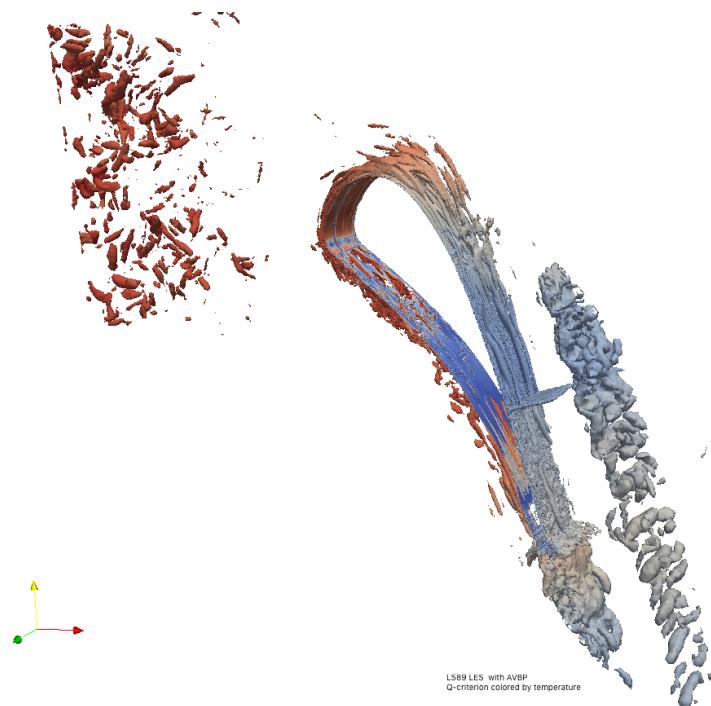


UNCERTAINTY QUANTIFICATION APPLIED TO TURBINE DESIGN

Advanced Mastère Thesis - MS SPA



Mentor:

Mrs. JEAN-CHRISTOPHE JOUHAUD,
LAURENT GICQUEL,
and FLORENT DUCHAINE.

School Representative:
M. GUILLAUME DUFOUR.

Trainee:
M. PAMPHILE ROY.

Dates of the internship:
from March 29th, 2016 to
October 14th, 2016

Acknowledgment

I have committed myself to this project. However, its completion would not have been possible without the help and support of all CFD's team. Furthermore, I would like to extend my thanks to all CERFACS employees for their welcoming.

Thank you to Mr. **THIERRY POINSOT** —head of CFD team— for accepting me as a trainee in his department.

Thank you to Mr. **JEAN-CHRISTOPHE JOUHAUD** —member of CFD team— for accepting me in his team and for his supervision during the whole internship.

Thank you to Mr. **BERTRAND IOOSS** —EDF R&T— for his help and advice.

Thank you to Mrs. **LAURENT GICQUEL** and **FLORENT DUCHAINE** —members of CFD team— who helped me with the simulation case.

Thank you to Mr. **Luis SEQUI** —PhD student— who provided me explanations and details on the case I had to use.

Thank you to all PhD students and to my co-worker **GHITA JETTOU** for their time, help and kindness.

Finally, I would like to thank my live-in companion and my family for their support through my studies and life.

Sommaire

Introduction	1
I Presentation of the Internship	2
1 Presentation of the CERFACS	3
II The Internship	4
2 Uncertainty quantification	5
2.1 Introduction	5
2.2 Probabilistic Framework	6
2.3 Sampling the Space of Parameters	7
2.4 Sensitivity Analysis	8
2.4.1 Morris's Method	8
2.4.2 Sobol' Indices	9
2.4.3 FAST method	10
2.5 Surrogate model	10
2.5.1 How to model physics	10
2.5.2 POD	11
2.5.3 Kriging	12
2.5.4 Radial Basis Functions (RBF)	14
2.5.5 Gaussian Process Regression	14
2.5.6 Measure the error	15
3 JPOD for Jack-POD	17
3.1 Presentation	17
3.2 External frameworks	17
3.2.1 scikit-learn	17
3.2.2 OpenTURNS	17
3.3 Add Uncertainty Quantification Capabilities	17
3.3.1 Sensitivity Analysis	17
3.3.2 Uncertainty Propagation	18
3.4 Re-Sampling Strategies	18
3.5 Miscellaneous	19
3.6 Validation cases	20
3.6.1 Analytical function	20
3.6.2 Channel Flow	23
4 UQ analysis applied to a LES case	26
4.1 Experimental case	26
4.2 LES simulation of the LS89	27
4.2.1 Simulation with AVBP	27
4.2.2 Mesh convergence	29
4.2.3 Some results	31
4.3 Space of Parameters	33
4.3.1 Passot-Pouquet turbulence spectrum	35
4.4 Setting up JPOD	35
4.5 Response Surface	36

4.6	UQ analysis	38
4.6.1	Uncertainty Propagation	38
4.6.2	Sensitivity Analysis	38
4.7	Conclusions and recommendations	39

III Conclusion	40
-----------------------	-----------

Introduction

Background

The aim of this report is to present the internship I have done at CERFACS in Toulouse, from March 29th to October 14th 2016. This experience was to put a period on my studies by demonstrating my abilities to work as a researcher. For six months, I have been working within the Computational Fluid Dynamic (CFD) team directed by Thierry POINSOT. I have been mentored by Jean-Christophe JOUHAUD and I had regular focal points with other seniors at CERFACS.

This report was requested on September 19th 2016 by ISAE's administration.

Objectives

Uncertainty Quantification is an ongoing topic of interest in the CFD field. It allows to determine which variables of the models are of interests, how the system will react after a perturbation or even after a geometrical modification. However, this statistical tool requires a lot of simulations.

The objective was to study the Uncertainty Quantification applied to costly CPU time numerical simulations. After reviewing the literature, the aim was to augment an existing tool with uncertainty quantification capabilities. From known analytical functions, the routine was to be applied in order to find concordant results. After that, it has been validated on a real complex application.

Step aside, as I had previously been working in the maintenance area, after that in a research department of a university and finally had a glimpse of an engineering department of a major company, I wanted to pursue with a reknown research laboratory. Thus, I would have had a fairly wide vision of the working area of a researcher.

Scope

First, I will present the enterprise, and then, I will expose the work I have done within the department during these six months.

Presentation of the Internship

1 | Presentation of the CERFACS

The European Centre for Research and Advanced Training in Scientific Computing (CERFACS) is a research civil company dedicated to the modelization and the numerical simulation. The company's shareholders and partners are Airbus Group, Cnes, EDF, Météo France, Onera, Safran and Total. Together, they are implied in major international projects and big scientific and technical challenges.

The CERFACS aims at developing and spreading its research and knowledge as to improve advanced computing methods. It relies on a highly qualified and renowned base of interdisciplinary researchers and engineers. As for the resources, it can rely on High Performance Computing (HPC) computers that can deliver a peak power of ~ 300 Tflop/s. Moreover, it benefits from external HPC resources such as GENCI's or PRACE's computers.



Figure 1.1 – Nemo Cluster / Peak Performance: 242 Tflop/s

The Internship

2 | Uncertainty quantification

2.1 Introduction

Real physical phenomena are complex and hardly entirely understood. As our understanding of the physics increases, so does our ability to describe it. The rise of the computing power helping, simulating physics is at reach.

But still, computing complex physics comes with lots of assumptions causing modelling errors at every step: *CAD* definition, modelling, discretizations, etc. The output variability of a simulation can be confusing regarding its variability, its uncertainties. Thus we have to determine the source of these uncertainties. Does it come from our model or from the physics? Uncertainties can be defined as:

Aleatoric: intrinsic to our system,

Epistemic: due to our lack of knowledge. We can reduce it by improving our model or adding simulations,

Numerical error: due to the construction of our numerical scheme.

As, in an experiment where a confidence interval associated with the data should always be plotted, we aim at describing the variability of the response surface. It is characterized using a statistical point of view. We are interested in finding the probability distribution of the physic. This way we can compute the mean, the standard deviation, etc. But when post-processing a simulation, we can't be sure of its convergence and thus of the non-physical uncertainty. Ideally, we would like to get all three errors we described. Using adjoint solutions for example, we can estimate the numerical error. Dealing with epistemic uncertainty, we can reduce our model as we are observing a physical problem. For example, when looking at the angle of attack of a cruising plane, we know that we don't have to put weight on high angles of attack as it should be below 5 degrees. We know that if we have high angles of attack, we could trigger a different physic, such as a stall, thus have much larger errors because of the inability of our model to assess these phenomena.

Ideally, our model should only contain aleatoric uncertainties as it is defined solely by the physics. The main objective of uncertainty studies is to be able to quantify this uncertainty in order to reduce the design margin or even perform complex system optimization. Thus, the first thing to come up with is to determine the type of uncertainty quantification (UQ) study to conduct:

- *Uncertainty Propagation*,
Propagate an initial perturbation within the system and observe its outputs.
- *Sensitivity Analysis*,
Rank the input parameters regarding their impact on the output.
- *Risk assessment*.
Observe the probability to exceed a threshold.

In any case, from perturbed input we are looking at the output response of a Quantity of Interest (QoI). This response of the model is called a response surface.

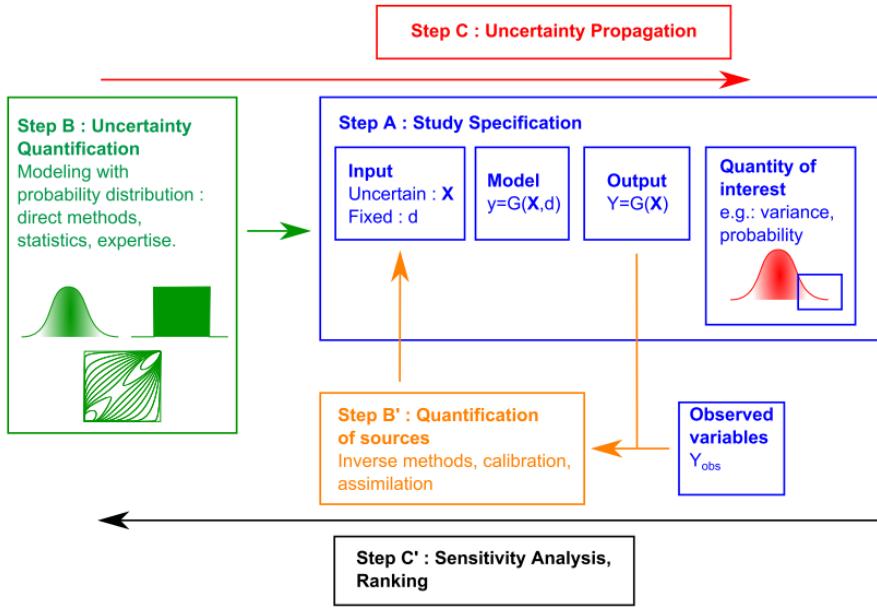


Figure 2.1 – UQ framework[3].

Furthermore, the solution is sensitive to the UQ method used. With the same set of data, we can use different methods in order to assess that. There is no ultimate solution. The fig. 2.1 is a visualization of the possible workflows.

The *Monte Carlo* methods [11] represent the golden standard for such analysis. However, as these methods requires a huge amount of simulations, they are usually used with simple models, which is to say with low CPU costs. Today, all realistic engineering applications are made using approximant methods. Interpolant methods based on polynomial interpolations are only used, up to now, for simple academic cases.

2.2 Probabilistic Framework

As stated previously, the objective is to observe, understand, the evolution of a response surface to uncertain inputs. This variability is characterized using probability density functions (PDF) —see fig. 2.2. A PDF gives information about the probability of occurrence $P(X)$ of a phenomenon:

$$f(x) \geq 0, \quad (2.1)$$

$$f(x) = P(X = x). \quad (2.2)$$

Also, the Cumulative Density Function (CDF) is a monotone increasing function defined as:

$$F(x) = P(a < X < b) = \int_a^b f(x)dx, \quad (2.3)$$

$$\int_{-\infty}^{+\infty} f(x)dx = 1. \quad (2.4)$$

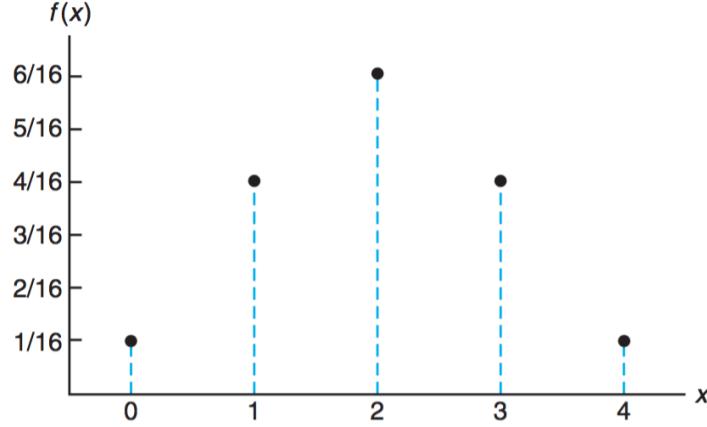


Figure 2.2 – Probability Density Function for a discrete distribution.

The shape of the PDF can be described using the statistical moments:

$$C_{\mu_1} = \frac{\mu_1}{\mu_0}, \quad (2.5)$$

$$C_{\mu_2} = \frac{\sqrt{\mu_2}}{\mu_1}, \quad (2.6)$$

$$C_{\mu_3} = \frac{\mu_3}{\mu_2^{\frac{3}{2}}}, \quad (2.7)$$

$$C_{\mu_4} = \frac{\mu_4}{\mu_2^2}. \quad (2.8)$$

Using the following notations:

$$E[X] = \int_{-\infty}^{+\infty} x f(x) dx, \quad (2.9)$$

$$\mu'_n = E[X^n], \quad (2.10)$$

$$\mu_n = E[(X - \mu'_1)^n], \quad (2.11)$$

$$\mu'_1 = \mu_1 = E[X]. \quad (2.12)$$

The first and second order moment are the most commonly used parameters. They are respectively known as the mean E or μ and the standard deviation σ . The skewness (third order moment eq. (2.7)) defines the position of the maximum relative to the mean. The more the skewness is, the more the PDF is shifted to the left. The flatness (fourth order moment eq. (2.8)) defines how flat the PDF is. The more the flatness is, the less the PDF is flat. If we only have the moments we can retrieve the PDF using Pearson's correspondence table [32].

Third and fourth orders need a lot of iterations in order for the error to be contained. It is why we restrict most of the time the analysis of the first and second order.

2.3 Sampling the Space of Parameters

Whatever method is used, the first step consists in defining how we are going to modify input variables to retrieve the evolution of the response surface. This is called a Design Of Experiments (DOE) as defined by Sacks et al. [28] —see fig. 2.3.

