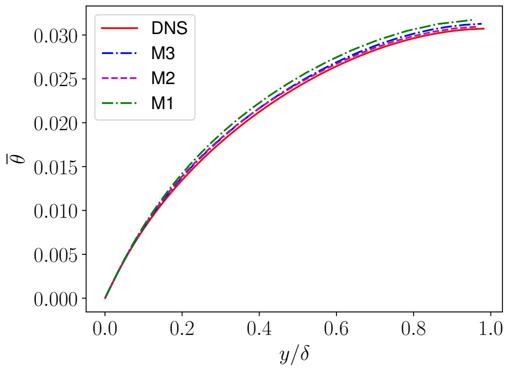


Figure 6.14: Dimensional mean temperature profiles for ($Re_\tau = 640$).

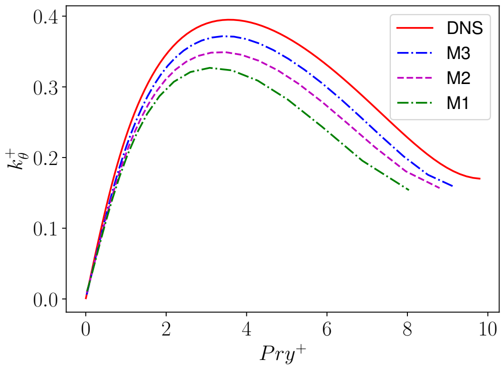


Figure 6.15: Temperature variance ($Re_\tau = 395$).

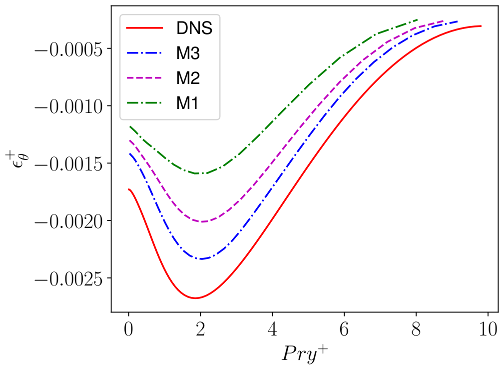


Figure 6.16: Dissipation rate of the temperature variance ($Re_\tau = 395$).

6.1.4 Thermal fields for $Pr = 0.01$

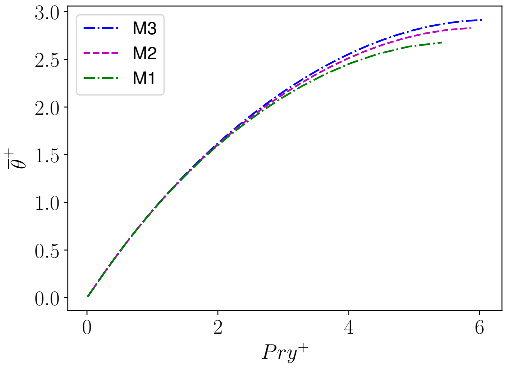


Figure 6.17: Mean temperature profiles for ($Re_\tau = 640$).

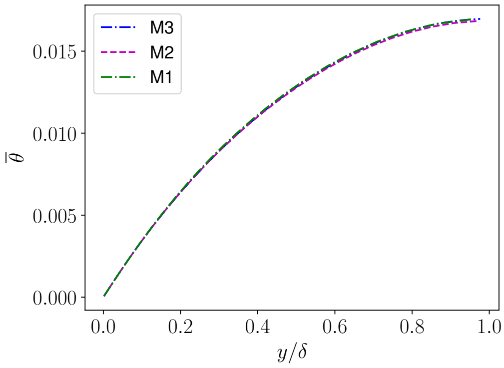


Figure 6.18: Dimensional mean temperature profiles for ($Re_\tau = 640$).

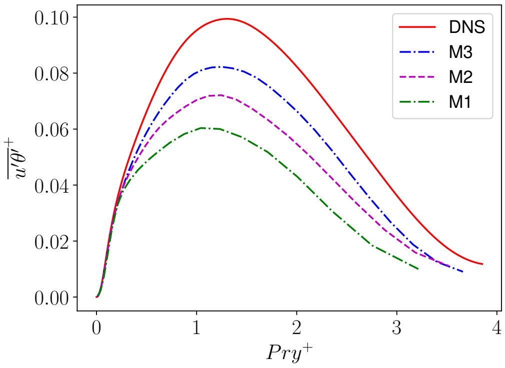


Figure 6.19: Streamwise THF ($Re_\tau = 395$).

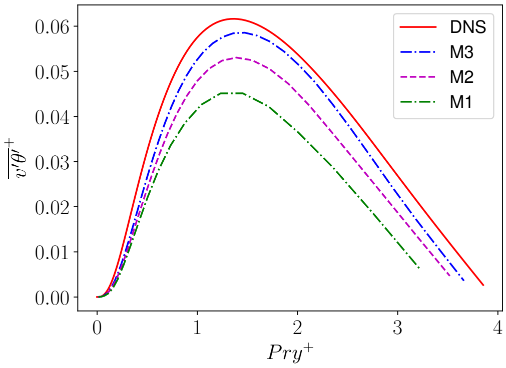


Figure 6.20: Wall-normal THF ($Re_\tau = 395$).

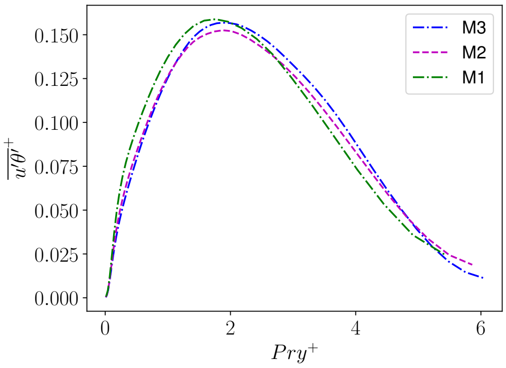


Figure 6.21: Streamwise THF ($Re_\tau = 640$).

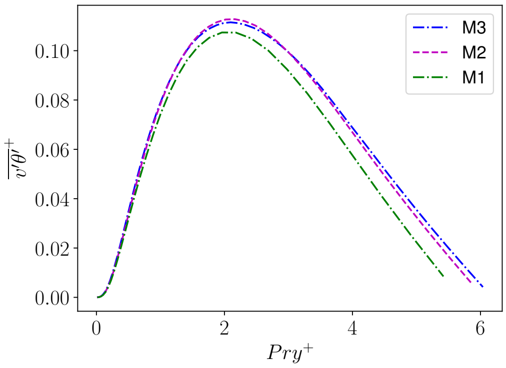


Figure 6.22: Wall-normal THF ($Re_\tau = 640$).

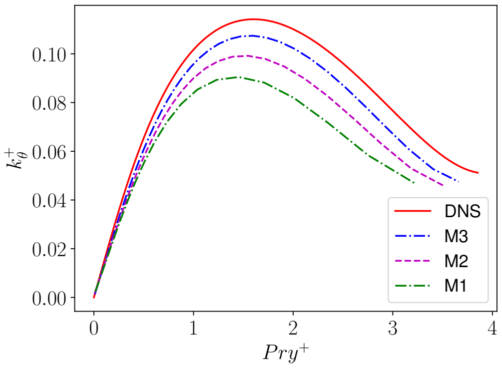


Figure 6.23: Temperature variance ($Re_\tau = 395$).

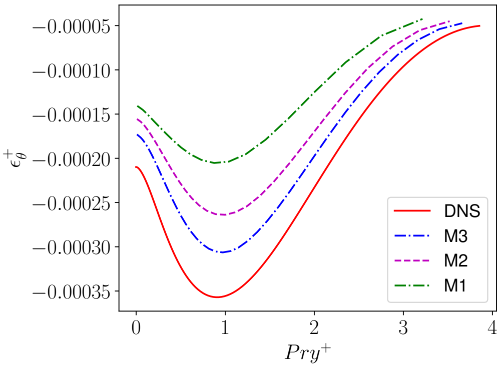


Figure 6.24: Dissipation rate of the temperature variance ($Re_\tau = 395$).

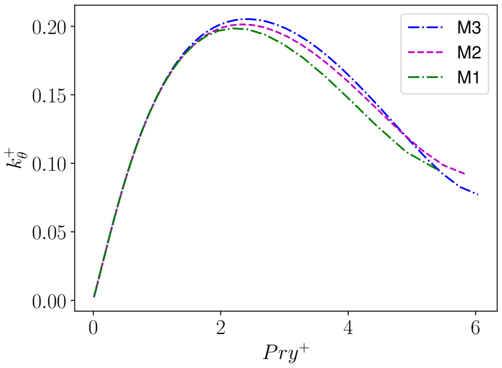


Figure 6.25: Temperature variance ($Re_\tau = 640$).

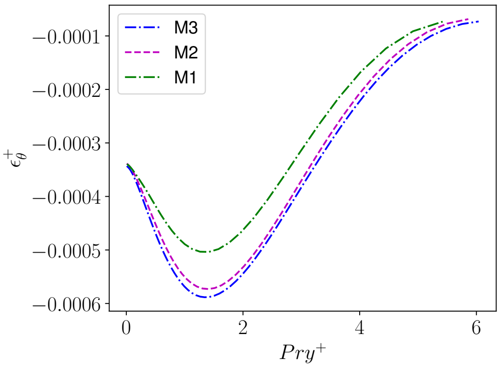


Figure 6.26: Dissipation rate of the temperature variance ($Re_\tau = 640$).

6.2 Instantaneous fields for $Re_\tau = 395$

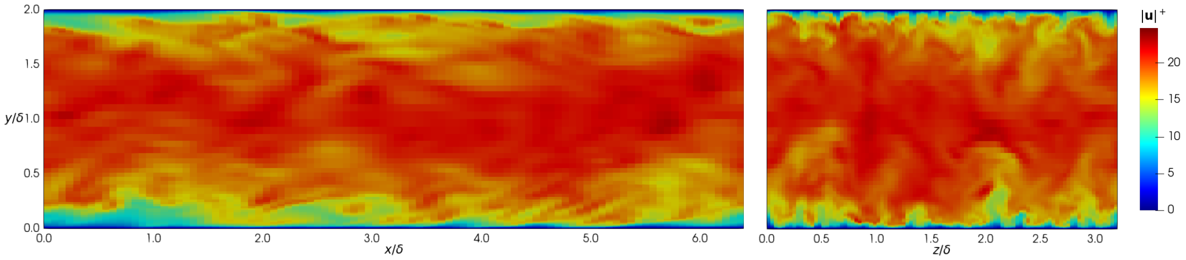


Figure 6.27: Streamwise (left) and spanwise (right) instantaneous velocity field magnitude (**M3**)

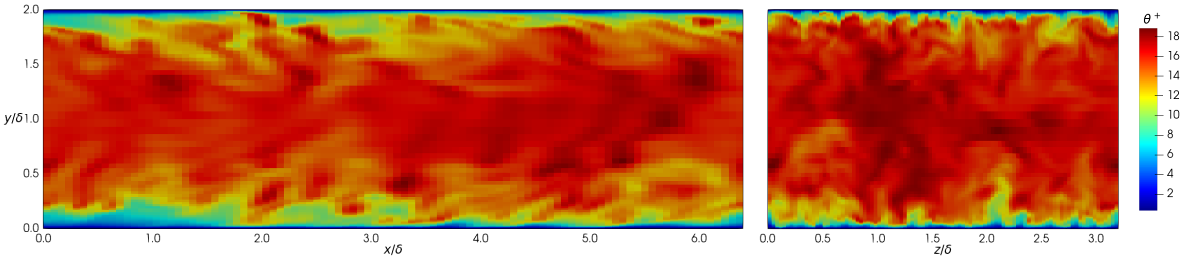


Figure 6.28: Streamwise (left) and spanwise (right) instantaneous temperature field for $Pr = 0.71$ (**M3**)

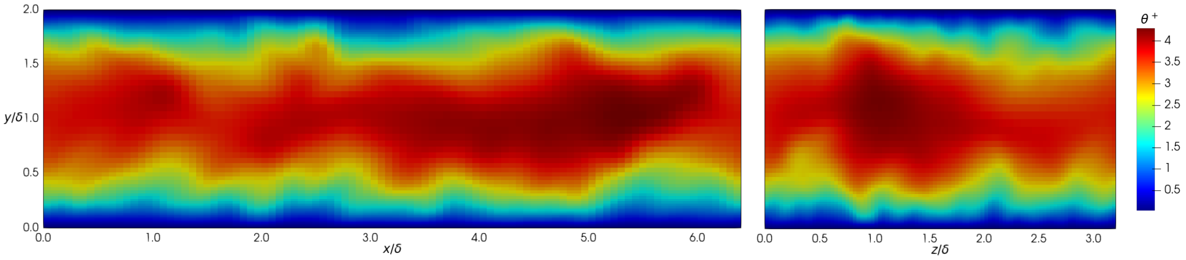


Figure 6.29: Streamwise (left) and spanwise (right) instantaneous temperature field for $Pr = 0.025$ (**M3**)

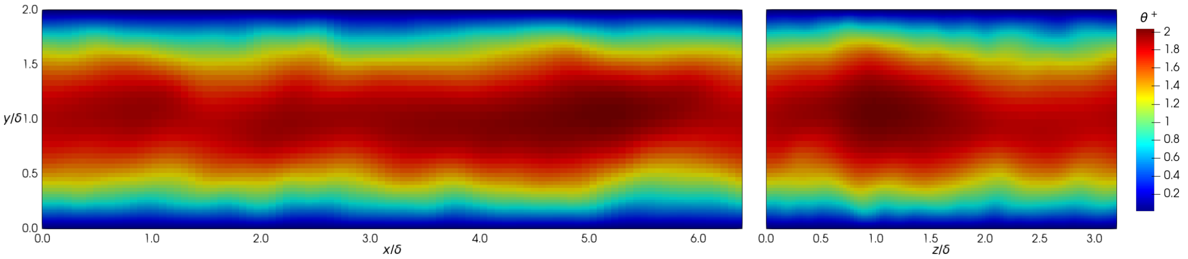


Figure 6.30: Streamwise (left) and spanwise (right) instantaneous temperature field for $Pr = 0.01$ (**M3**)

