Imperial College London

Error and Uncertainty Quantification in Statistics Computed from Direct Numerical Simulation

LAURA MISRACHI

laura.misrachi16@ic.ac.uk

UNDER THE SUPERVISION OF DR. SYLVAIN LAIZET

Abstract

Numerical simulation has been established as one of the most useful tool in the field of turbulence research. Notably, Direct Numercial Simulation (DNS) is widely used to understand the physics of flows, test hypothesis in turbulence as well as turbulence closure models. Just as it is the case for experimental data, it is of primary importance to estimate the uncertainty in statistics computed from DNS to understand their range of validity. Unfortunately, the available methods (standard Richardson extrapolation for example) to estimate the discretization error do not take into account sampling uncertainty which undoubtedly affects DNS data. This paper is based on the bayesian probabilistic model developed by Oliver and al. [12] which appears very promising with the provision of a framework to estimate the uncertainty in DNS data from its two main sources of error : discretization and finite sampling. Following the investigation of the mathematical model developed by Oliver, the model was implemented in Python and Matlab and our results were successfully tested against Oliver's. An estimator for the sampling uncertainty based on the work of Oliver and Trenberth [12][17] was developed and successfully tested as well. In a next step, the model was applied to the DNS of a turbulent channel flow at $Re_\tau \approx 180$ *performed by Dr. Sylvain Laizet. In addition to the work of Oliver, high-order quantities were investigated and it enabled us to gain some useful knowledge on the behavior of the uncertainty for these high-order quantities. Throughout the paper, several tests were developed to evaluate the robustness of the model. Notably, some issues were raised regarding the calibration of the model, which was found to fail in several cases. Some discretization error estimates were also set in default when the influence of the mean in* x *in* z *on the computed statistics was investigated. Still, a promising feature of the model was highlighted : the implementation of a non-informative prior in the bayesian model, which might be very useful when quantities with a priori unknown characteristics are investigated provided successful results for the centerline mean velocity. Several recommendations for future work were advised, notably the implementation of a more physical criteria to complete the mathematical model.*

Master thesis submitted in part fulfillment of the requirements

for the award of the MSc. in Advanced Aeronautical Engineering

Imperial College London, 2016 - 2017

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Sylvain Laizet for the continuous support of my Master thesis, for his patience, motivation and valuable guidance. I would like to thank him as well for the time he took to perform the numerous DNS simulations that were essential to this project. Last but not least, I would like to thank my family and friends, for their invaluable help and support all along the project and especially during the last intensive weeks of work.

CONTENTS

List of Figures

List of Tables

I. Introduction

I. Motivation and literature review

Numerical simulation has been established as one of the most useful tool in the field of turbulence research. It has been considered as a very interesting and promising alternative to experimental observation of fluid flows, notably because of its low cost. Even though experimental investigation of turbulence is known to provide highly accurate results, it remains quite expensive (model making, probe measurements) and complex notably in matching the non-dimensional numbers at the desired experimental scale. On the other hand, Computational Fluid Dynamics is substantially less expensive, allows flow simulation over complex geometries and the performance of dangerous experiments, within a reasonable period of time. The main drawback of CFD is its relatively poor accuracy and this explains why it is currently an active and intense field of research [1]. Yet very promising improvements have been reached and numerical simulation has become so powerful that it is not uncommon to test the accuracy of experimental data using DNS results as a reference [2].

In 1987,*Kim et al.* [3] performed a Direct Numerical Simulation of a turbulent channel flow, compared their results with available experimental data and introduced for the first time some statistical correlations. Their pioneering work marked the beginning of a phase of active research in the field of DNS. In DNS, the Navier-Stokes equations are numerically solved without the use of any turbulence model. This type of simulation aims at resolving all relevant physical scales, from the small dissipative ones, the so-called Kolmogorov scales, up to the integral length scale, associated with the largest eddies containing most of the turbulence energy [4]. DNS is an interesting tool to understand the physics of flows, test hypothesis in turbulence as well as turbulence closure models [5][6].

Just as it is the case for experimental data, it is of primary importance to estimate the uncertainty in statistics computed from DNS to understand their range of validity. Indeed, it is common to estimate the uncertainty in data derived from experiments : in 1979, *E. Rind* [7] introduced a method to estimate the error resulting from the instrumentation used to perform wind-tunnel experiments. Within his work, *E. Rind* insisted on the need to cross-validate the results of an experiment, through the comparison of several data sets. A coherent check of that kind may be achieved provided that an experimental error range has been defined. Obviously, all these considerations should apply for DNS as well. Yet, it is still quite uncommon to provide an error estimation for data computed from numerical simulation, such as DNS.

A common practice in DNS is to evaluate grid spacing requirements and required simulation time (etc), based on previous experiments or observations from numerical simulation [8]. A simple method may be used to estimate the discretization error : it is known as the Richardson extrapolation, for which outputs from simulations carried out on successively finer meshes must be available [9]. This method is frequently used to estimate the leading order error in numerical results, for both stochastic and deterministic problems. As most quantities computed from DNS are statistical, two main sources of error dominate : the discretization error due to the numerical

approximation of the Navier-Stokes equations and the sampling error due to the finite character of the sampling [10]. Yet, the classical Richardson extrapolation does not take into account sampling uncertainty. As a result, this method implicitly assumes that the sampling uncertainty is small relative to the discretization error. This approximation is very likely to be called into question, especially in the context of DNS, for which all relevant physical scales are solved. Hence, this classical method for error estimation is not directly applicable to DNS.

Estimating uncertainties in statistics computed from DNS is quite challenging. In his work, *S. Ghosal* [11] highlighted that standard approaches in error analysis cannot be directly applied to highly nonlinear problems, such as turbulence, because of the simultaneous presence of a continuum of space and time in terms of scales. More recently, *Oliver et al.* [12] focused on the question of uncertainty estimation in DNS and developed a bayesian probabilistic version of Richardson extrapolation to take into account the error due to finite sampling. The novelty in their work lies in the definition of an error model that accounts for both the discretization error (with an extension of Richardson extrapolation) and the sampling uncertainty. Hence, the parameters of Richardson extrapolation are processed as random variables and their distributions are inferred according to Bayes' theory.

II. Aims and objective

The work in this present paper focuses on the model of *Oliver et al.* [12] which provides a framework to estimate the uncertainty in statistics computed from DNS. The final goal of the project is to implement and test the model on data derived from the DNS of a turbulent channel flow at $\Re_\tau \approx 180$, with a particular emphasis on high-order quantities (skewness, flatness, turbulent kinetic energy dissipation) which were not necessarily investigated by *Oliver and al* [12]. The idea was also to test the model and understand to what extent it can be practically used for all sorts of DNS simulation. To achieve all of these objectives, the project was divided in the following intermediate stages:

- Investigation of the two main sources of error in DNS.
- Presentation and understanding of the bayesian model of *Oliver and al* [12].
- Implementation of the model and code validation.
- Application to the DNS of a turbulent channel flow at $Re_\tau \approx 180$ performed by Dr. Sylvain Laizet.
- Investigation and improvement of certain features of the model.

One aim of the project is to try to implement a more physical criteria in the model developed by *Oliver and al.* [12].

II. A FOCUS ON THE TWO MAIN SOURCES OF ERROR IN DNS

I. Discretization error

Discretization error in DNS arises out of the numerical approximation of the Navier-Stokes equations. By introducing spatial and temporal discretization schemes, the exact equations are transformed into discrete equations which are solved on a mesh of specific resolution h. Discretization error is the result of a complex interaction between the chosen discretization schemes, the mesh resolution and the mathematical behavior of the solution [9][13][14].

As mentioned above, Richardson extrapolation is one of the simplest method to evaluate the discretization error. Given outputs of simulations conducted on at least three meshes of successively finer resolutions, the discretization error may be computed. Let q and q_h respectively refer to the exact solution of a mathematical problem and its discrete approximation at resolution h. If one assumes that the exact solution may be approximated by the following error formula:

$$q = q_h + C_0 h^p + C_1 h^{p+1} + C_2 h^{p+2} + ... \tag{1}$$

and that q_{h_1}, q_{h_2} and q_{h_3} are the discrete approximations to the exact solution on successively finer meshes of respective resolutions h_1, h_2 and h_3, the standard Richardson extrapolation provides an estimate of the leading error order p through the following:

$$\frac{q_{h_3} - q_{h_2}}{q_{h_2} - q_{h_1}} = {t_1}^p \cdot \frac{{t_2}^p - 1}{{t_1}^p - 1} \tag{2}$$

where $t_1 = \frac{h_2}{h_1}$, $t_2 = \frac{h_3}{h_2}$ and the terms of order greater than p have been neglected ($O(h^{p+1})$). Unfortunately, this method does not take into account sampling uncertainty which certainly affects data derived from DNS and therefore its use in this context might lead to wrong estimates of p. Later, an improvement will be fitted to this model to include the effects of finite sampling, according to the work of *Oliver et al.* [12].

II. Sampling error

II.1 Overview

As most quantities computed in DNS are statistical, an additional type of error due to sampling must be acknowledged. A simulation is by definition bounded in time, and therefore the samples from which the quantities of interest may be computed are limited in numbers, which leads to more or less accurate estimations. This section aims at providing a method to evaluate the sampling error in data computed from DNS and is largely based on the work of *Oliver et al.* [12]. For this purpose, let X denote a scalar flow quantity (the axial velocity for example) and X_1, X_2,..., X_N, N samples derived from a DNS. One may assume that these samples are from a statistically stationary sequence of random variables (meaning that the statistics are time-independent), which is very likely to be a desired characteristic of DNS data to ensure the production of accurate statistics (????). The average of the N available samples can be computed with the unbiased estimator of the population mean:

$$\langle X \rangle_N = \frac{1}{N} \sum_{i=1}^{N} X_i \tag{3}$$

It is an estimator of the true mean, the one unaffected by sampling uncertainty, which we choose to denote $E[X]$. Hence, the sampling error e_N is simply the difference between the true mean and its estimation :

$$e_N = \langle X \rangle_N - E[X] \tag{4}$$

II.2 Variance estimation

Sampling uncertainty in turbulence is not very straightforward to predict, as the samples used to compute the relevant statistics are derived from a time history and space field which make them usually and a priori not independent. Obviously, if these samples were independent, identically distributed random variables, the Central Limit Theorem would provide an easy estimation of the sampling error, as $N \to \infty$,

$$e_N \to \mathcal{N}(0, \frac{\sigma^2}{N}) \tag{5}$$

where $\sigma^2 = var(X)$ could be estimated with classical methods. Extension of the Central Limit Theorem to the case of dependent, identically distributed random variables has been widely investigated and Billingsely [15] proposed a theorem for dependent variables subject to a strong mixing configuration, which means concretely, that these variables may be considered close to independent if they are taken temporally far apart from one another. This is very likely to be the case for turbulence data [16] and the theorem can be stated as follows:

$$e_N \underset{n \to +\infty}{\longrightarrow} \mathcal{N}(0, \frac{s^2}{N}) \tag{6}$$

where

$$s^2 = E[(X_1 - \mu)^2] + 2 \sum_{k=1}^{\infty} E[(X_1 - \mu)(X_k - \mu)] \tag{7}$$

Introducing the autocorrelation function at lag k,

$$\forall k, \ \rho(k) = \frac{E[(X_1 - \mu)(X_k - \mu)]}{E[(X_1 - \mu)^2]} \tag{8}$$

it gives:

$$Var(e_N) \approx \frac{\sigma^2}{N} \left(1 + 2 \sum_{k=1}^{\infty} \rho(k) \right) \tag{9}$$

The RHS term is the decorrelation separation distance T_0 which refers to the time distance between effectively independent samples. Thus, $N_{eff} = \frac{N}{T_0}$ is the effective number of independent samples and the extension of the Central Limit Theorem gets more relevant as :

$$Var(e_N) \approx \frac{\hat{\sigma}_N^2}{N_{eff}} = \frac{\hat{\sigma}_N^2 T_0}{N} \tag{10}$$

According to *Trenberth* [17], this exact formula can be modified to account for the fact that the number of available samples is finite and that the autocorrelation function is assessed by data:

$$Var(e_N) \approx \frac{\hat{\sigma}_N^2}{N} \left(1 + 2 \sum_{k=1}^{N} \left(1 - \frac{k}{N} \right) \hat{\rho}(k) \right) \tag{11}$$

Trenberth suggested a small variant for the estimator of $\hat{\sigma}_N$ to account for the effective number of independent samples :

$$\hat{\sigma}_N^2 = \frac{1}{N - T_0} \sum_{k=1}^{N} (X_i - \langle X \rangle_N) \tag{12}$$

II.3 Estimation of the autocorrelation function

The main difficulty in the estimation of the sampling error variance lies in the evaluation of the autocorrelation function. Following *Trenberth*, an estimate based on the mathematical definition of $\hat{\rho}$ can be set :

$$\hat{\rho}_k = \frac{1}{N} \sum_{i=k+1}^{N} (X_{i-k} - \mu)(X_i - \mu) \tag{13}$$

However, this straightforward technique is likely to lead to poor estimates of T_0 (see section 2. of [18]) as it tends to be noisy. Besides, one cannot put much confidence in the values of ρ for large k, since fewer pairs of (X_{i-k}, X_i) are available to compute the autocorrelation. As suggested by *Oliver et .al* [12], an autoregressive time series model was fitted to the sequence of observations $X_i, i \in [1, N]$, following Broersen [20]. A time series model of order p $AR(p)$ has the following form:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + u_t \tag{14}$$

where $u_t = \mathcal{N}(0, {\sigma_u}^2)$ is a white noise and ϕ_1, ϕ_2, ϕ_3, ..., ϕ_p are the p parameters of the model. This type of model states that the series depends linearly on its previous values together with a stochastic term. Thus, in order to estimate the autocorrelation function, the parameters $\phi_i, i \in [1, N]$, the variance of the white noise and the order of the autoregressive process must be evaluated. Even though *Oliver et. al* [12] provided a C++ package for the estimation of the variance from a sequence of correlated samples [19], I have chosen to code my own MATLAB program based on their recommendations. Given the order of the process p, several methods are available to estimate the parameters of the model and the white noise variance [20]. A quick benchmark of these methods to select the most appropriate follows:

Method	Characteristics
Yule-Walker	The model exhibit a relatively small variance but may reach very large bias. Not recommended for data with unknown characteristics.
Forward and Backward Least-squares	The estimated model is not guaranteed to be stationary, with possible poles of the estimated polynomial lying outside the unit circle.
Forward Least-squares	No guarantee for the obtention of a stationary model with all poles inside the unit circle.
Burg	Levinson Durbin recursion is used to estimate the parameters for increasing model orders, which ensure the model will be stationary with all roots within the unit circle. Method for which the smallest expected mean square error for the parameters is reached and less sensible to round-off errors.

Table 1: Benchmark of the method available for an AR parameter estimation.

Based on this table, Burg method [22], which is explained in more detail in Appendix A was implemented. Subsequently, the order of the autoregressive process was chosen according to a model selection criteria [21] that best fits our data. Several criteria have been investigated in the literature [21] to select the appropriate order of an AR process and the general form of the criterion to be minimized is:

$$criterion(v_{method}, N, p, \alpha) = ln(\hat{\sigma}_u(p)) + overfit(criterion, v_{method}, N, p, \alpha) \tag{15}$$

where v_{method} is the method-related estimate of the variance for the model of order p, p is the estimated order, N is the number of samples used for the estimation of the model parameters, $\hat{\sigma}_u$ is the residual variance estimate consistent with the parameters estimation for a model of order p and α is a method-specific coefficient. For a given criterion and a set of possible model order i, the best candidate minimizes the criterion. As the potential order of the AR process is greater than $0.1N$ (to be confirmed later in the paper), we are in the case of finite sampling and three criterion were therefore implemented : FIC (Finite Information Criterion), FSIC (Finite Sample Information Criterion) and CIC (Combined Information Criterion). The subsequent empirical formulae used to characterize the $overfit$ term in (15) are described in Appendix B.

II.4 Code implementation and validation

The code was implemented in MATLAB according to the mathematical considerations of Appendix A & B. A validation phase based on the comparison of the results obtained with our code and the package developed by *Rhys Ulerich* [12] was developed to ensure successful estimation of the error variance. The tested data set was made available on GitHub by *Rhys Ulerich* (rhoe.dat) [19] and the following table provides the model comparison:

Variables	**Rhys Ulerich Package**	**Our code**
Sample / data set size N	1753	1753
Criterion	CIC	CIC
Mean substracted to each value	True	True (choice made available to the user)
Parameter estimates	1.0	1.0
	-2.6990334396411866	-2.699035815820956
	2.8771681702855281	2.877172572053186
	-1.7247852051789097	-1.724788942073497
	0.75024605955486146	0.750249912924771
	-0.26866837869957461	-0.268672061616252
	0.06700587276734557	0.067007468593983
Residual variance $\hat{\sigma}_u^2$	$8.3374920647988362e-09$	$8.337383477446236e-09$
Best order of the process	6	6
Mean of the sample μ	0.20955287956200269	0.209552879562003
Decorrelation time T_0	62.190091348891279	62.123370203456425
Number of effectively independent data N_{eff}	28.18777014115533	28.218044099327800
Standard deviation of the sampling error	0.0011415499935005066	0.001140914957937

Table 2: Comparison of our results with *Oliver et al.* [12] for the error variance estimation.

As a conclusion, our results are in quite good agreement with *Oliver et al.* [12] and a relative error of only 0.055 % in the estimation of the standard deviation of the sampling error was reached. Let us emphasize on the fact that the large number of significant digits in the previous results has been chosen on purpose to ensure a precise comparison of our results with Oliver's.

III. A Bayesian probabilistic model to estimate the error in data computed from DNS

I. Overview

This section aims at highlighting the probabilistic model developed by *Oliver et al.* [12] to evaluate the error in statistics computed from DNS. Based on the previous discussions about the discretization error and the sampling uncertainty, a probabilistic model to estimate the error from these two sources has been set up. A bayesian approach is preferred to a deterministic one (Maximum Likelihood [24] for example) as it provides a higher level of knowledge on the uncertainty and allows an in-depth understanding of the correlation between our parameters. Hence, the parameters of the discretization error model (C_0, p) and sampling uncertainty (e_N) are treated as random variables and inferred from data thanks to Bayes' theory [25][26].

II. Mathematical model

Let $E[q]$ denote the true mean of a quantity q and $e_{h,N}$ the sampling error for the sample average computed from N correlated samples at resolution h. The latter can be expressed as:

$$e_{h,N} = E[q_h] - \langle q_h \rangle_N \tag{16}$$

where $E[q_h]$ is the true mean at resolution h and $\langle q_h \rangle_N$ is the sample average computed from N observations at resolution h. The discretization error at resolution h, ϵ_h may be expressed as $\epsilon_h = E[q] - E[q_h]$ and therefore one may obtain:

$$E[q] = E[q_h] + \epsilon_h = \langle q_h \rangle_N + \epsilon_h + e_{h,N} \tag{17}$$

Further introducing the error formula on which Richardson extrapolation is based (cf. equation (1)), one has a complete probabilistic model for the true mean $E[q]$:

$$E[q] = \langle q_h \rangle_N + C_0 h^p + e_{h,N} \tag{18}$$

where the terms of order greater than p have been neglected and $e_{h,N}$ is entirely determined by the variance estimator developed in section II.II. Let us emphasize on the fact that a specific model for the discretization error has been chosen here and that other models - of greater order notably - could be implemented. Rearranging (18) a little gives:

$$E[q] - \langle q_h \rangle_N - C_0 h^p = e_{h,N} \sim \mathcal{N}(0, var(e_{h,N})) \tag{19}$$

where $var(e_{h,N})$ is evaluated as in (11). For the sake of clarity and from now on, the true mean $E[q]$ will be referred to as $\bar{q}$. Based on equation (19), an inverse bayesian problem is set, where the vector of parameter $\theta = C_0, p, \bar{q}$ is the unknown and the M sample averages computed on mesh of distinct resolution h_i, $\hat{q}_i = \langle q_{h_i} \rangle_{N_i}, i \in [1, M]$ are referred to as the observed data and mentioned as vector $\mathcal{D}$. The conditional joint probability of $\theta | \hat{q}_1, ..., \hat{q}_M$, is computed according to Bayes' theorem:

$$\underbrace{\pi(\bar{q}, C_0, p | \hat{q}_1, ..., \hat{q}_M)}_{posterior} = \pi(\theta|\mathcal{D}) \propto \underbrace{\pi(\hat{q}_1, ..., \hat{q}_M | \bar{q}, C_0, p)}_{likelihood} \underbrace{\pi(\bar{q}, C_0, p)}_{prior} \tag{20}$$

Equation (19) highlights that the posterior has two components:

- A likelihood term which refers to the probability of effectively observing the data given specific values of the parameters $\bar{q}, C_0, p$.
- A prior term which holds any information available concerning the parameters, independently of the observations. For example, it is common to have some knowledge on the order of the error, which depends on the spatial and temporal discretization schemes used to solve the Navier-Stokes equations.