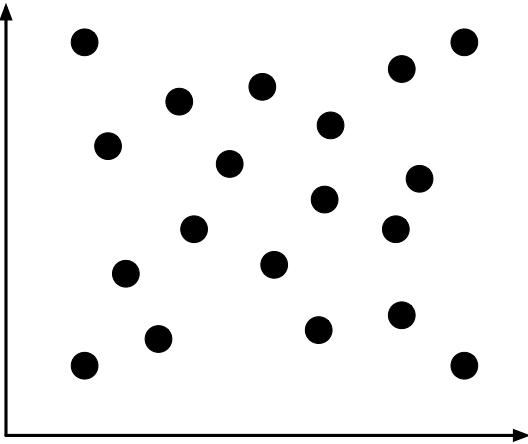


Figure 2.3 – Space of parameters with two input variables.

The quality of the DOE is paramount as it determines the physics that will be observed. If the space is not filled properly, homogeneously, we can bias our analysis and retrieve only a particular behaviour of the physic. This concept has been extensively been used in experiments, especially the one-at-a-time design, which consists of only changing only one parameter at a time. Doing so, the space is not filled properly and only simple behaviours can be recovered. In order to assess the quality of the sampling, the discrepancy is usually used. It is an indicator of the distance between the points within the space of parameters. The lower the discrepancy is, the better the design is.

As stated before, the golden standard would be to perform a *Monte Carlo* sampling but it would require a huge sampling which is unfeasible with costly numerical simulations. Therefore are found random (or quasi-random) sampling methods. Low discrepancy sequence has been designed to overcome this issue. These designs are built upon a pattern, a sequence, depending on factors such as prime numbers. This allows a fast generation of sampling space with good properties. A well-known method is the Latin Hypercube Sampling (LHS). The idea behind is to discretize the space to get a regular grid and sample randomly a point per zone.

In Damblin et al. [9] a comprehensive analysis of most common DOE is found. In the end, the *Sobol* or *Halton* DOE are sufficient when dealing with a small number of parameters (<5). With an increasing number of parameters, patterns start to appear.

2.4 Sensitivity Analysis

There are several methods in order to estimate the contribution of the parameters of the output. In order not to make extensive analysis which is not required, the first thing to do would be to check if the parameters are strongly correlated or not. If they are correlated, we can remove some parameters reducing the dimension of the problem.

Typically, sensitivity methods are based on the analysis of the variance. This allows to see the contribution of the parameters to the problem.

2.4.1 Morris's Method

The Morris method is a screening method [23]. If we have p parameters, we consider $p - 1$ elementary effect:

$$d_{X_j} = \frac{f(p_i) - f(p_{i-1})}{\Delta}, \quad (2.13)$$

with j the parameter and, i the repetition. We repeat the process n times and get respectively the mean of the effects and its spread:

Sensitivity:

$$\mu_j^* = E[| d_{x_j} |], \quad (2.14)$$

Interactions:

$$\sigma_j = \sigma(d_{x_j}). \quad (2.15)$$

We define a number of trajectory to follow. A trajectory consists of the successive function evaluation having modified one parameter at a time. This way we can fill the space and recover the elementary effects. Figure 2.4 represents a two input parameters case where five trajectories have been performed.

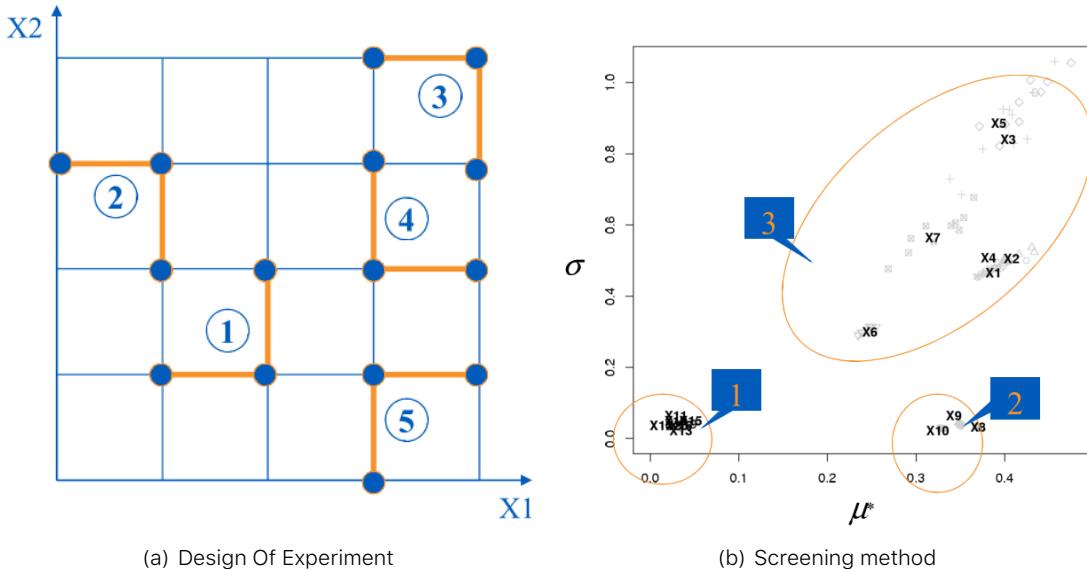


Figure 2.4 – Morris's method [14].

In this case, the first group of parameters are found to be of smaller interest as of small influence with the QoI. The second group shows a small coupling effect but a large sensitivity effect and finally the third group shows parameters which are highly correlated and with a high impact on the QoI.

This method can be used as a preliminary tool in order to reduce the number of uncertain parameters. Indeed, it offers a ranking of the parameters. After that, a further analysis of the physic can be drawn using a reduced space of parameters. Indeed, this analysis doesn't provide a ranking of the parameters' influence.

2.4.2 Sobol' Indices

Sobol' et al. in [31] introduced the Sobol' method. As opposed to Morris's method, it gives not only a ranking but also the quantification of the importance factor using the variance. This method only make the hypothesis of the independence of the input variables. For correlated input, the ANCOVA method can be used.

We use a functional decomposition of the variance of our function:

$$Var(y) = \sum_i^p V_i(y) + \sum_{i < j}^p V_{ij} + \dots + V_{1,2,\dots,p}, \quad (2.16)$$

$$V_i(y) = Var[E(y|x_i)] = \int_{-\infty}^{+\infty} x_i f(y) dx_i, \quad (2.17)$$

$$V_{ij} = Var[E(y|x_i x_j)] - V_i - V_j. \quad (2.18)$$

This way we can get Sobol' indices:

$$S_i = \frac{Var[E(y|x_i)]}{Var[y]} \quad S_{ij} = \frac{Var[E(y|x_i x_j)] - V_i - V_j}{Var[y]}. \quad (2.19)$$

Total indices are given by:

$$S_{T_i} = S_i + \sum_j S_{ij} + \sum_{j,k} S_{ijk} + \dots \simeq 1 - S_i. \quad (2.20)$$

Considering a toy function as an example:

$$y = f(x_1, x_2) = 4x_1^2 + 3x_2, \quad (2.21)$$

with x_1, x_2 using a uniform probability between $[-1/2, 1/2]$.

$$E(y) = \frac{1}{\frac{1}{2} + \frac{1}{2}} \int_{[-1/2, 1/2]} f(x_1, x_2) dx_1 dx_2 = \frac{1}{3}, \quad (2.22)$$

$$E(y|x_1) = 4x_1^2, \quad (2.23)$$

$$E(y|x_2) = 3x_2, \quad (2.24)$$

$$E(y|x_1, x_2) = 0. \quad (2.25)$$

Thus we have:

$$Var[E(y|x_1)] = \int_{-1/2}^{1/2} (4x_1^2)^2 dx_1 - \left(\frac{1}{3}\right)^2 = 0.08, \quad (2.26)$$

$$Var[E(y|x_2)] = \int_{-1/2}^{1/2} 9x_2^4 dx_2 - \frac{1}{9} = 0.08, \quad (2.27)$$

$$Var[y] = Var[E(y|x_1)] + Var[E(y|x_2)] = 0.838, \quad (2.28)$$

$$S_1 = \frac{Var[E(y|x_1)]}{Var[y]} = 0.106, \quad (2.29)$$

$$S_2 = \frac{Var[E(y|x_2)]}{Var[y]} = 0.894. \quad (2.30)$$

The conclusion is that x_1 contribute to up to 10.6% of the total output's variance and x_2 to up to 89.4%.

2.4.3 FAST method

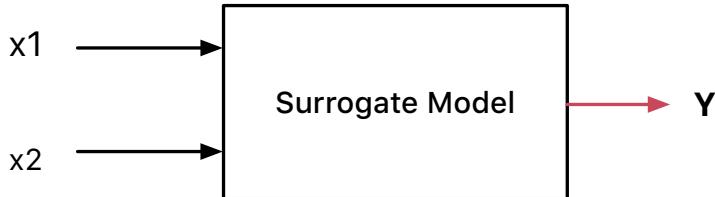
As a reference, Saltelli et al. [29] put forward a faster and more robust method, the *Fourier Amplitude Sensitivity Testing* (FAST) method. The idea behind is to use a Fourier decomposition on the model response.

2.5 Surrogate model

2.5.1 How to model physics

The use of Computational Fluid Dynamics (CFD) has proven to be reliable, faster and cheaper than experimental campaigns—in an industrial context. However, sensitivity analysis needs a large amount of simulation which is not feasible when using complex codes that are time and resources consuming. This is even more true in *LES* context as we are trying to have a representative simulation. The only solution to overcome this issue is to construct a model that would estimate a given QoI in a given range. This model requires a realistic amount of evaluation of the detail code. The general procedure to construct it consists of:

1. Generating a sample space,
Generate a set of data from which to run the code. A solution is called a snapshot.
2. Learning the link between the input and the output,
From the previously generated set of data, we can compute a model also called a response surface. A comprehensive set of method is presented by Forrester et al. [12].
3. Predicting a solution from a new set of input data.
The model can finally be used to estimate a new snapshot from a new set of input data.



- The model cannot be used for extrapolation. Indeed, it has been constructed using a sampling of the space of parameters. If we want to predict a point which is not contained within this space, the error is not contained as the point is not balanced by points surrounding it. As a famous catastrophe, an extrapolation of the physical properties of an o-ring of the *Challenger* space shuttle lead to an explosion during lift-off [10].

Once this model has been constructed, using *Monte Carlo* sampling we can compute Sobol' indices, etc. Indeed, this model is said to be costless to evaluate, this is why the use of the *Monte Carlo* sampling is feasible. To increase convergence, we can still use the same methods as for the DOE.

Both *Proper Orthogonal Decomposition* (POD) and *Kriging* are techniques that can interpolate data using snapshots. The main difference being that *POD* compresses the data it uses to use only the relevant modes whereas *Kriging* method doesn't reduce the size of the used snapshots. On the other hand, *POD* cannot reconstruct data from a domain missing ones [13].

2.5.2 POD

The *Proper Orthogonal Decomposition* (POD) is a technique used to decompose a matrix and characterize it by its principal components which are called modes [1]. To approximate a function $z(x, t)$, only a finite sum of terms is required:

$$z(x, t) \simeq \sum_{k=1}^m a_k(t) \phi_k(x). \quad (2.31)$$

The function $\phi_k(x)$ have an infinite representation. It can be chosen as a Fourier series or Chebyshev polynomials, etc. For a chosen basis of function, a set of unique time-functions $a_k(t)$ arise. In case of the POD, the basis function are orthonormal. Meaning that:

$$\int_x \phi_{k_1} \phi_{k_2} dx = \begin{cases} 1 & \text{if } k_1 = k_2 \\ 0 & \text{if } k_1 \neq k_2 \end{cases}, \quad (2.32)$$

$$a_k(t) = \int_x z(x, t) \phi_k(x) dx. \quad (2.33)$$

The principle of the POD is to choose $\phi_k(x)$ such that the approximation of $z(x, t)$ is the best in a least squares sense. These orthonormal functions are called the *proper orthogonal nodes* of the function.

When dealing with CFD simulations, the size of the domain m is usually smaller than the number of measures, snapshots, n . Hence, from the existing decomposition methods, the *Singular Value Decomposition* (SVD) is used. It is the snapshots methods [8].

The Singular Value Decomposition (SVD) is a factorization operation of a matrix expressed as:

$$A = U\Sigma V^T, \quad (2.34)$$

with V diagonalizes $A^T A$, U diagonalizes AA^T and Σ is the singular value matrix which diagonal is composed by the singular values of A . Knowing that a singular value is the square root of an eigen value. u_i and v_i are eigen vectors of respectively U and V which form an orthonormal basis. Thus, the initial matrix can be rewritten:

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T, \quad (2.35)$$

r being the rank of the matrix. If taken $k < r$, an approximation of the initial matrix can be constructed. This allows to compress the data as only an extract of u and v need to be stored.

2.5.3 Kriging

Kriging is a geostatistical interpolation method that use not only the distance between the neighbouring points but also the relationships among these points, the autocorrelation. The method has been created by D.G. Krige [17] and has been formalized by G. Matheron [21].

In order to predict an unmeasured location \hat{Y} , interpolation methods use the surrounding measured values Y_i and weight them:

$$\hat{Y} = \sum_{i=1}^N \lambda_i Y_i. \quad (2.36)$$

The advantage of this method is that the interpolation is exact at the sampled points and that it gives an estimation of the prediction error. Ordinary *Kriging* consists in the Best Linear Unbiased Predictor (BLUP) [27]:

Best: it minimizes the variance of the predicted error $Var(\hat{Y} - Y)$,

Linear: a linear combination of the data as in eq. (2.36),

Unbiased: it minimizes the mean square error $E[\hat{Y} - Y]^2$ thus $\sum_{i=1}^N \lambda_i(x) = 1$,

Predictor: it is an estimator of random effects.

λ_i are calculated using the spatial autocorrelation of the data, it is a variography analysis. Plots can be constructed using semivariance, covariance or correlation. An empirical variogram plot allows to see the values that should be alike because they are close to each other [4]. The empirical semivariogram is given by:

$$\gamma(h) = \frac{1}{2} \times \frac{1}{n} \sum_{i=1}^N (Y_i - Y_{i+h})^2. \quad (2.37)$$

A fitting model is then applied to this semivariogram. Hence, the variability of the model is inferior to data's. Kriging smooths the gradients. The exponential model is written as:

$$\gamma(h) = C(0) + C \left(1 - \exp \left(-\frac{h}{r} \right) \right), \quad (2.38)$$

with C the correlation matrix and the parameter r is optimized using the sample points.