6.3 Π profiles

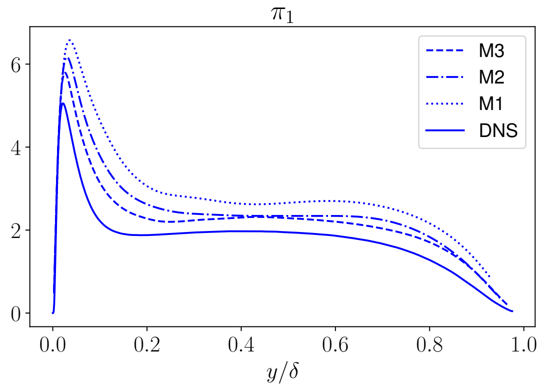


Figure 6.31: π_1, $Re_\tau = 395$

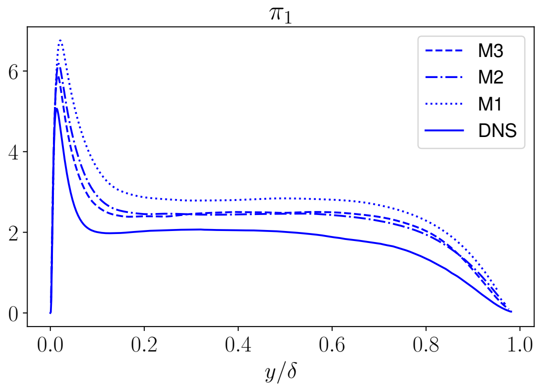


Figure 6.32: π_1, $Re_\tau = 640$

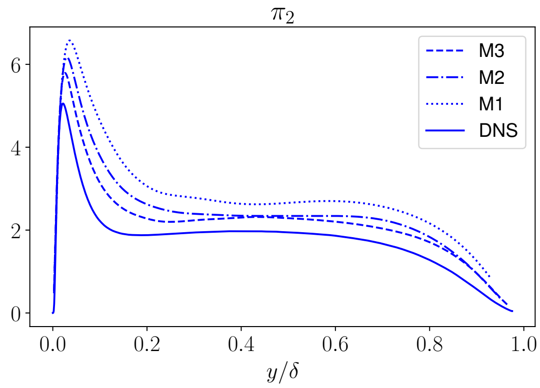


Figure 6.33: π_2, $Re_\tau = 395$

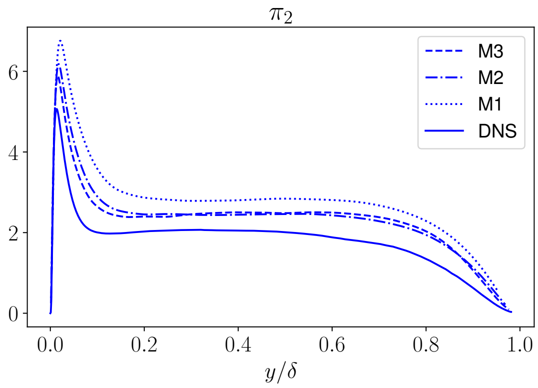


Figure 6.34: π_2, $Re_\tau = 640$

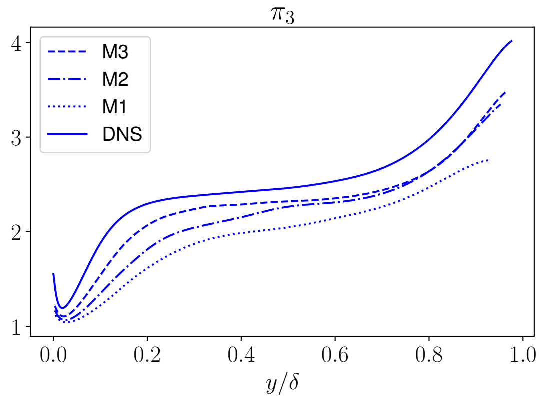


Figure 6.35: π_3, $Re_\tau = 395$

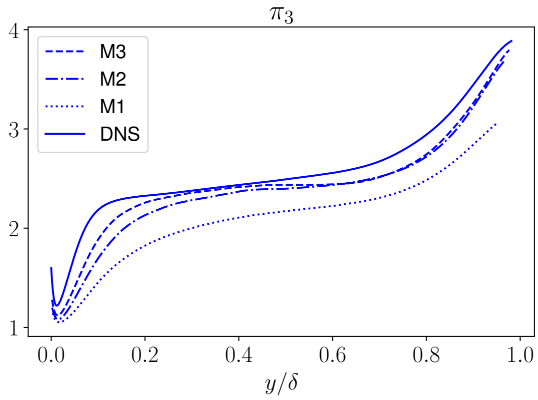


Figure 6.36: π_3, $Re_\tau = 640$

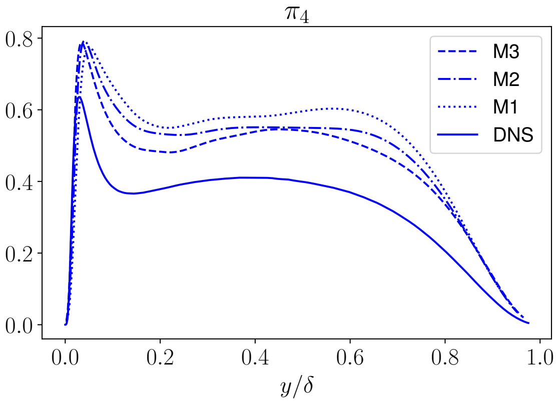


Figure 6.37: π_4, $Re_\tau = 395$

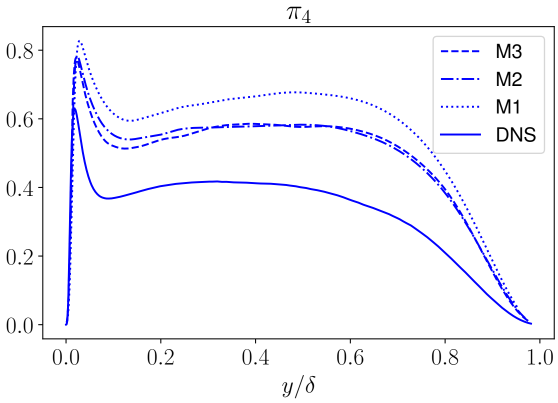


Figure 6.38: π_4, $Re_\tau = 640$

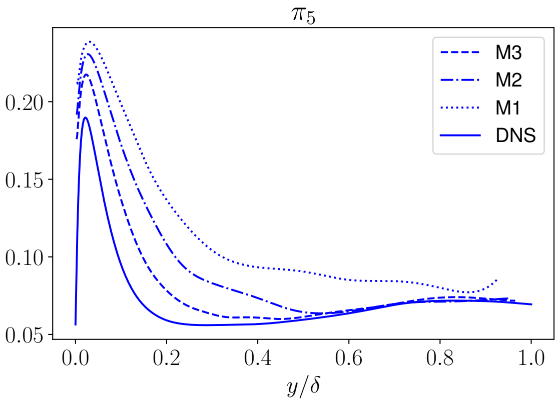


Figure 6.39: π_5, $Re_\tau = 395$

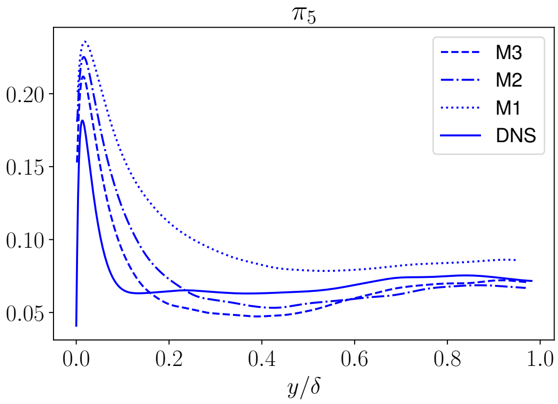


Figure 6.40: π_5, $Re_\tau = 640$

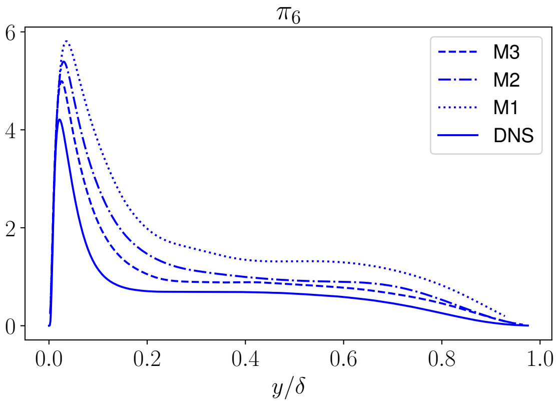


Figure 6.41: π_6, $Re_\tau = 395$

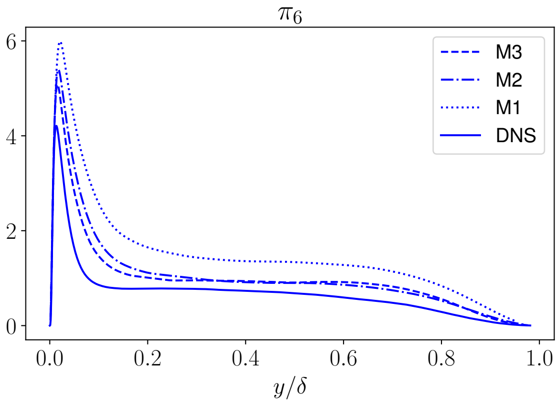


Figure 6.42: π_6, $Re_\tau = 640$

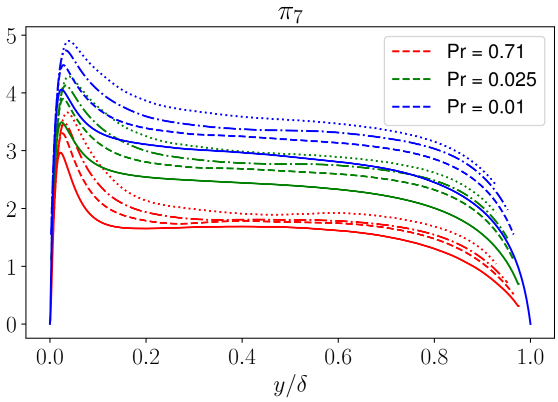


Figure 6.43: π_7, $Re_\tau = 395$

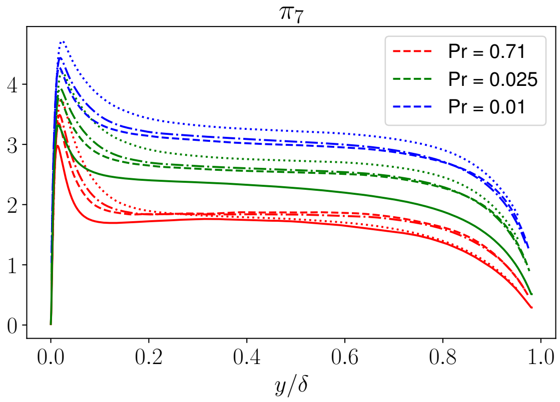


Figure 6.44: π_7, $Re_\tau = 640$

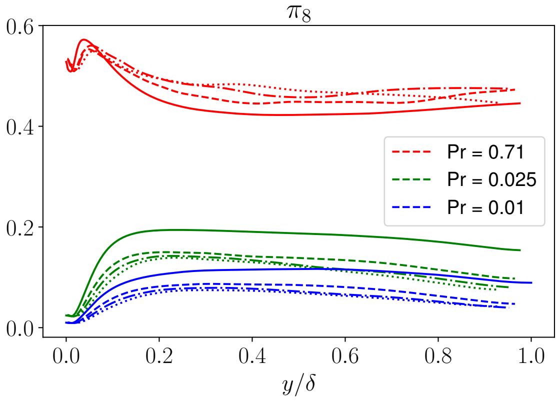


Figure 6.45: π_8, $Re_\tau = 395$

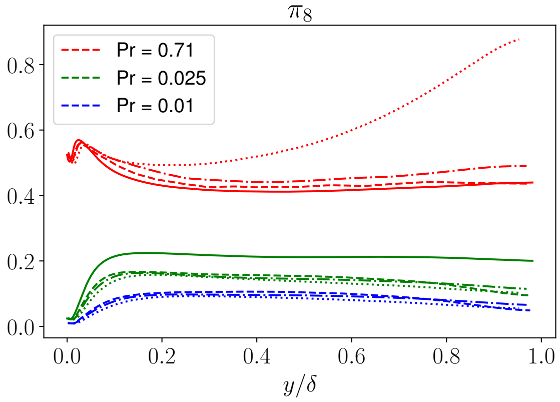


Figure 6.46: π_8, $Re_\tau = 640$

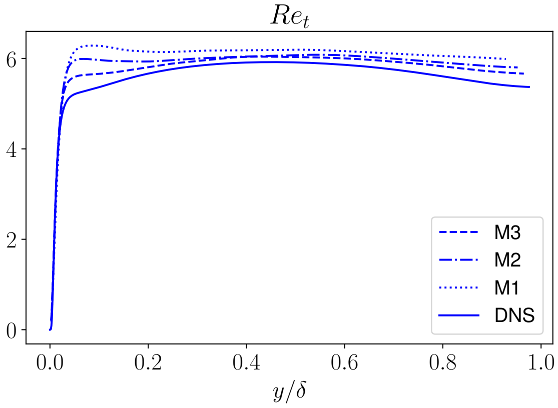


Figure 6.47: Re_t, $Re_\tau = 395$

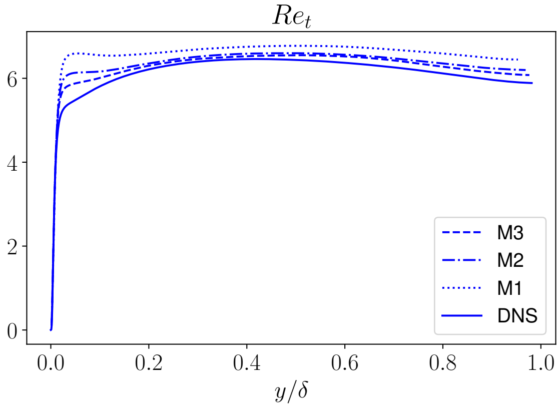


Figure 6.48: Re_t, $Re_\tau = 640$

6.4 Integrated Gradients

6.4.1 First linear path : $\mathbf{x}'_{(Re_\tau=640,Pr=0.71)} \longrightarrow \mathbf{x}_{(Re_\tau=395,Pr=0.025)}$

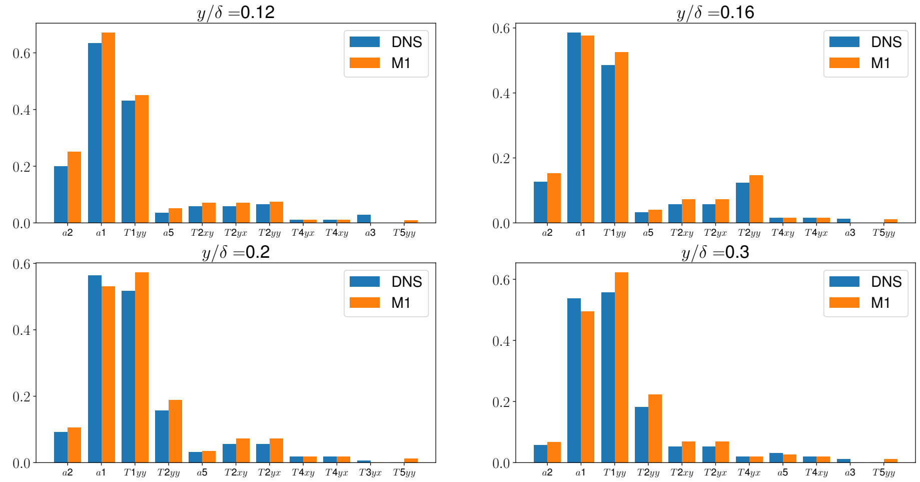


Figure 6.49: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between a first model trained with a DNS database and a second model trained with an **M1** database and initialised with the weights of the first model.

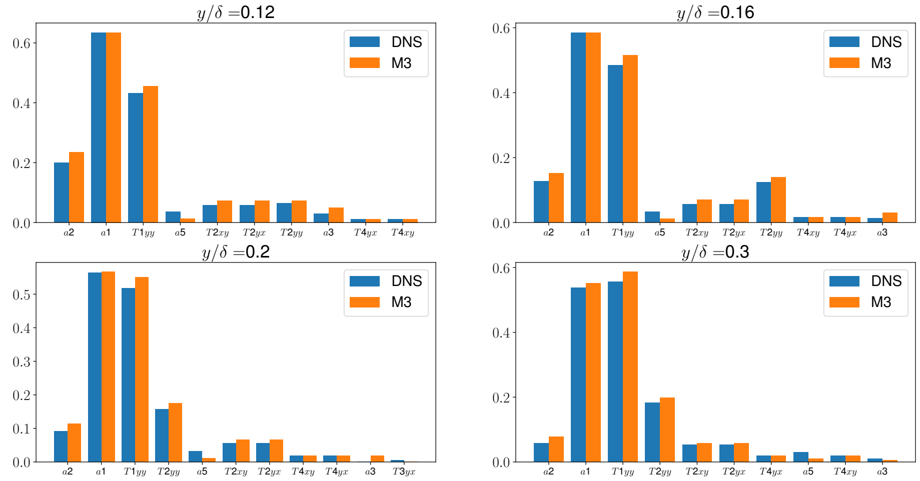


Figure 6.50: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between a first model trained with a DNS database and a second model trained with an **M3** database and initialised with the weights of the first model.

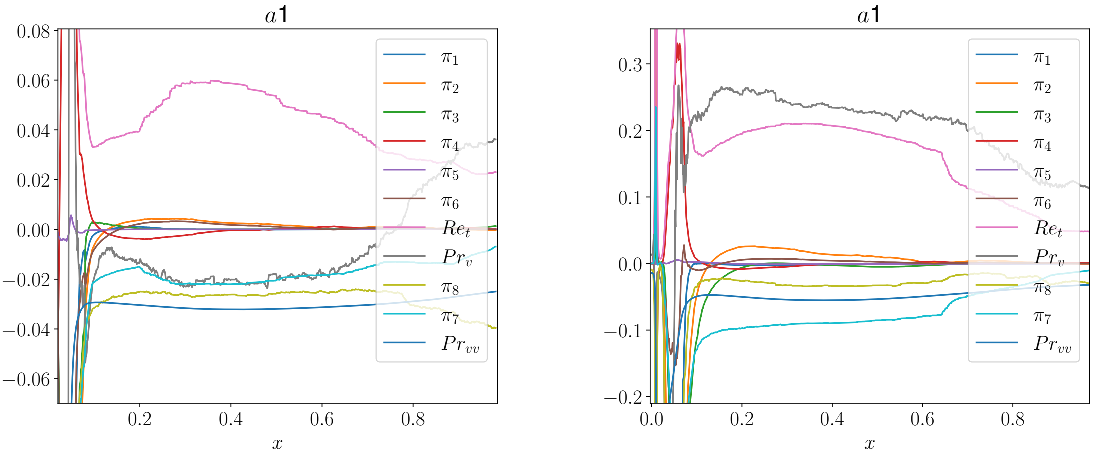


Figure 6.51: Attributions of the variation of a_1 to the invariant basis. model trained with an **M3** database (left) and model trained with an **M1** database both initialised with the weights of a model trained with a DNS data (right). Pr_v and Pr_{vv} designate the Prandtl number injected respectively in the first and second branch of the ANN.

6.4.2 Second linear path : $\mathbf{x}'_{(Re_\tau=640,Pr=0.71)} \longrightarrow \mathbf{x}_{(Re_\tau=640,Pr=0.025)}$

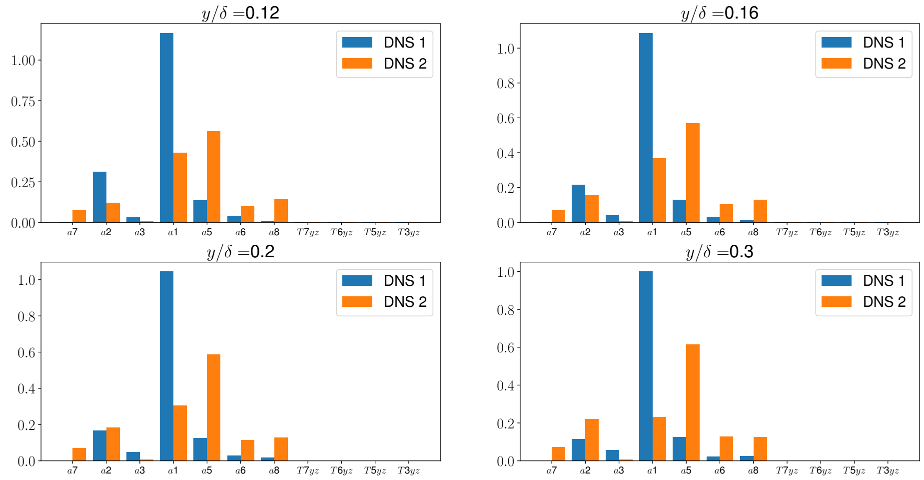


Figure 6.52: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between the base model trained with a DNS database and another model trained with the same database and initialised with random weights.

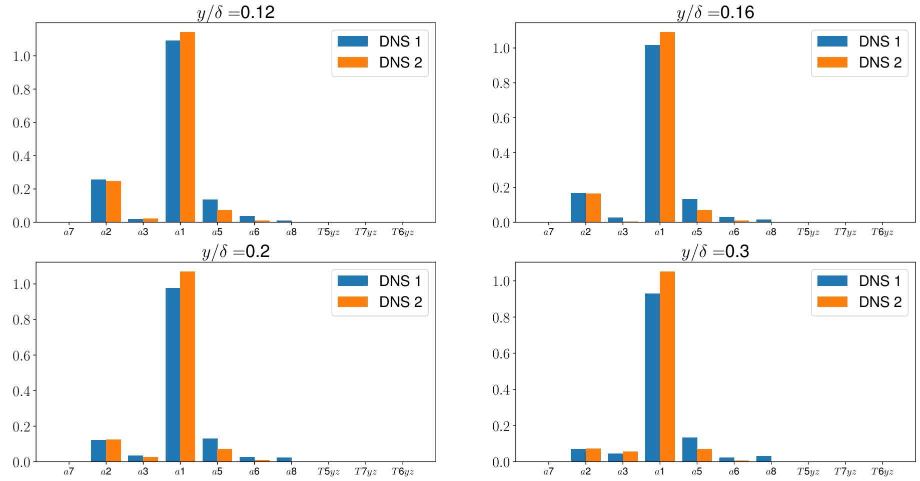


Figure 6.53: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between the base model trained with a DNS database and another model trained with the same database and initialised with the weights of the base model.

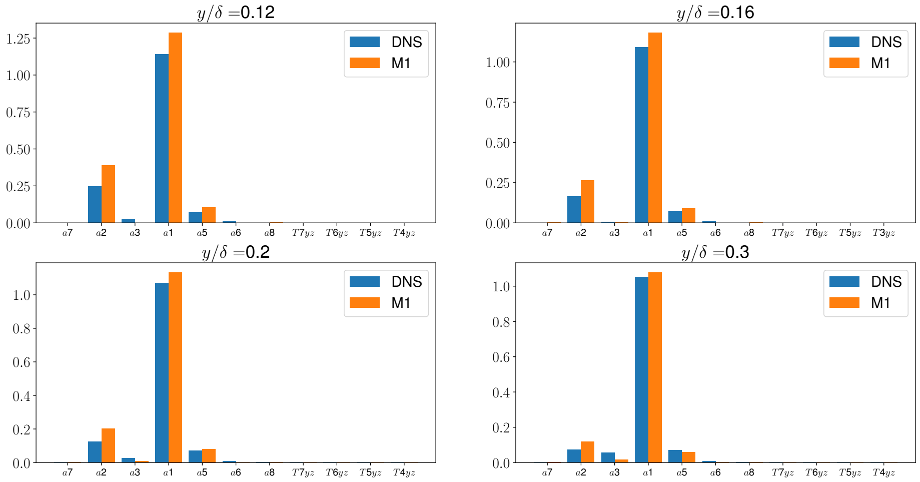


Figure 6.54: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between a first model trained with a DNS database and a second model trained with an **M1** database and initialised with the weights of the first model.

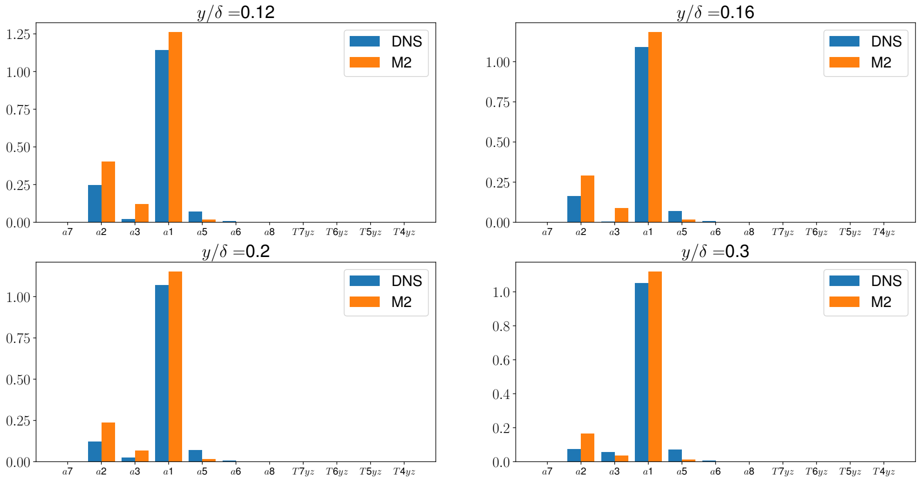


Figure 6.55: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between a first model trained with a DNS database and a second model trained with an **M2** database and initialised with the weights of the first model.

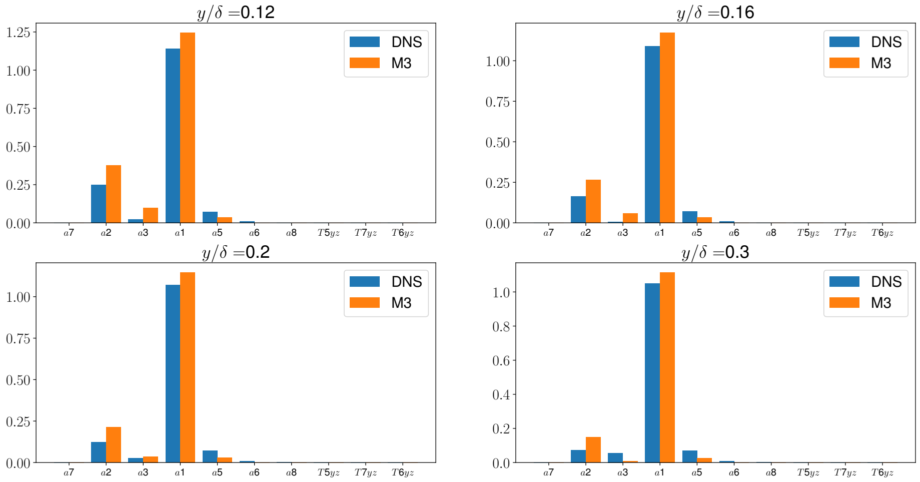


Figure 6.56: Maximum 10 attributions of the variation of the component D_{yy} to the coefficients a_i and tensors T_i at different distances from the wall. Comparison between a first model trained with a DNS database and a second model trained with an **M3** database and initialised with the weights of the first model.