In a sense, Bayes' theory predicts the posterior distribution in a quite subjective manner as prior information is required. One may assume that the sampling errors for different resolution h_i are independent, and thus:

$$\pi(\hat{q}_1, ..., \hat{q}_M | \bar{q}, C_0, p) = \prod_{i=1}^{M} \pi(\hat{q}_i | \bar{q}, C_0, p) \tag{21}$$

Rewriting (19) as $\hat{q}_i = \bar{q} - e_{h_i,N} - C_0 h_i^p$ and applying a basic change of variable, one holds the following:

$$\pi(\hat{q}_i | \bar{q}, C_0, p) = \frac{1}{\sigma_i} \Phi\left(\frac{\bar{q} - \hat{q}_i - C_0 h_i^p}{\sigma_i}\right) \tag{22}$$

where Φ is the normal density function defined as $\forall x \in [-\infty; +\infty], \Phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$ and σ_i is the standard deviation of the sampling error at resolution h_i, computed according to (11).

III. Prior information

To finalize the bayesian estimation of the posterior, one must incorporate prior distributions for the parameters. For the purpose of simplicity, $\bar{q}, C_0$ and p are taken to be independent in the prior. Since many DNS statistics are available in the literature, with some simulation results considered as strong reference, setting prior information is quite straightforward. It should also incorporate the features of the numerical approximation (true error order for instance) which has been individually set by each user. It is clear that if relevant information concerning the parameters is available, it should take part in the definition of the prior to ensure a more accurate estimation of the posterior. However, this is not always the case, especially when more complex statistics are computed (dissipation, higher-order quantities), unreferenced Reynolds number are involved or simulation over complex geometries are studied. In these situations, one might try to implement uninformative prior, such as uniform distribution. This shall be discussed later with practical test cases. For now, as precise information on the true mean is usually available, the prior on $\bar{q}$ will be modeled by a Normal law and p will be approached by a Gamma distribution to give plausibility to a wide range of orders. Finally, as it is not absurd to consider C_0 to be symmetric in the prior, it will be modeled by a Normal law of mean zero. Thus:

$$\bar{q} \sim \mathcal{N}(q_0, \sigma_q^2) \qquad C_0 \sim \mathcal{N}(0, {\sigma_{C_0}}^2) \qquad p \sim Gamma(\alpha, \beta) \tag{23}$$

where $prior = q_0, \sigma_q, \sigma_{C_0}, \alpha, \beta$ are to be precisely defined in different practical test cases. The Gamma distribution takes the following form:

$$\pi(p) = \frac{\beta^\alpha p^{\alpha-1}}{\Gamma(\alpha)} \exp(-\beta p) \tag{24}$$

Therefore, equation (20) can be modified to give the following posterior probability density:

$$\pi(\theta | \mathcal{D}, prior) \propto \frac{\beta^\alpha p^{\alpha-1}}{\Gamma(\alpha)\sigma_q\sigma_{C_0}} \exp(-\beta p)\ \Phi\left(\frac{\bar{q} - q_0}{\sigma_q}\right) \Phi\left(\frac{C_0}{\sigma_{C_0}}\right) \prod_{i=1}^{M} \frac{1}{\sigma_i}\ \Phi\left(\frac{\bar{q} - \hat{q}_i - C_0 h_i^p}{\sigma_i}\right) \tag{25}$$

IV. Code implementation

To gain knowledge on the uncertainty of the computed statistics, it is of interest to obtain the empirical probability density function (PDF) of our model parameters $\bar{q}, C_0$ and p, which are commonly referred to as the marginal PDFs:

$$\pi(\bar{q}|\mathcal{D}, prior, C_0, p) = \iint_{\Omega_{C_0}, \Omega_p} \pi(\theta|\mathcal{D}, prior)\ dp\ dC_0 \tag{26}$$

$$\pi(p|\mathcal{D}, prior, C_0, \bar{q}) = \iint_{\Omega_{C_0}, \Omega_{\bar{q}}} \pi(\theta|\mathcal{D}, prior)\ d\bar{q}\ dC_0 \tag{27}$$

$$\pi(C_0|\mathcal{D}, prior, \bar{q}, p) = \iint_{\Omega_{\bar{q}}, \Omega_p} \pi(\theta|\mathcal{D}, prior)\ dp\ d\bar{q} \tag{28}$$

After examination of these expressions, it appears that an analytical computation of the integrals is intractable, meaning that numerical methods are required to compute the marginal densities. Monte Carlo methods which are widely used in probabilistic inference to integrate complex distributions in high dimension [24] were therefore investigated as a solution.

IV.1 Monte Carlo and Markov Chain Monte Carlo method

The Monte Carlo method relies on repeated random sampling to compute a deterministic quantity. Integral calculation is one of its main application. It is assumed that one wishes to evaluate the following integral, in a continuous framework where X is a random variable with density f_X according to the Lebesgue measure and g an arbitrary function:

$$I = \int_{\mathcal{R}^d} g(x) f_X(x) dx \tag{29}$$

One can write the previous integral as an expectation for the random variable $g(X)$, provided that it is well-defined:

$$I = \mathbb{E}[g(X)] = \int_{\mathcal{R}^d} g(x) f_X(x) dx \tag{30}$$

The Law of Large Numbers is as follows:

Theorem 1 (*Law of Large Numbers*). *Let $X_1, X_2, ...$ be a sequence of identically distributed random variables, f_X denoted as their density function and let g: $\mathcal{R}^d \to \mathcal{R}^d$ be a function such that $m = \mathbb{E}[g(X_1)]$ exists. Then*

$$\lim_{n \to +\infty} \frac{g(X_1) + g(X_2) + ... + g(X_n)}{n} = \int_{\mathcal{R}^d} g(x) f_X(x) dx = m \tag{31}$$

Therefore, the key idea of Monte Carlo simulation consists in generating M samples $\left(x^{(1)}, x^{(2)}, ..., x^{(M)}\right)$ drawn independently from the distribution f_X and to compute an estimator for I based on the Law of Large Numbers:

$$I = \mathbb{E}[g(X)] = \frac{1}{M} \sum_{i=1}^{M} g\left(x^{(i)}\right) \tag{32}$$

A Markov chain is a stochastic process exhibiting the Markov property which states that the conditional probability of the future state of a given process relies only upon the current state, and not the previous ones. Mathematically, this can be expressed as, $\forall n \geq 0,\ \forall (i_0, ..., i_{n-1}, i, j) \in E^{n+2}$

$$\mathbb{P}(X_{n+1} = j \mid\mid X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}, X_n = i) = \mathbb{P}(X_{n+1} = j \mid\mid X_n = i) \tag{33}$$

A Markov chain is initialized at a certain state $x^{(1)}$ and a transition matrix $q\left(x^{(t)} \mid x^{(t-1)}\right)$ is used to determine the next state $x^{(2)}$, conditional on the previous state. And so on to form a sequence of states which is called a Markov chain:

$$x^{(1)} \rightarrow x^{(2)} \rightarrow x^{(3)} \rightarrow ... \rightarrow x^{(n-1)} \rightarrow x^{(n)} \tag{34}$$

Some useful remarks on Markov Chain follows [28]:

- If several chains are initialized with different conditions, all of them will first remain in a state close to the initial conditions. This is referred to as the ***burn-in period***.
- But after a sufficiently long sequence of transitions, the state of the chain is no longer affected by its initial state. At this stage, the chain has reached a steady state and the samples may be considered from a stationary distribution.

The goal of Markov Chain Monte Carlo (MCMC) is to design a Markov chain such that the stationary distribution of the chain is the distribution we aim to sample from. It is referred to as the target distribution p. For this purpose, the idea is to set up an adequate transition function to ensure a convergence to the target distribution for any initialization of the chain. Several algorithms designed to perform these tasks are available in the literature.

IV.2 Some classical algorithms for MCMC

Two major algorithms have been developed to perform MCMC: Metropolis-Hastings and Gibbs sampler [29]. The Metropolis-Hastings procedure starts with the initialization of the first state $\theta(1)$, then a proposal point θ^* is sampled from the proposal distribution $q\left(\theta^* \mid \theta(t)\right)$, the new state being conditional on the previous one. This new state is either accepted or rejected, with the following probability of acceptance :

$$\alpha = min\left(1, \frac{p\left(\theta^*\right) q\left(\theta(t)|\theta^*\right)}{p\left(\theta(t)\right) q\left(\theta^*|\theta(t)\right)}\right) \tag{35}$$

It is quite straightforward to make sense out of this formula:

- if $p(\theta^*) > p(\theta(t))$, the proposal point is very likely to be accepted which is consistent with the fact that the sampler is moving towards region of high probability of the target distribution p.
- if $p(\theta^*) < p(\theta(t))$, low probability region of the distribution are investigated. But still, there is a chance for this "poor" proposal point to be accepted, depending on the ratio of the proposal density. This effectively ensures that all the state space is explored, including the tail of the target distribution p.

To make the decision on whether to accept or reject a point, a sample from a standard uniform distribution is generated as $u \sim \mathcal{U}(0,1)$ and:

- if $u < \alpha$, the proposal point is accepted and the next state is set at θ^*.
- if $u > \alpha$, the proposal point is rejected and the next state is kept at $\theta(t)$.

The Metropolis-Hastings algorithm has the advantage to only require the computation of the ratio of the targeted density p (cf. (35)). Hence, no knowledge on the normalizing constant which appears in the exact formula of Bayes' theorem is needed, which substantially simplifies the posterior sampling, as this constant is often impractical to obtain [24]. It is the case here. The compulsory selection of a proposal distribution is one of the main drawback

of this procedure. Even though some recommendation are available, such as the choice of a PDF that holds the same support as the target distribution, this task remains quite complex. Another issue is that a non-negligible part of the computation is spent producing rejected samples that are not integrated in the final sequence of samples from the posterior. The Metropolis-Hastings algorithm for an univariate distribution (1D) is resumed in the following:

Algorithm 1 *Metropolis-Hastings algorithm in 1D*

1: *Generate a proposal point* $\theta^* \sim q(\theta^*|\theta(t))$
2: $t \leftarrow \frac{p(\theta^*)q(\theta(t)|\theta^*)}{p(\theta(t))q(\theta^*|\theta(t))}$ ▷ *Computationally expensive*
3: $u \leftarrow \mathcal{U}(0,1)$
4: **if** $u < min\,(1,t)$ **then**
5: $\quad \theta(t+1) \leftarrow \theta^*$
6: $\quad \theta(t+1) \leftarrow \theta(t)$
7: **end if**

The Gibbs sampler is another alternative, that does not require the introduction of any proposal density and for which all the samples are accepted. However, this procedure requires the knowledge of the full conditional distribution of each variable, conditioned on all others. This kind of information is not systematically available, as it is the case here.

IV.3 The Python package chosen for the project : emcee

Some algorithms with a faster convergence to a stationary sequence of samples from the posterior distribution are available in the literature. Given the complexity of the posterior we aim to sample from, I have chosen to follow the suggestion of *Oliver et al.* [12] and to investigate *Goodman and Weare's* affine invariant algorithm [30] which was implemented in PYTHON by *Foreman-Mackey* as the **emcee** package [31]. This procedure relies on an ensemble of K walkers $S = \{X_k\}$ which operate a simultaneous walk in the state space and where the proposal density for one walker is based on the current "positions" of the $K-1$ walkers in the complementary ensemble $S_{[k]} = \{X_j, \forall j \neq k\}$. The "position" of a walker is materialized by a vector $V \in \mathcal{R}^{N_{dim}}$, where N_{dim} is the dimension of the parameter space (three in our case). The new position of a walker at position X_k is updated as:

$$X_k(t) \rightarrow Y = X_j + Z[X_k(t) - X_j] \tag{36}$$

where X_j is taken randomly from the complementary ensemble $S_{[k]}$ and Z is a random variable drawn from a distribution g. This random walk has been designed so that g satisfies the following:

$$g(z^{-1}) = zg(z) \tag{37}$$

In this case, it is clear that the proposal distribution is symmetric and therefore the proposal point is to be accepted with the following probability :

$$\alpha = min\left(1,\ Z^{N_{dim}-1}\frac{p(Y)}{p\left(X_k(t)\right)}\right) \tag{38}$$

In [30], the efficiency of the algorithm was investigated with the autocorrelation time, which is an estimation of the number of computation of the posterior needed to draw independent samples from the target density. The

autocorrelation time was found to be substantially smaller for this algorithm in comparison with the classical Metropolis procedure over several test cases [30]. The stretch move algorithm from *Goodman and Weare* is presented in the following [31]:

Algorithm 2 *Goodman and Weare algorithm*

for $k = 1, ..., K$ **do**
  Randomly draw a walker X_j *from the complementary ensemble* $S_{[k]}(t)$
  Draw z *from the distribution suggested by Goodman and Weare*
  $Y \leftarrow X_j + z[X_k(t) - X_j]$
  $t \leftarrow z^{N_{dim}-1} \frac{p(Y)}{p(X_k(t))}$ ▷ *Computationally expensive*
  $u \leftarrow \mathcal{U}(0, 1)$
  if $u < min\,(1, t)$ **then**
    $X_k(t+1) \leftarrow Y$
    $X_k(t+1) \leftarrow X_k(t)$
  end if
end for

The distribution suggested by *Goodman and Weare* for g is as follows:

$$g(z) \propto \begin{cases} \frac{1}{\sqrt{z}} & \text{if } z \in \left[\frac{1}{a}, a\right] \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

IV.4 Settings

To achieve an efficient sampling of the posterior distribution from equation (25), the **emcee** package was used. The subsequent settings implemented in PYTHON are discussed here. First, in accordance with the package, the logarithm of the PDFs were encoded [32]. Some thoughts were then given to the initialization of the walkers, as one might find natural that a good intialization of the chain is likely to ensure a fast convergence. A maximum likelihood estimation was set up to find the values of the parameter that maximize the likelihood of effectively obtaining our observed values [33]. This was done by minimizing the negative log-likelihood of the posterior (equation (22)) in PYTHON. The walkers were then initialized in a tiny gaussian ball around the maximum likelihood results, as it was advised in the **emcee** package API documentation [32].

For each simulation, a burn-in phase of 10000 iterations over which the samples were discarded was run and the chain was subsequently reset. The sampling was then conducted with 10000 iterations and 100 walkers, which is equivalent to a total run length of 1000000. These particular values were selected to ensure a convergence of the chain with a small statistical error according to [34]. In practice, to achieve this, approximatively $20\ T_0$ of the data needs to be discarded at the beginning of the MCMC simulation, and a total run length of approximatively $1000\ T_0$ should be reached, where T_0 is the autocorrelation time. Thus, for each simulation, the autocorrelation time, which has a dedicated module in the **emcee** package, was inspected and compared with the total run length. After several tests, the settings presented earlier were found to provide adequate ratios of $\frac{totalrunlength}{T_0}$.

V. Model calibration and validation

As it is, the model described in section III.II. needs to undertake a calibration phase to ensure that the obtained results make sense. Reworking equation (18) gives the following:

$$\langle q_h \rangle_N = E[q] - C_0 h^p - e_{h,N} \tag{40}$$

In other words, if one holds the PDFs of the true mean $E[q]$, the error constant C_0, the order of the discretization error p and the sampling error $e_{h,N}$, prediction of the observed value of q on a mesh of resolution h can be achieved according to equation (40). The PDFs previously mentioned are obtained from data of a number of meshes M. Therefore, the idea is to perform a DNS on a finer mesh h_{finest}, which will provide an observation for q : $\langle q_{h_{finest}} \rangle_{obs}$. If this value is included in the 90 % credibility interval of the predicted observed value on the finest mesh $\langle q_{h_{finest}} \rangle_N$, then the model is validated. Otherwise, the discretization error prediction cannot be taken with confidence. The estimation of the PDF of $\langle q_h \rangle_N$ according to (40) follows a change of variable which is explained in more details in Appendix C.

IV. Code validation

As *Oliver et al.* [12] made their DNS results and variance estimation available [35], I had the idea to compare their results to the one I obtained for their data with my code. This stands as an efficient way to validate (or invalidate) my Python implementation. This validation phase was undertaken for two types of quantities : a scalar, the centerline mean velocity U_{cl} and a single-point statistics, the mean velocity $\langle u \rangle$. The PDFs of the error parameters q, C_0 and p are presented here for the scalar quantity, and, for the sake of clarity, they were omitted for the single-point statistics. For both quantities, the normalized discretization error $\frac{\epsilon_h}{q_{obs,h}}$ was computed on the nominal mesh and the whole PDF is displayed for the centerline mean velocity U_{cl}, whereas the 90 % credibility interval only is presented for the mean velocity $\langle u \rangle$. The computation of the normalized discretization error

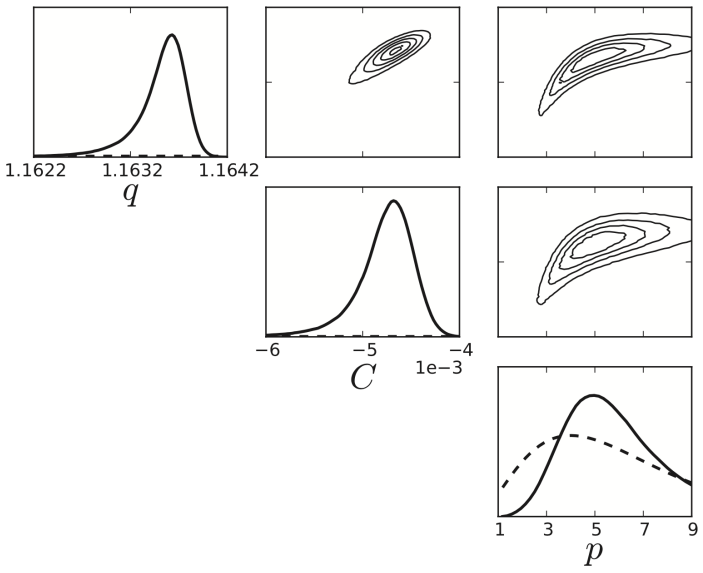


Figure 1: PDFs of the parameters from Oliver et al. [12] for the centerline mean velocity. The diagonal plots are the posterior marginal densities of the parameters and the off-diagonal plots are the joint posterior projected on the parameter space.

follows a change of variable which is explained in depth in Appendix D. Finally, for both quantities, the results of the model validation introduced in section III.V are showed.

I. A scalar quantity : the centerline mean velocity U_{cl}

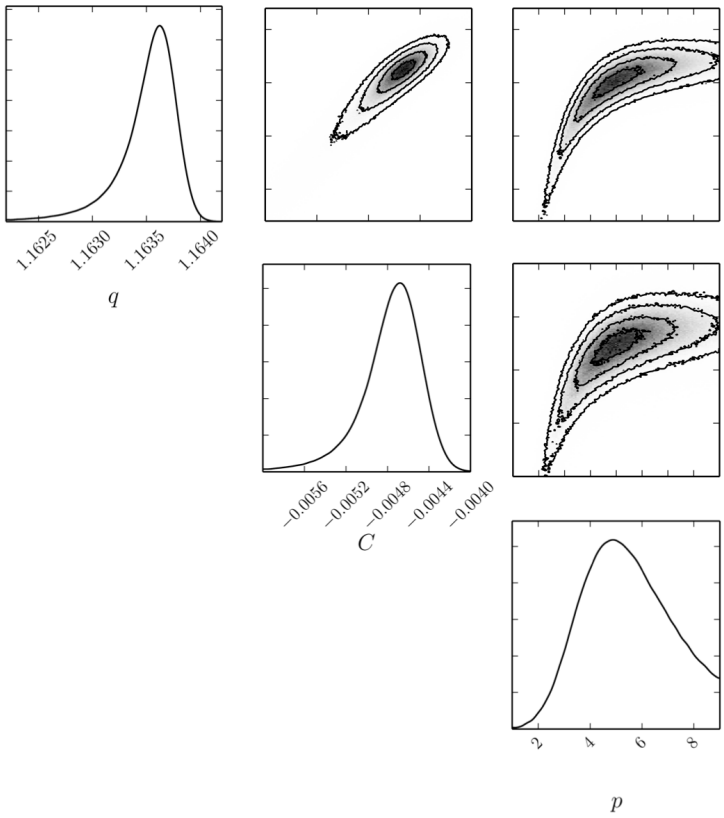


Figure 2: PDFs of the parameters obtained with my code for the centerline mean velocity. The diagonal plots are the posterior marginal densities of the parameters and the off-diagonal plots are the joint posterior projected on the parameter space.

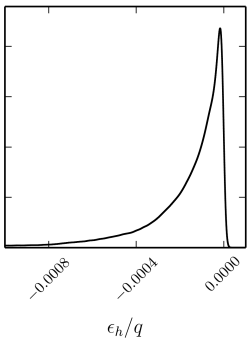


Figure 3: PDF of the normalized discretization error for U_{cl}, obtained from my code.