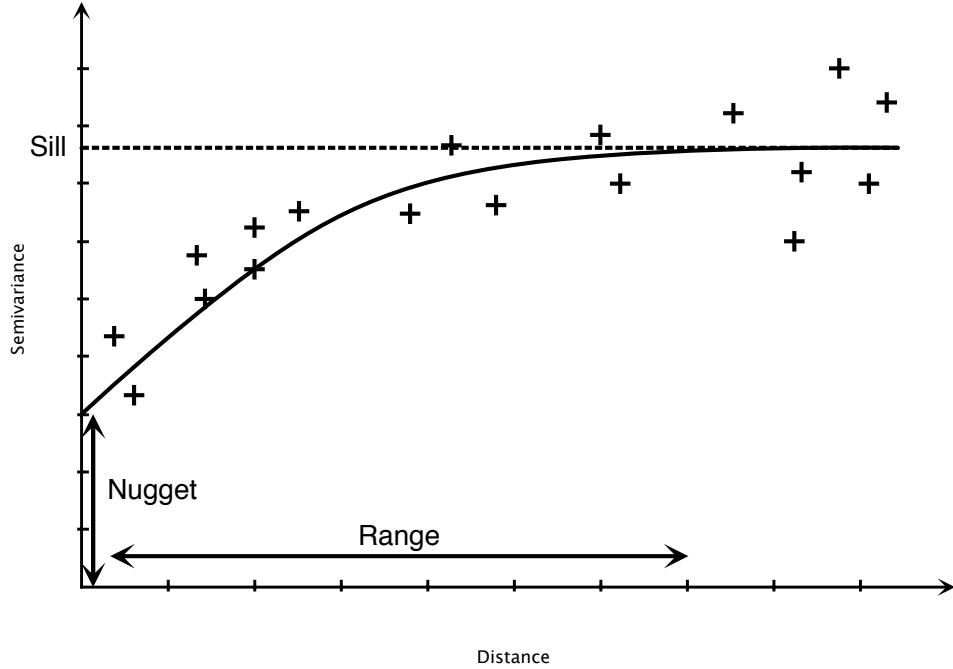


Figure 2.5 – Exponential model of a semivariogram.

A model is described using:

Sill: it corresponds to the maximum of γ . It defines the end of the range.

Range: it is the zone of correlation. If the distance is superior to the range, there is no correlation, whereas if the distance is inferior to it, the sample locations are autocorrelated.

Nugget: if the distance between the points is null, γ should be null. However, measurement errors are inherent and cause a nugget effect. It is the y-intercept of the model.

Once the model is computed, the weights are determined to use the *MSE* condition and gives:

$$\lambda_i = K^{-1}k, \quad (2.39)$$

K being the covariance matrix $K_{i,j} = C(Y_i - Y_j)$ and k being the covariance vector $k_i = C(Y_i - Y)$ with the covariance $C(h) = C(0) - \gamma(h) = Sill - \gamma(h)$.

$$\begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1j} \\ \vdots & \ddots & \vdots \\ \gamma_{i1} & \cdots & \gamma_{nn} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} \gamma_{1X} \\ \vdots \\ \gamma_{nX} \end{pmatrix}. \quad (2.40)$$

Furthermore we can express the field Y as $\hat{Y} = R(S) + m(S)$ which is the residual and the trend components [5]. Depending on the treatment of the trend, there are several *Kriging* techniques —ordinary Kriging being the most used—:

Simple: the variable is stationary, the mean is known,

Ordinary: the variable is stationary, the mean is unknown,

Universal: the variable is non-stationary, there is a tendency.

Ordinary Kriging is the most used method. In this case, the covariance matrix is augmented:

$$\begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1j} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \gamma_{i1} & \cdots & \gamma_{nn} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ -\mu \end{pmatrix} = \begin{pmatrix} \gamma_{1X} \\ \vdots \\ \gamma_{nX} \\ 1 \end{pmatrix}. \quad (2.41)$$

Once the weights are computed, its dot product with the residual $R_i = Y_i - m$ at the known points gives the residual $R(S)$. Thus we have an estimation of \hat{Y} . Finally, the error is estimated by the second order moment:

$$\sigma^2 = \sum_{i=1}^N \lambda_i \gamma_{iX} - \mu. \quad (2.42)$$

Some care has to be taken with this estimation of the variance. Being a good indicator of the correctness of the estimation, this is only an estimation of the error based upon all surrounding points.

2.5.4 Radial Basis Functions (RBF)

The idea is to recover the solution using radial basis function:

$$\tilde{f}(x) = \sum_{i=1}^{n_p} \sigma_i \psi(\|x - x^i\|), \quad (2.43)$$

ψ is a basis function and σ_i are the modes of this function. Then we can reconstruct the solution with:

$$[A] \{\sigma\} = \{F\}. \quad (2.44)$$

A and σ respectively represents the basis functions and the mode matrices. The method is thus a simple interpolation using a sum of basic functions. The basis function can be defined as linear, cubic, Gaussian, etc. The problem is that in its initial form it cannot interpolate linear functions correctly. The extended RBF method allows to address such problems.

2.5.5 Gaussian Process Regression

There are two approaches when dealing with regression problems. In simple cases, we can use simple functions in order to approximate the output set of data. On the other hand, when dealing with complex multidimensional problems with strong non-linearity, there are infinite possibilities of functions to consider. This is where the Gaussian process comes in.

As stated by Rasmussen et al. in [26], a process is a generalization of a probability distribution of functions. When dealing with *Gaussian* processes, they can simply be fully defined using the mean and covariance of the functions:

$$f(x) \sim GP(m(x), k(x, x')), \quad (2.45)$$

$$m(x) = \mathbb{E}[f(x)], \quad (2.46)$$

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]. \quad (2.47)$$

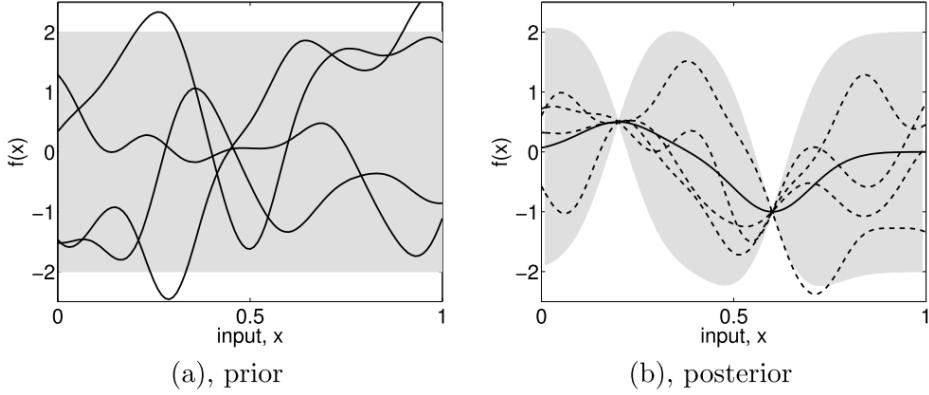


Figure 2.6 – (a) shows four samples from a prior distribution. (b) shows the situation after two observations have been made. [26].

Starting from a prior distribution of functions, it represents the belief we have on the problem. Without any assumption, the mean would be null. If we are now given a dataset $D = \{(x_1, y_1), (x_2, y_2)\}$, we only consider the function that actually pass through or close to these points, as in fig. 2.6. This is the learning phase. The more points are added, the more the model will fit the function. Indeed, as we add observations, the error is reduced at these points.

The nature of the covariance matrix is of great importance as it fixes the properties of the functions to consider for inference. This matrix is also called *kernel*. Many covariance functions exist and they can be combined to fit specific needs. The Gaussian hypothesis formulation is equivalent to the Kriging. A common choice is the squared exponential covariance kernel:

$$k(x, x') = \sqrt{\pi} l \sigma_p^2 \exp -\frac{(x - x')^2}{2(\sqrt{2}l)^2}, \quad (2.48)$$

with l the length scale, an hyperparameter, which depends on the magnitudes of the parameters. When dealing with a multidimensional case and non-homogeneous parameters, it is of prime importance to adimentionize everything as one input could bias the optimization of the hyperparameters.

Then the Gaussian process regression is written as a linear regression:

$$\hat{f}(x_*) = \sum_{i=1}^n \alpha_i k(x_i, x_*), \quad (2.49)$$

$$\alpha = (K + \sigma_n^2 I)^{-1} y. \quad (2.50)$$

The Kriging method is one of the most employed as of today. We can even enhance the result of the regression if we have access to the derivative (or even the hessian) of the function [12]. This could be even more challenging if we don't have an adjoint solver to compute it. Another method is to use a multi-fidelity metamodel in order to obtain an even better solution. This can be performed if we have two codes that compute the same thing or if we have two grids to run from.

2.5.6 Measure the error

In order to assess the quality of the model, the expected value is compared with the estimation. Depending on the metric used, significantly different results are observed. A common indicator is to compute the coefficient of determination Q_2 [19]:

$$Q_2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}. \quad (2.51)$$

When dealing with a non-analytical function, the expected values are not known. Mainly, there are two methods. First of all, a new sample can be evaluated and then compared to the prediction

of this sample using the model. Secondly, the quality can be estimated by *Leave-One-Out Cross Validation* (LOOCV). It consists of removing a snapshot from the database and estimate a new model on this reduced base. Then an evaluation of the value of the removed snapshot (\hat{Y}) is compared to the actual snapshot (Y).

3 | JPOD for Jack-*POD*

3.1 Presentation

JPOD is python application that allows to construct a surrogate model from a complex code [6]. This code can either be a simple *1D* function or a complex code like *AVBP*. It has been developed internally at CERFACS since 2007.

The strategy used by JPOD consists of:

1. Use POD reconstruction in order to compress data,
2. Use Kriging interpolation on POD's coefficients,
3. Interpolate missing data.

The purpose of the internship has been to augment JPOD with UQ capabilities in order to treat a LES case. Several open source codes have been used to that aim.

3.2 External frameworks

3.2.1 scikit-learn

scikit-learn is a Python module for machine learning [24]. It is used by JPOD to construct the surrogate model. More precisely, it is the *GaussianProcessRegressor* class —which implements the algorithm presented in section 2.5.5— that is being used.

3.2.2 OpenTURNS

Open source initiative to Treat Uncertainties, Risks'N Statistics (OpenTURNS) is a *C++* framework which objective is to conduct uncertainty quantification analysis [3]. It is the result of a partnership between *EDF*, *Airbus Group*, *IMACS* and *Phimeca*. The user interface consists of a python framework. In order to measure the importance of the simulation's parameters, it provides several tools. It is used by JPOD to generate the sampling space, compute Sobol' indices and generate samples from the surrogate.

3.3 Add Uncertainty Quantification Capabilities

3.3.1 Sensitivity Analysis

In JPOD, SA is done using Sobol' or FAST method. The implementation uses OpenTURNS sensitivity class. Class's outputs are the first, second and total order indices (no second order indices with FAST by design) giving two samples generated using the surrogate. Furthermore, if a vectorial output is given, indices will be computed all along the vector allowing spatial (or temporal) analysis which output is called a sensitivity map as in [20].

At the time of the implementation, aggregated indices were not available within the framework. It is a composite indicator which represents the global influence of a parameter upon the variability of the QoI over a domain of space or time. They have been implemented as described by Marrel et al. [18] as the mean of the indices weighted by the variance at each point or temporal step:

$$S_i = \frac{\sum_{l=1}^p Var[Y_l] S_i^l}{\sum_{l=1}^p Var[Y_l]} \quad (3.1)$$

As a complementary indicator, *block* indices are also computed. They consist of integrating the function over the space to have a scalar value to compute indices from.

3.3.2 Uncertainty Propagation

Aside from these indices, we want to compute the mean and the standard deviation of the output. Which is error propagation. It has been tested using a normal input on a simple *1D* function $f(X) = X$. The mean and the variance has been correctly retrieved. The tool has been set so the user can choose any PDF for the input parameters.

3.4 Re-Sampling Strategies

As stated in section 2.3, sampling the space of parameters is paramount as it is used to construct the model. Moreover, in case of a costly simulation environment, it constrains the number of simulation that can be performed. The strategy is then to explore most of the space with as little points as possible and then refine the exploration around interest zones. Using Scheidt's work in her thesis [30], several technic has been added:

- Mean Square Error (MSE),

As stated in section 2.5.5, one of the main advantages of Gaussian processes is to give an insight about the variance of the solution. The first method is thus to use this data and weight it with the modes of the POD:

$$\sum \sigma = \sum S_i^2 \times \sigma_i. \quad (3.2)$$

Global optimization on this indicator gives the new point to simulate.

- Leave-One-Out Cross Validation (LOOCV) and MSE,

A LOOCV is performed on the POD. This gives the point where the model is the more sensitive if deleted. The strategy is thus to add a new point around it. As in fig. 3.1, to do so, a hypercube bounded by the closest point is created. Within it, a global optimization over the MSE is conducted giving the new point.

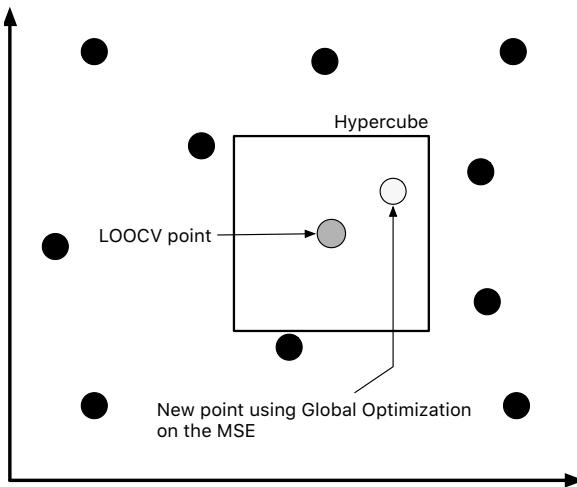


Figure 3.1 – Sketch of a Hypercube.

- LOOCV-Sobol',

Using the same point as with the LOOCV-MSE method, the hypercube around the point is here truncated using prior information about Sobol' indices. It requires that indices are closed from converged in order not to bias results.

- Extrema,
Based solely on the points comprised in the DOE, extrema are found and the minimal and maximal values around these two points are added. Which makes 4 new points to add to the DOE. Hence, the variability of the response surface can be verified. Indeed, extrema values could be the results of the model and not from the physics.

- Hybrid.
This last method consists of a navigator composed by any combination of the previous methods.

A detailed comparison has not been made yet. However, the MSE seems to add points where the discrepancy is high, which is coherent as the expected MSE would be located far from the sampling points. Thus, it can be used when a further exploration is required.

3.5 Miscellaneous

Aside from these improvements, the code has been largely refactored. Starting from the installation procedure. The application can now be easily installed like any Python module. It allows an easier deployment and simplify modification or improvement testing.