6.5 Complete statistical analysis

This section provide the statistical analyses not presented in chapter 4.

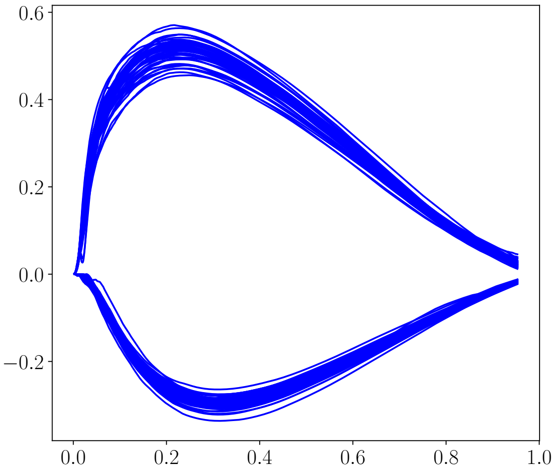


Figure 6.57: LES training statistical sample for $Pr = 0.025$ and $Re_\tau = 640$

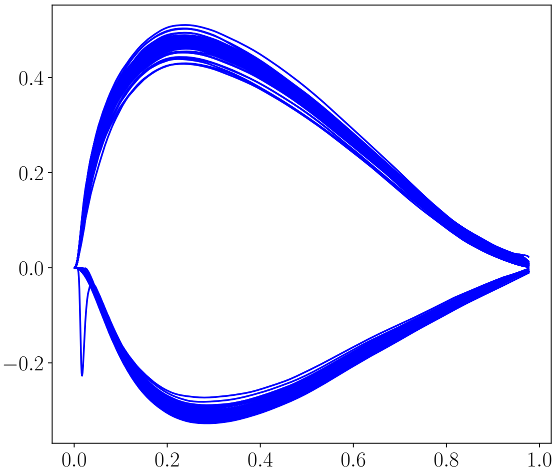


Figure 6.58: LES training statistical sample for $Pr = 0.025$ and $Re_\tau = 395$

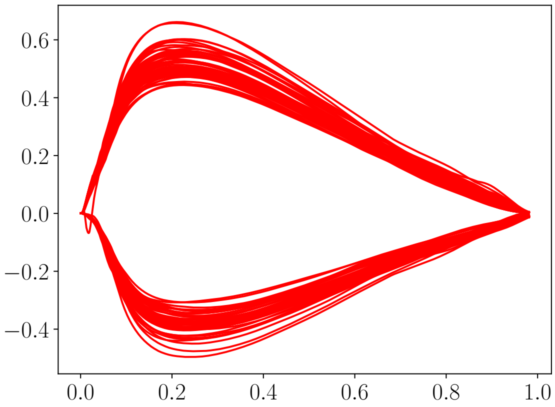


Figure 6.59: DNS validation statistical sample for $Pr = 0.025$ and $Re_\tau = 640$

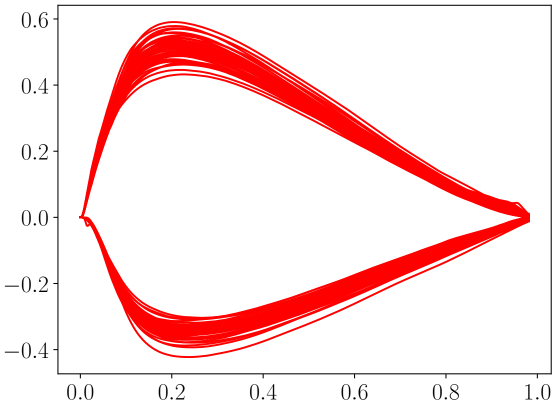
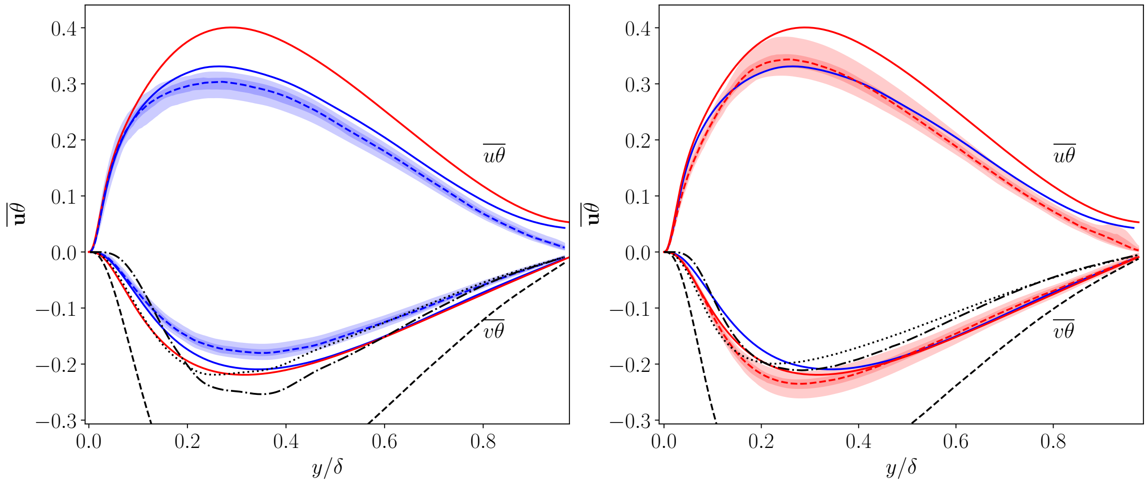
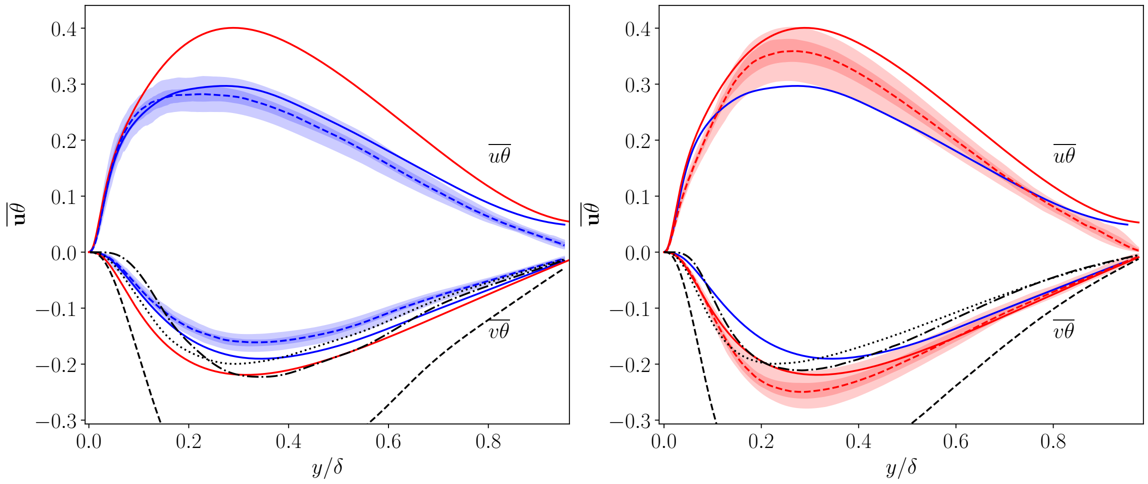


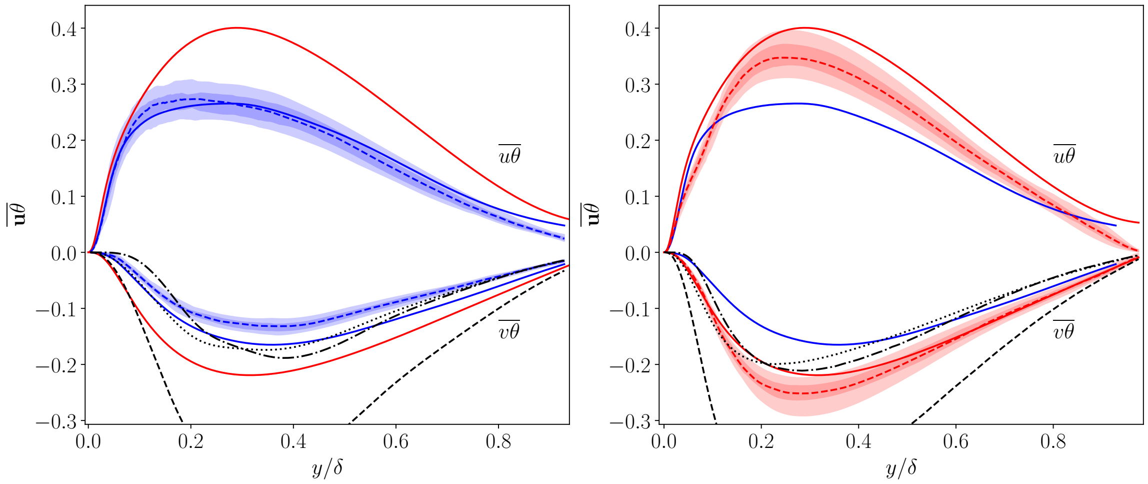
Figure 6.60: DNS validation statistical sample for $Pr = 0.025$ and $Re_\tau = 640$



(a) LES training (**M3**).   (b) DNS validation (**M3**).

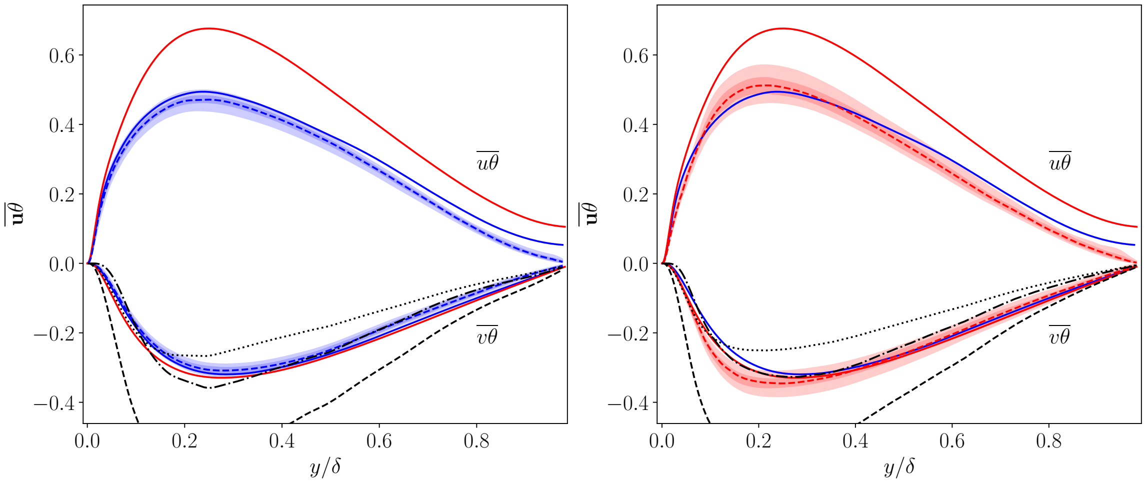


(c) LES training (**M2**).   (d) DNS validation (**M2**).



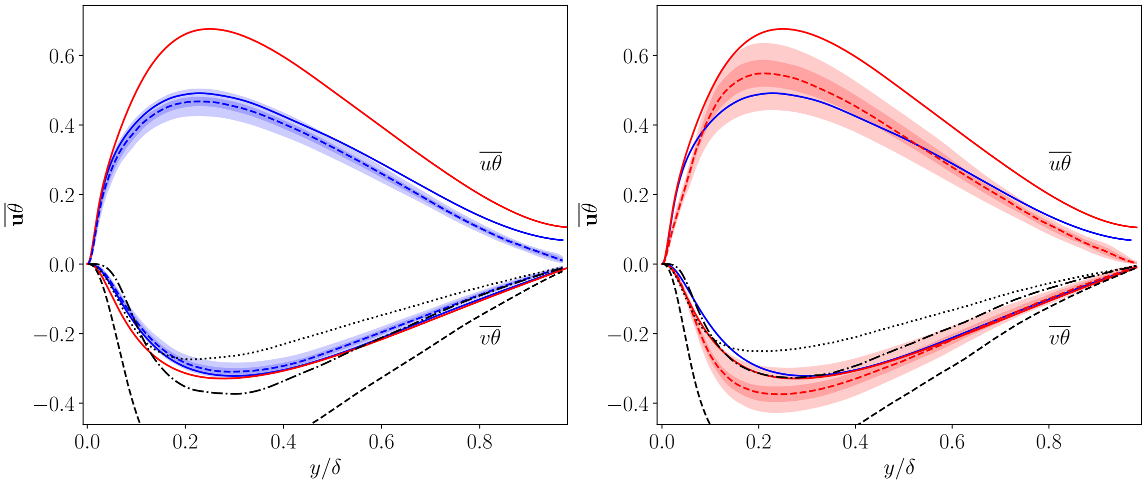
(e) LES training (**M1**).   (f) DNS validation (**M1**).

Figure 6.61: Statistical analysis of the THF predicted for $Re_\tau = 395$ and $Pr = 0.025$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



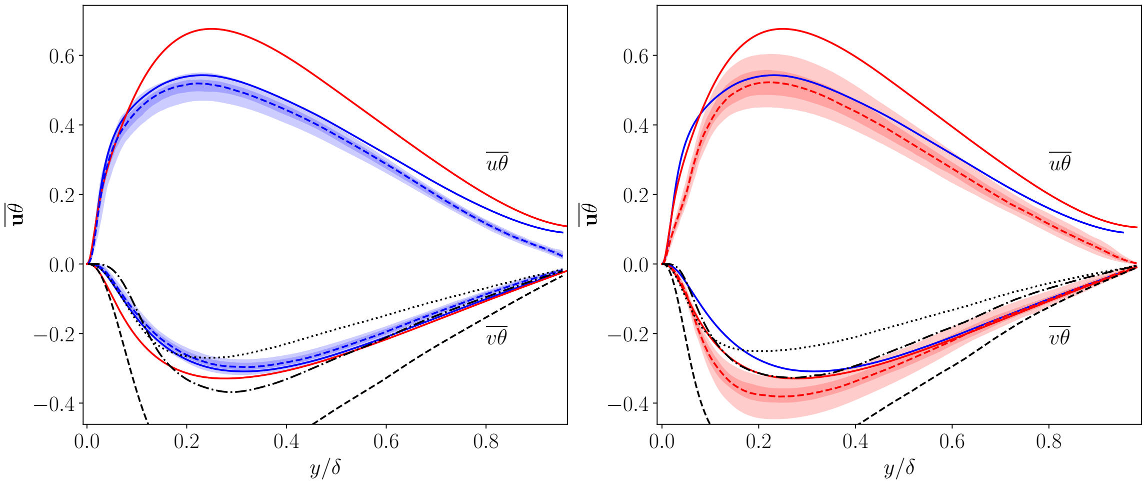
(a) LES training (**M3**).

(b) DNS validation (**M3**).



(c) LES training (**M2**).

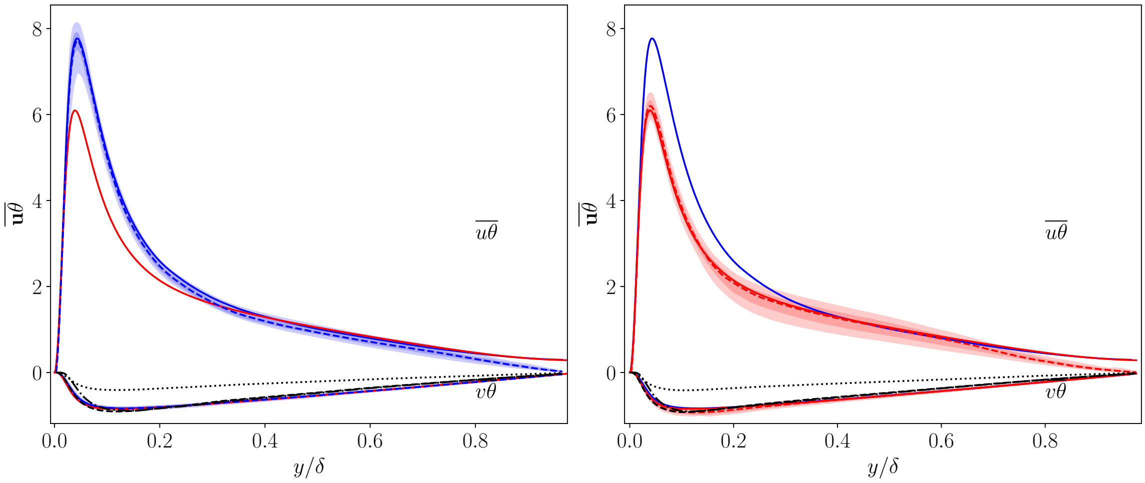
(d) DNS validation (**M2**).



(e) LES training (**M1**).

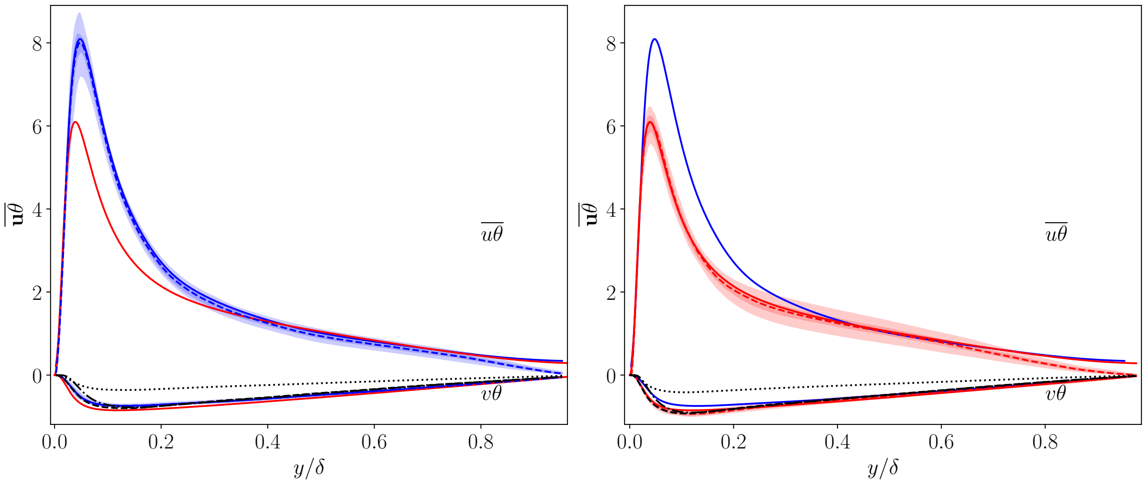
(f) DNS validation (**M1**).

Figure 6.62: Statistical analysis of the THF predicted for $Re_\tau = 640$ and $Pr = 0.025$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



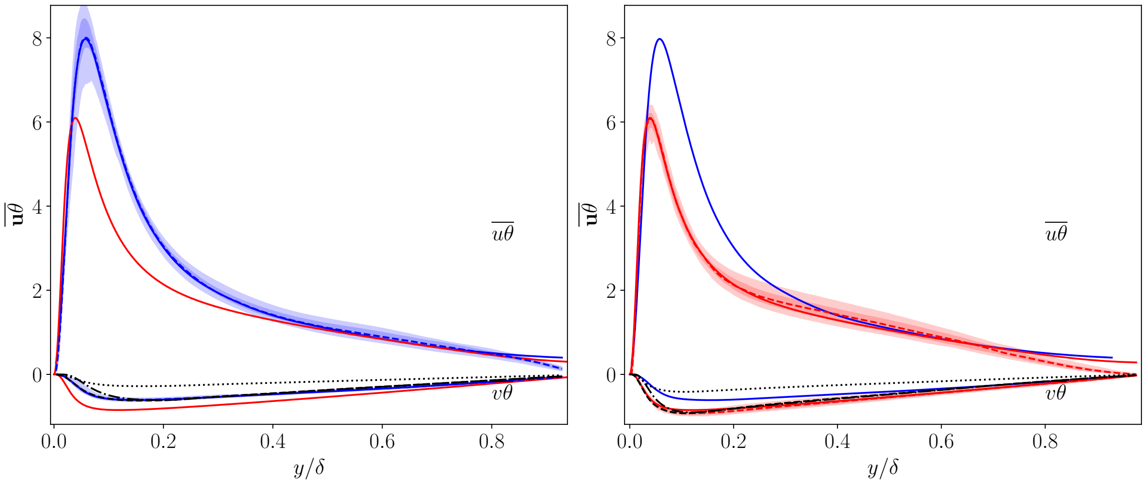
(a) LES training (**M3**).