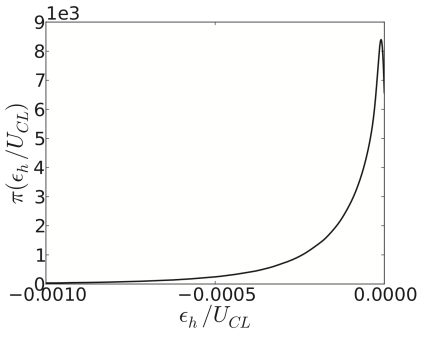


Figure 4: PDF of the normalized discretization error for U_{cl}, obtained from Oliver et al. [12].

Figure 1, 2, 3 4, 5 and 6 show that the obtained results are in very good agreement with Oliver's. A tiny difference between figure 1 and figure 2 may be noticed in the joint posterior contours : it seems to come from the different axis set up.

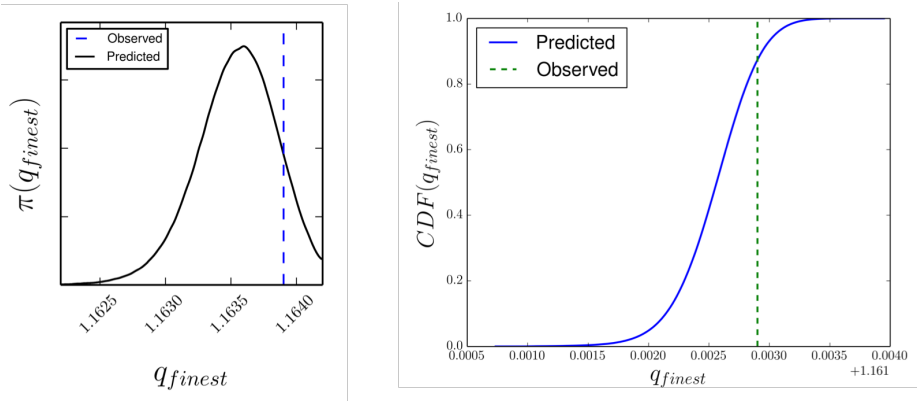


Figure 5: PDF (left) and CDF (right) of the prediction of q_{finest} computed according to equation (40) with my code. In both cases, the observed value is plotted in dashed lines.

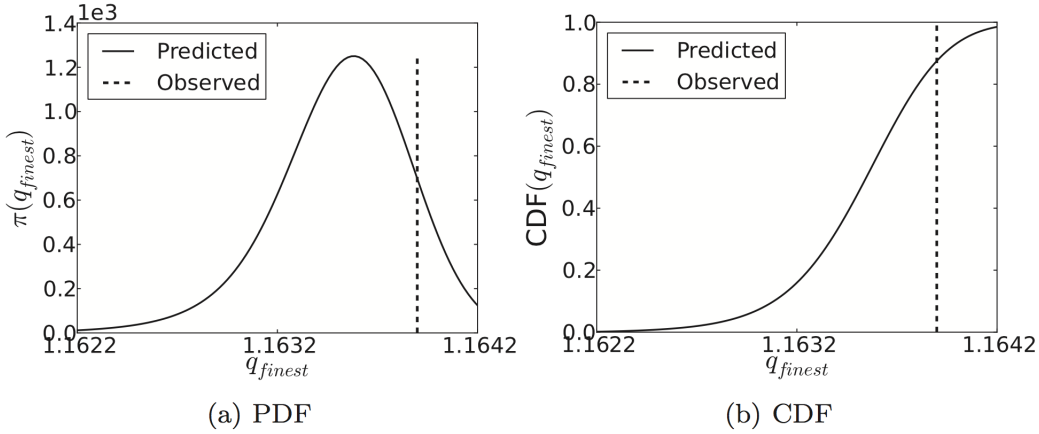


Figure 6: PDF (left) and CDF (right) of the prediction of q_{finest} computed according to equation (40), from Oliver et al. [12]. In both cases, the observed value is plotted in dashed lines.

II. Single-point statistics : the mean velocity $\langle u \rangle$

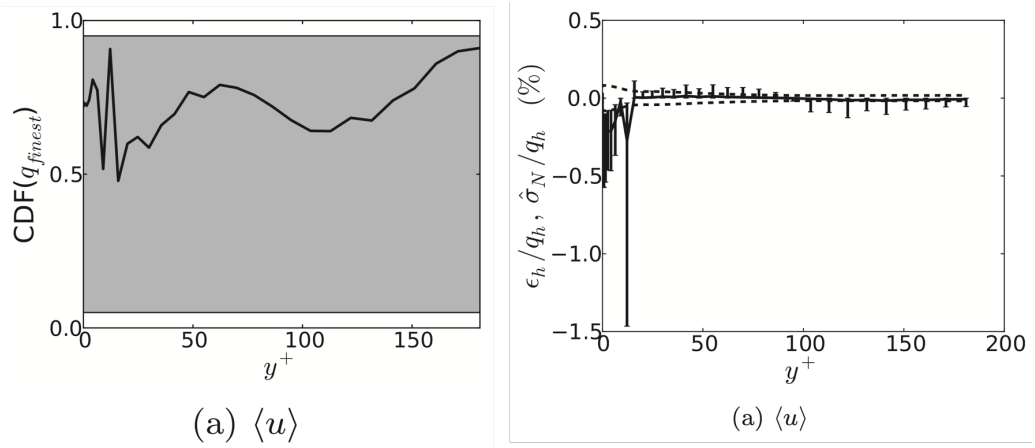


Figure 7: Model calibration according to Oliver et al. [12] (left) : CDF value of the observed data across the channel height. Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in dashed lines) (right).

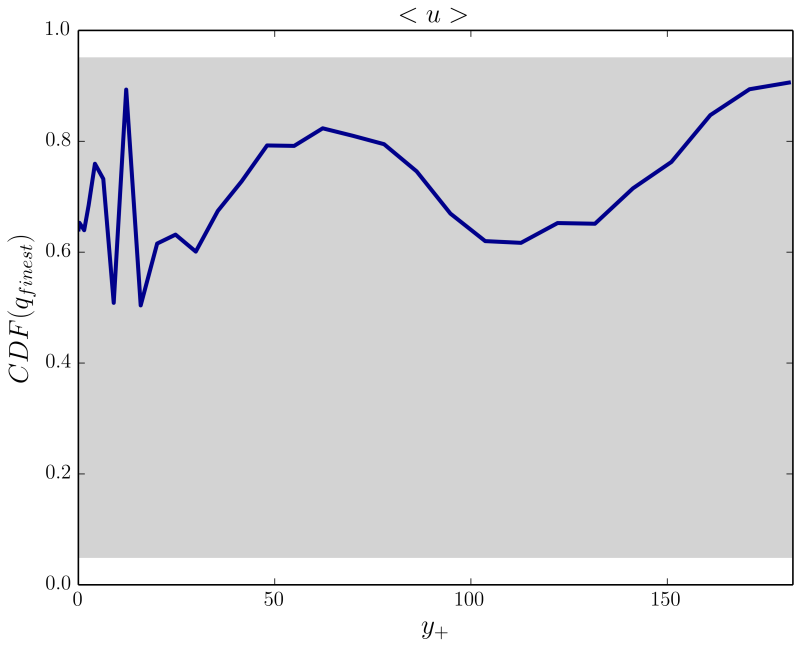


Figure 8: Model calibration obtained with my code : CDF value of the observed data across the channel height.

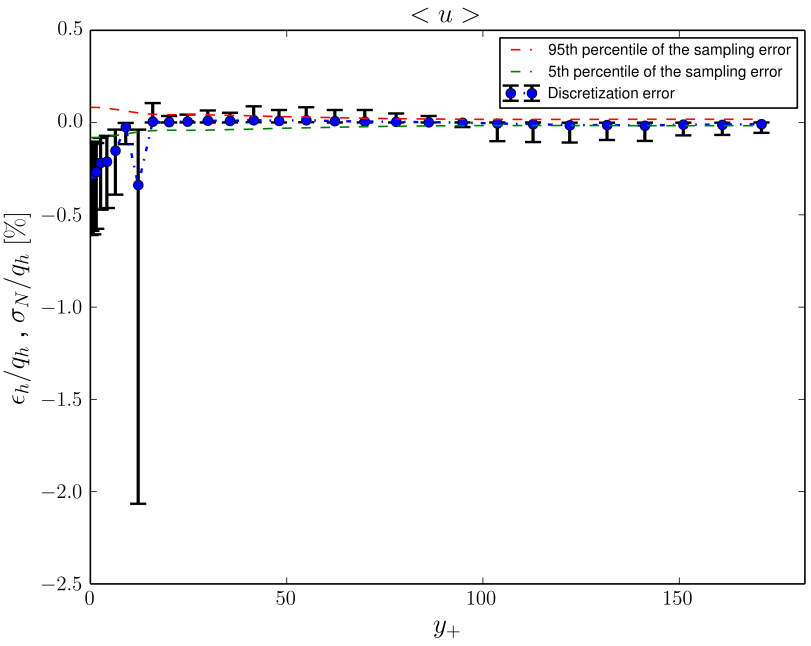


Figure 9: Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in green and red dashed lines), obtained with my code.

The examination of figure 7, 8 and 9 highlights again a very good agreement between our results and Oliver's. The calibration plots are extremely similar, while a minor difference in the discretization error can be spotted at $y^+ \approx 15$, for which a negative peak is observed at 1.5% in Oliver's result, compared to 2% for ours. This might be due to the randomness of the process and it does not look like our code should be questioned for that matter.

As it is not the purpose of this section, no further comments on the obtained results are made here. This will follow in the next sections. Overall this validation phase was successful and the outputs of our code may be taken to match the model with confidence.

V. Application to the DNS of a turbulent channel flow at $Re_\tau = 180$

In this section, the error model highlighted in section IV. is applied to the DNS of a turbulent channel flow at $Re_\tau = 180$. Several quantities are investigated : the mean centerline velocity as well as a range of single-point statistics, such as the mean velocity, the Reynolds stresses, higher-order statistics with the skewness and flatness, along with the turbulence kinetic energy dissipation.

I. Characteristics of the DNS

The DNS was performed by *Dr. Sylvain Laizet* with the high-order flow solver Incompact3D [36][37] which browses a sixth-order finite-difference scheme for the spatial discretization and a third-order Runge-Kutta scheme for the time advancement. The bulk Reynolds number is $Re_b = \frac{U_b \delta}{\nu} = 4200$ and $Re_\tau = \frac{u_\tau \delta}{\nu} \approx 180$, where δ is the channel height. If no specific mention is made, the quantities are assumed to be normalized by U_b and δ, whereas the presence of the subscript "+" refers to the standard normalization by u_τ and $\delta_\nu = \frac{\nu}{u_\tau}$. To apply the error model based on Richardson extrapolation, a nominal and coarsest mesh were defined, along with a finest mesh dedicated to the calibration of the model. Let us emphasize on the fact that, in order to adequately perform the "deterministic" Richardson extrapolation, three meshes of different resolutions at least are needed. However, as DNS is computationally expensive, the probabilistic model was tested here with two meshes only. The mesh resolutions along with the domain characteristics were taken to be similar to the ones of *Oliver and al.*[12] to allow for an easy comparison of the results. In order to minimize the computational cost due to data storage, the outputs of the simulation were saved every five time steps and a constant time step $\Delta t = 0.005 \times 5 = 0.025$ was used. The simulation was run for 250000 times steps on the whole (equivalent to 50000 time steps when considering the saved files). The following table resumes the DNS characteristics :

Grid	L_x	L_z	N_x	N_y	N_z	Δx^+	Δz^+	Δy^+_{wall}	Δy^+_{CL}	Re_τ
Coarsest	4π	2π	96	65	96	23.4	11.7	0.98	31.08	178.96
Nominal	4π	2π	192	129	192	11.7	5.8	0.99	7.81	178.00
Finest	4π	2π	384	257	384	5.8	2.9	0.99	1.98	177.87

Table 3: Characteristics of the DNS simulation at $Re_b = 4200$ and $Re_\tau \approx 180$

II. Sampling parameters

Dr. Laizet provided me with gross results of twelve tensors $(u_x, u_y, u_z, \frac{du_i}{dx_i}, \forall i \in [1,3])$ to enable me to compute the relevant statistics outlined earlier. To reach an acceptable level of file sizes, for each mesh resolution, the outputs were provided with respectively $n_x = 32$ and $n_z = 2$ values in x and z. This is far from what would usually be used to compute the relevant statistics (all the produced data would be used to increase accuracy), but this stands as a first attempt to estimate the uncertainty in DNS statistics. One should therefore bear that in mind when considering the reached level of error. The number of samples N along with their length n_T and

the spacing between them Δt_s must now be given some thought. According to *George* [4], the relative error in estimating the mean of the velocity $\langle u \rangle$ is :

$$\epsilon_T = \sqrt{\frac{2I}{n_T}} \frac{\sigma_u}{\langle u \rangle} \tag{41}$$

where I is the integral time scale defined as $I = \int_0^\infty \rho(\tau) d\tau$, with ρ the autocorrelation function, n_T is the length of a sample and σ_u the standard deviation of the mean velocity which can be directly estimated from the data. The integral time scale was computed using the MATLAB code developed for the estimation of the autocorrelation function (see section II.II.3) along with a trapezoidal method to approximate the integral. Then, several level of accuracy and their subsequent sample size were tested. Introducing the turbulence intensity as $T_u = \frac{\sigma_u}{\langle u \rangle}$, the following table resumes our findings:

ϵ_T [%]	I [-]	T_u [-]	n_T [-]
0.5	751.4	$9.8e^{-4}$	58
0.1	”	”	1445
0.05	”	”	5780

Table 4: Length of sample T needed to achieve a statistical error ϵ_T for the mean velocity $\langle u \rangle$

As higher-order statistics are investigated in this work, higher sample length are needed [4] to ensure small statistical error. Therefore, after several tests, a sample length of $n_T = 9000$ was selected. This choice effectively gives a good match of the Reynolds stresses (second-order moments) with the data of MKM [38], which is considered as a strong reference in the literature. However, given the total length of our signal (45000 time steps), only $N = 5$ samples can be produced with this choice of sample length. This is relatively low, especially when one aims at using the variance estimator developed in section II.II which is based on an extension of the Central Limit Theorem, valid as $N \to \infty$. One way to overcome this issue would be to run the DNS simulation for a longer time, in order to collect more samples of a fixed length $n_T = 9000$. The interval between the samples was taken to be $\Delta t_s = 100$, meaning that our samples are very likely to be correlated, which is fine with our variance estimator, designed for correlated samples.

III. Prior information

Prior information needs to be encoded for the parameter of our model $\bar{q}, C_0$ and p:

$$\bar{q} \sim \mathcal{N}(q_0, \sigma_q^2) \qquad C_0 \sim \mathcal{N}(0, {\sigma_{C_0}}^2) \qquad p \sim Gamma(\alpha, \beta) \tag{42}$$

q_0 was set to the data from MKM at $Re_\tau \approx 180$ [38], σ_q and σ_{C_0} were respectively set to $\sigma_q = 0.25 q_0$ and $\sigma_{C_0} = 4\sigma_q$. These value were taken from *Oliver et al.*[12] as they provide a coherent dispersion of the data around their mean. For the discretization error order p, α and β were respectively set to 3 and $\frac{1}{2}$ to give probability to a wide range of orders. This is a reasonable choice given the mix of orders of the schemes present in the Incompact3D solver.

IV. Results

In the following, results of the bayesian probabilistic model applied to several DNS statistics are highlighted.

IV.1 Centerline mean velocity

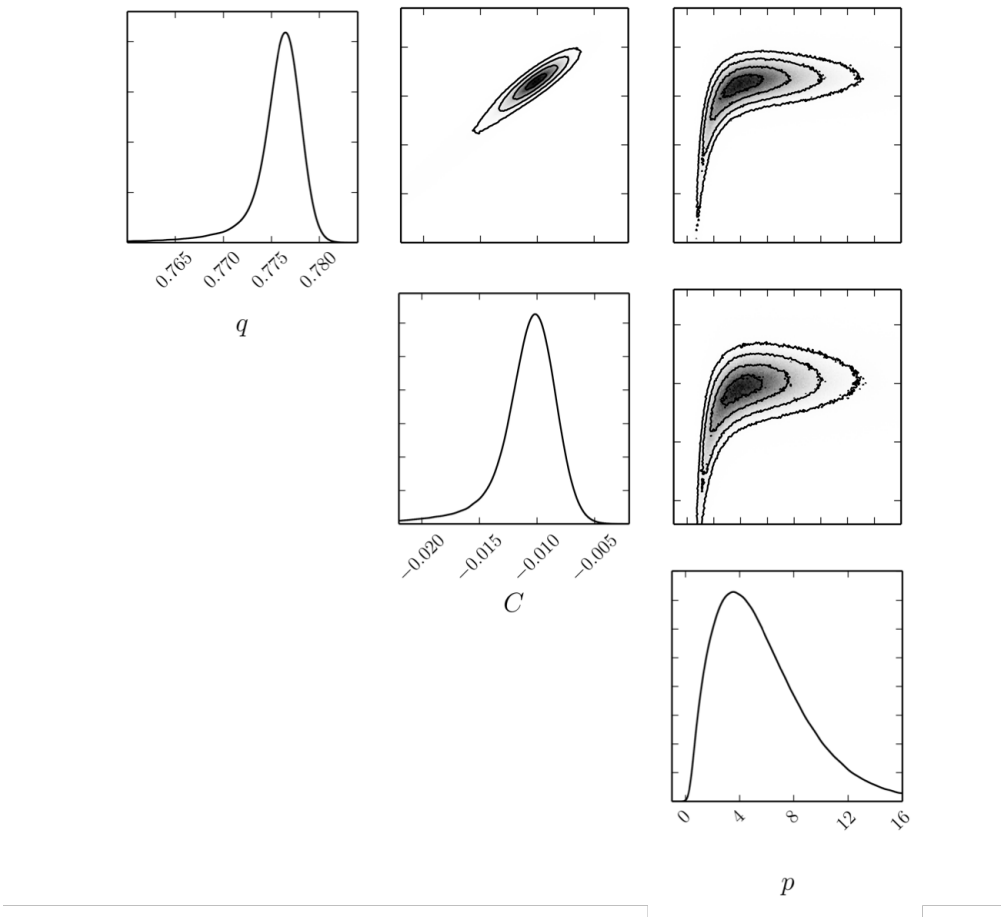


Figure 10: PDFs of the parameters for the centerline mean velocity U_{cl}. The diagonal plots are the posterior marginal densities of the parameters and the off-diagonal plots are the joint posterior projected on the parameter space.

Figure 10 shows that the uncertainty in the true mean is quite small : the gap between the 5th and 95th percentile to the mean value reaches 1.1%, which is not too bad considering the low numbers of values available in x and z to average our data. A large uncertainty in the error order p can be noticed, with a respective first and third quartiles at 3.11 and 7.56. The most probable value is at $p = 5$, which can make sense given the mix of order in the schemes (6 and 3 respectively for the spatial and time discretization) of the Incompact3D solver. Concerning the error constant C_0, one might notice that zero probability is given to positive values of C_0. This means that the model is entirely certain of its sign. Figure 11 highlights a quite small discretization error with a high probability assigned to values less than 0.2%. The shape of this PDF is quite particular, with zero probability assigned to positive values and high probability assigned to very small negative values. This is consistent with the PDF of C_0 which is bounded away from zero. And, as $\epsilon_h = C_0 h^p$, only C_0 can influence the sign of the discretization error. The large peak of probability near zero is due to several features of the joint probability distribution of C_0 and p:

- The uncertainty on p is quite large with non-negligible probability assigned to high values of p. Therefore,

as $h \leq 1$, h^p gets smaller as p increases.

- In the meantime, as p increases, C_0 does not vary much (see Figure 10), which leads to smaller values of the discretization error.

Concerning the calibration of the model, figure 12 and 13 show that it is successfully validated for the centerline mean velocity, with a CDF of the observed value around 0.89, which is still in the 90% credibility interval.

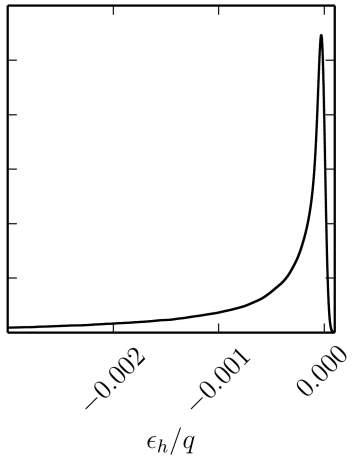


Figure 11: PDF of the normalized discretization error for U_{cl}.