To use and maintain the project, a comprehensive documentation has been created. The documentation is mainly generated automatically from the docstrings within the code using an open-source tool called [Sphinx](#). Sphinx is a tool originally created for the [Python documentation](#). The documentation is written using a markup language called *reStructuredText* and then converted into *HTML*. Furthermore, it uses a theme provided by [Read the Docs](#). This web platform hosts the documentation of several well-known open source projects such as *Django*. To cover up, the documentation is clear, well formatted, complete and uses the latest technologies available—as seen in fig. 3.2.

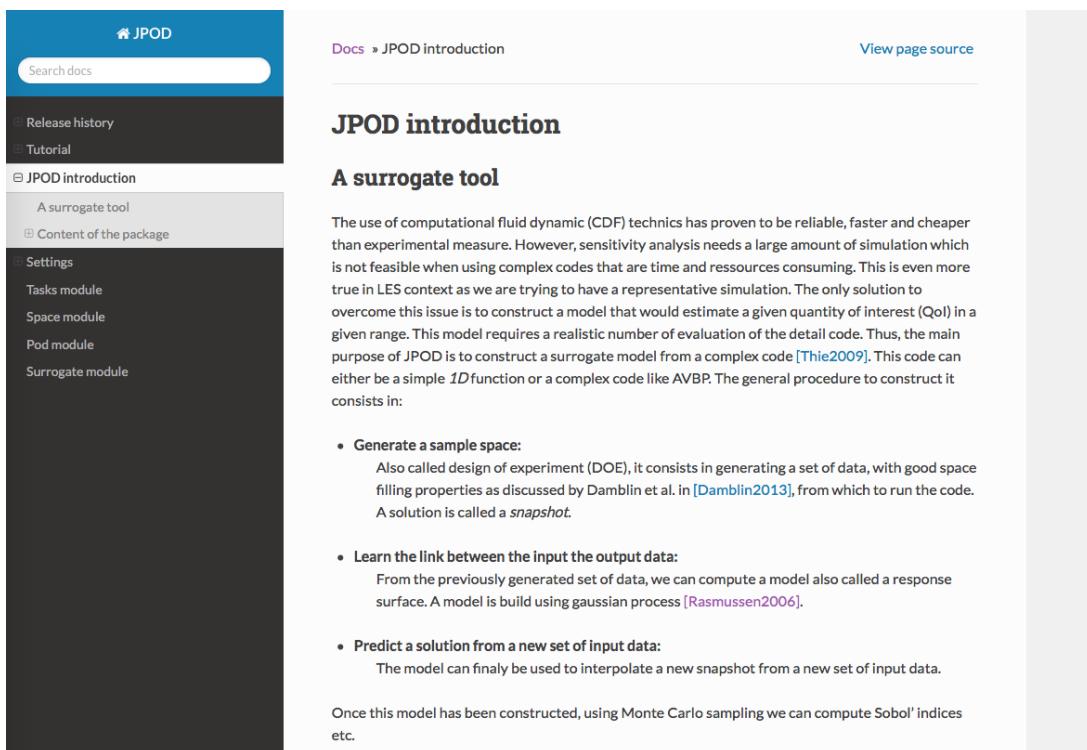


Figure 3.2 – Documentation window.

3.6 Validation cases

3.6.1 Analytical function

In order to test the code augmentation, the *Ishigami* function [16] was used on $[-\pi, \pi]^3$. It is known to have nonlinearities and it is non-monotone:

$$h(X_1, X_2, X_3) = \sin X_1 + 7 \sin^2 X_2 + 0.1X_3^4 \sin X_1. \quad (3.3)$$

Its theoretical indices are well known so we can compute the L_2 norm —also called the root mean square error—in order to assess the quality of the model:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2}. \quad (3.4)$$

	Sobol indices	Value
First order	S_1	0.314
	S_2	0.442
	S_3	0.
Second order	S_{12}	0.
	S_{13}	0.244
	S_{23}	0.
Third order	S_{123}	0.
Total order	S_{T_1}	0.557
	S_{T_2}	0.443
	S_{T_3}	0.244

Table 3.1 – Theoretical values of first-order Sobol indices as in [15].

Thus, this simple case, allows to assess the correctness of the tools. The convergence of the model has been checked regarding the learning sample size. It can be noted that a 3 parameters equation required around 100 snapshots in order to get $Q_2 > 0.8$ (fig. 3.3).

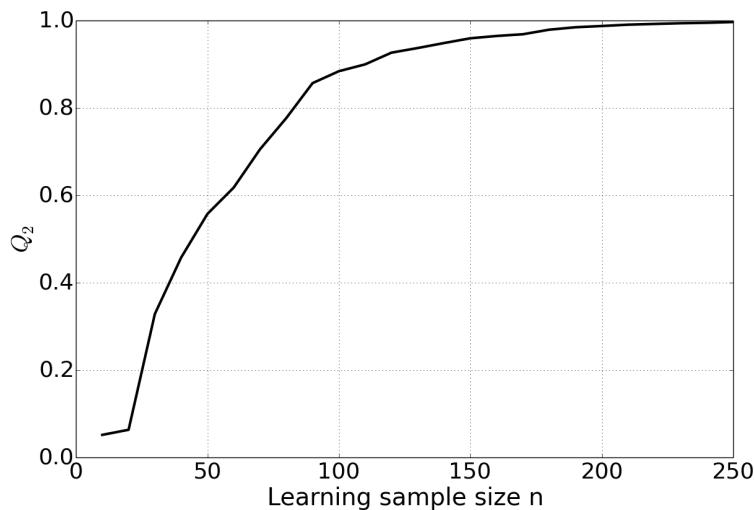
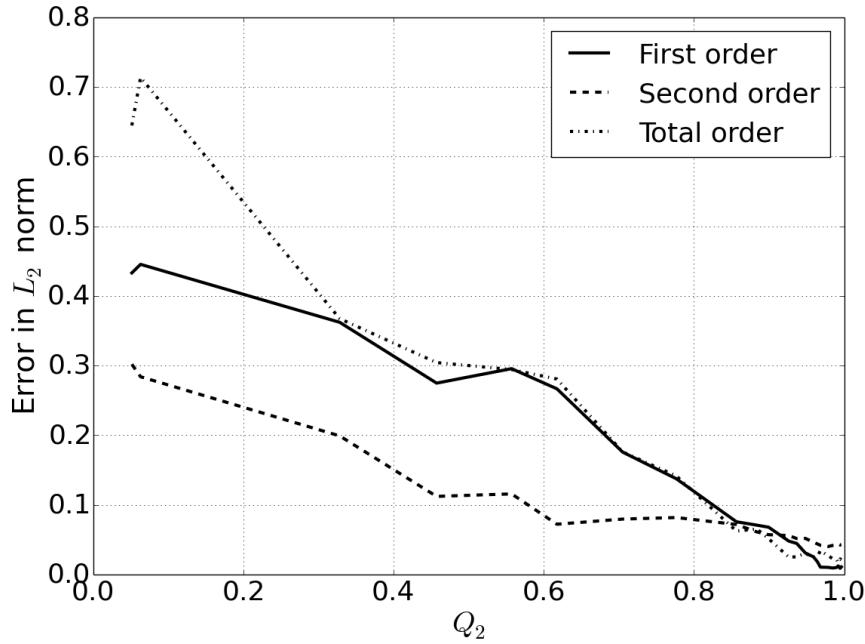
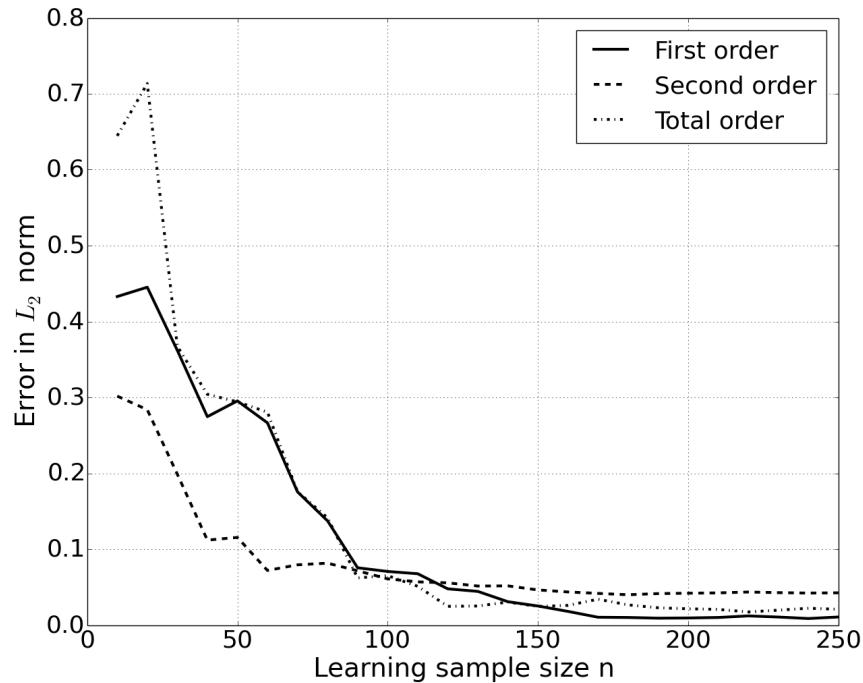


Figure 3.3 – Correlation function of the size of the learning sample.

From the following figures, it has been shown that a $Q_2 > 0.8$ is required in order to get all Sobol' indices with an error $L_2 < 10\%$. This set of figures has been generated with a large statistical space of size 50000. In the end, the tool is able to retrieve Sobol' indices correctly as shown in fig. 3.5.



(a) Error function of the POD's correlation



(b) Error function of the learning sample size

Figure 3.4 – Convergence of the Indices.

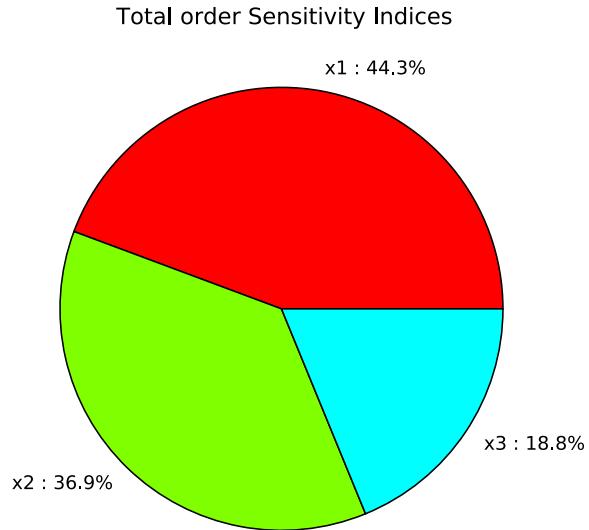


Figure 3.5 – Visual representation of Sobol indices.

Indeed, from fig. 3.6, it is noticeable that a 10% error is reachable upon 25000 samples. Here, the learning sample size was 200 in order to ensure the correctness of the model —fig. 3.7.

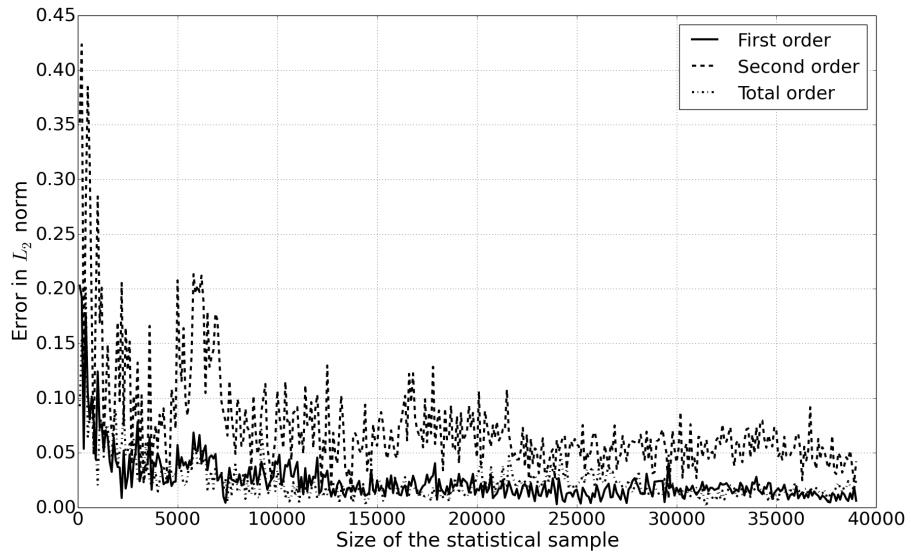


Figure 3.6 – Sobol' indices error function of the statistical space.

These results are in coherence with Marrel et al. [19] and demonstrate that the tool works as expected. The same process has been validated with other optimization functions [22].

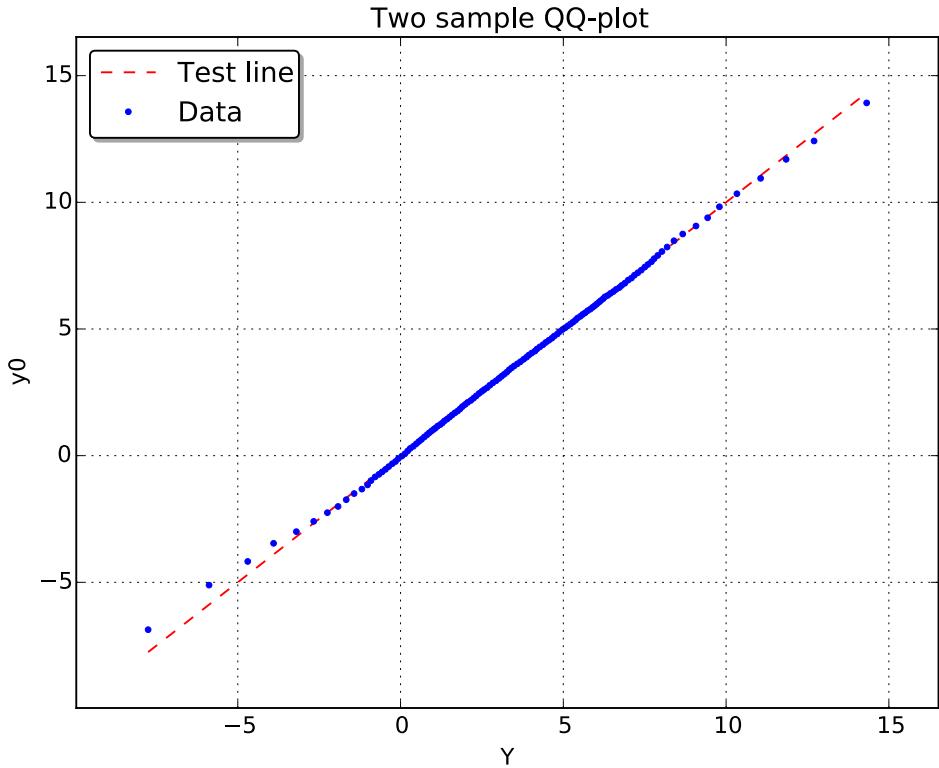


Figure 3.7 – Regression visualization on Ishigami function.