(b) DNS validation (**M3**).



(c) LES training (**M2**).

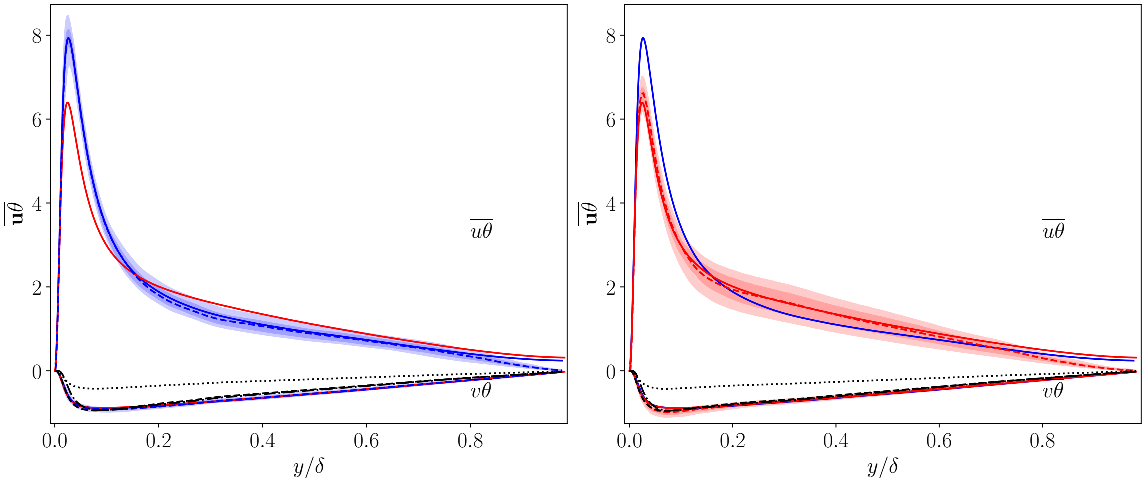
(d) DNS validation (**M2**).



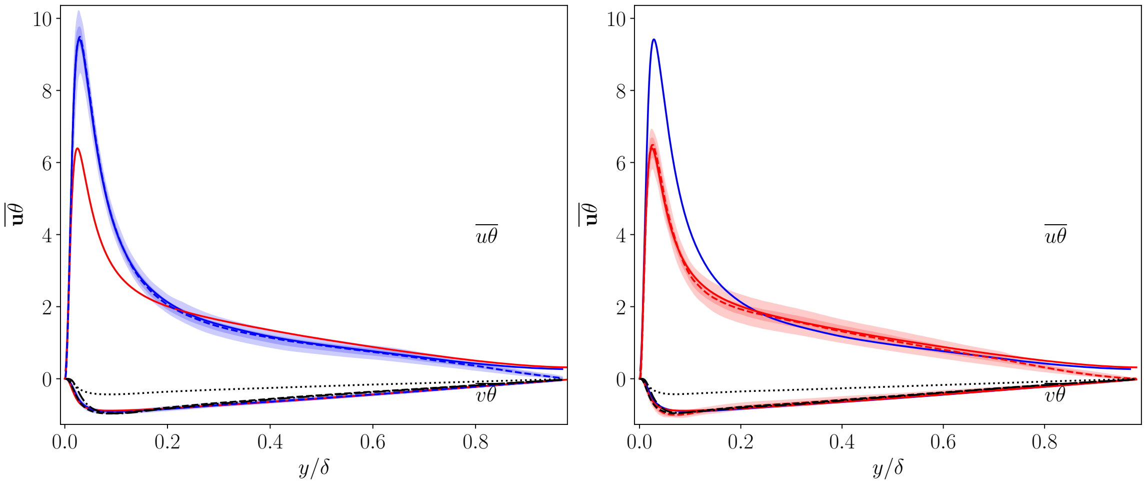
(e) LES training (**M1**).

(f) DNS validation (**M1**).

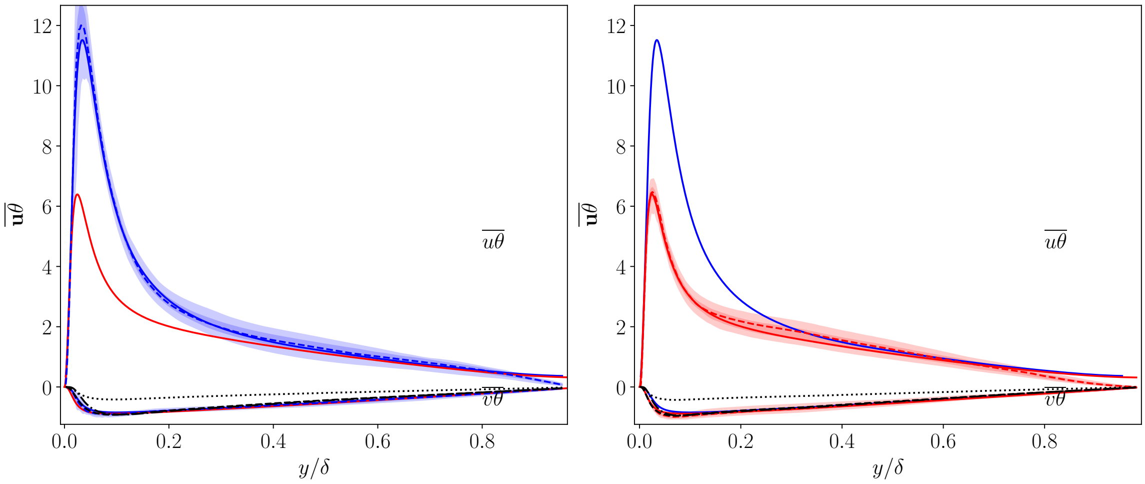
Figure 6.63: Statistical analysis of the THF predicted for $Re_\tau = 395$ and $Pr = 0.71$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



(a) LES training (**M3**). (b) DNS validation (**M3**).

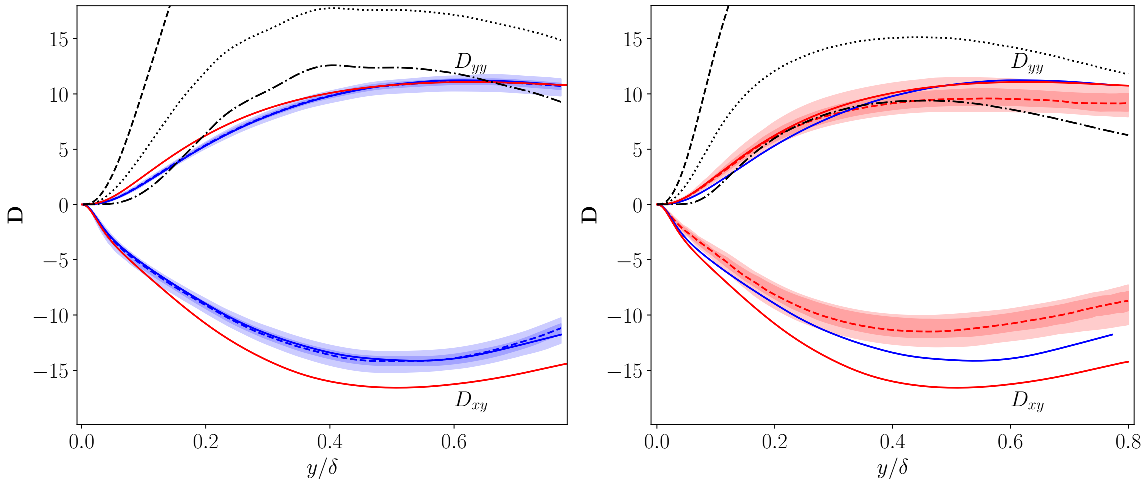


(c) LES training (**M2**). (d) DNS validation (**M2**).

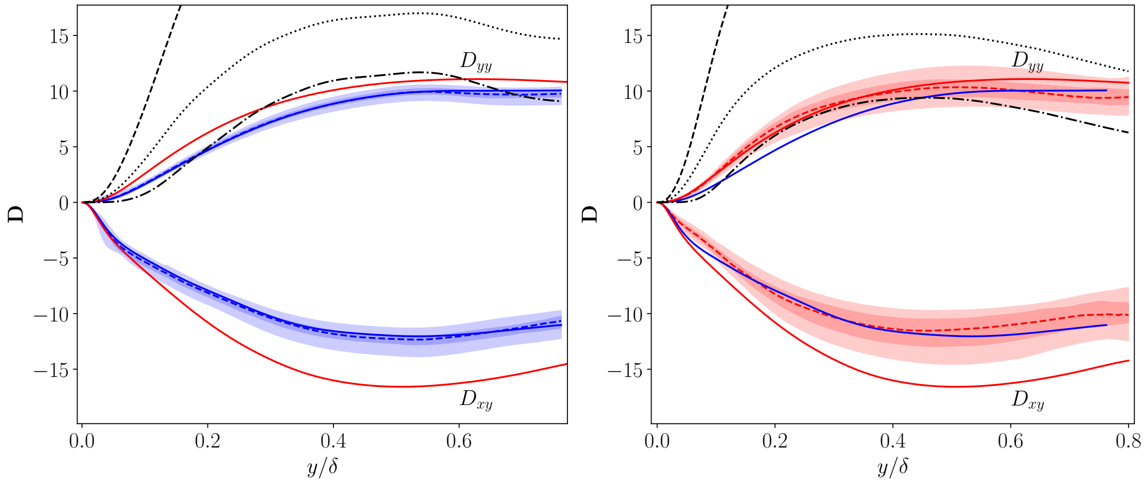


(e) LES training (**M1**). (f) DNS validation (**M1**).

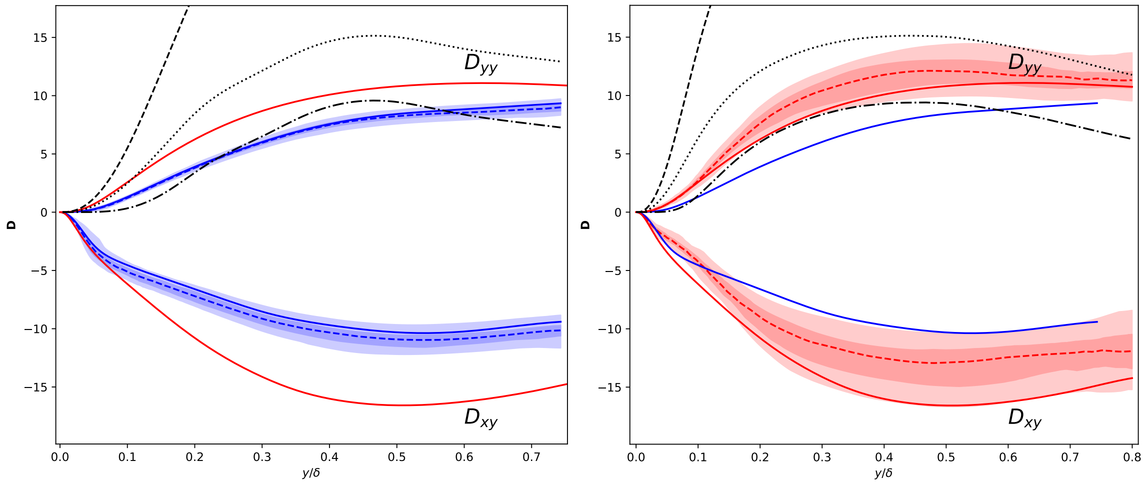
Figure 6.64: Statistical analysis of the THF predicted for $Re_\tau = 640$ and $Pr = 0.71$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



(a) LES training (**M3**). (b) DNS validation (**M3**).

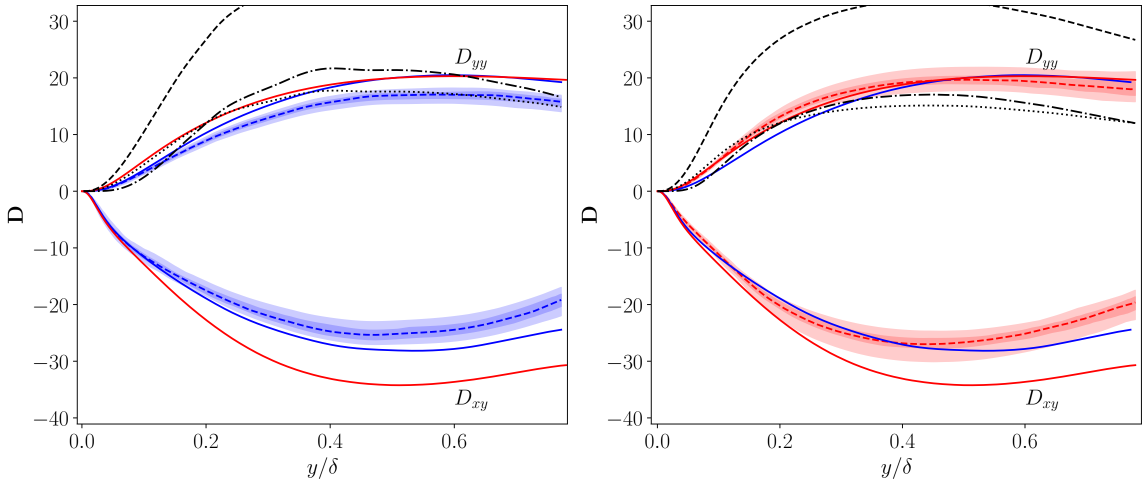


(c) LES training (**M2**). (d) DNS validation (**M2**).

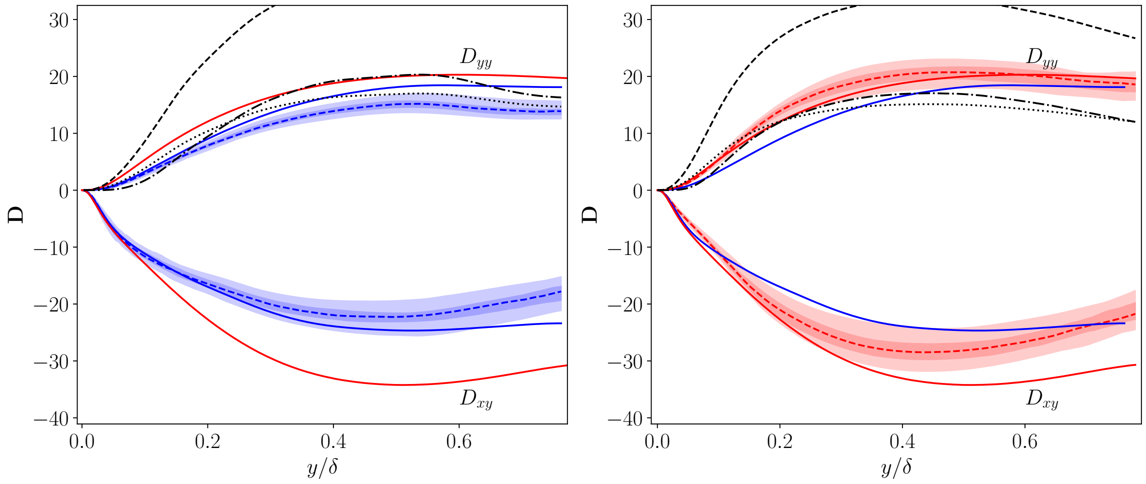


(e) LES training (**M1**). (f) DNS validation (**M1**).

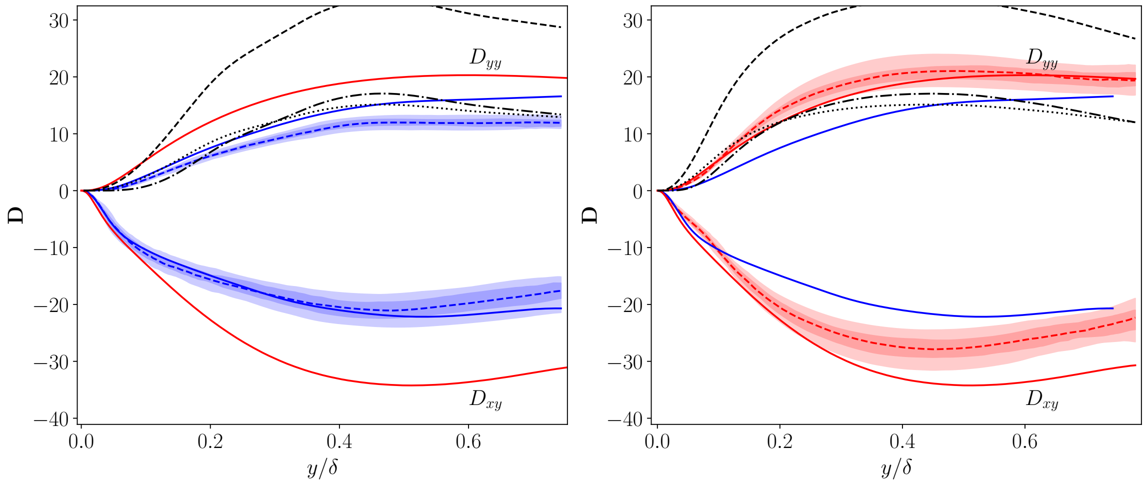
Figure 6.65: Statistical analysis of the relevant components of the **D** tensor for $Re_\tau = 395$ and $Pr = 0.01$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



(a) LES training (**M3**). (b) DNS validation (**M3**).

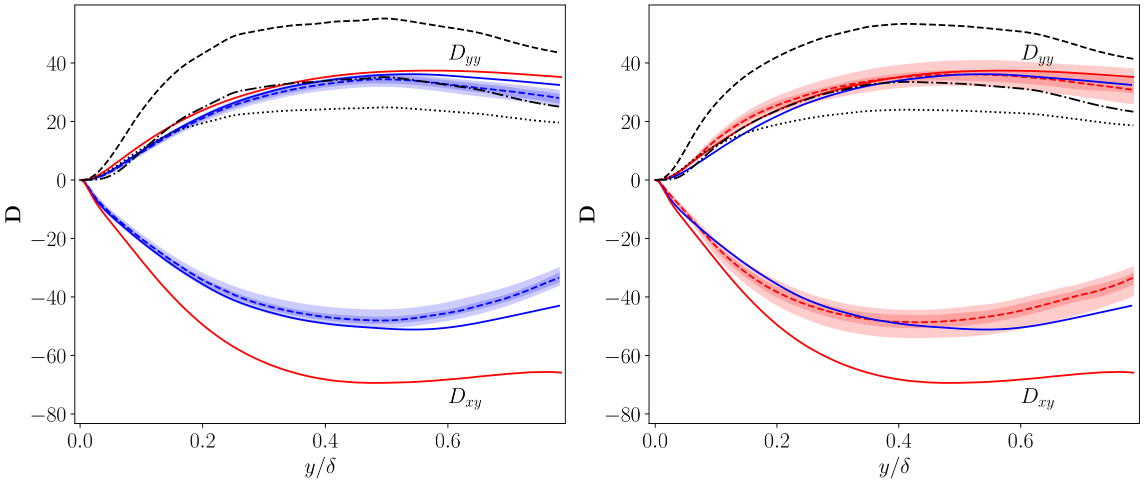


(c) LES training (**M2**). (d) DNS validation (**M2**).



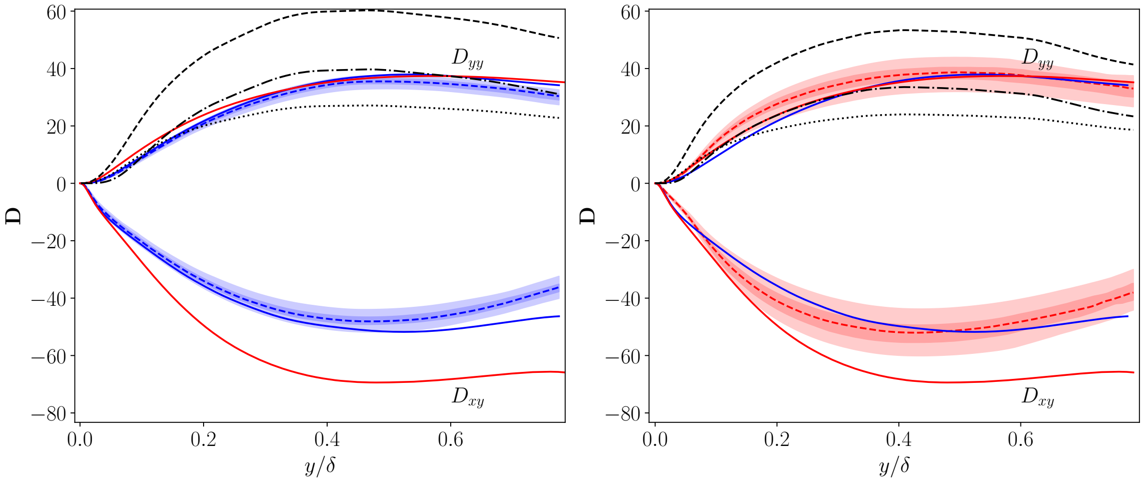
(e) LES training (**M1**). (f) DNS validation (**M1**).