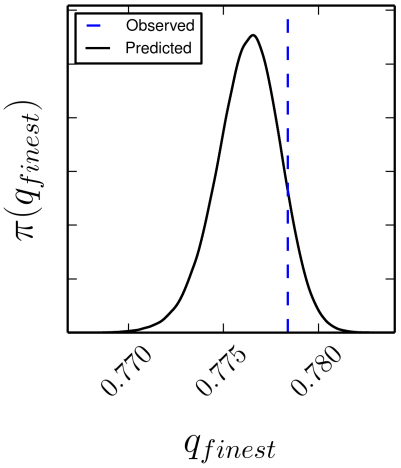


Figure 12: PDF of the prediction of q_{finest} for U_{cl}. The observed value is plotted in dashed lines.

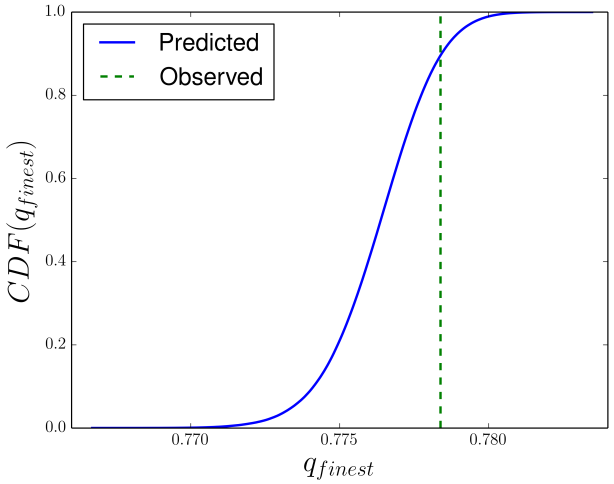
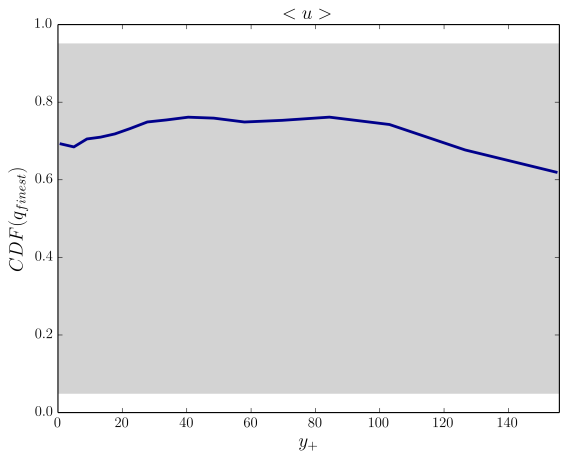


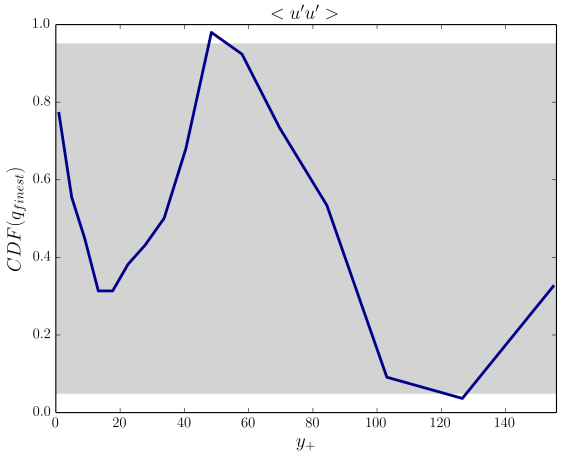
Figure 13: CDF of the prediction of q_{finest} for U_{cl}. The observed value from the simulation is plotted in dashed lines.

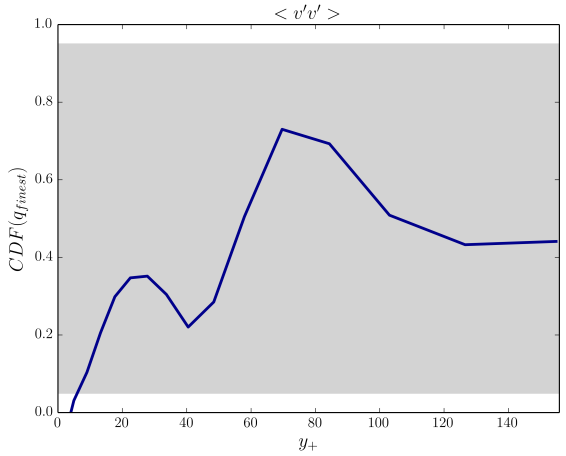
IV.2 Single-point statistics

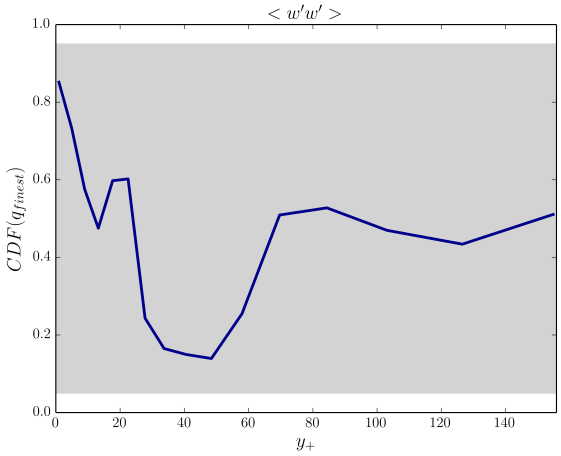
For the single-point statistics, the calibration model results are first highlighted. Then follows a summary of the discretization and sampling error results for these quantities, which will be commented in more details.

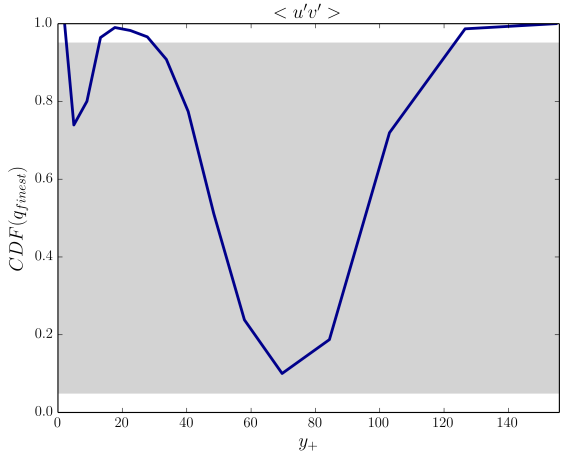
Calibration model results

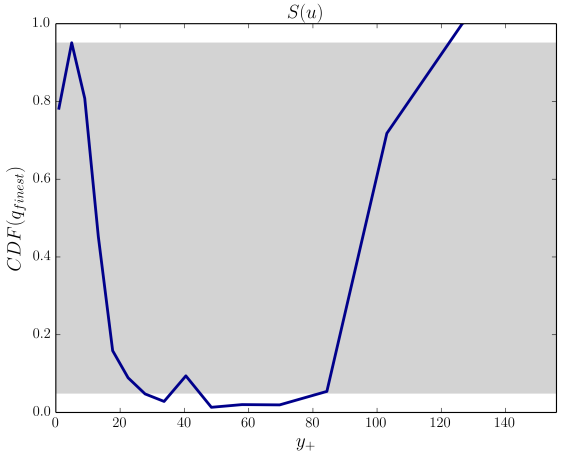












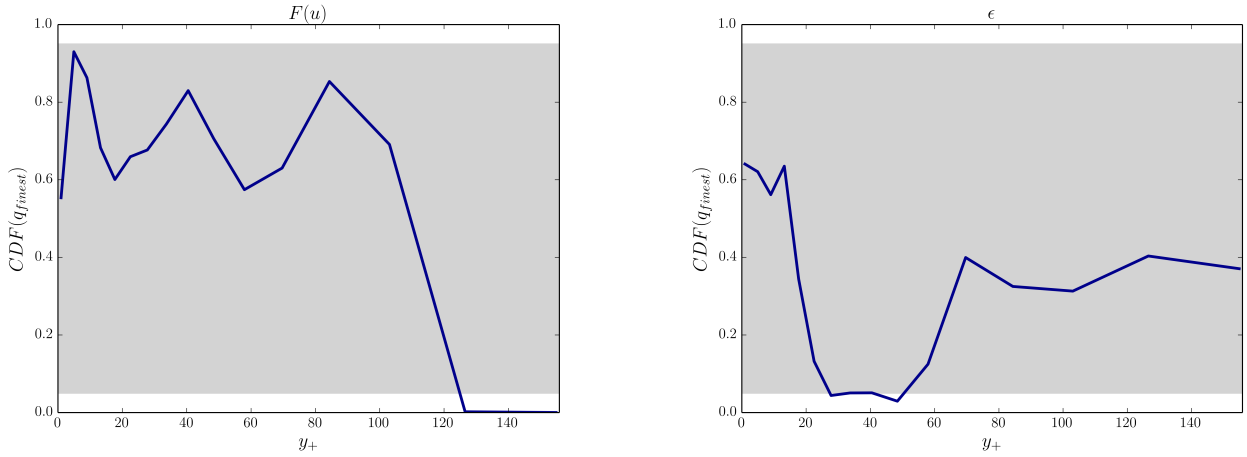


Figure 14: Results of the calibration model for the single-point statistics. The blue curve is the CDF value of the observed data on the finest mesh at each y^+ position. The grey rectangle defines the 90% credibility interval.

As one may notice on figure 14, the model is not successfully validated for certain quantities (at specific y^+ positions at least). Indeed, if the CDF value of the observed data falls out of the the grey region, which represents the 90% credibility interval, it means that the data effectively observed on the finest mesh is drawn from the tail of the distribution of the prediction q_{finest}. In this case, our simulation observations and the model are in poor agreement and the discretization error estimation cannot be trusted with high confidence. This appears as a strong limitation of the model presented here and it is therefore important to try to understand the origin of this issue.

Oliver et al. [12] had the same issues with part of their data and therefore provided some useful and interesting explanations for this feature of the calibration phase. They observed that for most points where an issue occurred, the data were not converging monotonically with increasing mesh refinement. The following graph resumes the difference between non-monotonic and monotonic convergence: For example, for $\frac{\langle u'u' \rangle}{U_b^2}$, the two

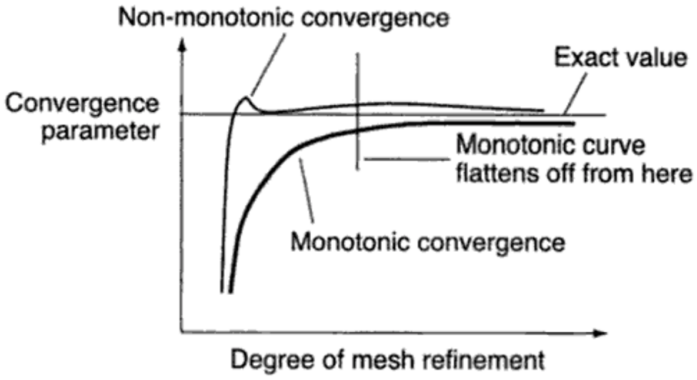


Figure 15: Characteristics of monotonic and non-monotonic convergence, taken from [39].

points at $y^+ \approx 48.4, 126.6$ which are not validated by the model exhibit the following data on the coarsest, nominal and finest mesh:

location	Coarsest	Nominal	Finest
$y^+ \approx 48.4$	3.038	3.037	3.169
$y^+ \approx 126.6$	1.202	1.013	1.047

Table 5: Observed data from the coarsest, nominal and finest mesh for several y^+ location of $\frac{\langle u'u' \rangle}{U_b^2}$

The first two rows of the above table highlight the non-monotonic convergence with increasing mesh refinement : in both cases, an increase of the finest value compared to the nominal value is observed, while the nominal value is decreased compared to the coarsest. *Oliver et al.* [12] suggested that the discretization error model with one term only for the error function, $\epsilon_h = C_0 h^p$ cannot properly capture a non-monotonic convergence. This makes sense as $\epsilon_h = C_0 h^p$ is a monotonic function of h. Richardson extrapolation uses the assumption that the first term in the Taylor expansion largely dominates over the other and, as we have just seen here, it is not always the case. A way to properly capture this non-monotonic convergence would be to use more complex model for the discretization error [40]. Typically, the introduction of a second-term in the definition of ϵ_h as $\epsilon_h = C_0 h^p + C_1 h^{p+1}$ might solve the issue. Indeed, if the two constant of the error C_0 and C_1 compensate each other in a certain way, non-monotonic behavior might appear in the evolution of the discretization error.

Still, when examining the data from the coarsest, nominal and finest mesh for the skewness $S(u)$, we noticed that the model was not validated for points that were exhibiting a monotonic convergence. This is quite surprising and *Oliver et al.* [12] did not report this kind of issue. Let us remind that, in order to properly perform Richardson extrapolation when the true order of the discretization error is unknown, three grid solutions in the asymptotic regime are required [40]. The asymptotic regime refers to the case for which the leading error term dominates the error formula. The fact that we used data from only two refined meshes is causing some issues, in my opinion. Even though this stochastic model works fine with data from only one mesh (as it is stochastic and not deterministic), it is not clear that it actually makes sense to use less than three grid solutions, as there are three unknowns in the model C_0, p and $\bar{q}$. This might be a reason to explain why the model is not validated for data exhibiting monotonic behavior.

From the previous considerations, it seems that the model failure is due to the fact that the discretization error is not in the asymptotic regime. According to *Oliver et al.* [12] this does not necessarily imply that the discretization resolution or the averaging period for the computation of the statistics are not sufficient. For example, the model failure could be explained by the mesh-dependence of the dominant error and the subsequent error order. This could occur as the spatial error might dominate on a certain mesh whereas the temporal would on another one. And, by definition, the discretization error should not vary from one mesh to another for a given quantity, thus explaining why our model is invalidated. Finally, the second-order discretization error model $\epsilon_h = C_0 h^p + C_1 h^p$ could be implemented, provided that data from four refined meshes would be available. Indeed, as mentioned above, it is highly recommended to introduce more data to inform the error when a new unknown is added in the error formula. Just as it is the case for the deterministic method, the number of mesh-related solutions should be equal to the number of unknowns. Even though some quantities were invalidated by the model, it is usually the case for only a few y^+ locations. Unlike *Oliver et al.* [12], who have decided not to show

any results in this case, I though it would be quite extreme to discard all the results. Therefore, I have decided to highlight the discretization error results, with a particular mention when the model was invalidated.

Discretization and sampling error results

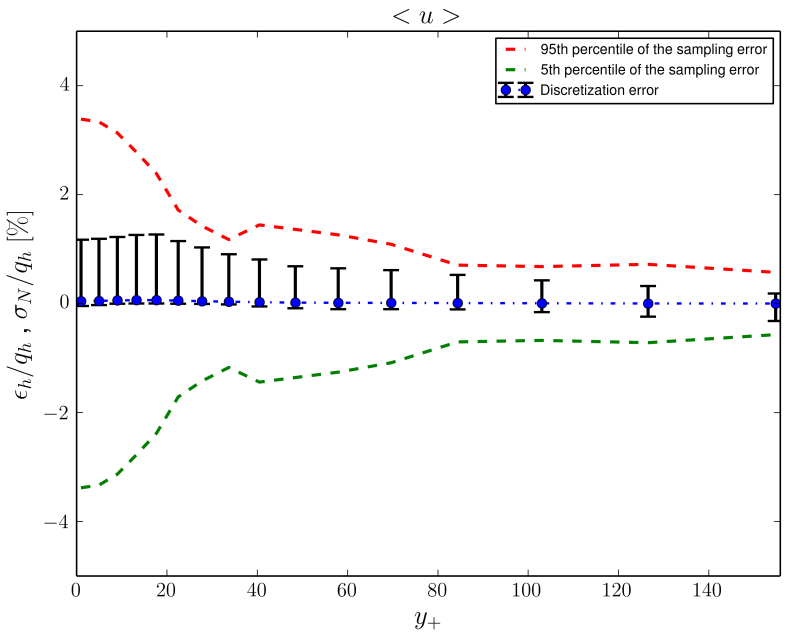


Figure 16: Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in green and red dashed lines) for the mean velocity $\langle u \rangle$ on the nominal mesh.

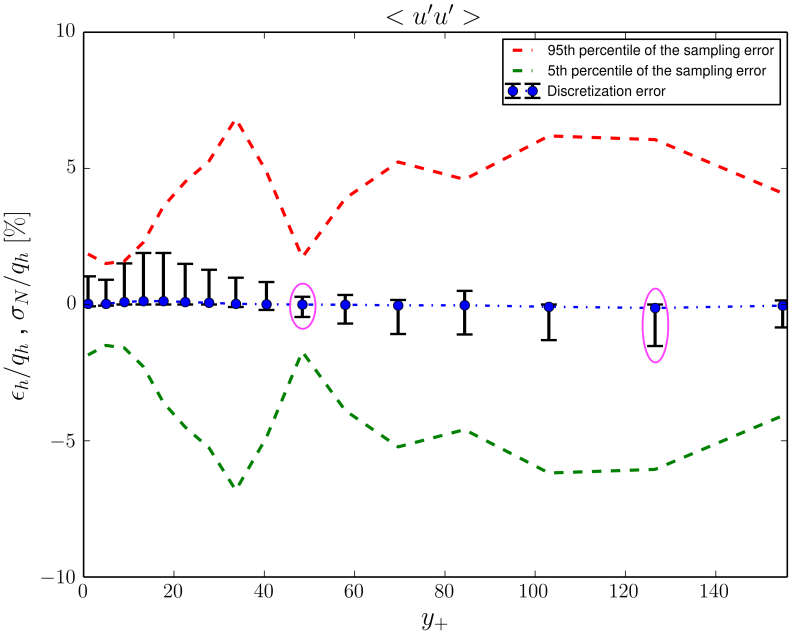


Figure 17: Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in green and red dashed lines) for $\langle u'u' \rangle$ on the nominal mesh. The pink circle refers to the points for which the calibration model failed.

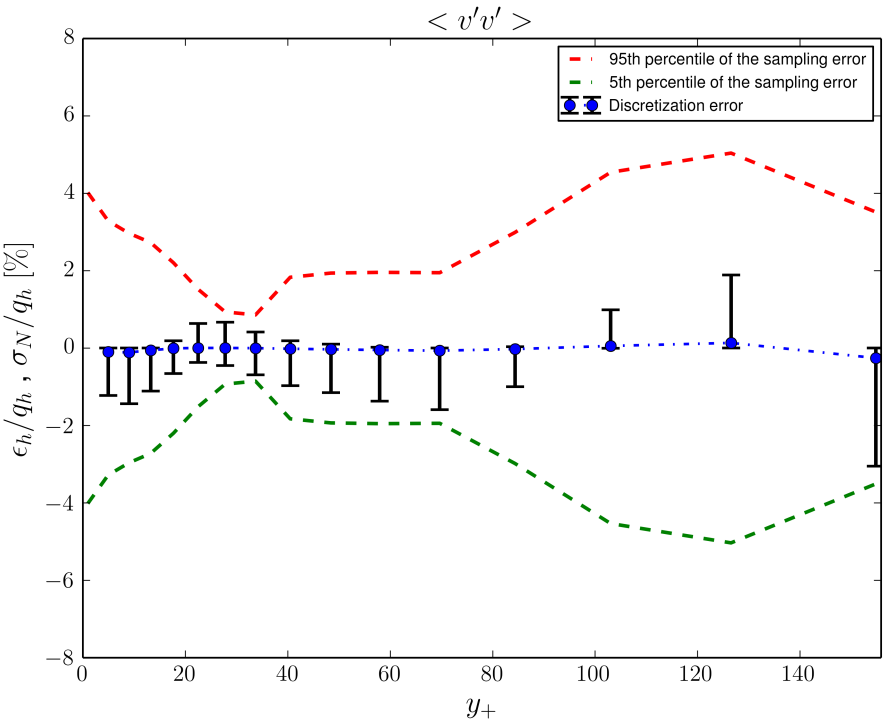


Figure 18: Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in green and red dashed lines) for $\langle v'v' \rangle$ on the nominal mesh.

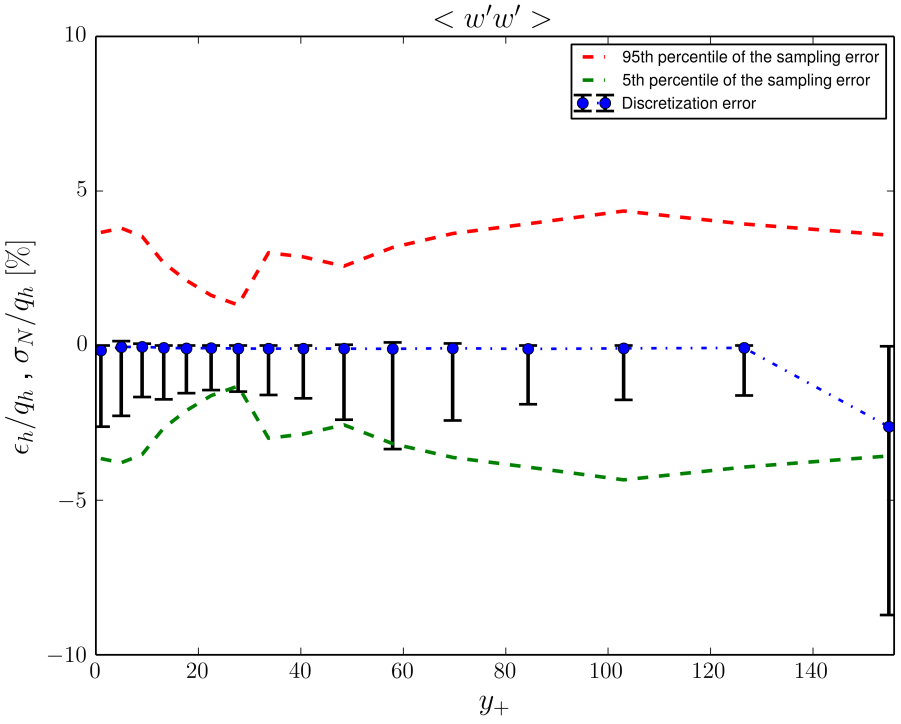


Figure 19: Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in green and red dashed lines) for $\langle w'w' \rangle$ on the nominal mesh.