3.6.2 Channel Flow

The tool has been validated in a complex case but the Ishigami function only returns a scalar value. The purpose of this second test was to demonstrate JPOD capabilities to deal with a 1D vectorial output. Moreover, even if simple and easy to compute, this case is representative of a real physics while still being non-linear. Channel flow equations are based on *Saint Venant* equations:

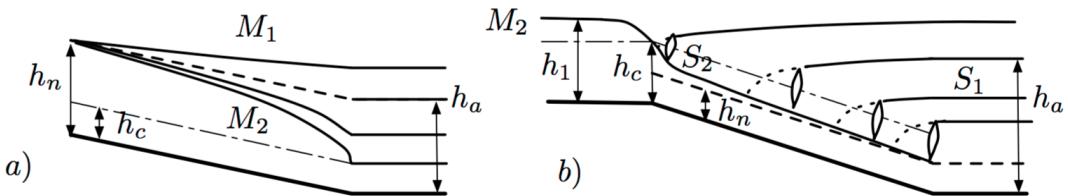


Figure 3.8 – a) Low slope. b) High slope.

The purpose of the problem is the determination of the water elevation h along the channel. The distance from the initial point is noted s and h is given by the differential equation:

$$\frac{dh}{ds} = I \frac{1 - \left(\frac{h}{h_n}\right)^{-10/3}}{1 - \left(\frac{h}{h_c}\right)^{-3}}, \quad (3.5)$$

$$h_c = \left(\frac{q^2}{gL^2}\right)^{1/3} \quad h_n = \left(\frac{q^2}{IL^2K_s^2}\right)^{3/10}, \quad (3.6)$$

with:

Q : Mass flow rate,

I : Slope,

K_s : Strickler's friction coefficient,

L : Width of the channel,

g : earth gravitational constant.

The sensitivity analysis has shown a higher influence of Strickler's coefficient over the mass flow. Indeed, the fig. 3.9 showing spatial Sobol' indices, indicates that the influence of both parameters is spatially consistent. Meaning that along all the channel, Strickler's coefficient has a greater influence compared to the mass flow rate. At the end of the domain, the divergence of the curves is due to the boundary conditions which are constant. This leads to a wrong approximation of the variance, thus a wrong statistical analysis of the indices. Furthermore, aggregated indices confirm the previous analysis done with the curves:

S_Q : 0.065,

S_{T_Q} : 0.085,

S_{K_s} : 0.905,

$S_{T_{K_s}}$: 0.926.

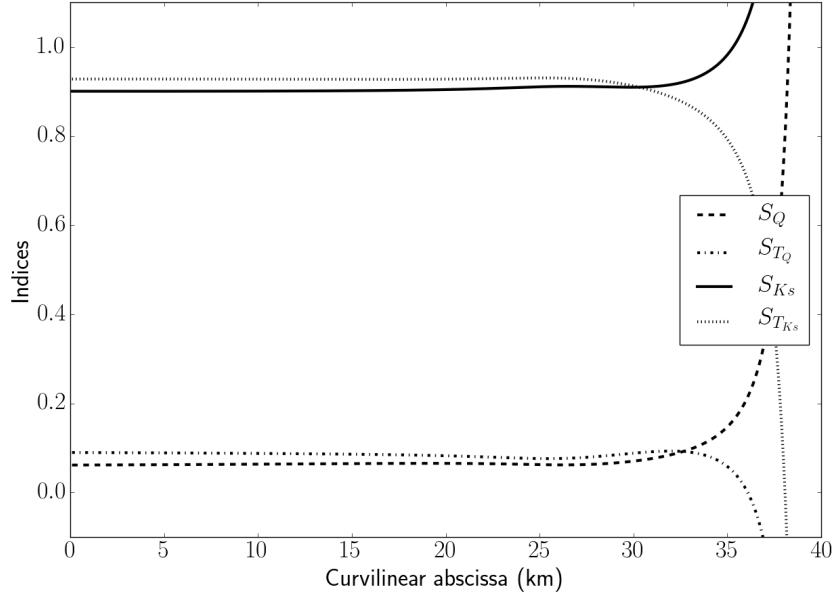


Figure 3.9 – Spatial Sobol' indices.

Finally, the last two figures show how uncertainties have been propagated. Here a normal law was used for the mass flow and a uniform one was used for Strickler's coefficient:

$$Q : N(4035, 400), \quad (3.7)$$

$$K_s : U(15, 60). \quad (3.8)$$

On fig. 3.10, the PDF is plotted along with the mean response of the model which is bounded by the standard deviation and the extrema curves. It allows to see the probability of getting a response

over another one. As a reminder, the propagation highly depends on the nature of the input PDF. In this case, the mean value is not far from the highest probability region and the response is quite flat as expected (the most influent parameter being K_s and its distribution being uniform).

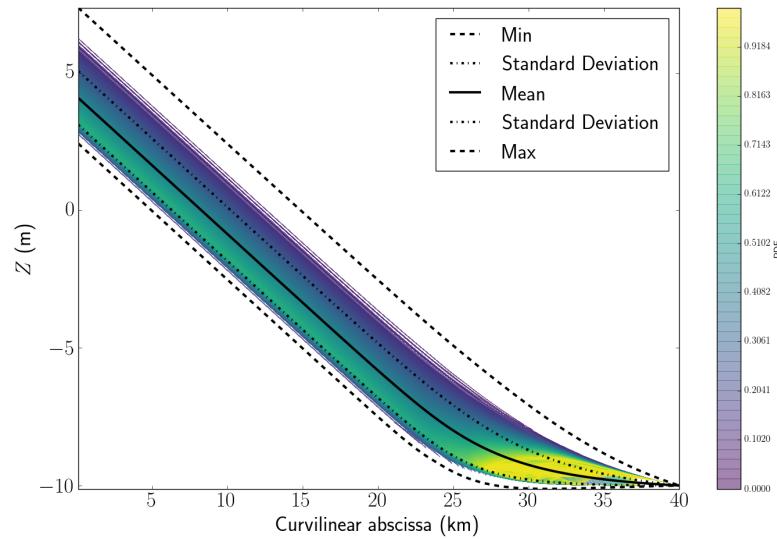


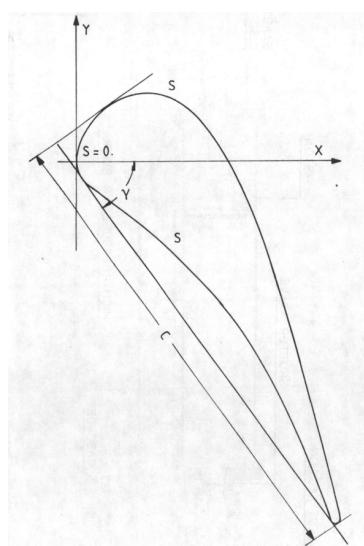
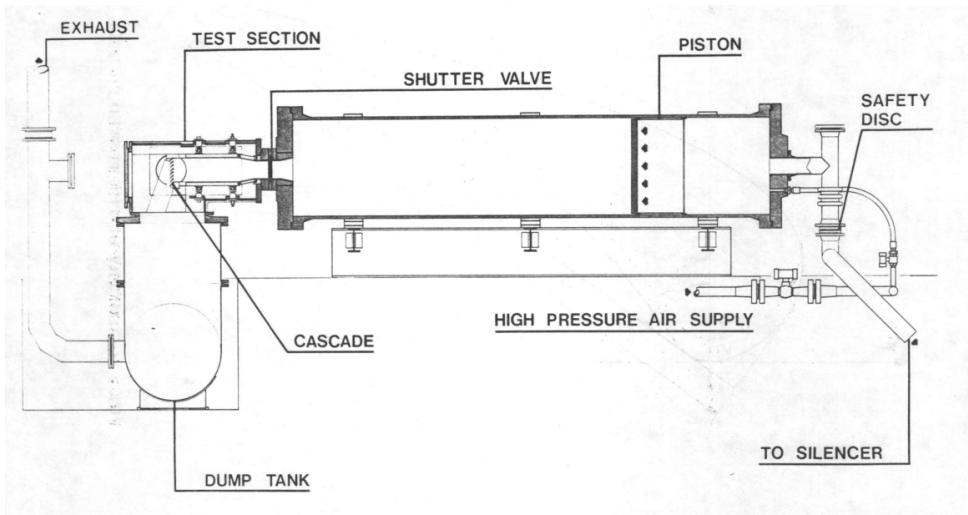
Figure 3.10 – PDF of the outputs.

4 | UQ analysis applied to a *LES* case

4.1 Experimental case

The *LS89* case is a well-addressed experimental case which aims at getting a representative heat transfer over a blade cascade located at the exit of the combustion chamber of the turbomachine. The set up has carefully been achieved so that it is commonly used to validate CFD models.

The experimental case has been conducted at the Von Karman Institute for Fluid Dynamics (VKI) by Arts et al. as described in [2]. The cascade consists of five high-pressure turbine profiles —see fig. 4.1.



c (mm)	67,647
g/c	0,850
γ (degr.)	55,0
o/c	0,2207
r_{LE}/c	0,061
r_{TE}/c	0,0105

(c) Geometrical Characteristics

Figure 4.1 – LS89 Setup.

In order to set up the simulation, we used the *MUR235* case —see table 4.1 and table 4.2.

Total temperature (K)	413,30
Total inlet pressure (bar)	1,828
Static inlet pressure (bar)	1,800
Static outlet pressure (bar)	1,049
Wall temperature (K)	301,15
Free stream turbulence (%)	6,0
Incidence angle (degr.)	0,0

Table 4.1 – Measured parameters.

	Inlet	Outlet
Total temperature (K)	413,3	
Total pressure (bar)	1,828	
Mach number	0,150	0,927
Reynolds number	$2,6471 \cdot 10^5$	$1,1521 \cdot 10^6$
Temperature (K)	411,45	352,69
Pressure (bar)	1,800	1,049
Density (kg/m ³)	1,524	1,036
Velocity(m/s)	61,00	349,02
Dynamic viscosity (kg/ms)	$2,33750 \cdot 10^{-5}$	$2,1240 \cdot 10^{-5}$
Kinematic viscosity(m ² /s)	$1,5589 \cdot 10^{-5}$	$2,0494 \cdot 10^{-5}$

Table 4.2 – Free stream conditions.

Hence we can compute the Reynolds number:

$$R_{e_2} = 1,15 \cdot 10^6. \quad (4.1)$$

Also, the through-flow time (the time for a particule to reach the outlet from the inlet) would be:

$$T = 1,11 \text{ ms}. \quad (4.2)$$

The free stream turbulence intensity is defined as:

$$T_{u_\infty} = \frac{\sqrt{u'^2}}{\bar{u}}. \quad (4.3)$$

From this experiment has been extracted the convective heat transfer coefficient defined as:

$$H = \frac{\dot{q}_{wall}}{T_0 - T_{wall}}. \quad (4.4)$$

The uncertainty on this parameter is of the order of $\pm 5\%$ and the repeatability of the results remained within 1%.

4.2 LES simulation of the LS89

4.2.1 Simulation with AVBP

The simulations have been performed using AVBP [25] which is one of the most advanced and renown CFD software. This parallel code solves the three-dimensional compressible Navier-Stokes equations for both steady state and unsteady reacting flows. The unsteady prediction is based on a

Large Eddy Simulation (LES) approach. One of the forces of AVBP is to handle hybrid unstructured meshes which allows various complex applications. Basically, the numerical schemes employed are based on a *Lax-Wendroff* (LW) or a *Taylor-Galerkin* (TTG) discretization. Several tools have been built upon AVBP in order to facilitate its use.

As in this study, the main quantity of interest has been the heat transfer coefficient at the blade, the simulation needs a well-resolved boundary layer. However, at the wall the flow is not Homogeneous Isotropic Turbulent (HIT) as the wall is not moving. The environment of the wall is defined using the distance from it. The sheer stress on the wall allows to compute the friction velocity and then the local wall *Reynolds*:

$$\tau_w = -\mu \frac{du}{dy}, \quad (4.5)$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \quad (4.6)$$

$$y^+ = \frac{u_\tau y}{\nu}. \quad (4.7)$$

The first layer is the viscous sub-layer controlled by the viscous forces ($y^+ < 5$), there is a linear evolution of the velocity here. Whereas the inertial sub-layer is controlled by inertial forces ($y^+ > 50 \rightarrow 200$), this is a log evolution. Between the two zones is an intermediate layer.

There are two approaches. If we want to calculate everything —wall resolved—, we have to go down to $y^+ 5$. If we can't afford such a mesh or we are not concerned about the boundary layer (no interaction with it), we can use a model and then we only need to go to $y^+ 50 - 100$. If we resolve to the wall, we need to check that with $y^+ 5$ we have $\nu_t \rightarrow 0$. We can verify $\nu_t \propto (y^+)^3$ and that we approximate well the sheer stress at the wall. It can be found by:

$$C_f = 2(2,236 \ln Re - 4,639)^{-2}, \quad (4.8)$$

$$\tau_w = C_f \times \frac{1}{2} \rho U_b^2. \quad (4.9)$$

$$(4.10)$$

At the wall, the Smagorinsky model doesn't give good results as the flow is not isotropic anymore. For this model to perform well, it usually requires reasonably large grid to have HIT flow thus we must activate wall models. WALE model, on the other hand, gives good results in this context. As it needed a well-resolved boundary layer. Thus, a resolved LES has been performed using the WALE model.

Regarding the numerical scheme, usually, the more the order of the scheme is, the better the solution is. However, we can have coupling effect due to the spatial discretization. Thus we need a good order and a good space discretization. The error of the scheme is divided in a dispersion term which is the real part of it and the dissipation is the imaginary part of it. Only centred schemes provide no dissipation. The Von Neumann method is utilized to determine the stability of the scheme both spatially and temporally. Only the modulus $\hat{G} = \frac{U_i^{n+1}}{U_i^n}$ is important for stability analysis. Even if the scheme is stable at high CFL it doesn't mean that the solution will not be dissipated. This simple analysis can only be done on simple PDE. Otherwise we use benchmark of the code trying to convect a pulsation, for example.

Regarding LES, the constrains are numerous as the schemes need to handle complex geometries, different scales, etc. A modification of the TTG (Two-step Taylor-Galerkin) centred scheme is used for convection. It is a 3rd order space and 3 to 4 orders in time scheme. As the LW (Lax-Wendroff) scheme is cheaper, this scheme is used to begin the computation and detect possible errors.