Figure 6.66: Statistical analysis of the relevant components of the **D** tensor for $Re_\tau = 395$ and $Pr = 0.025$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



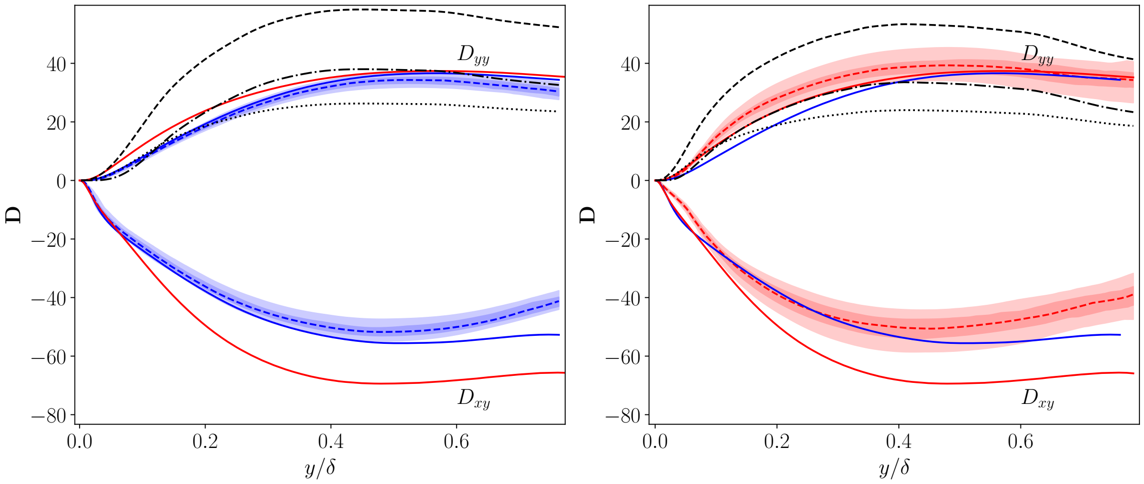
(a) LES training (**M3**).

(b) DNS validation (**M3**).



(c) LES training (**M2**).

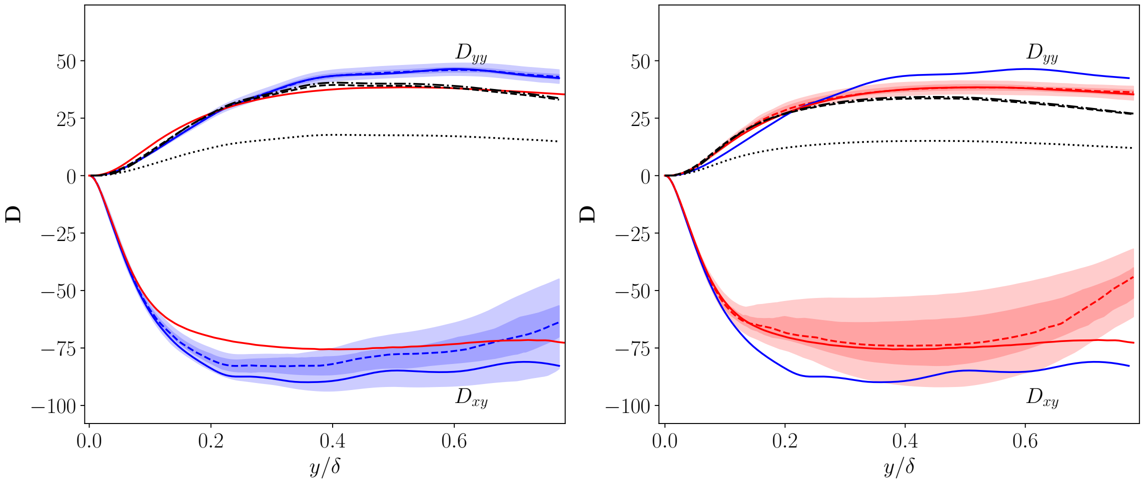
(d) DNS validation (**M2**).



(e) LES training (**M1**).

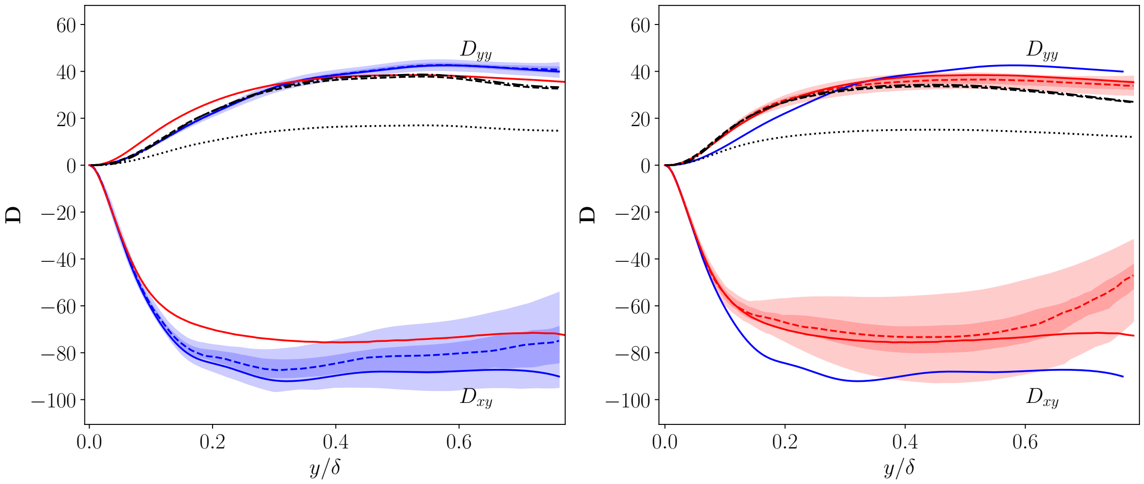
(f) DNS validation (**M1**).

Figure 6.67: Statistical analysis of the relevant components of the $\mathbf{D}$ tensor predicted for $Re_\tau = 640$ and $Pr = 0.025$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



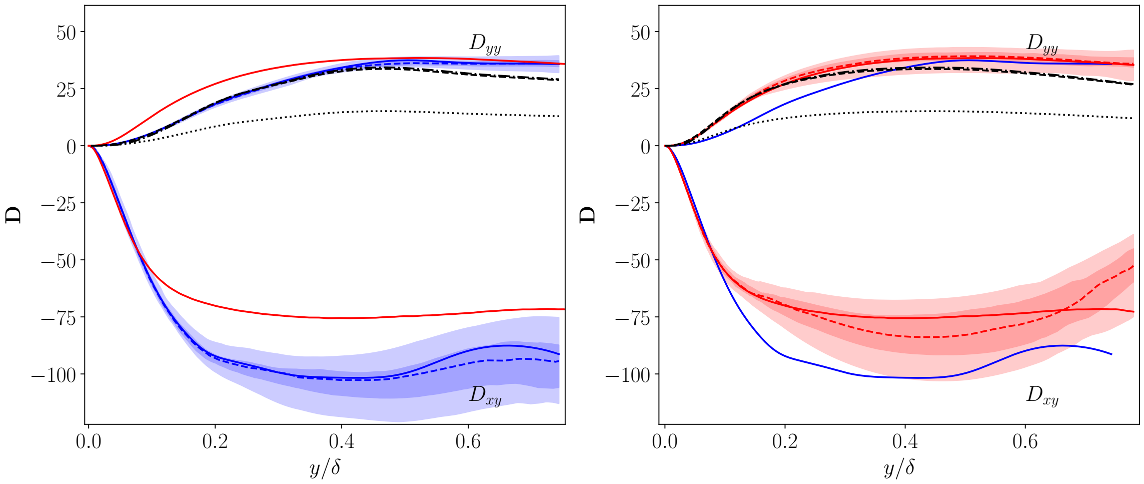
(a) LES training (**M3**).

(b) DNS validation (**M3**).



(c) LES training (**M2**).

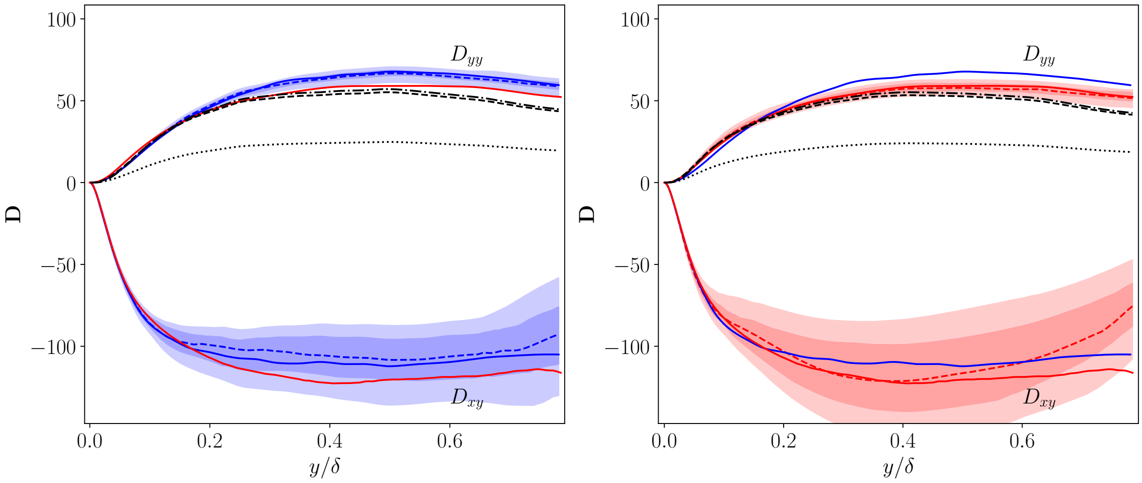
(d) DNS validation (**M2**).



(e) LES training (**M1**).

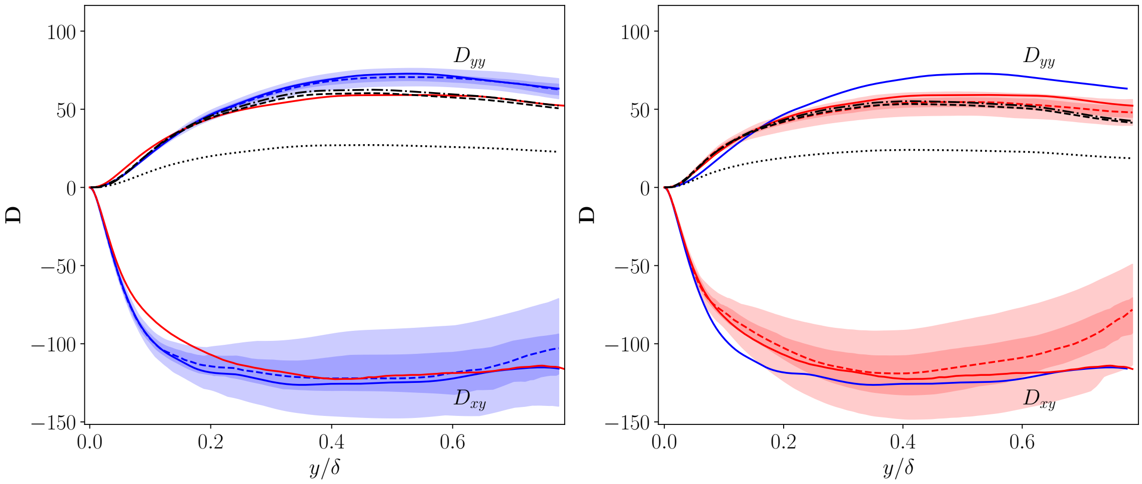
(f) DNS validation (**M1**).

Figure 6.68: Statistical analysis of the relevant components of the $\mathbf{D}$ tensor predicted for $Re_\tau = 395$ and $Pr = 0.71$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



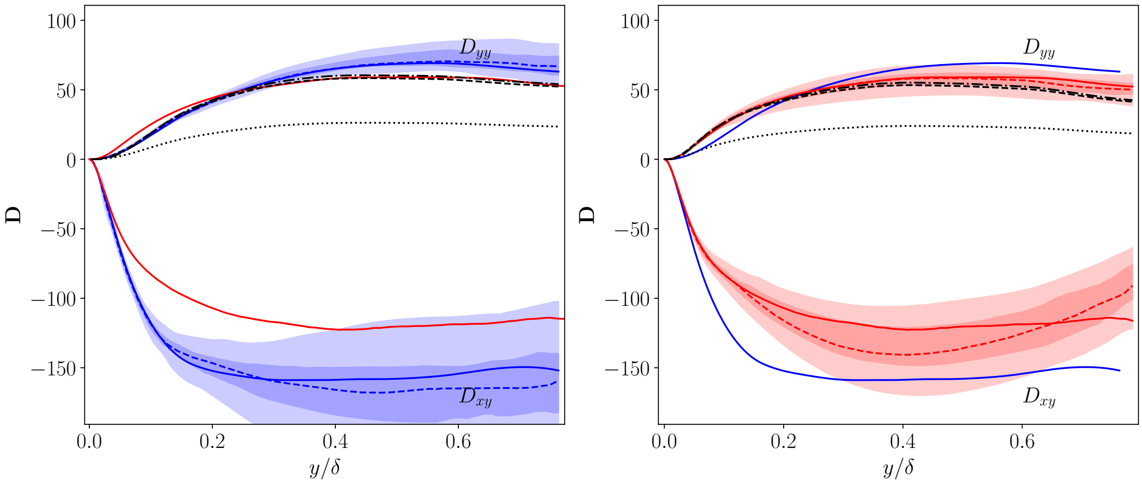
(a) LES training (**M3**).

(b) DNS validation (**M3**).



(c) LES training (**M2**).

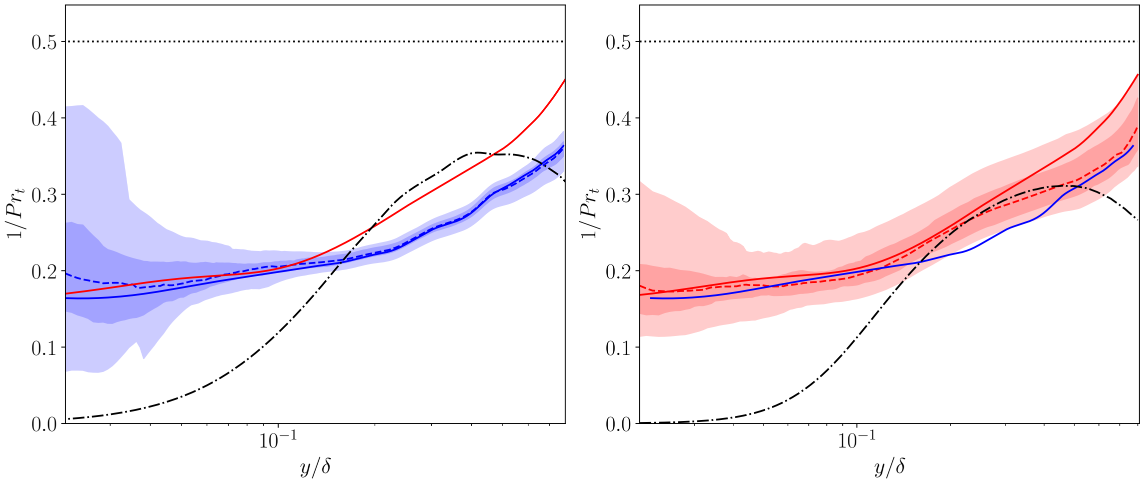
(d) DNS validation (**M2**).



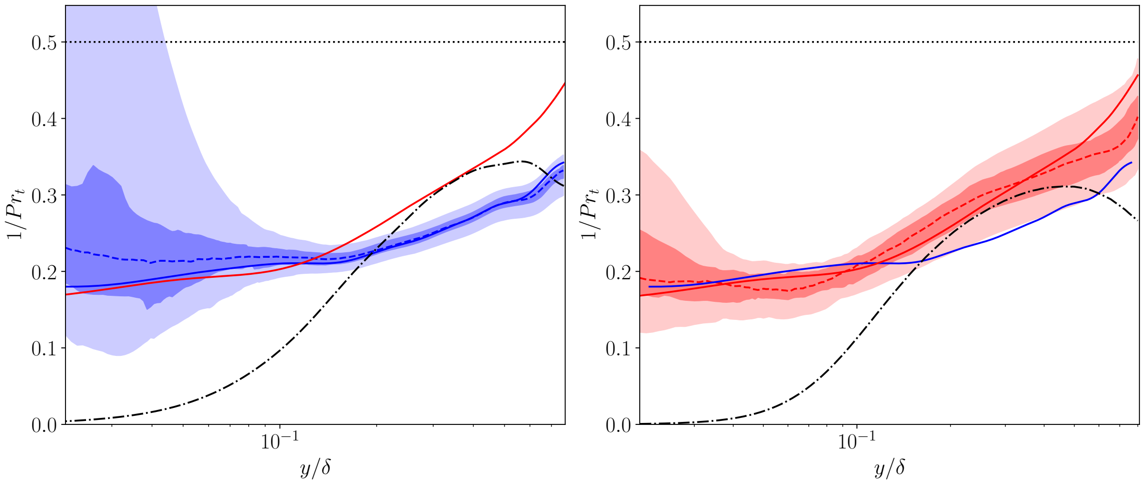
(e) LES training (**M1**).

(f) DNS validation (**M1**).

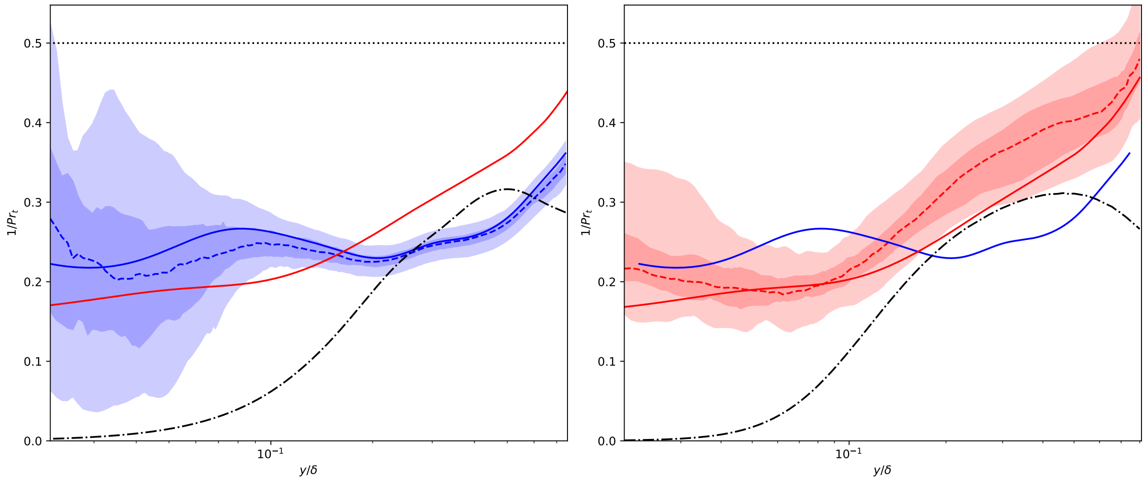
Figure 6.69: Statistical analysis of the relevant components of the $\mathbf{D}$ tensor predicted for $Re_\tau = 640$ and $Pr = 0.71$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



(a) LES training (**M3**). (b) DNS validation (**M3**).