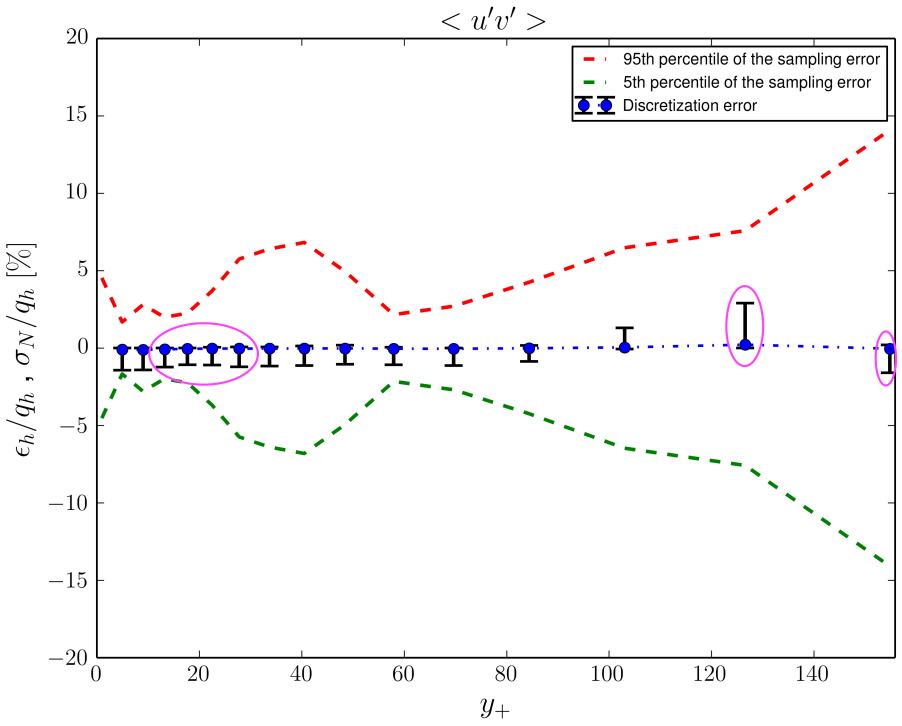


Figure 20: Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in green and red dashed lines) for $\langle u'v' \rangle$ on the nominal mesh. The pink circle refers to the points for which the calibration model failed.

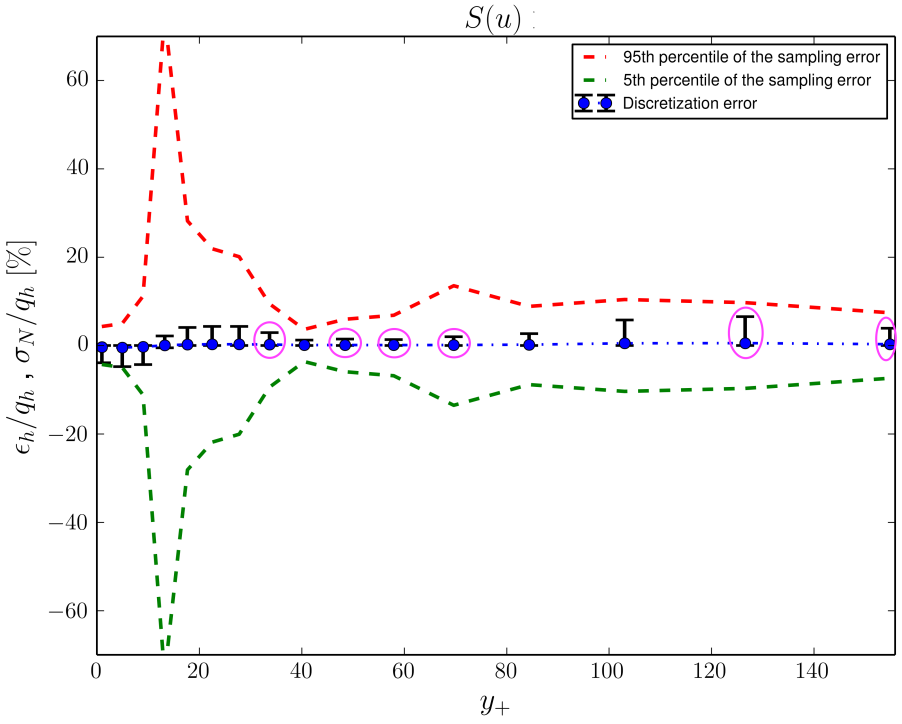


Figure 21: Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in green and red dashed lines) for the skewness on the nominal mesh. The pink circle refers to the points for which the calibration model failed.

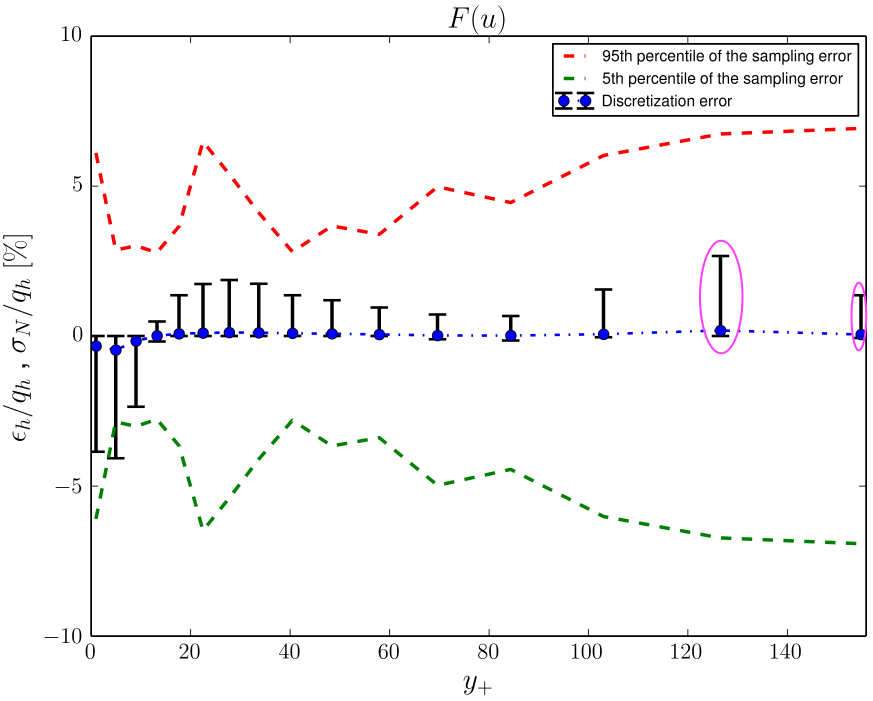


Figure 22: Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in green and red dashed lines) for the flatness on the nominal mesh. The pink circle refers to the points for which the calibration model failed.

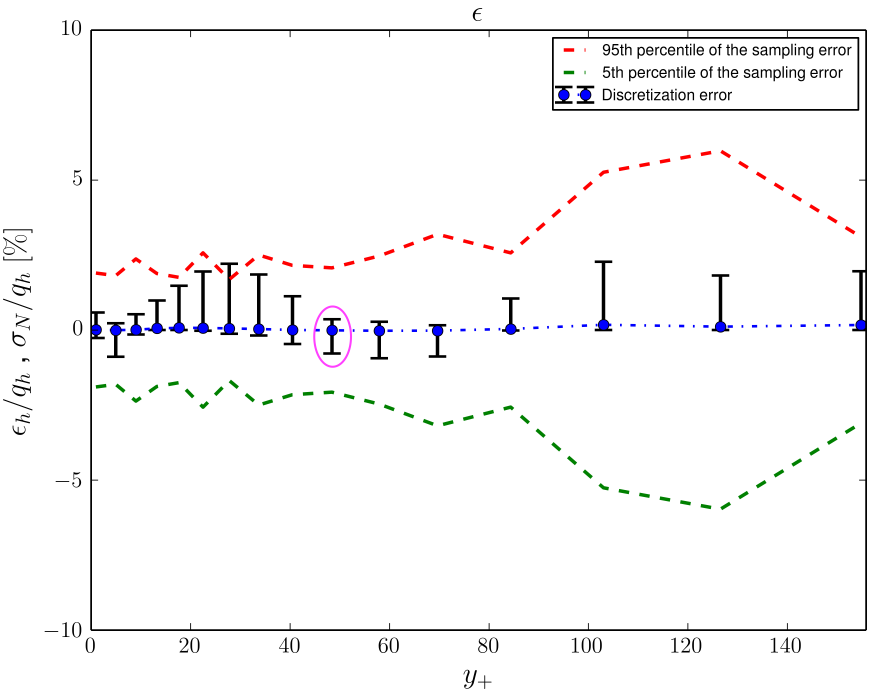


Figure 23: Normalized discretization error (error bars) and sampling uncertainty (5th and 95th percentile in green and red dashed lines) for the turbulence energy dissipation ϵ on the nominal mesh. The pink circle refers to the points for which the calibration model failed.

From the previous results, several interesting comments can be made. First, one may notice that the sampling uncertainty is increasing overall with the order of the computed moments. This is quite as expected as it seems reasonable that more samples are needed to maintain a certain level of accuracy for high-order moments quantities. The difference in sampling uncertainty between statistics of order one and two is globally quite small. For the skewness, the sampling uncertainty reaches very high values ($\geq 60\%$), whereas the flatness exhibits substantially lower sampling uncertainty ($\leq 10\%$). This is not very surprising and might be due to the fact that moments of even orders (such as the flatness) are always positive. An interesting work would be to test the sampling and discretization uncertainty for the fifth and sixth order moments, to check if this trend is confirmed. Unfortunately, no reference results for these high-order moments are available, which makes it difficult to perform an error estimation based on our model as prior information needs to be encoded. One could still test the implementation of an non-informative prior, but it might be a little ambitious to test it with high-order quantities as a first attempt.

Besides, one might find the level of sampling uncertainty of the turbulence kinetic energy dissipation ϵ quite low: it lies between the level of uncertainty of order one and two quantities. Indeed, $\epsilon = 2\nu\langle s'_{ij}s'_{ij}\rangle = \nu\left(\langle\frac{\partial u'_i}{\partial x_j}\frac{\partial u'_i}{\partial x_j}\rangle + \langle\frac{\partial u'_i}{\partial x_j}\frac{\partial u'_j}{\partial x_i}\rangle\right)$ is the sum of products of velocity derivatives which is expected to be quite difficult to sample. However, the results for ϵ were matching MKM data quite well, even with low numbers of points to compute the average, namely $n_x = 32$ and $n_z = 2$ values in x and z. This result might therefore be considered as coherent.

Based on the previous plots, sampling uncertainty seems to dominate over the discretization error. On the contrary, *Oliver et al.* [12] showed in their paper that the predominance of the sampling uncertainty over the discretization error, commonly acknowledged by the research community could be questioned. Still, our results can be explained by the low number of points used to compute the averages : $n_x = 32$ and $n_z = 2$. It would therefore be interesting to compute the discretization and sampling error in the case where $n_x = 192$ and $n_z = 192$ (for the nominal mesh).

Comments on the shape of the two types of error can be made as well. The sampling uncertainty is symmetric, which is coherent with the definition of $e_{h,N}$ as a normal law of mean zero $e_{h,N} \sim \mathcal{N}(0, \hat{\sigma}_N)$. Concerning the discretization error, one might notice that it is of a particular sign most of the times, with very asymmetric error bars. This is due to the shape of the PDF of C_0, which tends to be bounded away from zero, thus making the model very confident of the sign of the discretization error. This was already observed for the centerline mean velocity (see figure 10 and 11).

Final plot with total error

Even though the discretization error is not valid everywhere, we decided to plot the data obtained from simulation for the different quantities along with an error range. The error range was taken to be the range between the 5th percentile and 95th percentile of the true mean. Let us remind that the true mean is given by the following:

$$E[q] = \langle q_h\rangle_N + \epsilon_h + e_{h,N} \tag{43}$$

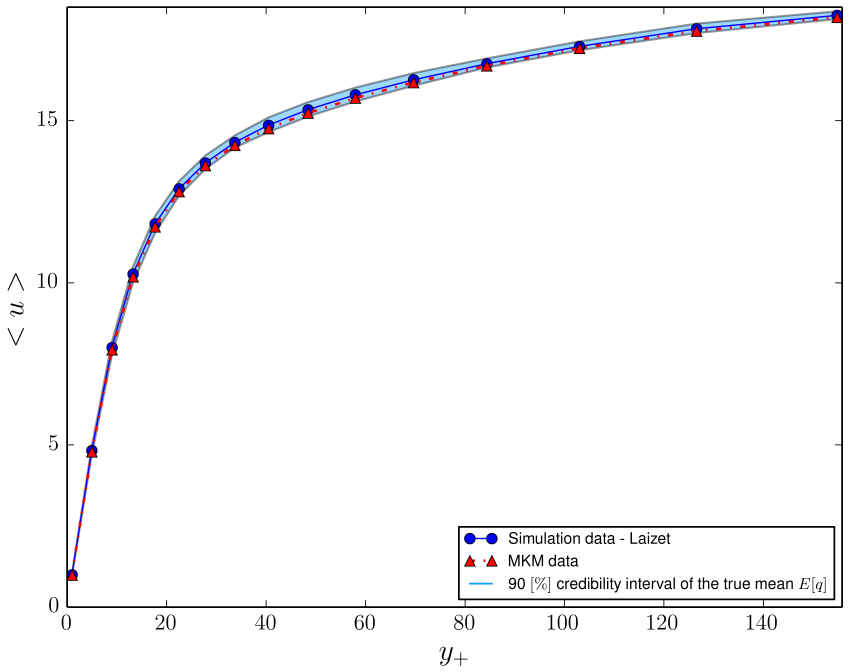


Figure 24: 5th and 95th percentile of the true mean of $\langle u \rangle$ in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM.

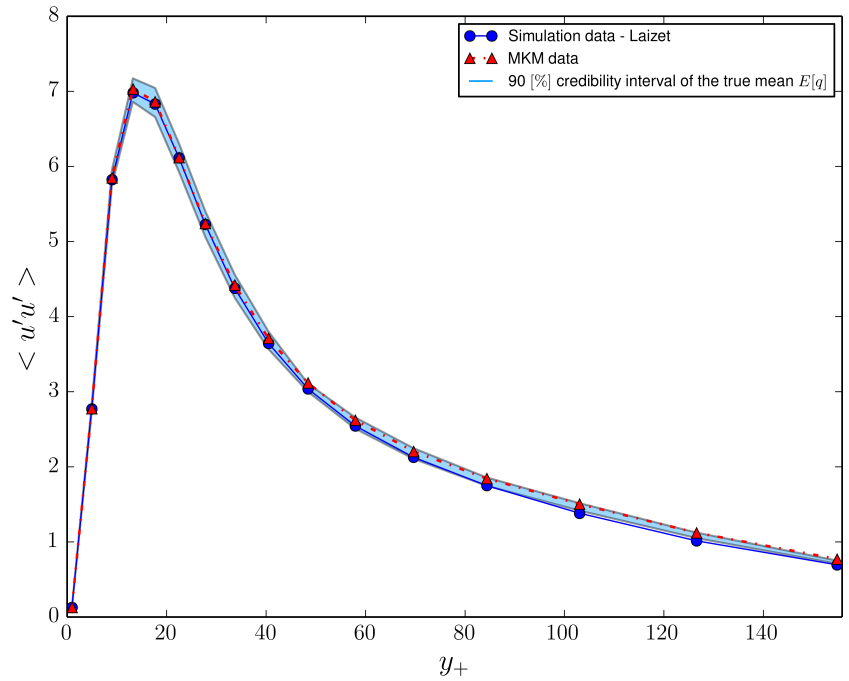


Figure 25: 5th and 95th percentile of the true mean of $\langle u'u' \rangle$ in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM.

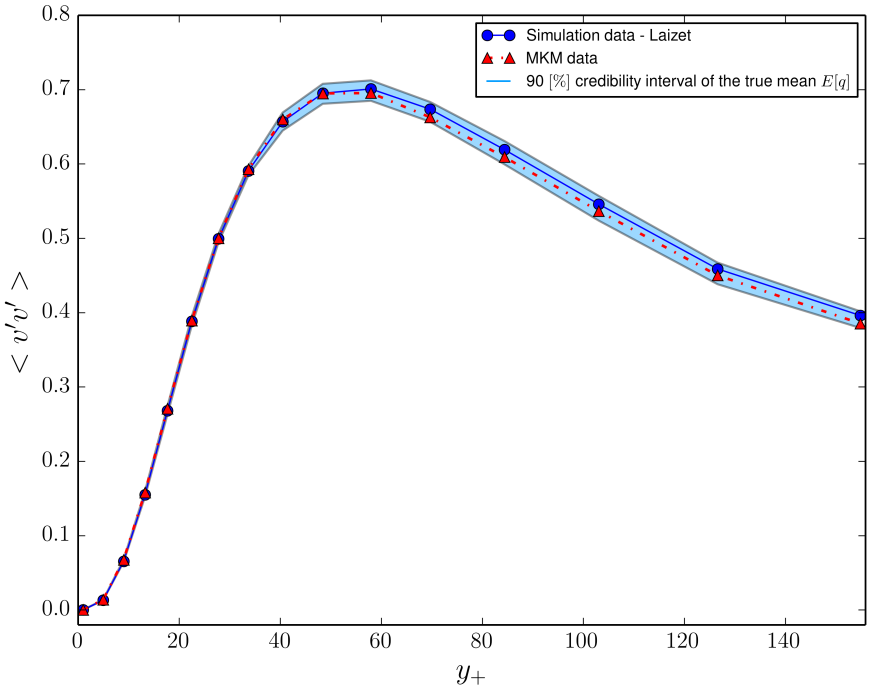


Figure 26: 5th and 95th percentile of the true mean of $\langle v'v' \rangle$ in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM.

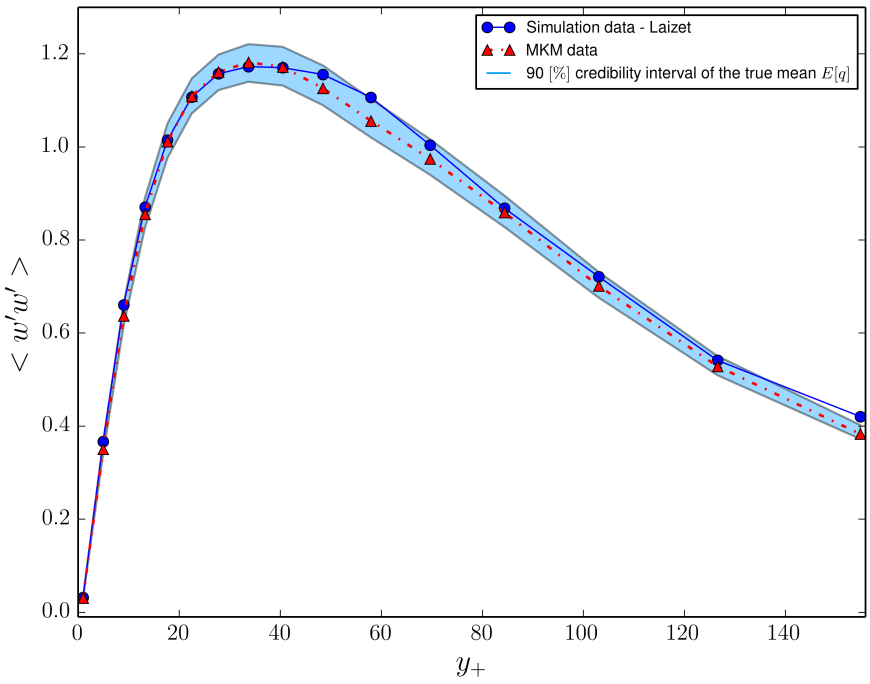


Figure 27: 5th and 95th percentile of the true mean of $\langle w'w' \rangle$ in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM.