An artificial diffusion scheme is added to the spatial scheme. A fourth order artificial diffusion `snu4` is applied to all the domain and damp the small wiggles between nodes whereas the 2nd order `snu2` is only applied when needed and it damps the sharp gradients. Thus the `snu4` need to be the smallest value otherwise everything will be damped. These two parameters are important and should be as little as possible as it could damp a physical phenomenon. See section 4.2.2 for scheme convergence analysis.

To sum up, the configuration in table 4.3 was used.

Governing Equations	Navier-Stokes
Flow regime	Turbulent
Turbulence	WALE
Chemistry model	no chemistry
Number of species	1 (AIR)
Convective scheme	TTG4A

Table 4.3 – Flow parameters.

4.2.2 Mesh convergence

Several meshes have been generated using CENTAUR a non-structured mesher. It is a semi-automatic tool which can be combined several types of elements. Indeed, most configuration include *tetrahedric* and *prism* elements. The main advantage of the code is that it generates by default a mesh that usually works with lots of solvers.

Number of cells (M)	10	20	25
Time step (s)	$2,79 \cdot 10^{-8}$	$1,94 \cdot 10^{-8}$	$8,60 \cdot 10^{-9}$
Iterations	40 000	57 000	129 000
Cost (h(n-nodes))	~3 (10)	~5 (15)	~17 (15)

Table 4.4 – Meshes information to simulate $\sim 1,2$ ms.

All three meshes share the same topology. The blade vicinity is constituted of 5 layers of prisms and the rest of the domain is composed of tetrahedric elements.

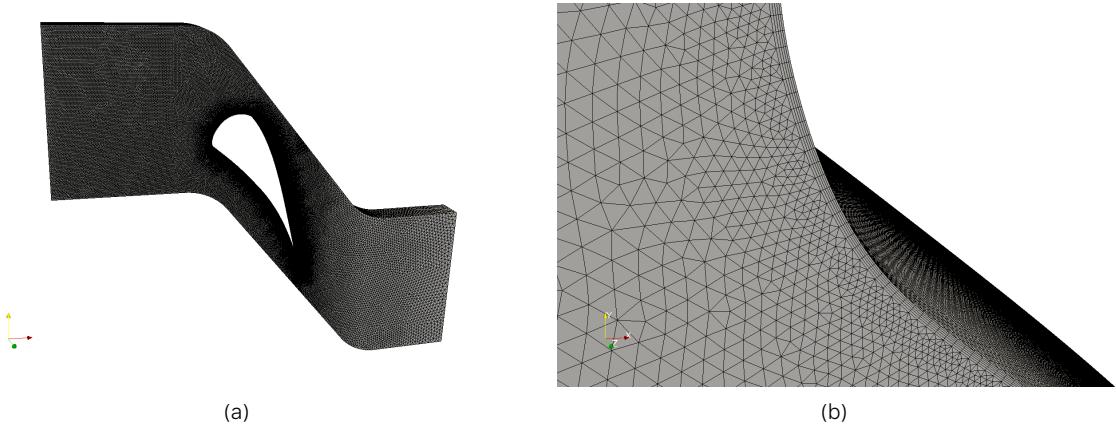
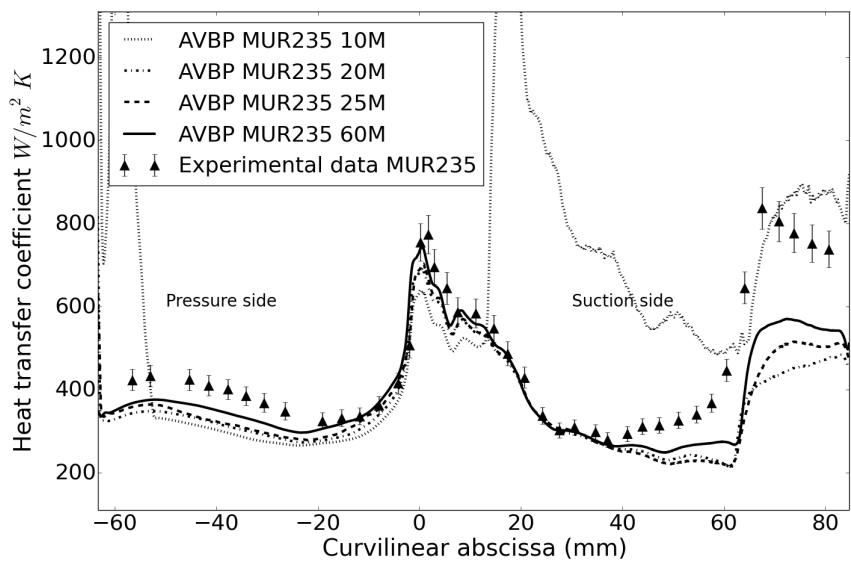
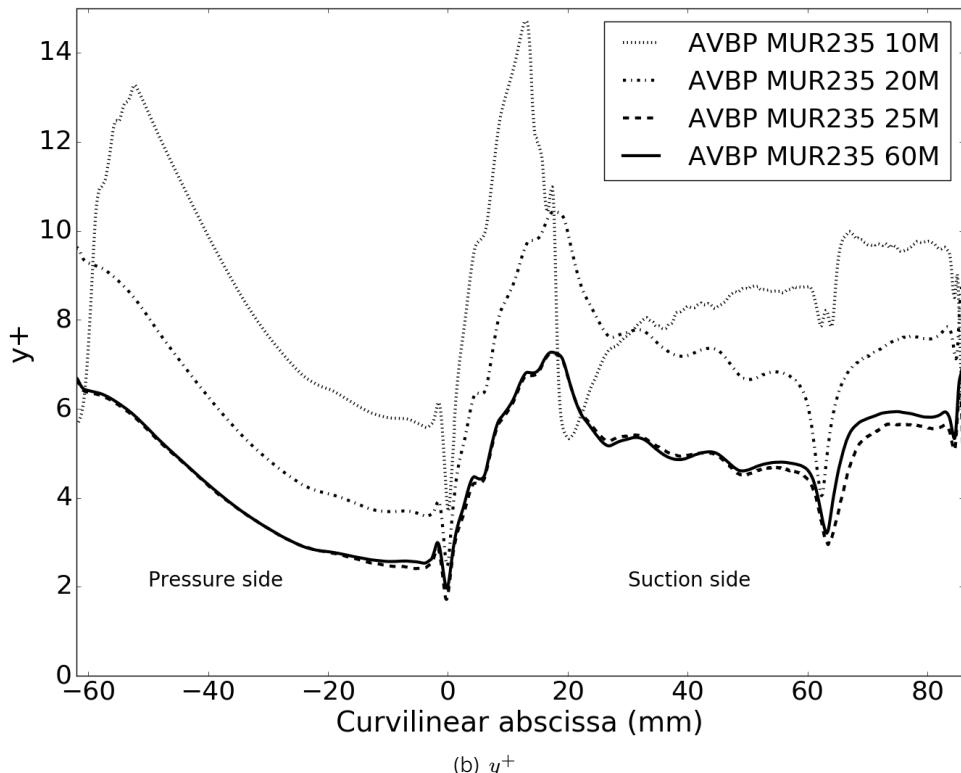


Figure 4.2 – 20M cells mesh.

Compared to the baseline with 60M cells, the convergence analysis as shown that the 20 million cells mesh could represent enough physics to be used —as [7].



(a) Heat transfer coefficient



(b) y^+

Figure 4.3 – Mesh convergence.

Aside from the mesh, a LW scheme has been tested with no luck. Thus the TTG4A scheme has been used and performed correctly.

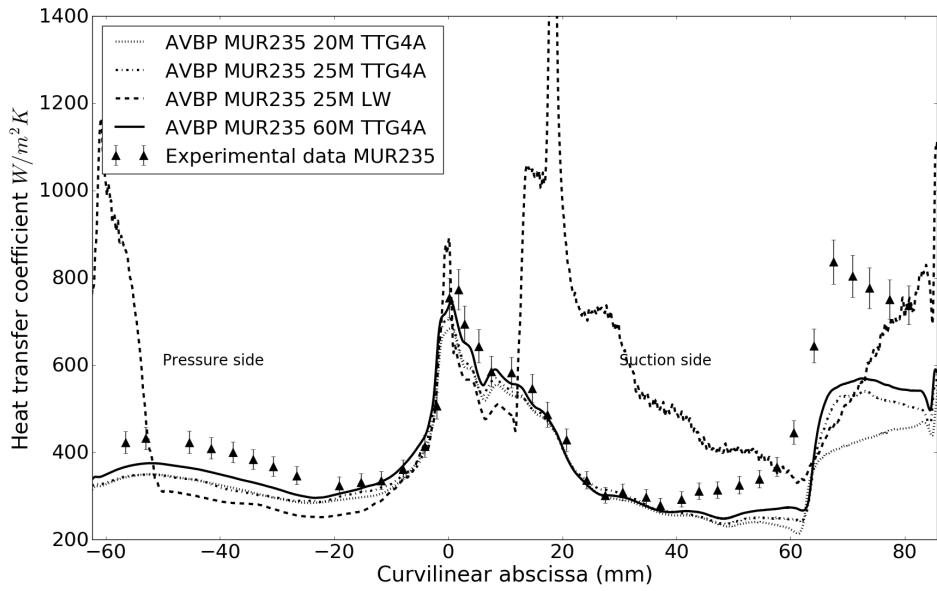


Figure 4.4 – Scheme convergence.

4.2.3 Some results

Following are some results of the simulation. The fig. 4.5 highlights the presence of a shock on the extrados of the blade. Also, the turbulence injection at the inlet of the domain is visualized. On fig. 4.7, the Kármán vortex streets are noticeable.

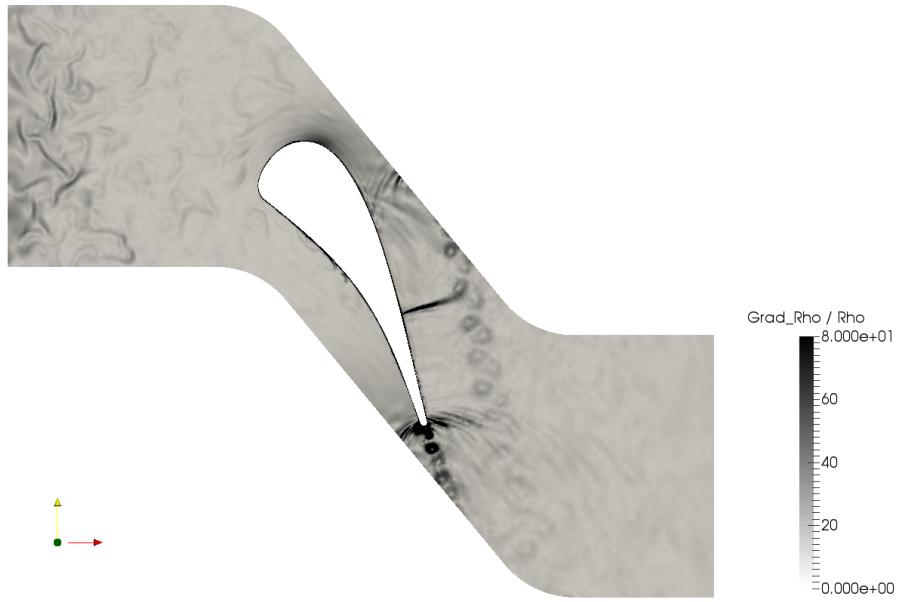


Figure 4.5 – $\frac{\nabla \rho}{\rho} (m^{-1})$ with $T_u = 30\%$.

The Q-criterion —fig. 4.6— demonstrates how the flow is turbulent by showing vortices. Here again the shock is easily noticeable.

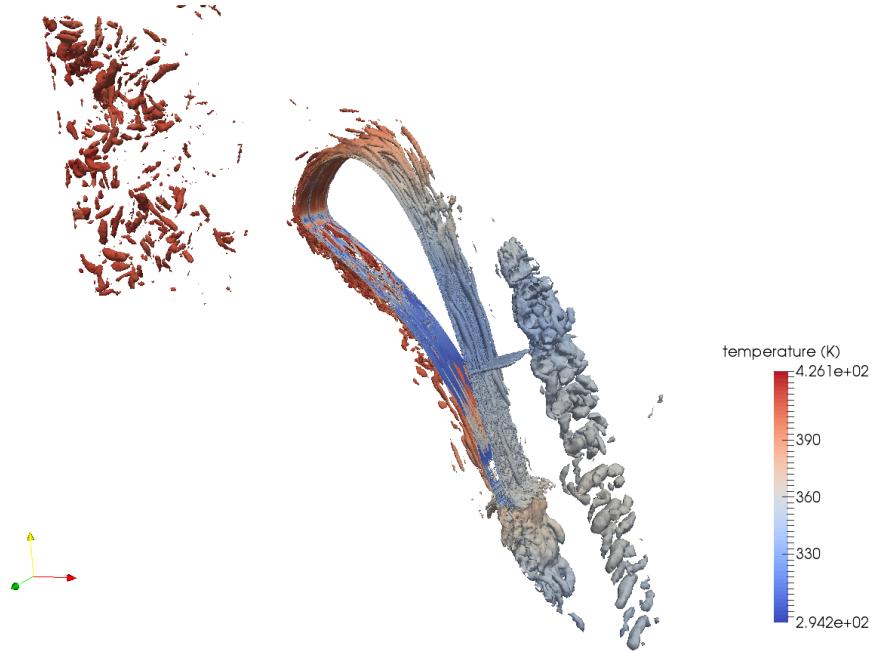


Figure 4.6 – Q -criterion iso-surfaces coloured by temperature.

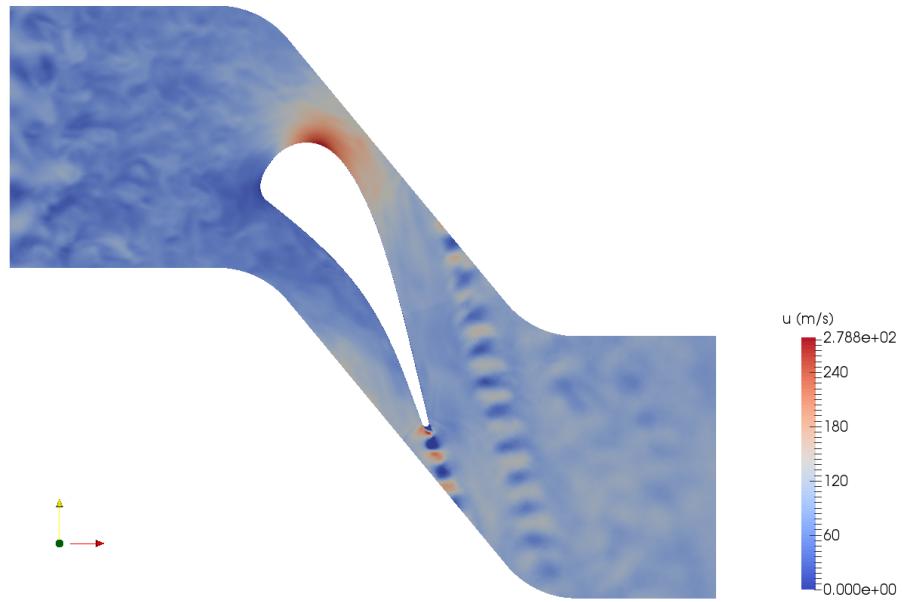


Figure 4.7 – Axial velocity (m/s) with $T_u = 30\%$.