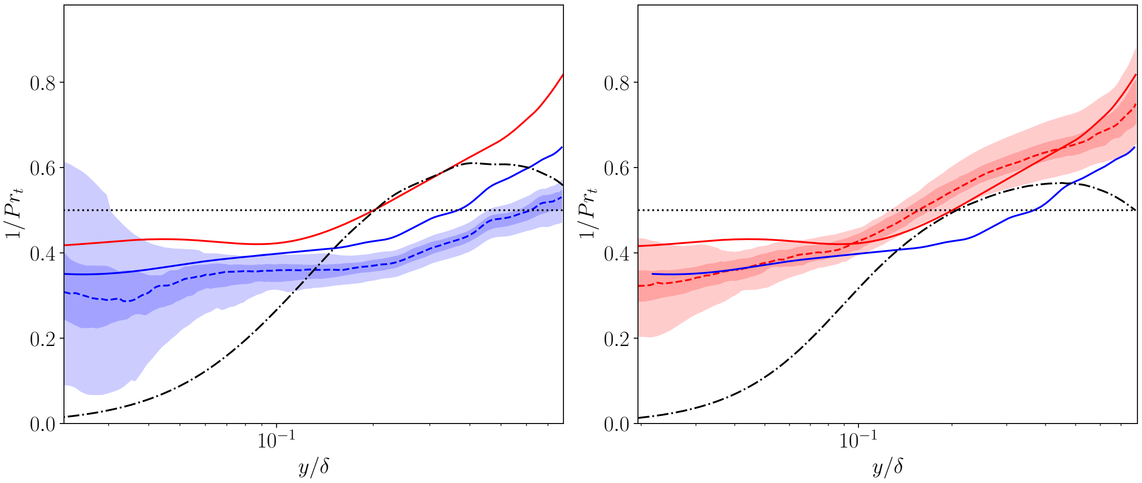


(c) LES training (**M2**). (d) DNS validation (**M2**).

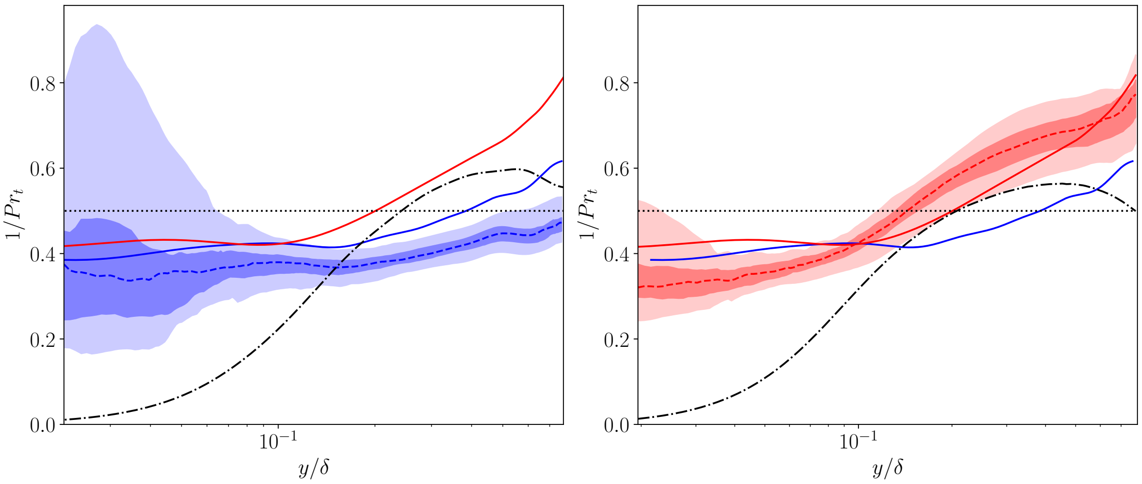


(e) LES training (**M1**). (f) DNS validation (**M1**).

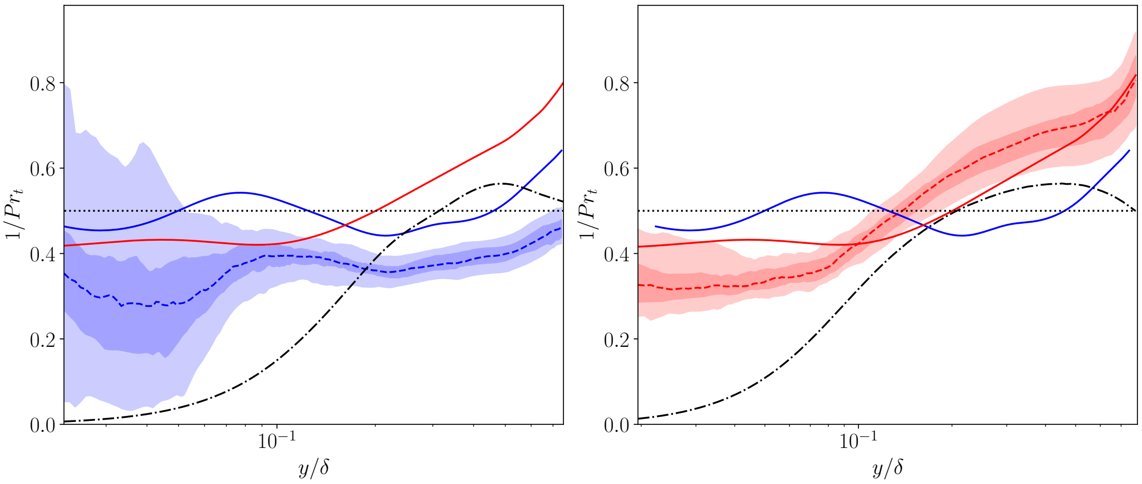
Figure 6.70: Statistical analysis of the inverse of Pr_t predicted for $Re_\tau = 395$ and $Pr = 0.01$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



(a) LES training (**M3**).
(b) DNS validation (**M3**).

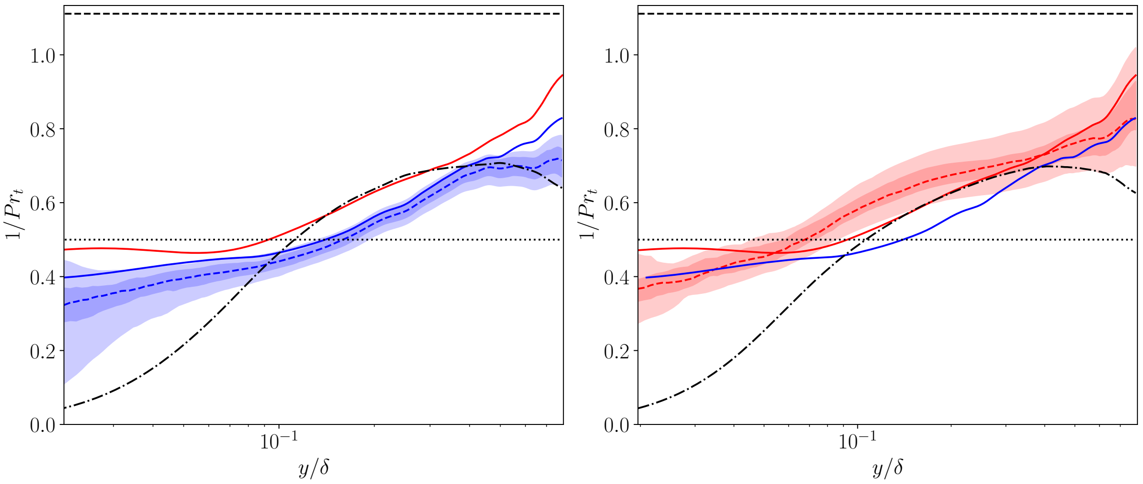


(c) LES training (**M2**).
(d) DNS validation (**M2**).

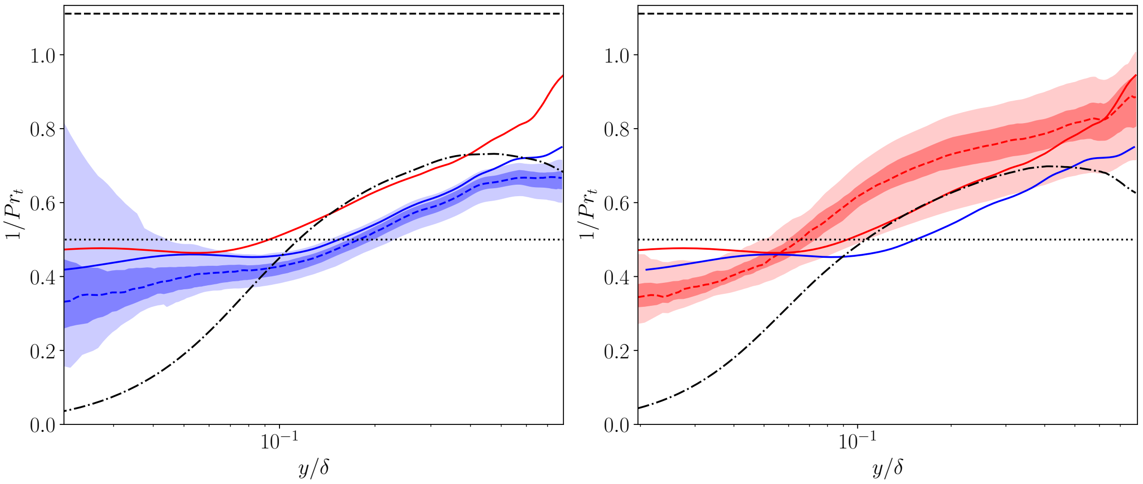


(e) LES training (**M1**).
(f) DNS validation (**M1**).

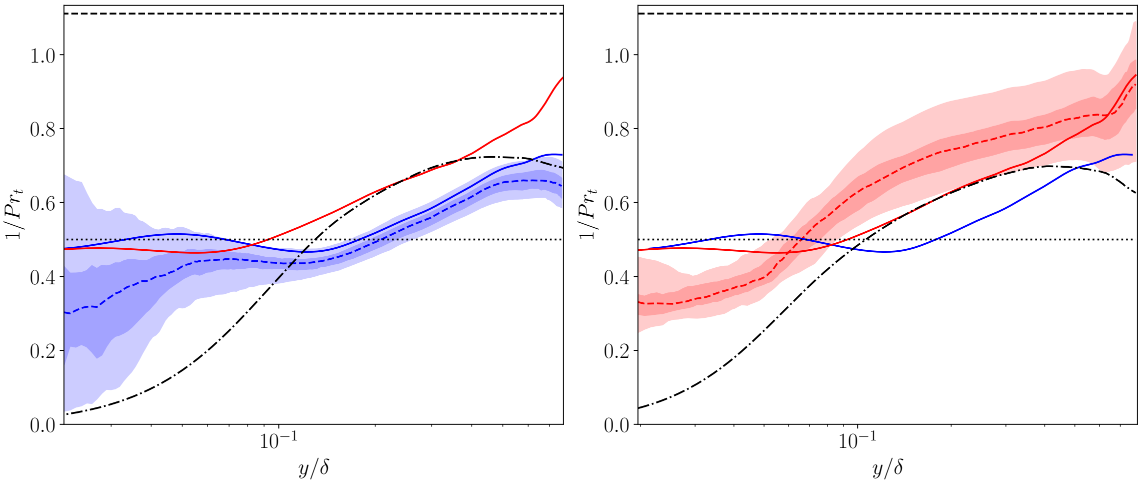
Figure 6.71: Statistical analysis of the inverse of Pr_t predicted for $Re_\tau = 395$ and $Pr = 0.025$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



(a) LES training (**M3**). (b) DNS validation (**M3**).

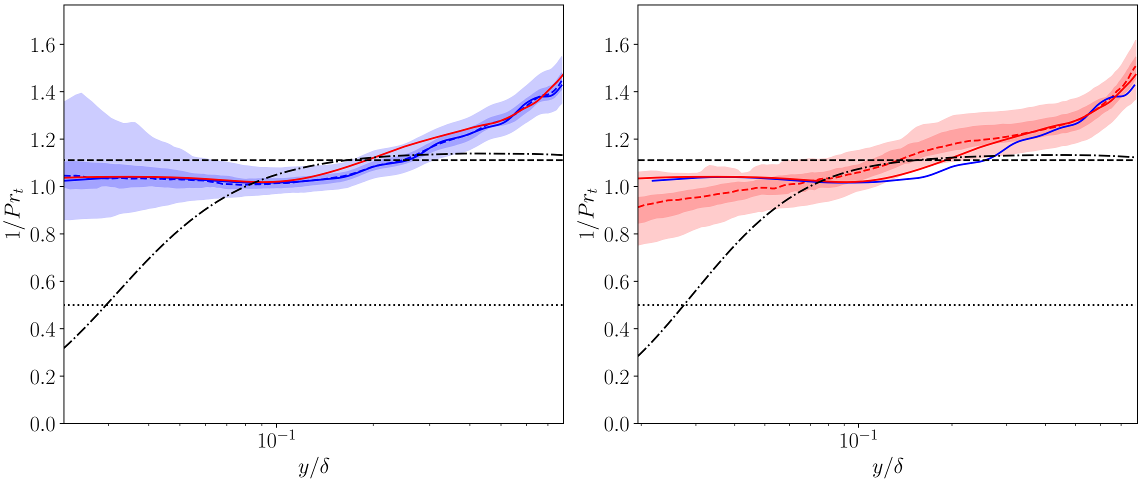


(c) LES training (**M2**). (d) DNS validation (**M2**).



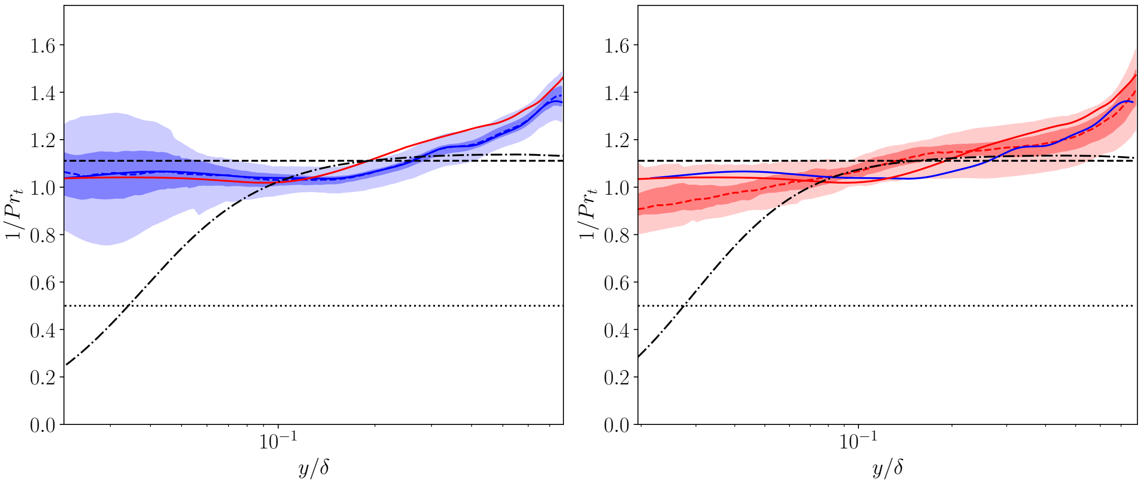
(e) LES training (**M1**). (f) DNS validation (**M1**).

Figure 6.72: Statistical analysis of the inverse of Pr_t predicted for $Re_\tau = 640$ and $Pr = 0.025$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



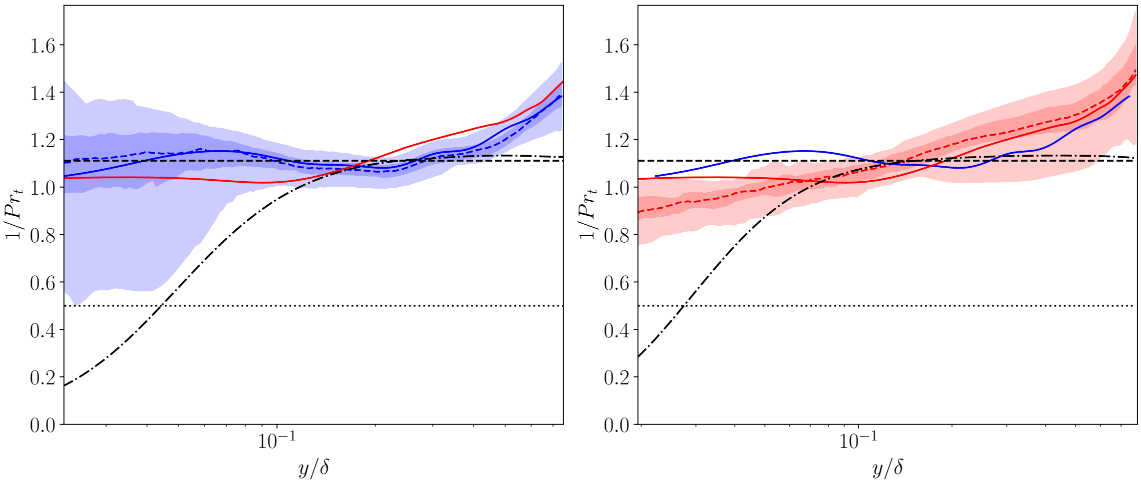
(a) LES training (**M3**).

(b) DNS validation (**M3**).



(c) LES training (**M2**).

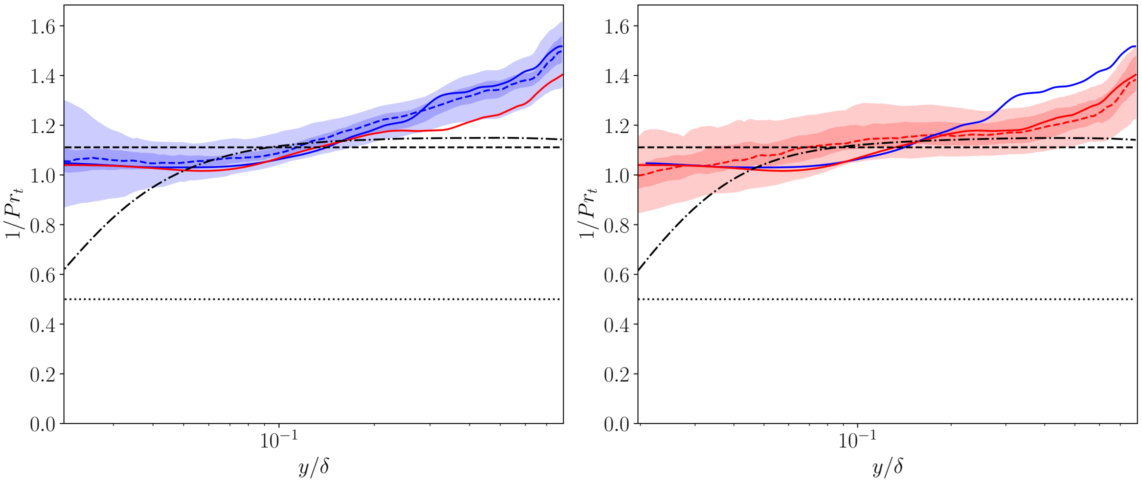
(d) DNS validation (**M2**).



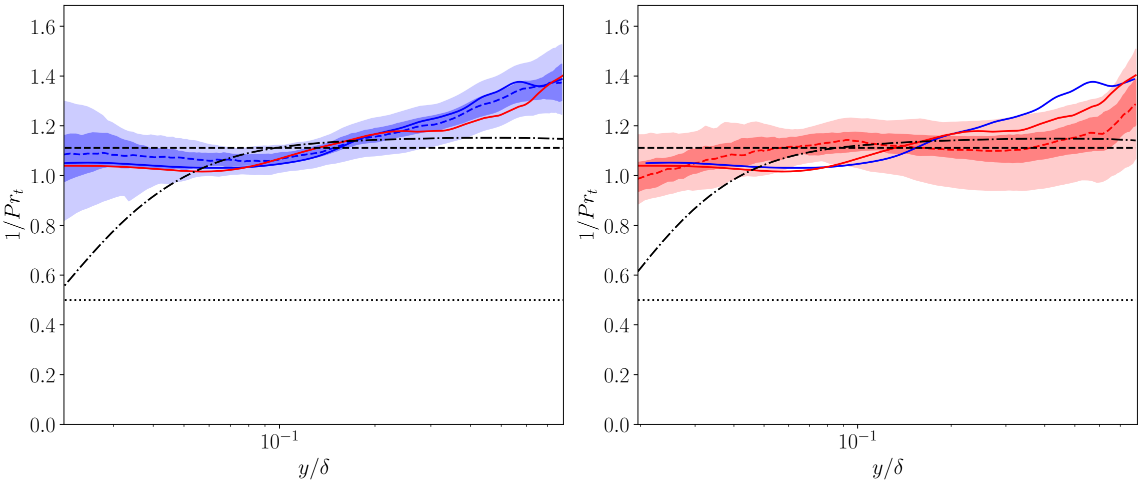
(e) LES training (**M1**).

(f) DNS validation (**M1**).