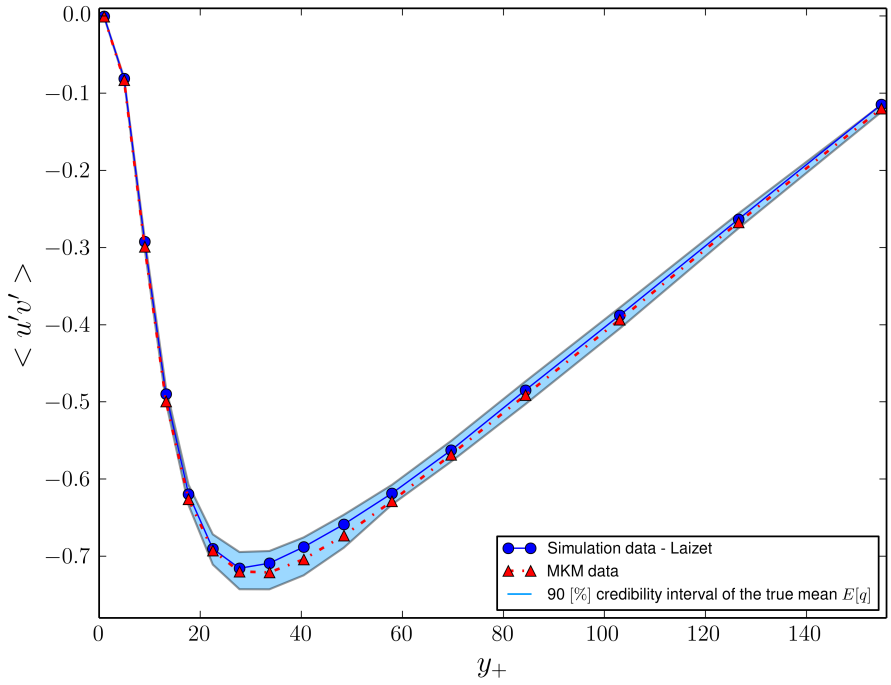


Figure 28: 5th and 95th percentile of the true mean of $\langle u'v' \rangle$ in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM.

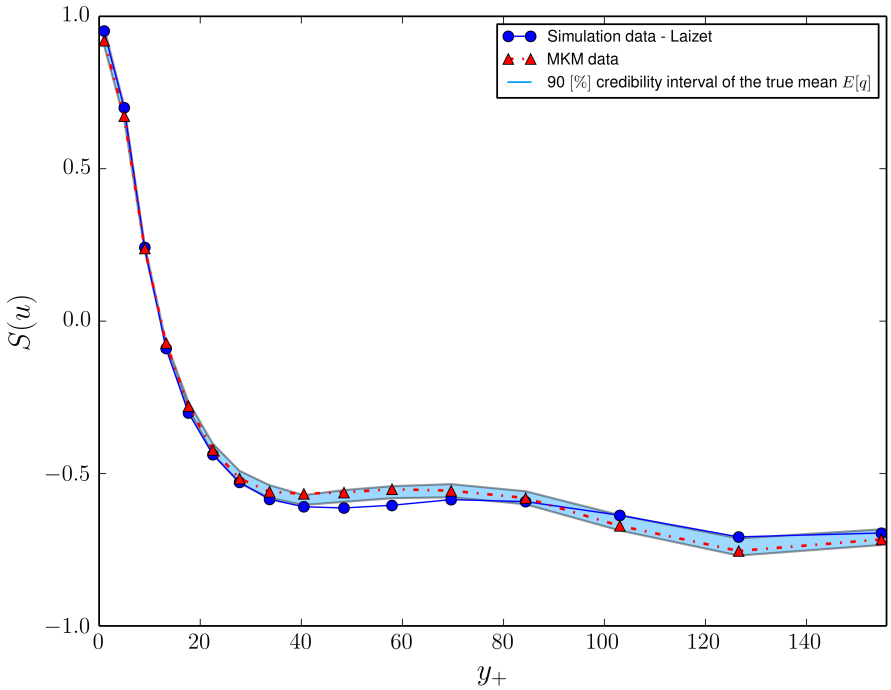


Figure 29: 5th and 95th percentile of the true mean of $S(u)$ in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM.

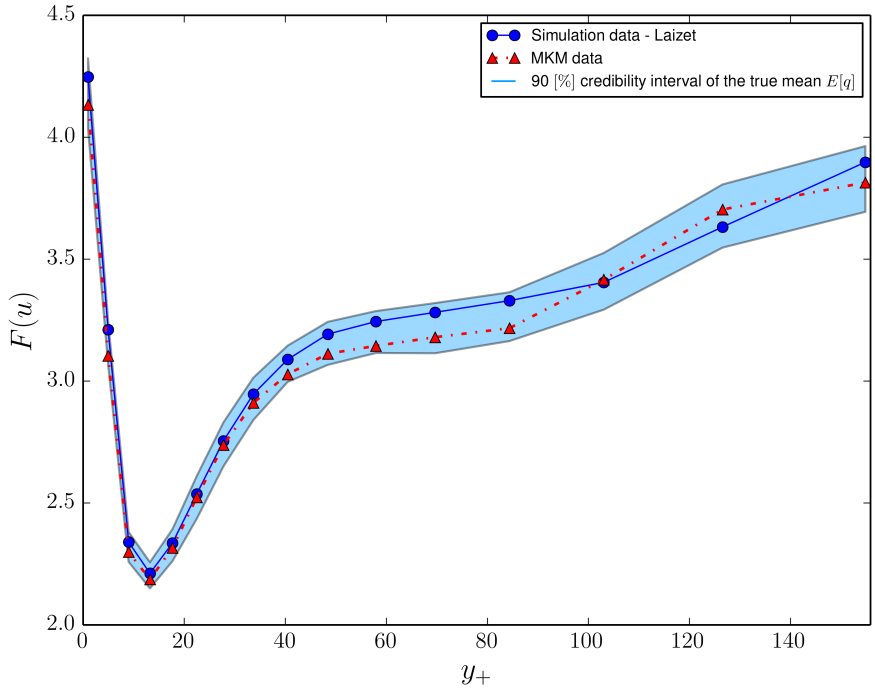


Figure 30: 5th and 95th percentile of the true mean of $F(u)$ in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM.

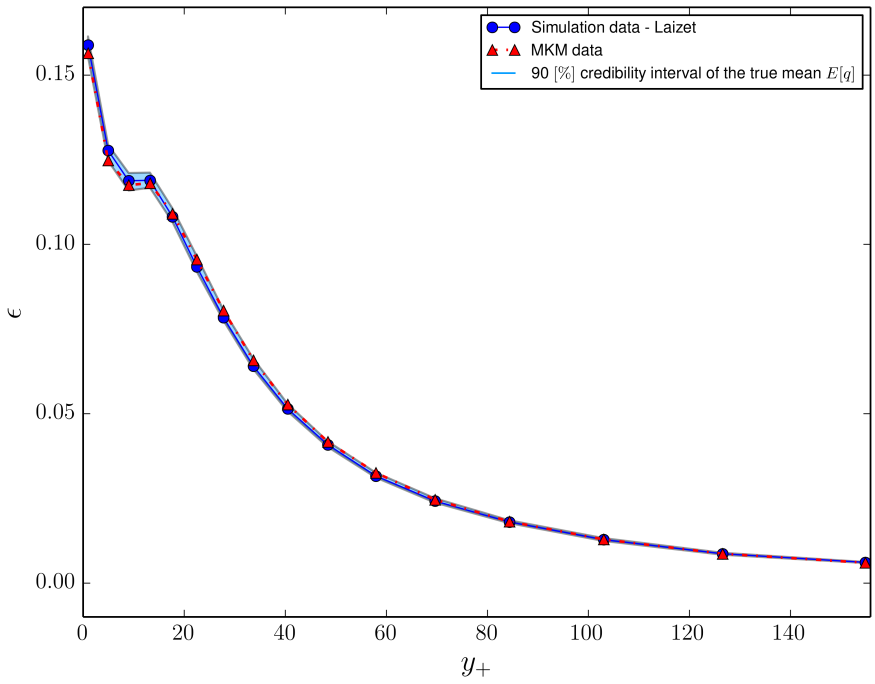


Figure 31: 5th and 95th percentile of the true mean of turbulence kinetic energy dissipation in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM.

The previous plots highlight that the total error range is globally increasing with the order of the computed moment, which seems logical as explained before. Still, the turbulence kinetic energy dissipation ϵ exhibits a substantially small error range, even compared to the mean velocity $< u >$. This is a quite surprising result, given that ϵ is a relatively high-order moment quantity (second order in velocity derivatives). An interesting comment valid for figure 24 to 31 is that MKM data always lie in the 90% credibility interval of the true mean $E[q]$, meaning that the simulation results are not too bad as their related error bars systematically include these values of reference. The error bar range seems overall quite large, especially if it is compared to Oliver's [12], but this is not very surprising given the number of points in x and z, $n_x = 32$, $n_z = 2$ used to compute the statistics. Some issues may be spotted in the plot of the skewness : indeed, the observed data from Laizet's simulation are not in the 90% credibility interval of the true mean which is very problematic and does not really make sense. However this occur for the points which did not successfully pass the model calibration phase and therefore for which the discretization error estimation cannot be taken with confidence.

VI. Investigation and improvement of certain features of the model

In this section, we aim at further investigating certain features of the model presented earlier. First, we shall have a look at the practical implementation of a non-informative prior for the bayesian model. Then, we will examine whether the introduction of a second term in the discretization error formula can lead to successful model calibration or not. Finally, the influence of the mean in x and z will be discussed.

I. Implementation of a non-informative prior

As highlighted in the previous sections, precise information on the statistics computed from DNS are not always available (for example the fifth and sixth order moments of the velocity). In these cases, an alternative solution is to introduce a uniform prior to finalize the bayesian model. This kind of implementation was tested for the centerline mean velocity, for which we already hold some results (see figure 10). Two types of prior were defined:

- A uniform prior π_1 defined as :

$$\pi_1(\bar{q}, C_0, p) = \begin{cases} \frac{1}{68} & \text{if } \ 0 \leq \bar{q} \leq 2, -1 \leq C \leq 1, -1 \leq p \leq 16 \\ 0 & \text{otherwise} \end{cases} \tag{44}$$

- A prior π_2 defined as :

$$\pi_2(\bar{q}, C_0, p) = \begin{cases} \frac{1}{4} \frac{\beta^\alpha p^{\alpha-1}}{\Gamma(\alpha)} \exp(-\beta p) & \text{if } 0 \leq \bar{q} \leq 2, -1 \leq C \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{45}$$

The first prior π_1 was tested and did not give successful results for the marginal PDF of p. Indeed, Figure 32 highlights that the posterior is highly influenced by the prior, as the marginal PDF of p is very close to a uniform distribution, which is very different from the PDF of figure 10. This seems to indicate that sampling from the posterior is not accurate with this type of prior, as it is clear that the data should not reflect this kind of behavior for the error order p. We tried to increase the number of samples computed in the MCMC phase, but the results barely changed. These poor results might also be due to initialization issues, which could cause the walkers to

be stuck in a certain region of the support of the distribution for example. Anyway, it is clear that one holds information on the true order of the error since the numerical schemes were selected and introduced directly by the user and this led me to implement the prior π_2. Indeed, this prior assumes a uniform distribution for $\bar{q}$ and C and a Gamma distribution for p, where α and β should be appropriately selected by the user, based on the introduced numerical schemes. The results are showed in figure 33 and they appear to be in very good agreement with what has been obtained with an informative prior (see figure 10).

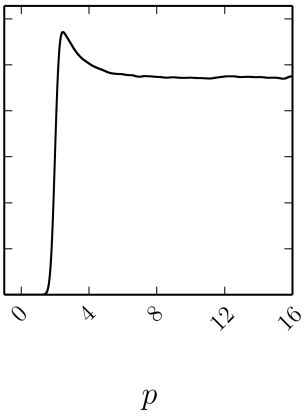


Figure 32: Marginal PDF of p following the implementation of the uniform prior π_1.

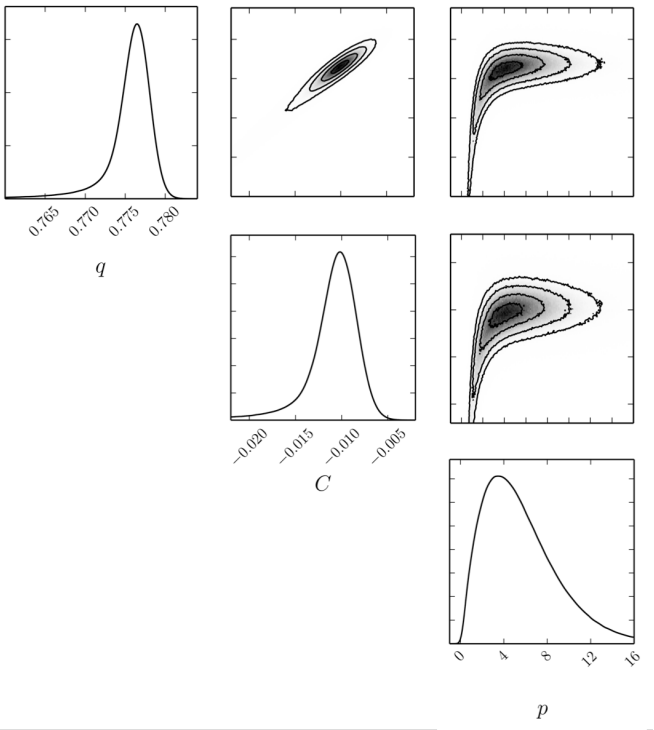


Figure 33: PDFs of the parameters for the centerline mean velocity U_{cl}, with the semi-informative prior π_2. The diagonal plots are the posterior marginal densities of the parameters and the off-diagonal plots are the joint posterior projected on the parameter space.

To confirm the apparent match between the informative and non-informative prior results, precise information of the parameters of the PDFs were compared and summarized in the following table: From table 6, one

Variable	Informative prior	Non-informative prior π_2
Mean of q	0.776155	0.776111
Mean of C_0	-0.010471	-0.010521
Mean of p	5.028749	4.957403
5th percentile of q	0.770229	0.768984
95th percentile of q	0.778852	0.778841
5th percentile of C_0	-0.016464	-0.017747
95th percentile of C_0	-0.007418	-0.007422
5th percentile of p	1.367616	1.205184
1st quartile of p	3.109986	3.017131
3rd quartile of p	7.557985	7.513183
95th percentile of p	12.357239	12.330538

Table 6: Comparison of the PDFs characteristics of q, C_0 and p with a informative a a non-informative prior.

understands that the results are in relatively good agreement between the two configurations. The outcomes of the implementation of a non-informative prior in this bayesian model are very promising and an interesting check would be to apply it to high-order moments quantities for which we hold results with an informative prior. The extension to quantities with unknown characteristics (such as the fifth and sixth order moments for example) could then be studied. It is important to remind that the development of such a model for error estimation in DNS should aim at providing results for all types of DNS, including the ones involving unreferenced Reynolds numbers. Therefore the success of this model and its practicability are highly conditioned by the possibility of using non-informative prior.

II. Enhanced discretization error model

As highlighted in figure 14, the calibration phase of the studied model was not successful for several quantities. *Oliver and al.*[12] also had this issue, in particular for the quantity $\frac{\langle u'u' \rangle}{U_b^2}$. The introduction of a second term in the definition of ϵ_h was therefore tested for $\frac{\langle u'u' \rangle}{U_b^2}$ with Oliver's data, as they provided results for a "fine" mesh [35], with grid resolution $h_{fine} = \frac{64}{180}$ and a number of points in each direction $N_{x_{fine}} = N_{z_{fine}} = 270$ and $N_{y_{fine}} = 180$. The following formula was used for the discretization error:

$$\epsilon_h = C_0 h^p + C_1 h^{p+1} \tag{46}$$

Following the introduction of an additional term in the discretization error formula, prior information on this additional term needs to be encoded. As very few information is available on C_0 and C_1, I have decided to implement non-informative priors for these two random variables:

$$C_0 \sim \mathcal{U}(-1, 1) \qquad C_1 \sim \mathcal{U}(-1, 1) \tag{47}$$

One could have been tempted to keep the normal law of mean zero for C_0, but this was decided for a discretization error model with a unique term. It is therefore safer to use less informed distribution for the priors of the two

constant terms. The results are quite surprising as the points that were initially causing some issues are

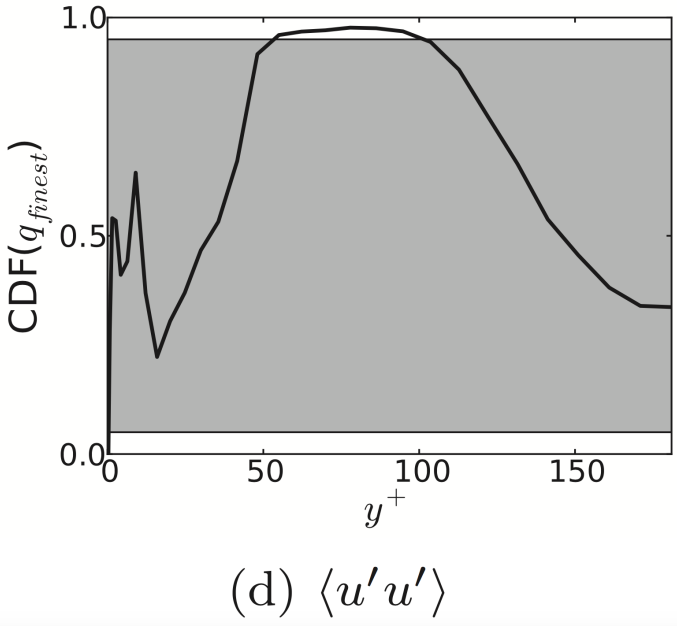


Figure 34: Calibration results for $\langle u'u' \rangle$ with Oliver's data from the coarsest, coarse and nominal meshes, with a discretization error as $\epsilon_h = C_0 h^p$.

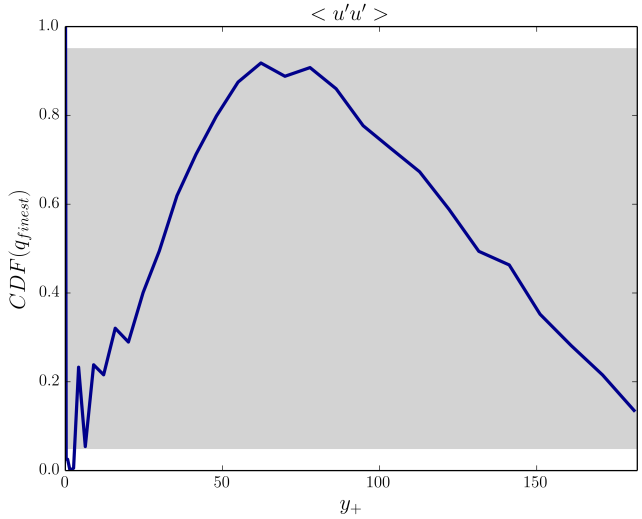


Figure 35: Calibration results for $\langle u'u' \rangle$ with Oliver's data from the coarsest, coarse, nominal and fine meshes, with a discretization error as $\epsilon_h = C_0 h^p + C_1 h^{p+1}$.

now correctly calibrated, whereas a non-negligible amount of points near the wall are now invalidated by the model. These results are not very straightforward to explain : I thought it might come from my estimation of q_0, which needs to be interpolated from Del Alamo and Jimenez and de-normalized with the appropriate wall Reynolds number used by *Oliver et al.* [12]. It is therefore difficult to conclude on the efficiency of this technique to correctly validate the model, especially since it is also computationally expensive to perform the additional simulation required.