Finally, the flow takes roughly $5,5 \cdot 10^{-4}$ s to pass the blade (leading to trailing edge). This time is the convective time and with a 20M cells mesh, it corresponds to:

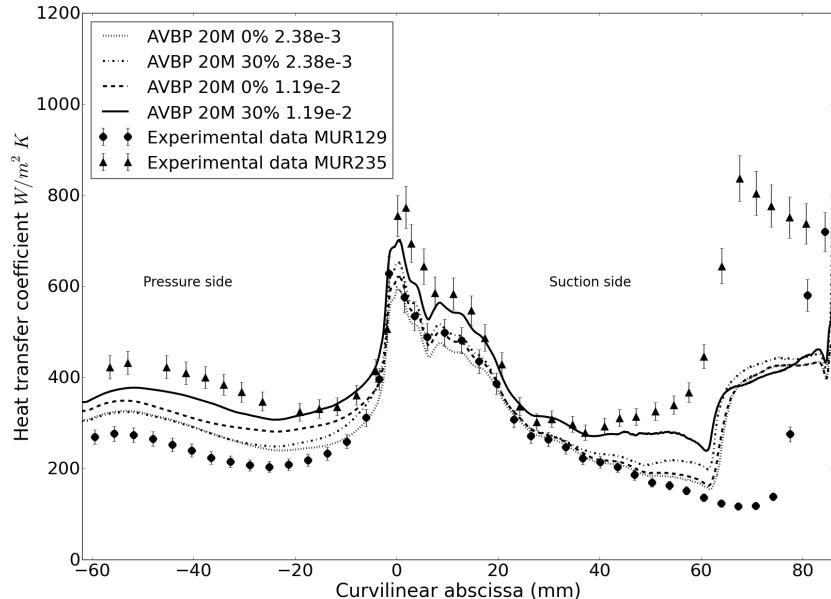
$$Iter = \frac{T}{Time\ step} \quad (4.11)$$

$$Iter = 28\,350,52 \quad (4.12)$$

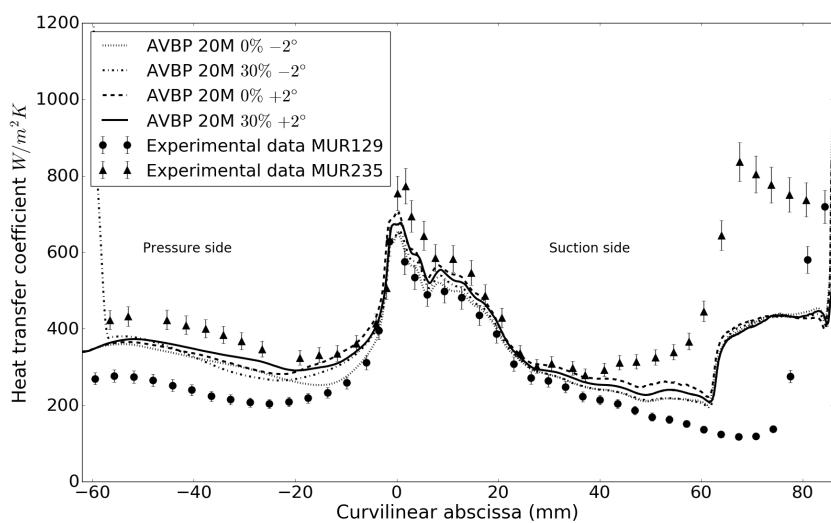
If we do the same calcul for the flow to travel from the inlet to the blade's leading edge, it needs $\sim 47\,000$ iterations.

4.3 Space of Parameters

In order to perform a UQ study in this case, two sets of variables have been considered. At first, we wanted to use the turbulence intensity with the most energetic wave length. However, the latter needs a specific mesh to be set. Indeed, for a specific length the mesh needs to be able to contain at least two vortices. Knowing that at least two nodes are required to represent a vortex, this gives a topological limitation to the range of possible values. This prevents a classical study to be performed where we would want to have continuous variables. Thus, the most energetic wave length has been replaced by the inflow angle α . The following figure is the result of simulations performed at the space parameters' corners. The initial solution consisted of a converged simulation of the MUR235 case.



(a) $T_u - \Delta_e$



(b) $T_u - \alpha$

Figure 4.8 – Walk through Space of Parameters.

First analysis upon the heat transfer coefficient can be drawn from here. Beginning with the case T_u and Λ_e , an increasing T_u or Λ_e cause an increase in the heat transfer coefficient H . Considering now the case T_u and α on the extrado (from 0 to 25 mm), an increasing T_u or α cause an increase of H . In both cases, T_u seems to impact more than Λ_e or α . These results are confirmed by Collado et al. [7]. A further analysis would require more computations to ensure that the response is smooth. This is why a UQ analysis is required.

Following was a convergence study in order to measure the requested time per simulation.

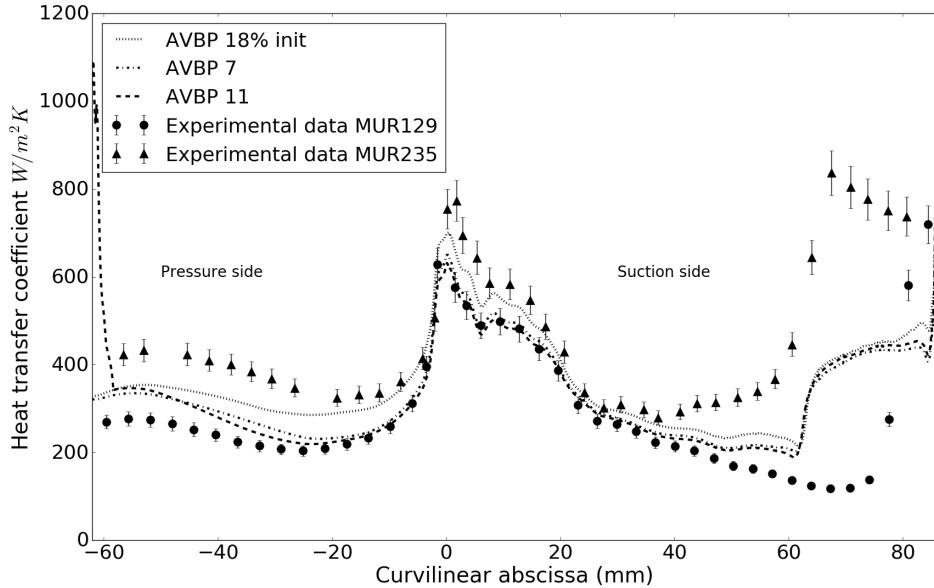


Figure 4.9 – Convergence Study for $T_u - \alpha$.

The fig. 4.9 shows that converged results need at least 210 000 iterations (AVBP 7) from -10 to 55 mm. After 60 mm, the presence of the shock is affecting to much the solution and the simulation is not able to retrieve the correct values. The MUR129 case was performed without turbulence injections, hence the evaluation of the convergence. Finally, this leads to a computational effort of $TCPU = 6\,600$ h per simulation. From section 4.2.2, this corresponds to 5.7 convective times.

To conclude, in this study, we are interested in the impact of the angle of attack and the turbulence intensity upon the heat transfer coefficient of the blade.

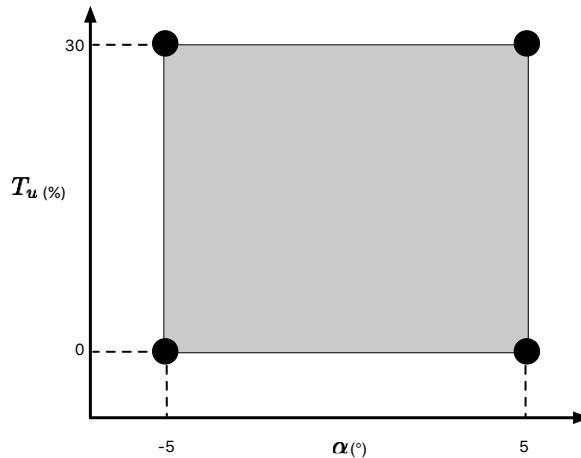


Figure 4.10 – Space of Parameters.

4.3.1 Passot-Pouquet turbulence spectrum

In order to inject turbulence during the experiment, a grid has been installed in the wind tunnel. This injection can be modelled using a homogeneous isotropic turbulence. Here, the energy spectrum used is a Passot-Pouquet synthetic spectrum of the form:

$$E(k) = A \left(\frac{k}{k_e} \right)^4 \exp -2 \left(\frac{k}{k_e} \right)^2, \quad (4.13)$$

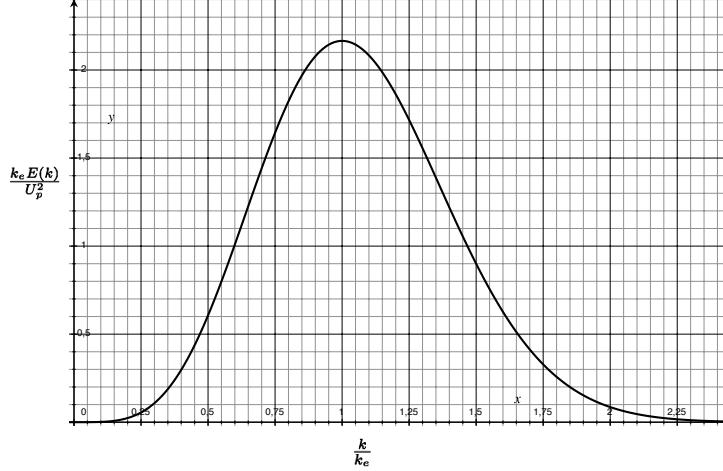


Figure 4.11 – Energy spectrum for the Passot-Pouquet expression.

where the amplitude of the spectrum $A = 16 \frac{u_p^2}{k_e}$ and the wave number of the most energetic mode $k_e = \frac{2\pi}{\lambda_e}$. A can be linked to the RMS velocity u_p and k_e by integrating the energy spectrum. The latter corresponds to the most energetic eddies and thus to the largest ones.

The integral scale Λ_f corresponds to the size of the largest eddies and is linked to the most energetic wave length by:

$$\Lambda_f = \frac{\sqrt{2\pi}}{k_e} = \frac{\lambda_e}{\sqrt{2\pi}} \simeq 0,4\lambda_e. \quad (4.14)$$

4.4 Setting up JPOD

As stated previously, JPOD has been complemented by uncertainty quantification capabilities. Being an all automated tool, it has been set up to create a model and perform a full UQ study. The space of parameters has been created using a low discrepancy sequence of Sobol' with initially 20 samples.

Every snapshot consists of a folder containing a whole AVBP setup with a boundary conditions file automatically adapted. The output of each snapshot consists of a tecplot file filled with 3220 values of H discretizing the blade profile. This post processing is also created automatically after the simulation completion.

Regarding UQ analysis, the Sobol' method has been used coupled with aggregated indices. To do so, these two uniform PDF has been set:

$$T_u : U(0, 30) \quad \alpha : U(-5, 5). \quad (4.15)$$

4.5 Response Surface

Before conducting any analysis, the response surface of the model has been plotted in fig. 4.12. The response being a vectorial output, it has been integrated over the chord line has the flux was normal to it.

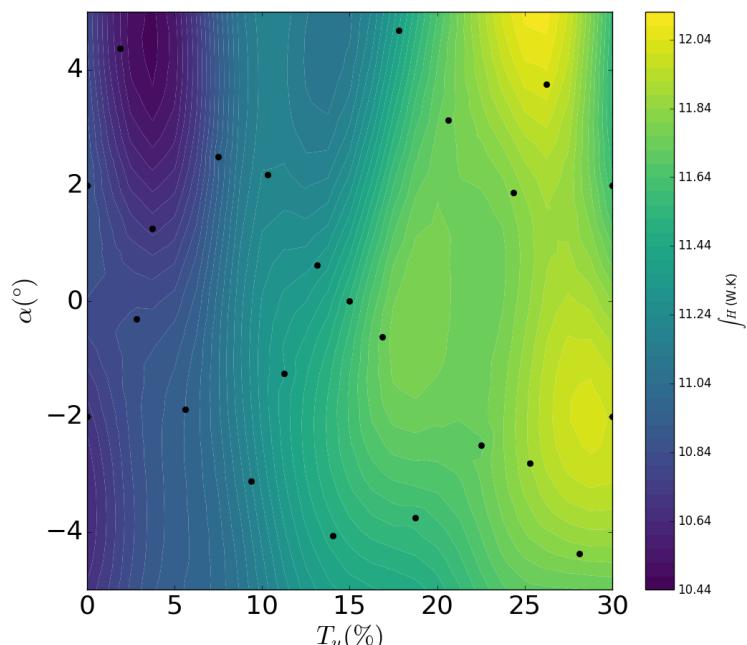
The first thing to notice is the correct repartition of the sample points within the space of parameters ensuring that most of the effects would be captured. Secondly, the predictions made using the model are in accordance with the observations made previously. The heat transfer coefficient increases with the turbulence intensity and is stable regarding the angle of the incoming flow. The model is said to be additive regarding the turbulence intensity. Thus, it seems that no further analysis would be required to assess the importance factors.