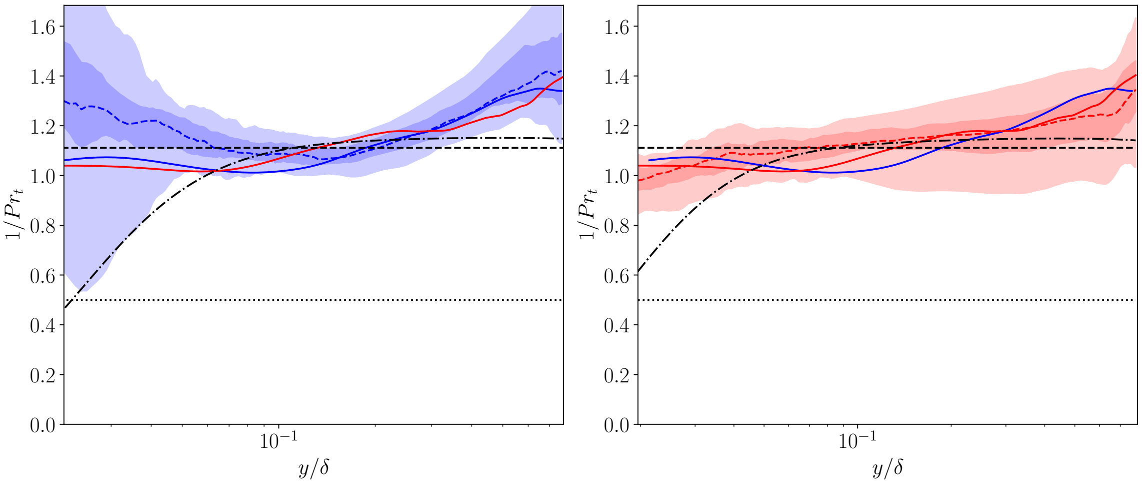
Figure 6.73: Statistical analysis of the inverse of Pr_t predicted for $Re_\tau = 395$ and $Pr = 0.71$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



(a) LES training (**M3**). (b) DNS validation (**M3**).

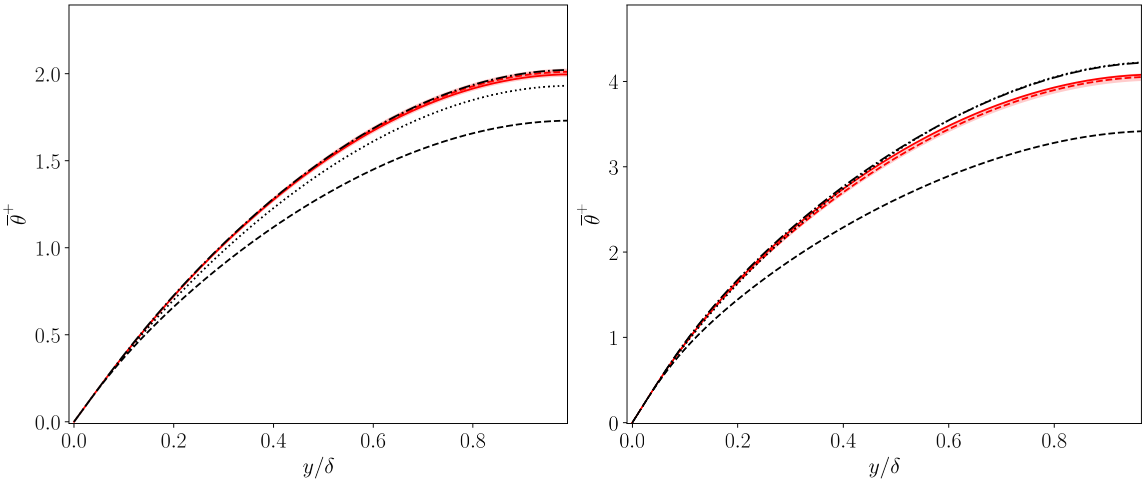


(c) LES training (**M2**). (d) DNS validation (**M2**).



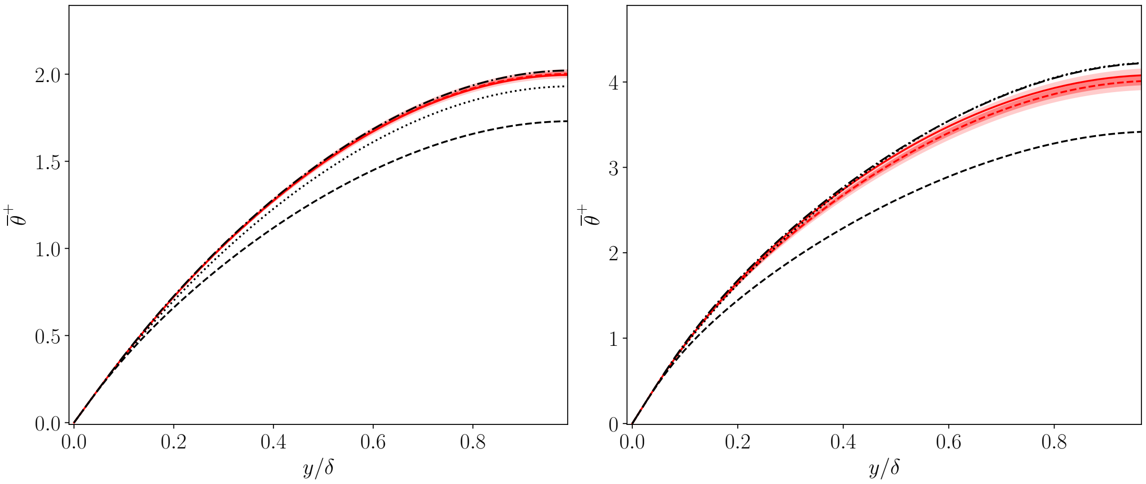
(e) LES training (**M1**). (f) DNS validation (**M1**).

Figure 6.74: Statistical analysis of the inverse of Pr_t predicted for $Re_\tau = 640$ and $Pr = 0.71$. DNS (solid red), LES data (solid blue), RA(dash black), RA_2(dot black), Kays(dot-dash black), median LES training(dash blue) and DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.



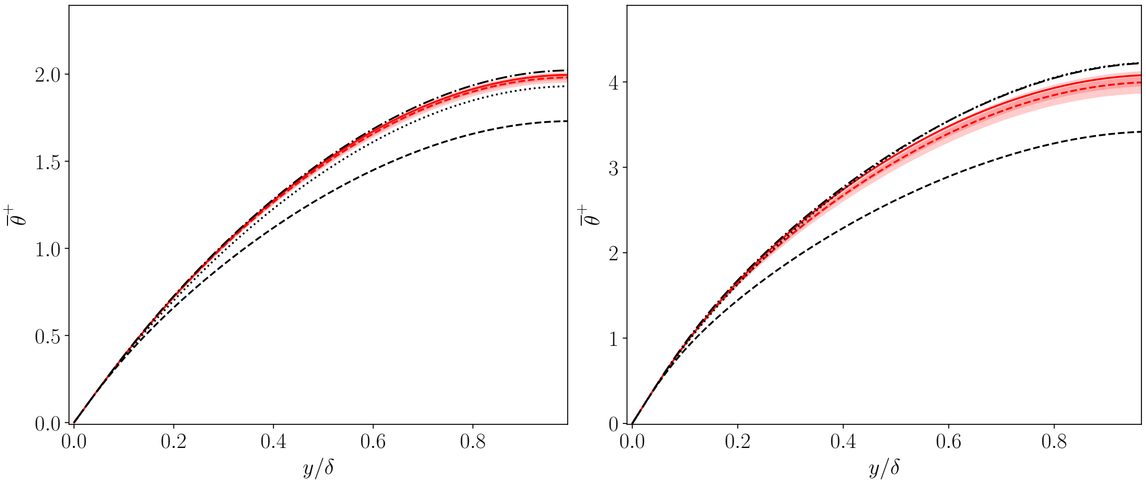
(a) $Pr = 0.01$, **M3**.

(b) $Pr = 0.025$, **M3**.



(c) $Pr = 0.01$, **M2**.

(d) $Pr = 0.025$, **M2**.

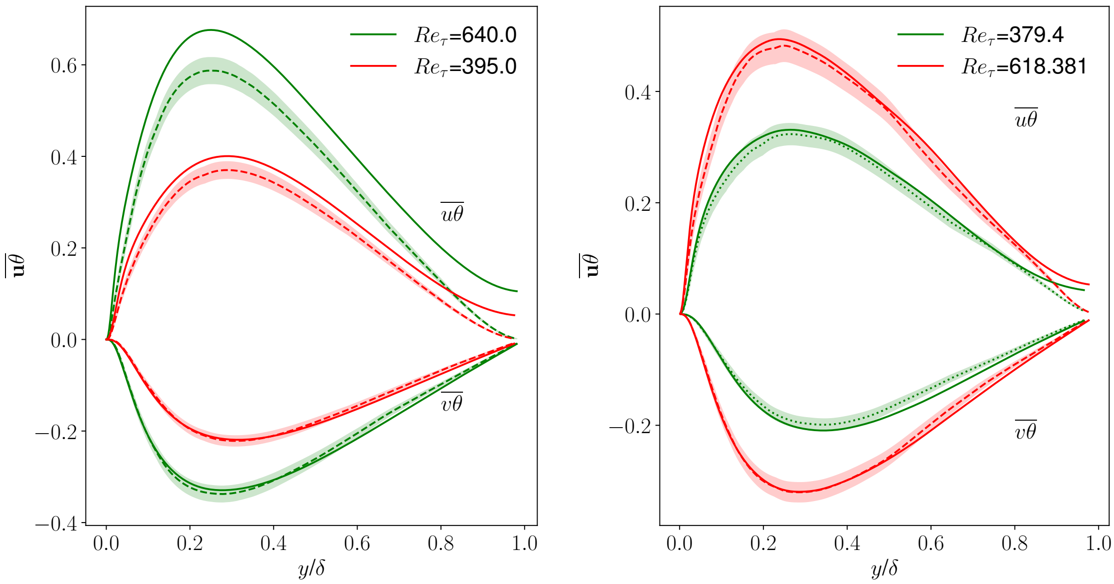


(e) $Pr = 0.01$, **M1**.

(f) $Pr = 0.025$, **M1**.

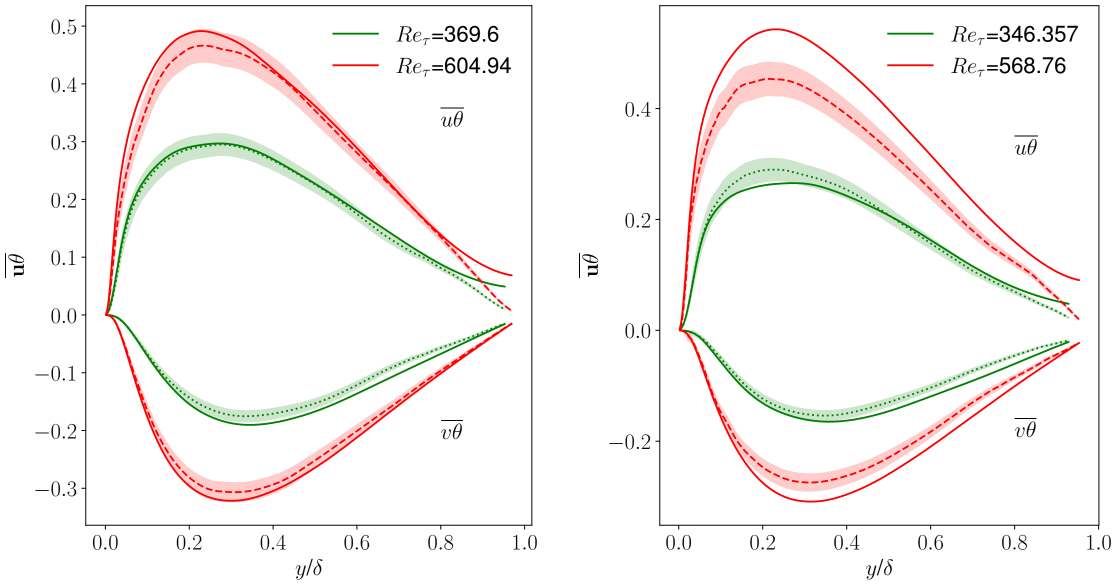
Figure 6.75: Statistical analysis of the temperature profile obtained by integration of exp. 4.4 for $Re_\tau = 395$. DNS (solid red), RA(dash black), RA_2(dot black), Kays(dot-dash black), median DNS validation (dash red), shaded areas encompasses 90% of the predictions while darker shaded areas encompasses 50% of the predictions.

6.6 Training using all data (predictions for $Pr = 0.025$ and $Pr = 0.71$)



(a) DNS

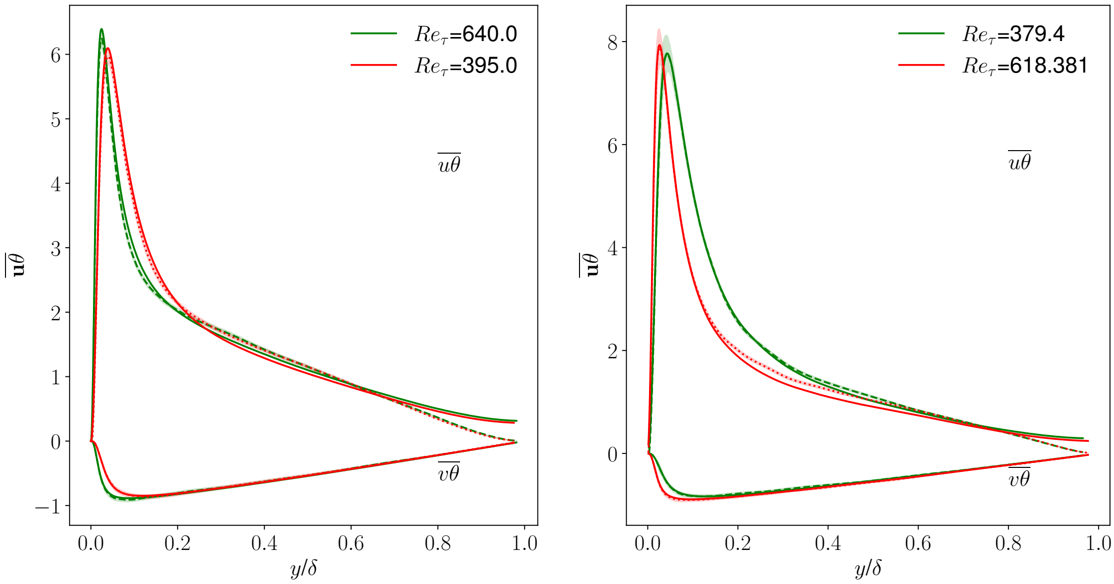
(b) **M3**



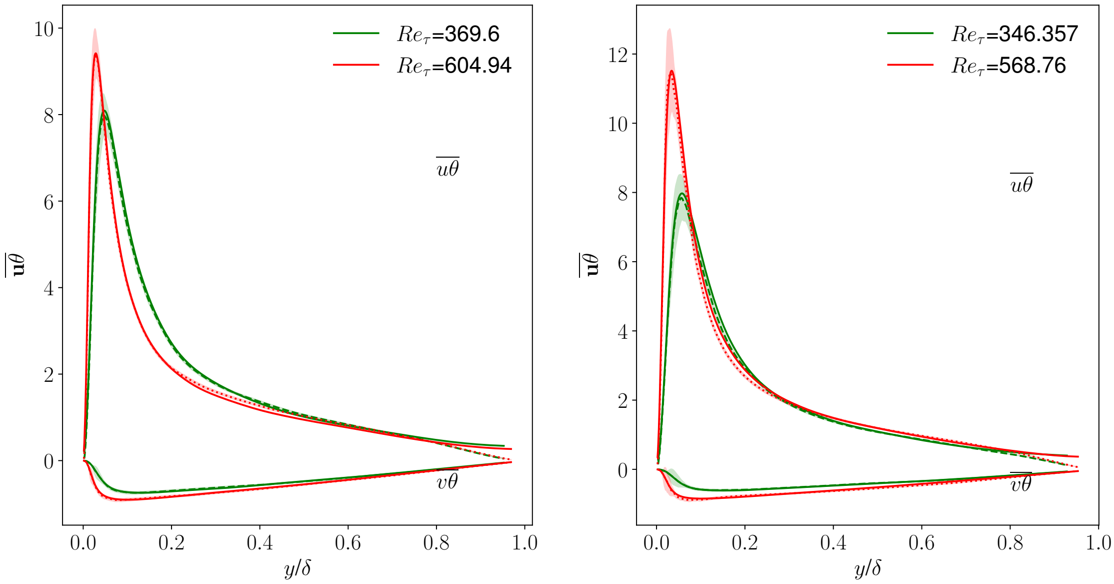
(c) **M2**

(d) **M1**

Figure 6.76: Predictions of the THF for $Pr = 0.025$ by a model trained with all data and without losses weighting

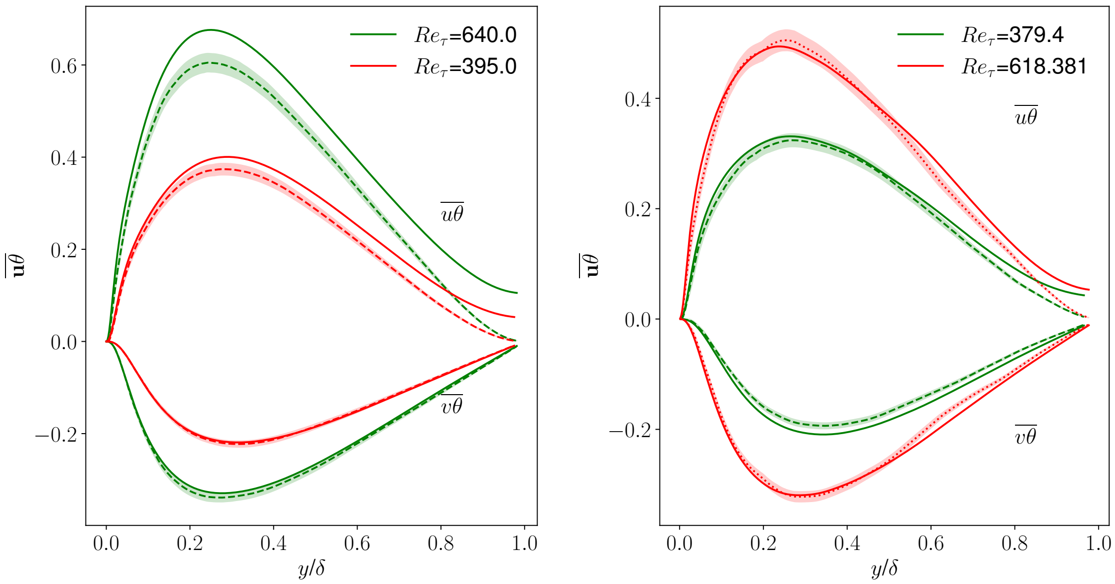


(a) DNS (b) **M3**



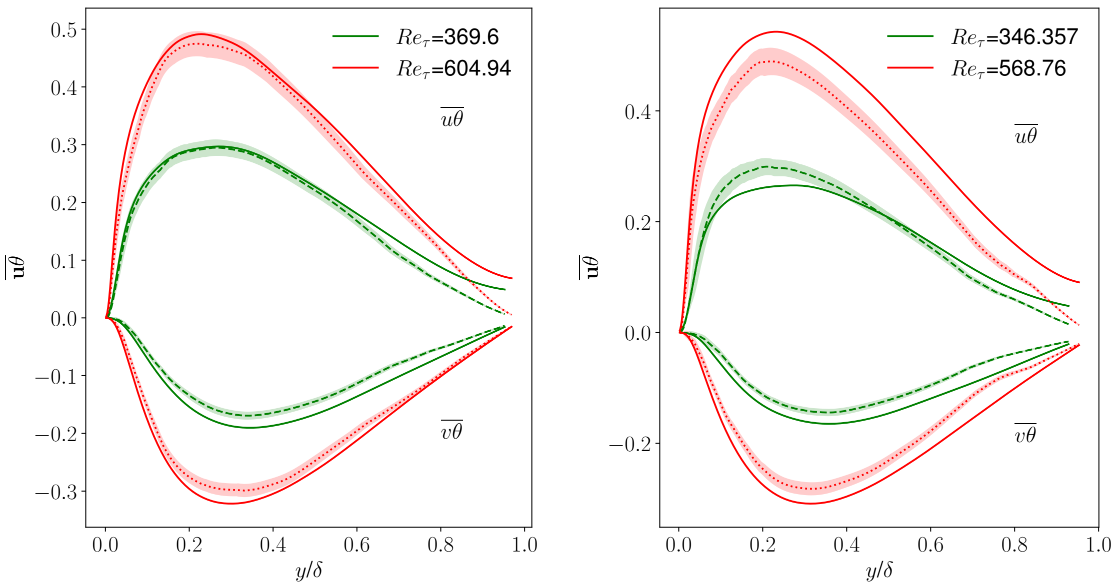
(c) **M2** (d) **M1**

Figure 6.77: Predictions of the THF for $Pr = 0.71$ by a model trained with all data and without losses weighting