III. Influence of the mean in x and z

As I had issues matching the statistics from Dr. Laizet's DNS simulation with MKM, even though it was acted that very good agreement between the two data sets could be reached [37], I had the idea to investigate the influence of the number of points in x and z used to compute the statistics. Later on, I found out that my mistake came from my way of averaging to compute the statistics : I had chosen to average first in space (x and z) and then in time. Usually, the opposite is done and it effectively gave a good agreement between the two data sets. This discovery enabled us to understand that the random process is not ergodic, and it is certainly the sign of a too short time simulation and/or too low number of points in x and z for the averaging process. Besides, in collaboration with Dr. Sylvain Laizet, we thought of a solution to perform error estimation on statistics computed with the usual number of points in x and z for the nominal resolution, i.e $N_{x_{nominal}} = N_{z_{nominal}} = 192$ and $N_{y_{nominal}} = 129$. To overcome memory storage issues which forced us earlier to work with a limited number of points in x and z, our idea has been to generate, for each specific (x, y, z) coordinate, raw results samples that have been collected over a certain time n_T and immediately after averaged over this time. Therefore, only the time averaged values were saved for each samples, which substantially decreased the computational cost of the memory storage process. The number of samples and the simulation time used to compute each of them were respectively fixed to $N = 5$ and $n_T = 9000$ in accordance with the considerations of section V.II. The influence of the mean in x and z on the discretization error and sampling uncertainty was examined for $\langle u \rangle$, $\langle u'u' \rangle$ and $\langle v'v' \rangle$ and is presented in the following figures:

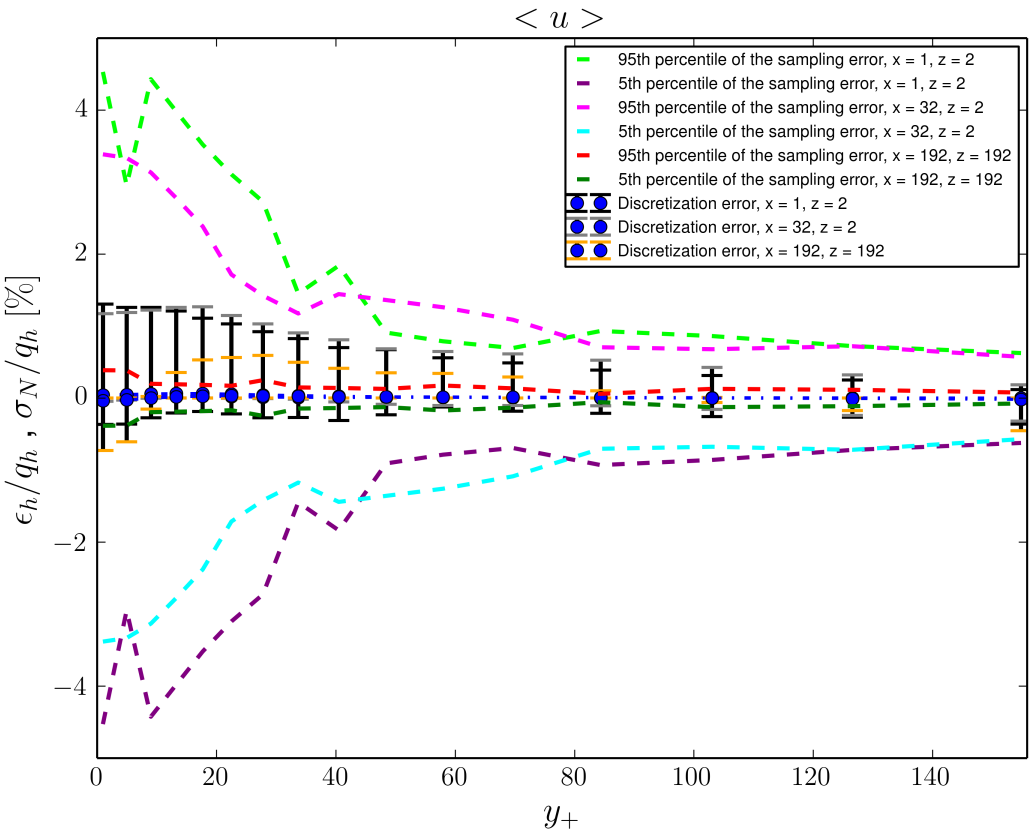


Figure 36: Discretization error and sampling uncertainty for $\langle u \rangle$ for different N_x, N_z.

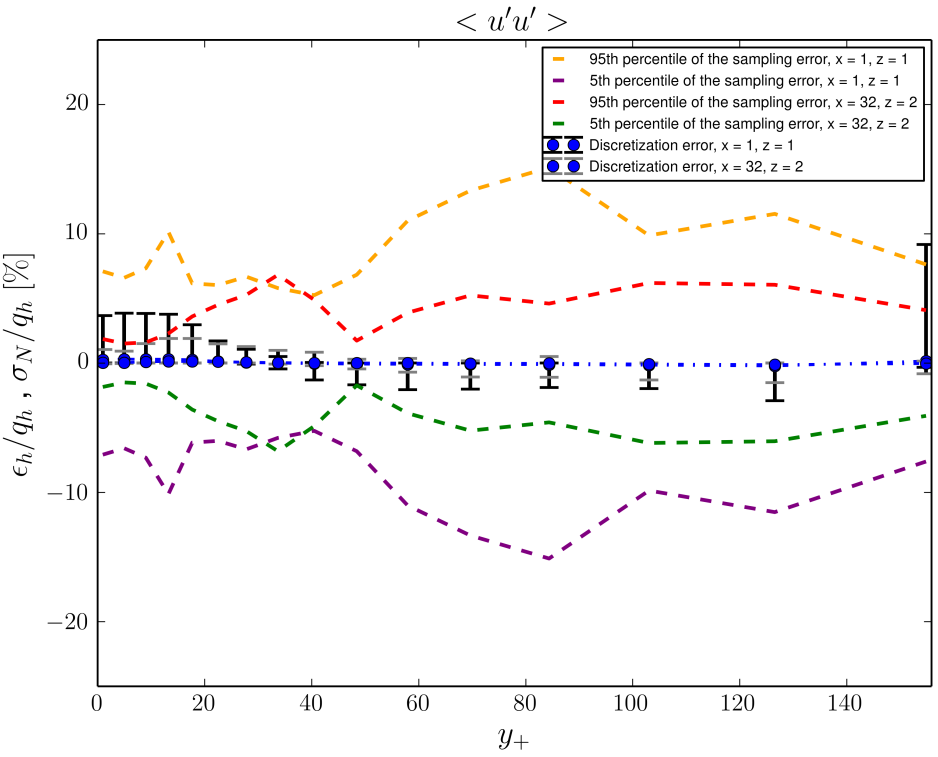


Figure 37: Discretization error and sampling uncertainty for $\langle u'u' \rangle$ for different N_x, N_z.

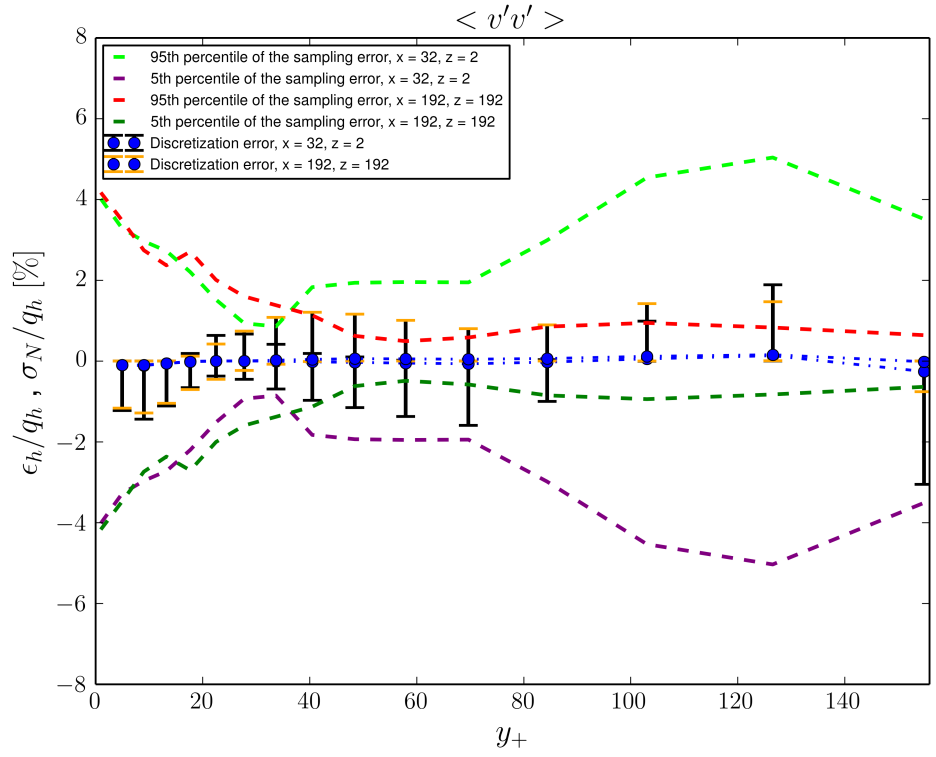


Figure 38: Discretization error and sampling uncertainty for $\langle v'v' \rangle$ for different N_x, N_z.

From these plots, one can notice that the sampling uncertainty is decreasing with the number of points in x and z used to compute the statistics. This is as expected, since adding more points in the computation of a given statistics will undoubtedly increase its accuracy. When $N_x = N_z = 192$, it is clear that the sampling uncertainty reaches very low levels and, in these circumstances (see figure 36 and 38), the sampling uncertainty and the discretization error are in the same order of magnitude. This is in line with the results of *Oliver et al* [12] and justifies the need to jointly estimate the uncertainty from these two sources of error. Regarding the discretization error, for a given mesh, it should not vary with the number of points used to compute the statistics. Our results are inconsistent with this statement, for $\langle u'u' \rangle$ and $\langle v'v' \rangle$ at least. This seems to indicate a significant issue in the discretization error model presented here. Indeed, even if certain points related to particular y^+ positions did not pass the calibration phase, most of them did (see figure 14 for $\langle u'u' \rangle$ and $\langle v'v' \rangle$), meaning that there are no clear issues with the discretization error bars presented in figure 36 to 38. For example, for $\langle u'u' \rangle$ and $y^+ < 20$, the grey and black error bars do not match at all. The same issue is observed for $\langle v'v' \rangle$ in the range $30 < y^+ < 80$. For $\langle u \rangle$, the discretization error bars are quite close for $(N_x, N_z) = (32, 2)$ and $(N_x, N_z) = (2, 1)$, but this is not true anymore when comparing with the case $(N_x, N_z) = (192, 192)$. To ensure that this inconsistency in the discretization error estimation was not the result of the use of only two mesh-related solutions to perform the bayesian Richardson extrapolation, the uncertainty on the nominal mesh was computed for $\langle u'u' \rangle$ with $(N_x, N_z) = (32, 2)$ and $(N_x, N_z) = (2, 1)$ with data from the coarsest, nominal and finest meshes. The discretization error bars were still found to be quite different (see Appendix E). As expected, the final plots of $\langle u \rangle$ and $\langle v'v' \rangle$ for $N_x = N_z = 192$ exhibit smaller error range (see figure 39 and 40). What could be more

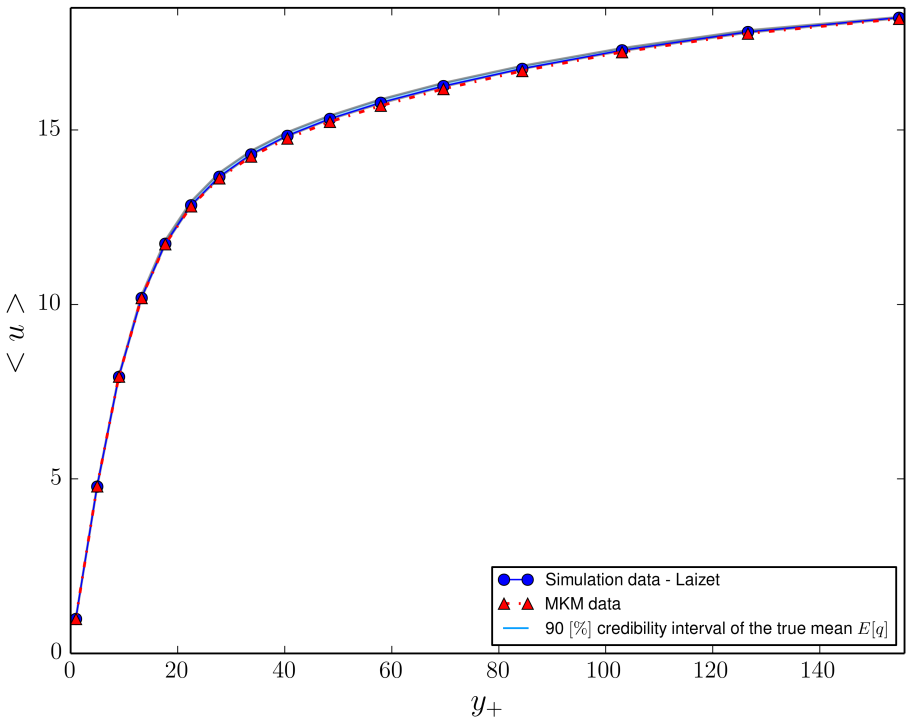


Figure 39: 5th and 95th percentile of the true mean of $\langle u \rangle$ in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM. $N_x = N_z = 192$

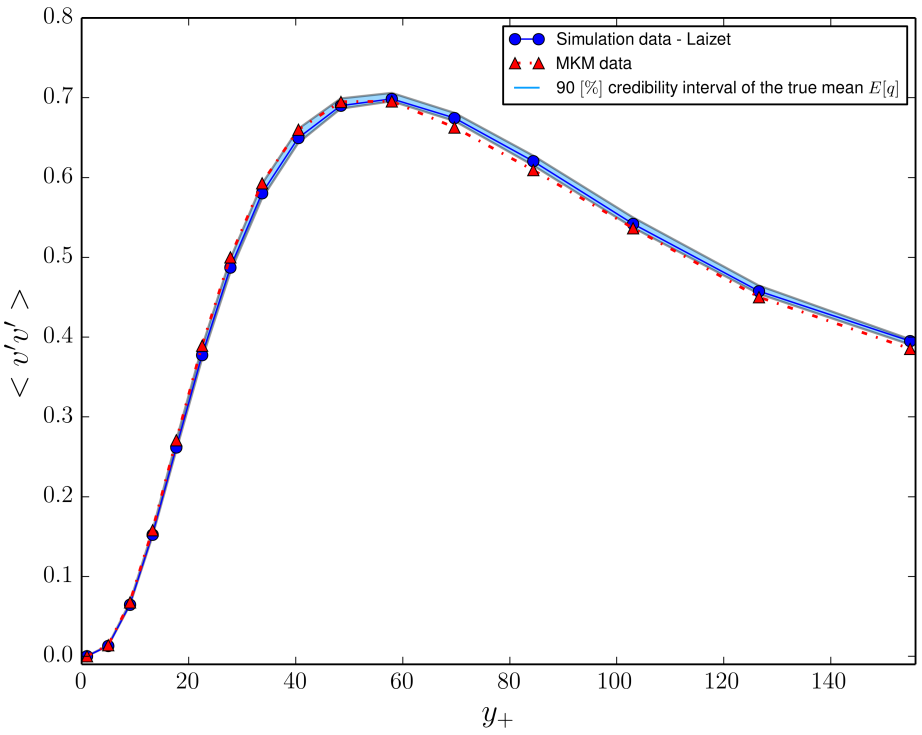


Figure 40: 5th and 95th percentile of the true mean of $\langle v'v' \rangle$ in filled blue, the blue navy circle and the red triangle are respectively the data from Dr. Laizet's simulation and from MKM. $N_x = N_z = 192$

surprising at first, is the fact that for both figure 39 and 40, MKM data do not systematically lie in the 90% credibility interval of the true mean, whereas it was the case for $(N_x, N_z) = (32, 2)$. However, as the accuracy of the computed statistics is increased (by acting on the sampling as it is the case here for example), it is natural to capture more precisely how our data compare to the reference simulation, in terms of error range. In other words, it is clear that quantities exhibiting significant error bars are very likely to include the values of MKM, but this does not make the results any better as they are highly uncertain. Therefore, under the assumption that this model is appropriate and makes physical sense, these figures highlight that the results of Dr. Laizet's simulation are not in such a good agreement with MKM data for certain y^+ positions ($y^+ > 60$ for $\langle v'v' \rangle$ for example), as the error range do not include these values of reference. Still, this statement must be nuanced as one should keep in mind that the model itself was already called into question (see previous comments in section VI.III and V.IV.) and does not appear fully reliable.

VII. Conclusion

I. Work overview

Error estimation in statistics computed from DNS was investigated first from a theoretical point of view, with a precise overview of the two main sources of error in DNS. This thesis was based on the bayesian probabilistic model developed by *Oliver et al.* [12], which appeared as a very promising tool to jointly estimate the discretization and sampling uncertainty. In a first step, the model was successfully implemented and our results were tested against

Oliver's. An estimator for the sampling uncertainty was developed and successfully tested as well. In a next step, the model was applied to the DNS of a turbulent channel flow at $Re_\tau \approx 180$ performed by Dr. Sylvain Laizet. In addition to the work of *Oliver*, high-order quantities were investigated and it enabled us to gain some useful knowledge on the behavior of the uncertainty for these high-order quantities. Throughout the project, several tests were developed to evaluate the robustness of the model. Notably, some issues were raised regarding the calibration of the model, which was found to fail in several cases. For now, no clear solutions were found to overcome this particular matter, even though the approach suggested by *Oliver* was tested. Some discretization error estimates were also set in default when the influence of the mean in x in z was investigated. Still, a promising feature of the model was highlighted : the implementation of a non-informative prior, which might be very useful when quantities with a priori unknown characteristics are investigated provided successful results for the centerline mean velocity. As a final remark, I would like to stress that for now, this model is impractical as it requires data from a finer mesh (than the nominal) to validate the model, which is computationally expensive, time-consuming and unrealistic for relatively large domain.

II. Future work

Some aspects of the presented work and features of the presented model could be further examined. This subsection aims at gathering my thoughts on the question, with the explicit goal of inspiring future research on the subject.

- First of all, in an attempt to be more rigorous in the estimation of the sampling uncertainty, it is essential to collect the raw data over longer simulation time in such a way to create more samples N of a fixed length ($n_T = 9000$). All of this goes towards a better approximation of the Central Limit Theorem which is valid as $N \to \infty$. Notably, the quality of the CLT approximation could be assessed with a confidence interval of the estimate. In other words, for a desired precision of the variance estimation, one should be able to identify the number of samples N required. As DNS simulations are computationally expensive, it was essential to notice this issue quite early, which, unfortunately, was not my case.

- As well, I believe it is paramount to provide data from at least three meshes to perform Richardson extrapolation, as advised in section V.IV.2. In this work, the model was tested with data from two meshes only and I became aware of the problem it triggered only at an advanced stage of the project. This is the sign of poor project management as performing simulation on an additional mesh is computationally expensive and time-consuming.

- Additional investigations on the use of non-informative priors in the bayesian model would be highly recommended. The following steps are suggested: test of the implementation for high-order statistics for which one holds results with informative prior to confirm the promising results of the centerline mean velocity, then tests for quantities with unknown characteristics such as the fifth and sixth order moments....

- Due to a lack of time, the total error estimation was not performed for all quantities with an acceptable level of points in x and z for the averaging process, i.e. $(N_x, N_z) = (192, 192)$ and obviously it would be very interesting to finalize it.

- Regarding the error estimation process in itself, the implementation of a more physical criteria would be an interesting improvement of this model which is based on mathematical considerations only. To ensure that the mesh is sufficiently resolved, a common and classical approach in DNS [1] relies on the comparison of the Kolmogorov length scale η_K with the mesh spacing in each direction. If $\Delta_x, \Delta_y, \Delta_z \leq \eta_K$, then it is clear that the mesh is able to capture the smallest scales of the turbulence phenomena, thus giving an acceptable lower limit for the mesh resolution. The Kolmogorov length scale can be computed from the turbulent kinetic energy ϵ as follows:

$$\eta_K = \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}} \tag{48}$$

  It is essential to note that the estimation of η_K, as a function of y^+ is not very accurate as it is highly mesh-dependent, since ϵ is itself mesh-dependent. Since the uncertainty on ϵ can be estimated with the model presented in this paper, according to the principle of propagation of uncertainty, the relative error committed on the estimation of the Kolmogorov scale is:

$$\frac{\Delta\eta_K}{\eta_K} = \nu^{\frac{3}{4}} \frac{1}{4} \frac{\Delta\epsilon}{\epsilon} \tag{49}$$

  where, $\frac{\Delta\epsilon}{\epsilon}$ is an output of our model for each y^+ positions considered. From the above considerations, the physical criteria of the error estimation could consist in the systematical comparison of the total error range (i.e. the length of the 90% credibility interval of the true mean for each y^+ positions) with the Kolmogorov length scale. Indeed, if the error bar would be in the same order of magnitude of the smallest physical scales η_K, then one could admit that the error is relatively marginal. This physical criteria might be a little naive, as η_K is susceptible to be quite small (10^{-3} - 10^{-2} when normalized), whereas the length of the error range interval can reach higher values (see figure 39 and 40). With regard to this observation, it is interesting to mention that numerical error compensation [41] is likely to occur in DNS, hence providing results that look actually better than what they really are. This may lead to bad resolution of the physical criteria, even though the obtained statistics are in good agreement with any values of reference.

References

1. F. Montomoli, Computational Fluid Dynamics, Lecture Notes, Imperial College London, 2016-2017.

2. M.P. Schultz, K.A. Flack, Reynolds-number scaling of turbulent channel flow, Physics of Fluids, 2013.

3. J. Kim, P. Moin and R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, Journal of fluid mechanics, 1987.

4. W.K. George, Lectures in Turbulence for the 21st Century, Lecture Notes, Imperial College London, 2016-2017.

5. P. Moin, K. Malesh, Direct numerical simulation: A tool in turbulence research, Annual Review of Fluid Mechanics, 1998.