(a) DNS

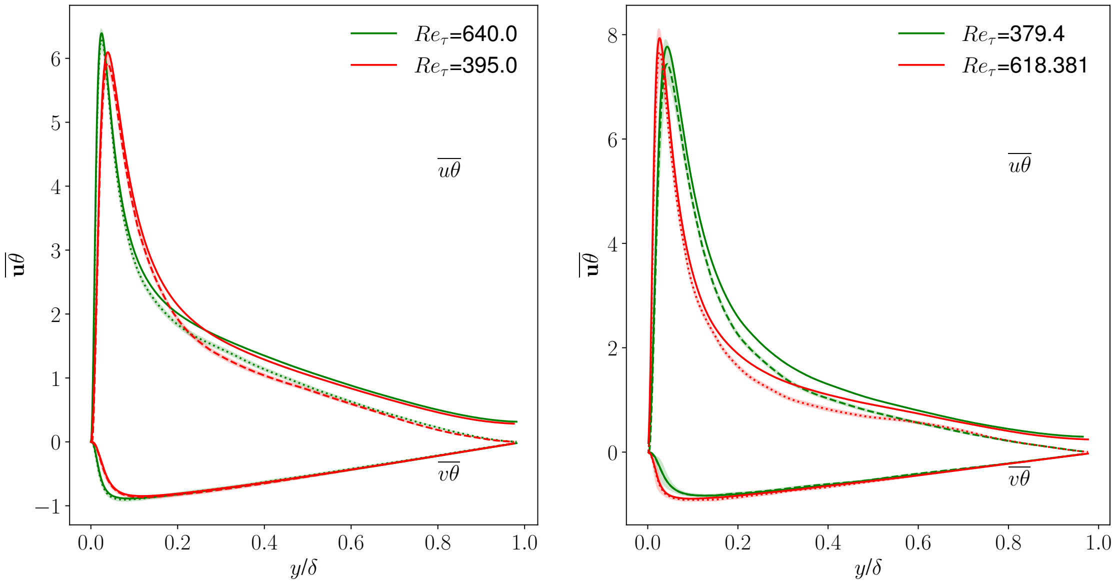
(b) **M3**



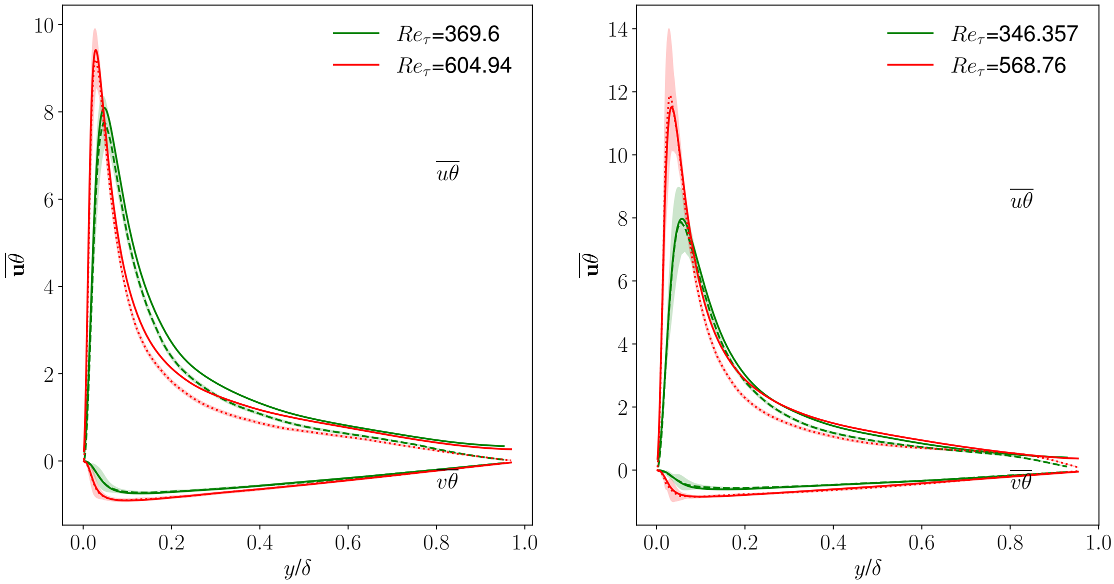
(c) **M2**

(d) **M1**

Figure 6.78: Predictions of the THF for $Pr = 0.025$ by a model trained with all data and with losses weighting using arbitrary weights.

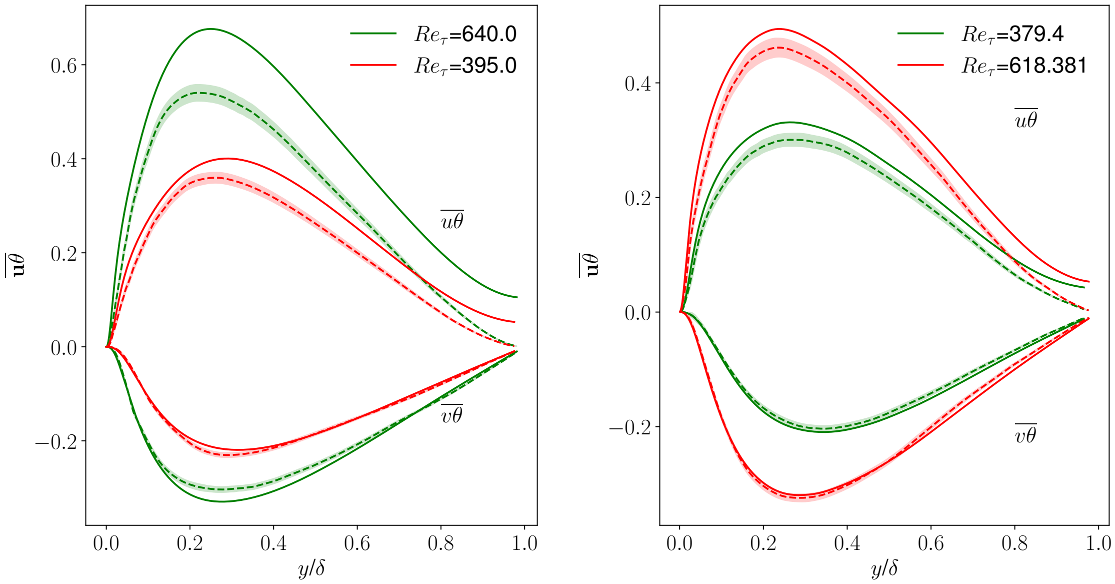


(a) DNS (b) **M3**



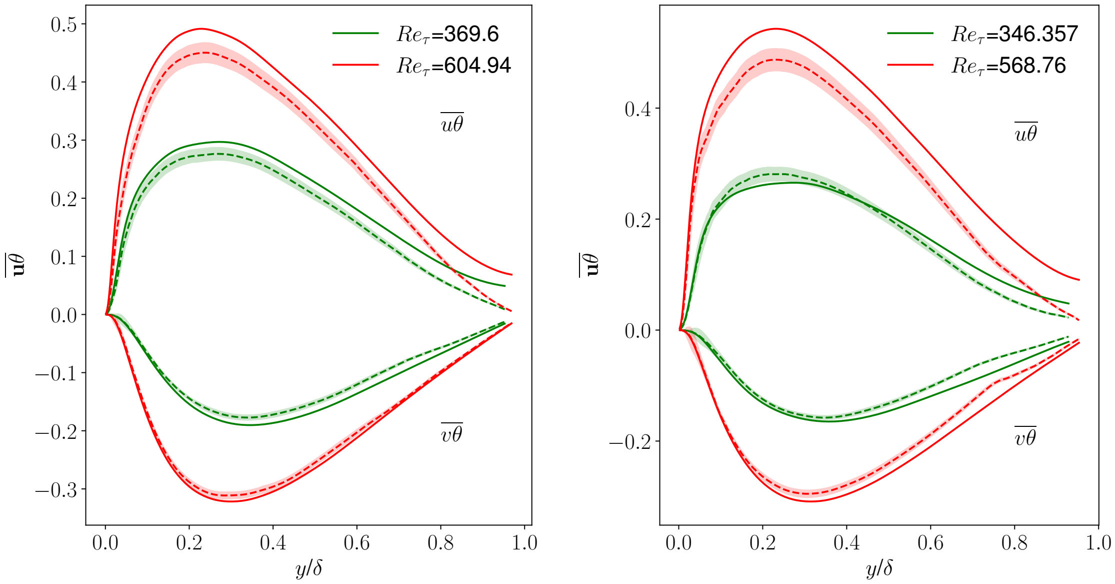
(c) **M2** (d) **M1**

Figure 6.79: Predictions of the THF for $Pr = 0.71$ by a model trained with all data and with losses weighting using arbitrary weights.



(a) DNS

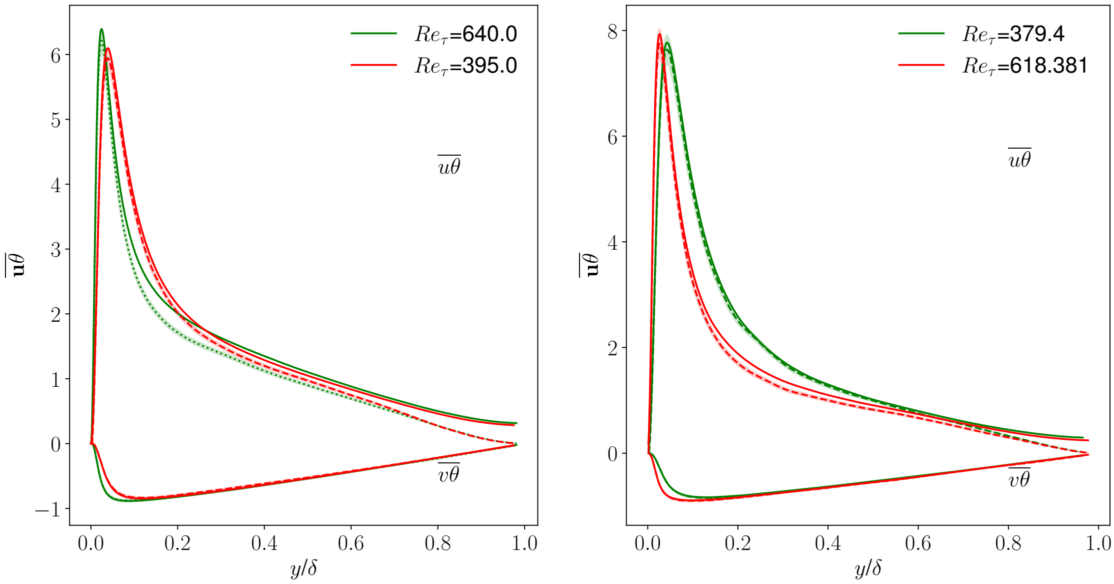
(b) **M3**



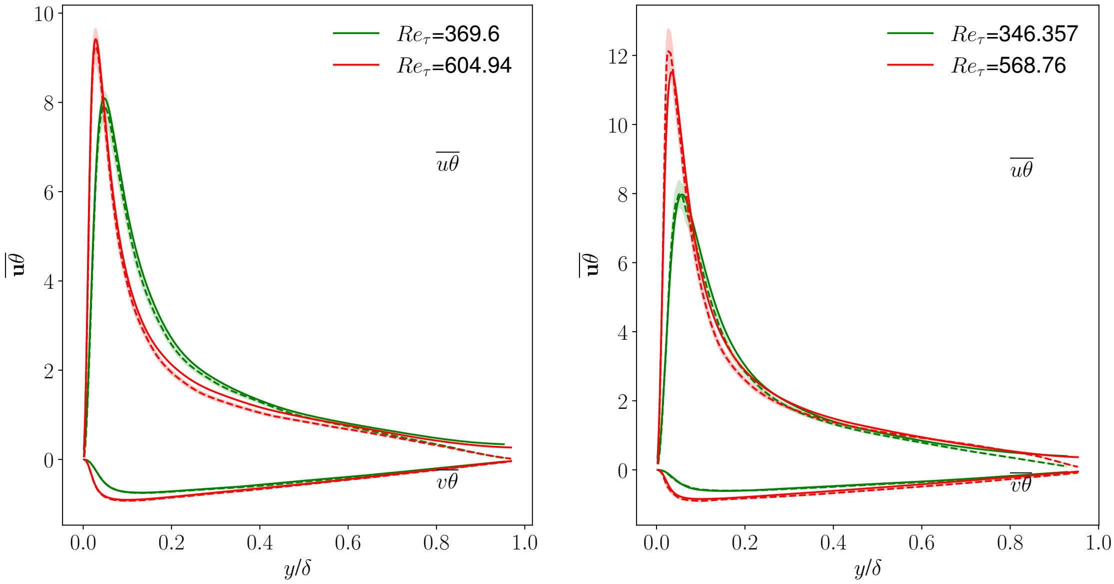
(c) **M2**

(d) **M1**

Figure 6.80: Predictions of the THF for $Pr = 0.025$ by a model trained with all data and with losses weighting using weights dependent on the modeled proportion of the TKE.



(a) DNS (b) **M3**



(c) **M2** (d) **M1**

Figure 6.81: Predictions of the THF for $Pr = 0.71$ by a model trained with all data and with losses weighting using weights dependent on the modeled proportion of the TKE.

Bibliography

[1] Thomas Wetzel, Julio Pacio, Luca Marocco, A. Weisenburger, A. Heinzel, Wolfgang Hering, Carsten Schroer, Georg M[U+9EEE]ler, J[U+9EF5]gen Konys, Robert Stieglitz, Joachim Fuchs, Joachim Knebel, C. Fazio, M. Daubner, and Frank Fellmoser. Liquid metal technology for concentrated solar power systems: Contribution by the german research program. AIMS Energy, 2:89–98, 03 2014.

[2] Y Du Terrail O Metais F Felten, Y Fautrelle. Numerical modelling of electromagnetically-driven turbulent flows using les methods, applied mathematical modelling. Applied Mathematical Modelling, 28, 15-27, 2004.

[3] J.M. Aurnou, Michael Calkins, Jonathan Cheng, Keith Julien, Eric King, David Nieves, Krista Soderlund, and S. Stellmach. Rotating convective turbulence in earth and planetary cores. Physics of the Earth and Planetary Interiors, 246, 07 2015.

[4] Afaque Shams, Ferry Roelofs, Emilio Baglietto, Sylvain Lardeau, and Sasa Kenjeres. Assessment and calibration of an algebraic turbulent heat flux model for low-prandtl fluids. International Journal of Heat and Mass Transfer, 79:589–601, 12 2014.

[5] Sandro Manservisi and Filippo Menghini. Triangular rod bundle simulations of a cfd -[U+03F5]−[U+03F5] heat transfer turbulence model for heavy liquid metals. Nuclear Engineering and Design, 273:251–270, 07 2014.

[6] Andrea Santis, Agustín Villa Ortiz, Afaque Shams, and Lilla Koloszar. Modelling of a planar impinging jet at unity, moderate and low prandtl number: Assessment of advanced rans closures. Annals of Nuclear Energy, 129:125–145, 02 2019.

[7] Afaque Shams, Ferry Roelofs, I Tiselj, J Oder, Yann Bartosiewicz, M Duponcheel, B Niceno, Wentao Guo, Enrico Stalio, Diego Angeli, A Fregni, Sophia Buckingham, Lilla Koloszar, Agustín Villa Ortiz, Philippe Planquart, Chidambaram Narayanan, Katrien Tichelen, Wadim Jäger, and Thomas Schaub. A collaborative effort towards the accurate prediction of turbulent flow and heat transfer in low-prandtl number fluids. 08 2019.

[8] Ferry Roelofs, editor. Thermal Hydraulics Aspects of Liquid Metal Cooled Nuclear Reactors. WP, 2018.

[9] M. Duponcheel, Laurent Bricteux, M. Manconi, Gregoire Winckelmans, and Yann Bartosiewicz. Assessment of rans and improved near-wall modeling for

forced convection at low prandtl numbers based on les up to re=2000re=2000. International Journal of Heat and Mass Transfer, 75:470–482, 08 2014.

[10] G. Grötzbach. Challenges in low-prandtl number heat transfer simulation and modelling. Nuclear Engineering and Design, 264:41–55, 11 2013.

[11] William Morrow Kays. Turbulent prandtl number : where are we ? Journal of Heat Transfer-transactions of The Asme, 116:284–295, 1994.

[12] M. Duponcheel, Laurent Bricteux, M. Manconi, Gregoire Winckelmans, and Yann Bartosiewicz. Assessment of rans and improved near-wall modeling for forced convection at low prandtl numbers based on les up to re=2000re=2000. International Journal of Heat and Mass Transfer, 75:470–482, 08 2014.

[13] Afaque Shams. The importance of turbulent heat transfer modelling in low-prandtl fluids. 11 2018.

[14] Matilde Fiore, Lilla Koloszar, Clyde Fare, Miguel Mendez, Matthieu Duponcheel, and Yann Bartosiewicz. Physics-constrained machine learning for thermal turbulence modelling at low prandtl numbers. International Journal of Heat and Mass Transfer, 194:122998, 09 2022.

[15] Wesley Maddox, T. Garipov, Pavel Izmailov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. ArXiv, abs/1902.02476, 2019.

[16] Hiroshi Kawamura, H. Abe, and K. Shingai. Dns of turbulence and heat transport in a channel flo w with different reynolds and prandtl numbers and boundary conditions. Proceedings of the 3rd International Symposium on Turbulence, Heat and Mass Transfer, 01 2000.

[17] Iztok Tiselj and Leon Cizelj. Dns of turbulent channel flow with conjugate heat transfer at prandtl number 0.01. Nuclear Engineering and Design, 253:153–160, 12 2012.

[18] Sophia Buckingham. Prandtl number effects in abruptly separated flows : LES and experiments on an unconfined backward facing step flow. PhD thesis, VKI, 2018.

[19] Mattias Liefvendahl Timofey Mukha. Large-eddy simulation of turbulent channel flow, 2015.

[20] G. Winckelmans V. Legat. LMECA1321 Mécanique des fluides et transfert I, notes de cours.

[21] Menters F. Best Practice: Scale Resolving Simulations in ANSYS CFD. 2015.

[22] Stanley Corrsin. On the spectrum of isotropic temperature fluctuations in an isotropic turbulence. Journal of Applied Physics, 22:469–473, 1951.

[23] Nicoud Franck and Frédéric Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. Flow Turbulence and Combustion, 62:183–200, 09 1999.

[24] Matthieu Duponcheel, Laurent Bricteux, and Yann Bartosiewicz. DNS and LES database of turbulent heat transfer in a channel at very low Prandtl number, 2020.

[25] E.M.J. Komen, Leonardo Honfi Camilo, Afaque Shams, Bernard Geurts, and Barry Koren. A quantification method for numerical dissipation in quasi-dns and under-resolved dns, and effects of numerical dissipation in quasi-dns and under-resolved dns of turbulent channel flows. Journal of Computational Physics, 345, 05 2017.

[26] Edmond Shehadi. Large Eddy Simulation of Turbulent Flow over a Backward-Facing Step. PhD thesis, Uppsala universiteit, 2018.

[27] Johan Meyers and Pierre Sagaut. Is plane-channel flow a friendly case for the testing of large-eddy simulation subgrid-scale models? Physics of Fluids - PHYS FLUIDS, 19, 04 2007.

[28] Saleh Rezaeiravesh and Mattias Liefvendahl. Effect of grid resolution on large eddy simulation of wall-bounded turbulence. Physics of Fluids, 30, 04 2018.

[29] Yacine Addad, U Gaitonde, and Dominique Laurence. Optimal unstructured meshing for large eddy simulations. In Quality and reliability of large-eddy simulations, Leuven, Belgium, 2007. Quality and reliability of large-eddy simulations, Leuven, Belgium ; Conference date: 01-01-1824.

[30] Richard Underhill and Adam Olive. Nuclear heat transfer and passive cooling, study a: Liquid metal cfd modelling of the tall-3d test facility, 2021.

[31] Arthur G. Kravchenko, Parviz Moin, and Robert D. Moser. Zonal embedded grids for numerical simulations of wall-bounded turbulent flows. Journal of Computational Physics, 127:412–423, 1996.

[32] Stefan Wallin Arne V. Johansson Thilo Knacke Matteo Montecchia, Geert Brethouwer. Improving les with openfoam by minimising numerical dissipation and use of explicit algebraic sgs stress model. Journal of Turbulence, pages 697–722, 2019.

[33] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. 03 2017.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**