6. J. Jimenez, R.D. Moser, What are we learning from simulating wall turbulence?, Philosophical Transactions of The Royal Society A Mathematical Physical and Engineering Sciences, 2007.

7. E. Rind, Instrument Error Analysis as It Applies to Wind-Tunnel Testing, NASA Archives, 1979.

8. P. Luchini, S. Russo, A fast algorithm for the estimation of statistical error in DNS (or experimental) time averages, APS Division of Fluid Dynamics, 2015.

9. C. J. Roy, Review of Discretization Error Estimators in Scientific Computing, AIAA, 2010.

10. P. Luchini, S. Russo, A fast algorithm for the estimation of statistical error in DNS (or experimental) time averages, APS Division of Fluid Dynamics, 2015.

11. S. Ghosal, An Analysis of Numerical Errors in Large-Eddy Simulations of Turbulence, Journal of Computational Physics, 1996.

12. T.A. Oliver, N. Malaya, R. Ulerich and R.D. Moser, Estimating uncertainties in statistics computed from direct numerical simulation, Physics of Fluids, 2014.

13. C. J. Roy, Review of code and solution verification procedures for computational simulation, Journal of Computational Physics, 2005.

14. W. L. Oberkampf, C. J. Roy, Verification and Validation in Scientific Computing, Cambridge University Press, 2010.

15. P. Billingsley, Probability and Measure, John Wiley & Sons, 1995.

16. J.L. Lumley, K. Takeuchi, Application of central-limit theorems to turbulence and higher-order spectra, Journal of Fluid Mechanics, 1976.

17. K.E. Trenberth, Some Effects of Finite Sample Size and Persistence on Meteorological Statistics, Monthly Weather Review, 1984.

18. D. B. Percival, Three Curious Properties of the Sample Variance and Autocovariance for Stationary Processes with Unknown Mean, 1993.

19. R. Ulerich, Autoregressive process modeling tools in header-only C++, [available online at $http://rhysu.github.io/ar/$], 2014.

20. P. M.T. Boersen, Automatic Autocorrelation and Spectral Analysis, Springer, 2006.

21. P. M.T. Boersen, Finite sample criteria for autoregressive order selection, IEEE Transactions on Signal Processing, 2000.

22. L. J. Faber, Commentary on the denominator recursion for Burg's block algorithm, Proceedings of the IEEE, 1986.

23. R. Ulerich, Autoregressive process modeling tools in header-only C++, rhoe.dat data set, [available online at $https://github.com/RhysU/ar$], 2014.

24. M. Steyvers, [$http://psiexp.ss.uci.edu/research/teachingP205C/205C.pdf$], Computational Statistics with Matlab, 2011.

25. M. E. Glickman and D. A. van Dyk, Basic Bayesian Methods, Bayesian inference, Methods in Molecular Biology, Humana Press Inc.

26. M. Dashti, A. M. Stuart, The Bayesian Approach to Inverse Problems, [$https://arxiv.org/pdf/1302.6989.pdf$], 2015.

27. L. Elie, B. Lapeyre, Introduction aux Methodes de Monte-Carlo, Note de cours, Ecole des Ponts et Chaussées, 2001.

28. G. Hamra, R. MacLehose and D. Richardson, Markov Chain Monte Carlo: an introduction for epidemiologists, International Journal of Epidemiology, 2013.

29. V. Mazet, Introduction aux Méthodes de Monte-Carlo par Méthode de Markov, Note de cours, Université Henri Poincare, 2003.

30. J. Goodman and J. Weare, Ensemble Samplers with Affine Invariance, Communications in Applied Mathematics and Computational Science, 2010.

31. D. Foreman-Mackey, D. W. Hogg, D. Lang, J. Goodman, emcee: The MCMC Hammer, [available online at $https://arxiv.org/pdf/1202.3665.pdf$], 2013.

32. D. Foreman-Mackey, API of the emcee package, [available online at $http://dan.iel.fm/emcee/current/$], 2013.

33. C. Hurlin, Maximum Likelihood Estimation, Lectures Notes in Advanced Econometrics, HEC Lausanne, 2013.

34. K. Sahlin, Estimating convergence of Markov chain Monte Carlo simulations, Master Thesis in Mathematical Statistics, 2011.

35. T.A. Oliver, N. Malaya, R. Ulerich and R.D. Moser, Data from the paper "Estimating uncertainties in statistics computed from direct numerical simulation, [available online at $http://turbulence.ices.utexas.edu/bayes.html$], 2014.

36. S. Laizet and al., Incompact3D Website, [available online at $http://www.incompact3d.com$], 2017.

37. S. Laizet and E. Lamballais, High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy, Journal of Computational Physics, 2009.

38. J. Kim, P. Moin and R. Moser, [$http://turbulence.ices.utexas.edu/MKM\_1999.html$], DNS Data for Turbulent Channel Flow 2010.

39. S. Hosder, B. Grossman, R. T. Haftka, W. H. Mason and L. T. Watson, Remarks in CDF Simulation Uncertainties, Computers and Fluids Journal, 2003.

40. I. Celik, J. Li, G. Hu and C. Shaffer, Limitations of Richardson Extrapolation and Some Possible Remedies, Journal of Fluids Engineering, 2005.

41. H. Schwenke, W.Knapp, H.Haitjema, A.Weckenmann, R.Schmitte and F.Delbressine, Geometric error measurement and compensation of machines - An update, 2008.

APPENDIX A : BURG RECURSION

Let $(x_n)_{n\in[0,N]}$ be a set of $N+1$ discrete values from which we want to fit a k order autoregressive process to approximate these original values by:

- a forward linear prediction: $y_n = -\sum_{i=1}^{k} a_i x_{n-i}$
- a backward linear prediction: $z_n = -\sum_{i=1}^{k} a_i x_{n+i}$

Burg method gives a way of finding the a_i, $i \in [1,k]$. The idea is to minimize the sum of squares between the original values and their estimates. We therefore introduce:

$$F_k = \sum_{n=k}^{N} (x_n - y_n)^2 = \sum_{n=k}^{N} \left(x_n - \left(-\sum_{i=1}^{k} a_i x_{n-i} \right) \right)^2 \tag{50}$$

$$B_k = \sum_{n=0}^{N-k} (x_n - z_n)^2 = \sum_{n=0}^{N-k} \left(x_n - \left(-\sum_{i=1}^{k} a_i x_{n+i} \right) \right)^2 \tag{51}$$

For the purpose of brevity, the coefficients f_k and b_k are introduced as follows:

$$f_k(n) = \sum_{i=0}^{n} a_i x_{n-i} \tag{52}$$

$$b_k(n) = \sum_{i=0}^{k} a_i x_{n+i} \tag{53}$$

Therefore, we have:

$$F_k = \sum_{n=k}^{N} (f_k(n))^2 \tag{54}$$

$$B_k = \sum_{n=0}^{N-k} (b_k(n))^2 \tag{55}$$

And then is introduced $A_{k+1} = (a'_n)_{n\in[1,k+1]}$ the vectors of the a_i at the iteration k as :

$$\forall n \in [1, k+1], \quad a'_n = a_n + \mu a_{k+1-n} \tag{56}$$

with $a_{k+1} = 0$. The coefficient μ is selected by minimizing $F_{k+1} + B_{k+1}$. It is therefore given by:

$$\mu = \frac{-2\sum_{n=0}^{N-k-1} f_k(n+k+1) b_k(n)}{D_k} \tag{57}$$

where D_k is given by:

$$D_k = F_k - f_k(k)^2 + B_k - b_k(N-k)^2 \tag{58}$$

And the iteration on D_k is as follows:

$$D_{k+1} = (1-\mu^2) D_k - f_{k+1}(k+1) - b_{k+1}(N-k-1)^2 \tag{59}$$

The vector μ is denoted as the reflexion coefficients and the a_i, $i \in [1,k]$ are the coefficients of the autoregressive model. Now, let p denote the effective order of the autoregressive process (it will be determined later with specific criteria, but for now let us assume it is known). Reworking the expression of the autoregressive model, one can obtain the following:

$$\forall k \in [0, p-1] \quad \rho_k + a_1 \rho_{k-1} + a_2 \rho_{k-2} + ... + a_p \rho_{k-p} = 0 \tag{60}$$

From equation (60), a linear system formed of Toeplitz matrix can be set. Its resolution gives the $\rho_k, \quad \forall k \in [0, p-1]$. Then, the other autocorrelation function values are given by:

$$\forall K \geq p \quad \rho_K = -a_1\rho_{K-1} - ... - a_p\rho_{K-p} \tag{61}$$

Appendix B: Model selection criteria and overfit term

Based on [21], the $overfit$ terms for the model selection criteria where implemented for three criteria : FSIC, FIC and CIC. The $overfit$ term is dependent on the Method used to compute the parameters and residual variance of the AR model. For Burg recursion, the empirical variance is given by:

$$v_{Burg}(N, i) = \frac{1}{N+1-i} \tag{62}$$

where N is the number of samples used to estimate the model parameters and i is the considered order at iteration i. And when the mean is substracted from the total sample (as it the case here, cf the variance estimation), $v_{Burg}(N, 0) = \frac{1}{N}$. The $overfit$ of the FIC criterion is given by:

$$overfit(FIC, v_{Burg}, N, p, \alpha) = \alpha(v(N, 0) - \Psi(N+1) + \Psi(N+1-p)) \tag{63}$$

where $\alpha = 3$ is recommended and Ψ refers to the digamma function. The overfit of the FSIC criterion is given by:

$$overfit(FSIC, v_{Burg}, N, p) = \frac{1+v(N,0)}{1-v(N,0)} \frac{\Gamma(-1-N+p)}{\Gamma(-1-N)} \frac{\Gamma(1-N)}{\Gamma(1-N+p)} - 1 \tag{64}$$

where Γ is the Gamma function. And finally, the $overfit$ of the CIC criterion is given by:

$$overfit(CIC, v_{Burg}, N, p) = \max(overfit(FSIC, v_{Burg}, N, p) \ , \ overfit(FIC, v_{Burg}, N, p, 3)) \tag{65}$$

All these criterion were implemented in MATLAB to find the optimal order p that minimizes the expression highlighted in the main section of the paper.

Appendix C: Change of variable for the calibration phase

In this appendix, we aim at highlighting the change of variable to set to obtain samples from the random variable $W = \langle q_h \rangle_N$ defined by:

$$W = \langle q_h \rangle_N = Q - C_0 h^p - e_{h,N} \tag{66}$$

provided that the PDFs of Q, C_0, p and $e_{h,N}$ are known. Let us remind that $e_{h,N} = \mathcal{N}(0, \sigma^2_{h,N})$ and that the mesh resolution h is fixed. The calibration model is applied to the finest mesh, so in our case $h = h_{finest}$. Let us consider the following set

$$A_w = \{(Q, C_0, p, e_{h,N}) \qquad Q - C_0 h^p - e_{h,N} \leq w\} \tag{67}$$

It is equivalent to write that:

$$A_w = \{(Q, C_0, p, e_{h,N}) \qquad Q \leq w + C_0 h^p + e_{h,N}\} \tag{68}$$

The distribution function of W is given by:

$$F_W(w) = \mathbb{P}(W \leq w) = \int_{q=-\infty}^{w+C_0h^p+e_{h,N}} \int_{p=0}^{+\infty} \int_{e_{h,N}=-\infty}^{+\infty} \int_{C=-\infty}^{+\infty} f_{C_0,p,Q,e_{h,N}}(C_0,p,q,e_{h,N}) \;\; dC_0dpdqde_{h,n} \tag{69}$$

where $f_{C_0,p,Q,e_{h,N}}$ is the joint distribution of the four random variables Q, C_0, p and $e_{h,N}$. The following change of variable is introduced, with $Y = (u,v,x,y)$ and $X = (Q,p,e_{h,N},C_0)$ and $Y = g(X)$ where g is a function $g: \;\; \mathbb{R}^4 \rightarrow \;\; \mathbb{R}^4$

$$(Q,p,e_{h,N},C_0) \mapsto (Q - Ch^p - e_{h,N}, p, e_{h,N}, C_0) \tag{70}$$

The inverse of g, g^{-1} is the function from $\mathbb{R}^4 \rightarrow \;\; \mathbb{R}^4$ defined as:

$$(u,v,x,y) \mapsto (u + yh^u + x, v, x, y) \tag{71}$$

The jacobian of g^{-1} is given by:

$$det(Jac(g^{-1}(u,v,x,y))) = \begin{vmatrix} 1 & yln(h)h^v & 1 & h^v \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = 1 \tag{72}$$

Therefore, one can obtain the following:

$$F_W(w) = \mathbb{P}(W \leq w) = \int_{u=-\infty}^{w} \int_{v=0}^{+\infty} \int_{x=-\infty}^{+\infty} \int_{y=-\infty}^{+\infty} f_{C_0,p,Q,e_{h,N}}(u + yh^v + x, v, x, y) \;\; dudvdxdy \tag{73}$$

Now, by differentiating the previous expression with respect to w, one obtains the distribution function:

$$f_W(w) = \int_{v=0}^{+\infty} \int_{x=-\infty}^{+\infty} \int_{y=-\infty}^{+\infty} f_{Q,C,p,e_{h,N}}(w + yh^v + x, v, x, y) \;\; dvdxdy \tag{74}$$

Or also:

$$f_W(w) = \int_{p=0}^{+\infty} \int_{e_{h,N}=-\infty}^{+\infty} \int_{C=-\infty}^{+\infty} f_{Q,C,p,e_{h,N}}(w + Ch^p + e_{h,N}, p, e_{h,N}, C_0) \;\; dpde_{h,N}dC_0 \tag{75}$$

If one assumes that $e_{h,N}$ is independent of $Q = E[q], C_0$ and p, which is likely to be the case as the discretization and sampling uncertainty are a priori not linked, the following can be obtained:

$$f_{Q,C_0,p,e_{h,N}}(w + Ch^p + e_{h,N}, p, e_{h,N}, C_0) = f_{Q,C_0,p}(w + Ch^p + e_{h,N}, p, C) \times f_{e_{h,N}}(e_{h,N}) \tag{76}$$

And according to the fact that $e_{h,N} \sim \mathcal{N}(0, \hat{\sigma_N}^2)$, one has that:

$$f_{e_{h,N}}(x) = \frac{1}{\sqrt{(2\pi)}} \frac{1}{\hat{\sigma}_N} \exp\left(-\frac{1}{2}\frac{x^2}{\hat{\sigma}_N^2}\right) \tag{77}$$

Therefore, we need to sample from the density of equation (76) which is feasible with the **emcee** package (see section III.IV.), given that our model provides an analytical expression for $f_{Q,C_0,p}$ (see equation (25)).

Appendix D : Change of variable for the discretization error formula

In this section, we assume that C_0, p and $Q = E[q]$ have the joint density $f_{C_0,p,Q}$. Our aim is to find the PDF of $Z = \frac{C_0h^p}{q_{obs}}$, where h and q_{obs} are two constants. Let us consider the following set:

$$A_z = \{(C_0,p) \qquad \frac{Ch^p}{q_{obs}} \leq z\} = \{(C_0,p) \qquad C \leq \frac{zq_{obs}}{h^p}\} \tag{78}$$

Therefore, the distribution function of the random variable Z is given by:

$$F_Z(z) = \mathbb{P}(Z \leq z) \int_{C_0=-\infty}^{\frac{zq_{obs}}{h^p}} \int_{p=0}^{+\infty} f_{C_0,p}(C_0, p) \;\; dC_0 dp \tag{79}$$

The following change of variable is introduced, with $Y = (u, v)$ and $X = (C_0, p)$ and $Y = g(X)$ where g is a function $g : \mathbb{R}^2 \to \mathbb{R}^2$

$$(C_0, p) \mapsto \left(\frac{C_0 h^p}{q_{obs}}, p \right) \tag{80}$$

The inverse of g, g^{-1} is the function from $\mathbb{R}^2 \to \mathbb{R}^2$ defined as:

$$(u, v) \mapsto \left(\frac{u q_{obs}}{h^v}, v \right) \tag{81}$$

The jacobian of g^{-1} is given by:

$$det(Jac(g^{-1}(u, v))) = \begin{vmatrix} \frac{q_{obs}}{h^v} & \frac{-u q_{obs} ln(h)}{h^v} \\ 0 & 1 \end{vmatrix} = \frac{q_{obs}}{h^v} \tag{82}$$

Therefore, the change of variable gives the following:

$$F_Z(z) = \int_{u=-\infty}^{z} \int_{v=0}^{+\infty} f_{C_0,p} \left(\frac{u q_{obs}}{h^v}, v \right) \mid \frac{q_{obs}}{h^v} \mid \;\; du dv \tag{83}$$

If one differentiate the previous expression with respect to z, the density of Z is obtained:

$$f_Z(z) = \int_{v=0}^{+\infty} f_{C_0,p} \left(\frac{z q_{obs}}{h^v}, v \right) \frac{\mid q_{obs} \mid}{h^v} \;\; dv = \int_{p=0}^{+\infty} f_{C_0,p} \left(\frac{z q_{obs}}{h^p}, p \right) \frac{\mid q_{obs} \mid}{h^p} \;\; dp \tag{84}$$

Finally, by using the definition of the marginal probabilities, one obtains the finale expression for the density of Z:

$$f_Z(z) = \int_{p=0}^{+\infty} \int_{Q=-\infty}^{+\infty} f_{C_0,p,Q} \left(\frac{z q_{obs}}{h^p}, p, q \right) \frac{\mid q_{obs} \mid}{h^p} dp dq \tag{85}$$

Then it is straightforward to estimate the PDF of the normalized discretization error since one just needs to apply MCMC algorithm to the posterior defined by $h_{v,q_{obs}} : \mathbb{R}^3 \to \mathbb{R}$

$$(z, p, q) \mapsto f_{C_0,p,Q} \left(\frac{z q_{obs}}{h^p}, p, q \right) \frac{\mid q_{obs} \mid}{h^p} \tag{86}$$

and then integrate over p and Q, which can easily be done with the PYTHON package **corner** for example.

APPENDIX E: DISCRETIZATION ERROR AND SAMPLING UNCERTAINTY FOR $\langle u'u' \rangle$ WITH DATA FROM THREE MESHES

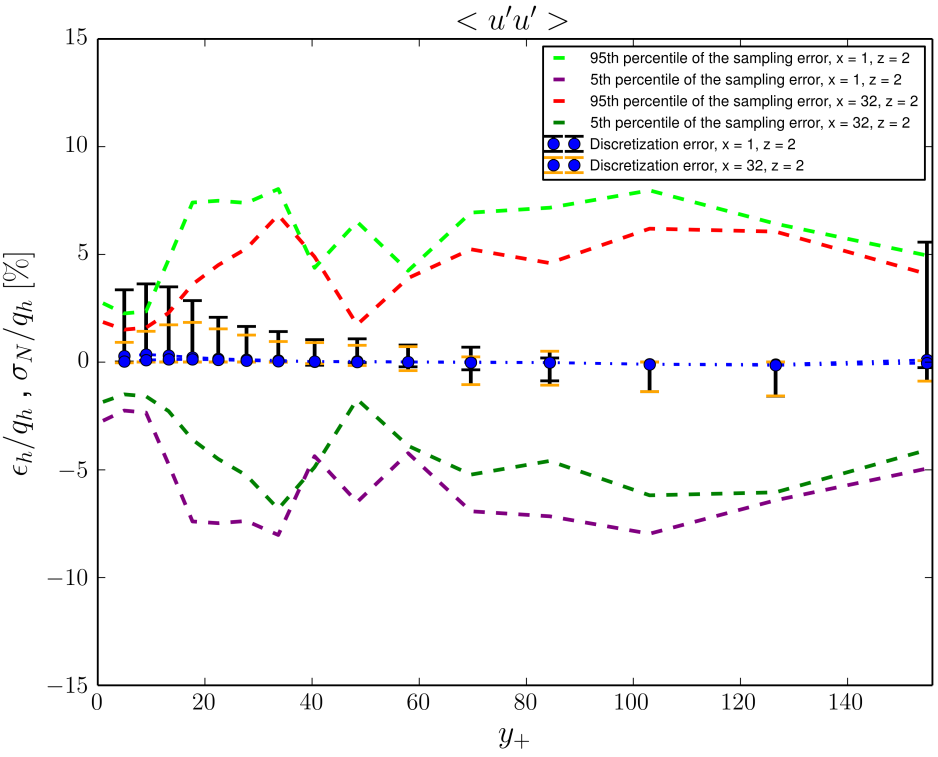


Figure 41: Discretization and sampling uncertainty for $\langle u'u' \rangle$ with data from the coarsest, nominal and finest meshes and $(N_x, N_z) = (32, 1)$.

As highlighted in figure 41, the discretization error bars are still very dependent on the number of points used to compute the statistics (especially for $0 < y^+ < 40$), even when data from three meshes are used.