As described in section 2.5.2, the model quality has been evaluated by Leave-One-Out as:

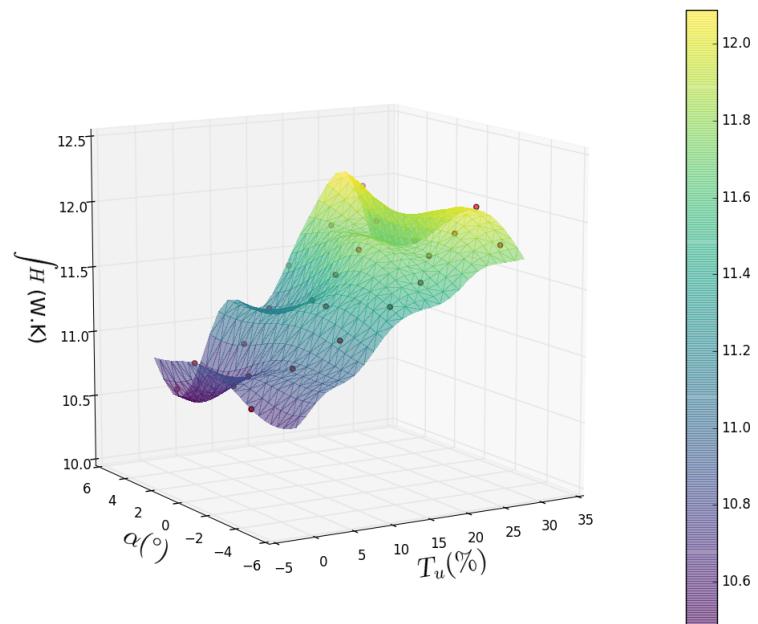
$$Q_2 = 0.684. \quad (4.16)$$

Regarding the POD itself here is JPOD's output:

```
1 POD summary:  
2 modes filtering tolerance : 0.99  
3 dimension of parameter space : 2  
4 number of snapshots : 24  
5 number of data per snapshot : 3200  
6 maximum number of modes : 100  
7 number of modes : 18  
8 modes :  
9 [ 3510.32660929    886.4432429    449.30945659    387.35370385    274.49359826  
10      226.59407407    189.98350168    138.45086113    112.51637688    77.82942743  
11      68.26135103    61.38800221    40.69511082    34.11905374    29.79768229 ]
```



(a) 2D



(b) 3D

Figure 4.12 – Heat Transfer coefficient response surface.

4.6 UQ analysis

4.6.1 Uncertainty Propagation

The fig. 4.13 reveals nothing out of the expected regarding the heat transfer coefficient propagation. As the two input distributions are uniform and the model is additive, the mean is centred between the extrema values.

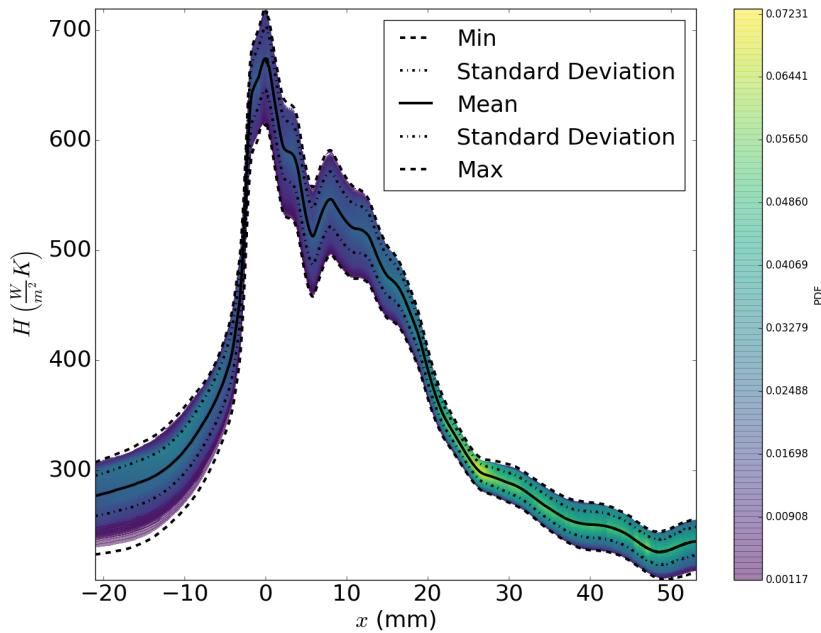


Figure 4.13 – Probability Density Function and moments along the chord line.

4.6.2 Sensitivity Analysis

As the response surface suggested, the heat transfer coefficient is mainly affected by the variation of the turbulence intensity. Total aggregated indices are:

$$T_u = 0.998, \quad (4.17)$$

$$\alpha = 0.112. \quad (4.18)$$



- Again, this is the result of a model which is able to represent $\sim 70\%$ of the variability based on Q_2 by LOOCV and considering two uniform input distribution of parameters.

Regarding the spatial evolution of the indices, the fig. 4.14 confirm that the response is not spatially dependent as the indices are stable all along the curvilinear abscissa.

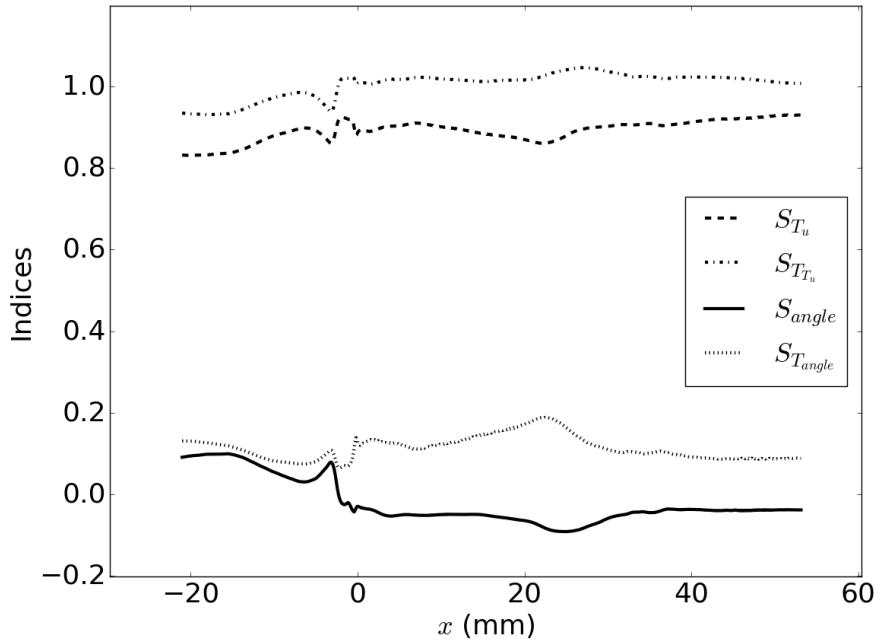


Figure 4.14 – Sobol' indices along the chord line.

4.7 Conclusions and recommendations

These cases demonstrated the ability of JPOD to deal with a high CPU time simulation of a realistic engineering configuration. Thus, it would be possible to adapt the workflow to any AVBP simulations as it is planned to be.

Even if the tool is working as designed, here are some improvement strategies:

- OpenTURNS has recently been released, the new sensitivity class contains several improvements that would benefit JPOD,
- Todays implementation is still bugged and not suited for HPC. Refactoring the code will soon be of prime importance,
- All resampling strategies haven't been tested. A further analysis would be required,
- The UQ class is only able to deal with scalar and 1D vectorial output.

Conclusion

Importance of the Internship

The objective of my mission was essential for the team in a sense that the *CERFACS* quickly wants to gain uncertainty capabilities. As the tool is working as expected, it will allow any simulation to be used and analysed in this statistical framework.

Use of the Knowledge

During all the implementation of the tool, I have been able to use and enhance my developing skills. I am now confident in creating applications with *Python* and relative technologies. The implementation of the documentation was also interesting, as it is of prime importance when it comes to maintaining a project.

Relationship With the Team

Even if I was working with relative autonomy, I was part of a small team and tried to understand the work of everyone. I had the chance to contribute in a workshop and attended several thesis dissertations.

Self-Enrichment

This internship had been a great experience in any way. In terms of knowledge, I learned a lot about Computational Fluid Dynamic and greatly benefited from all training sessions. I enjoyed using my computing skills in order to solve a physical problem. Moreover, Uncertainty Quantification being an ongoing topic of interest, it was challenging and interesting to work on it.

I was glad to be able to contribute to a project of this dimension at *CERFACS*. It has been a privilege for me to be here as a trainee and I now have a good insight of the company.

This was a great opportunity and I had a pleasant time working with everyone. I have been proposed to conduct a PhD at *CERFACS* which I am pleased to do.

Bibliography

- [1] Anindya Chatterjee. "An introduction to the proper orthogonal decomposition". In: *Current Science* 78.7 (2000).
- [2] Tony Arts, M Lambert de Rouvroit, and A W Rutherford. *Aero-Thermal Investigation of a Highly Loaded Transonic Linear Turbine Guide Vane Cascade*. Tech. rep. 174. von Karman Institute for Fluid Dynamics, 1990.
- [3] Michaël Baudin et al. "OpenTURNS: An industrial software for uncertainty quantification in simulation". In: (June 2015). arXiv: [1501.05242](https://arxiv.org/abs/1501.05242).
- [4] Geoff Bohling. *Introduction to Geostatistics and Variogram Analysis*. Tech. rep. October. 2005, pp. 1–20. doi: [10.1162/0162287054769931](https://doi.org/10.1162/0162287054769931).
- [5] Geoff Bohling. *Kriging*. Tech. rep. October. 2005, pp. 1–20.
- [6] Thierry Braconnier and Marc Ferrier. *Jack Proper Orthogonal Decomposition (JPOD) for Steady Aerodynamic Model*. Tech. rep. 2009, pp. 1–33.
- [7] E. Collado Morata et al. "Effects of free-stream turbulence on high pressure turbine blade heat transfer predicted by structured and unstructured LES". In: *International Journal of Heat and Mass Transfer* 55.21–22 (Oct. 2012), pp. 5754–5768. issn: 00179310. doi: [10.1016/j.ijheatmasstransfer.2012.05.072](https://doi.org/10.1016/j.ijheatmasstransfer.2012.05.072). URL: <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2012.05.072> http://linkinghub.elsevier.com/retrieve/pii/S0017931012003936.
- [8] L Cordier and M Bergmann. "Réduction de dynamique par décomposition orthogonale aux valeurs propres (POD)". In: *Ecole de printemps OCET* 7563 (2006). URL: <http://www.math.u-bordeaux1.fr/%7B-%7Dmbergman/PDF/OuvrageSynthese/OCET06.pdf>.
- [9] Guillaume Damblin et al. "Numerical studies of space filling designs : optimization of Latin Hypercube Samples and subprojection properties". In: *Journal of Simulation* (2013), pp. 276–289.
- [10] David Draper. "Assessment and Propagation of Model Uncertainty". In: *Journal of the Royal Statistical Society B* 57.1 (1995), pp. 45–97.
- [11] George S. Fishman. *Monte Carlo: concepts, algorithms, and applications*. Springer S. Springer, 1996. ISBN: 978-1-4419-2847-4. doi: [10.1007/978-1-4757-2553-7](https://doi.org/10.1007/978-1-4757-2553-7).
- [12] Alexander I J Forrester and Andy J. Keane. "Recent advances in surrogate-based optimization". In: *Progress in Aerospace Sciences* 45.1–3 (2009), pp. 50–79. issn: 03760421. doi: [10.1016/j.paerosci.2008.11.001](https://doi.org/10.1016/j.paerosci.2008.11.001).
- [13] Hasan Gunes, Sirod Sirisup, and George Em Karniadakis. "Gappy data: To Krig or not to Krig?" In: *Journal of Computational Physics* 212.1 (2006), pp. 358–382. issn: 00219991. doi: [10.1016/j.jcp.2005.06.023](https://doi.org/10.1016/j.jcp.2005.06.023).
- [14] Bertrand Iooss. "Revue sur l'analyse de sensibilité globale de modèles numériques". In: *Journal de la Société Française de Statistique* 152.1 (2010), pp. 3–25.
- [15] Bertrand Iooss, Mathieu Ribatet, and Amandine Marrel. "Global Sensitivity Analysis of Stochastic Computer Models with joint metamodels". In: 2006 (2008). arXiv: [0802.0443](https://arxiv.org/abs/0802.0443). URL: <http://arxiv.org/abs/0802.0443>.
- [16] T. Ishigami and T. Homma. "An importance quantification technique in uncertainty analysis for computer models". In: *IEEE* (1990), pp. 398–403. doi: [10.1109/ISUMA.1990.151285](https://doi.org/10.1109/ISUMA.1990.151285).
- [17] D G Krige, M Guarascio, and F A Camisani-Calzolari. "Early South African geostatistical techniques in today's perspective". In: *Geostatistics* 1 (1989), pp. 1–19.
- [18] Amandine Marrel, Nathalie Saint-Geours, and Matthias De Lozzo. "Sensitivity Analysis of Spatial and/or Temporal Phenomena". In: *Handbook of Uncertainty Quantification*. Ed. by Roger Ghanem, David Higdon, and Houman Owhadi. Cham: Springer International Publishing, 2015, pp. 1–31. ISBN: 978-3-319-11259-6. doi: [10.1007/978-3-319-11259-6_39-1](https://doi.org/10.1007/978-3-319-11259-6_39-1).

- [19] Amandine Marrel et al. "Calculations of Sobol indices for the Gaussian process metamodel". In: *Reliability Engineering and System Safety* 94.3 (2009), pp. 742–751. **ISSN:** 09518320. **DOI:** [10.1016/j.ress.2008.07.008](https://doi.org/10.1016/j.ress.2008.07.008). arXiv: [0802.1008](https://arxiv.org/abs/0802.1008).
- [20] Amandine Marrel et al. "Global sensitivity analysis for models with spatially dependent outputs". In: *Environmetrics* 22.3 (2011), pp. 383–397. **ISSN:** 11804009. **DOI:** [10.1002/env.1071](https://doi.org/10.1002/env.1071). arXiv: [0911.1189](https://arxiv.org/abs/0911.1189).
- [21] Georges Matheron. "Principles of Geostatistics". In: *Economic Geology* 58 (1963), pp. 1246–1266.
- [22] Marcin Molga and Czesław Smutnicki. "Test functions for optimization needs". In: c (2005), pp. 1–43. **URL:** <http://www.bioinformaticslaboratory.nl/twikitdata/pub/Education/NBICResearchSchool/Optimization/VanKampen/BackgroundInformation/TestFunctions-Optimization.pdf>.
- [23] Max D Morris. "Factorial Sampling Plans for Preliminary Computational Experiments". In: *Technometrics* 33.2 (1991), pp. 161–174. **ISSN:** 0040-1706. **DOI:** [10.2307/1269043](https://doi.org/10.2307/1269043).
- [24] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.2825–2830 (Jan. 2012). arXiv: [1201.0490](https://arxiv.org/abs/1201.0490). **URL:** <http://arxiv.org/abs/1201.0490>.
- [25] Thierry Poinsot. *AVBP formation*. 2016.
- [26] CE Rasmussen and C Williams. *Gaussian processes for machine learning*. MIT Press, 2006. **ISBN:** 026218253X.
- [27] G. K. Robinson. "That BLUP is a good thing: the estimation of random effects". In: *Statistical Science* 6.1 (1991), pp. 15–32. **ISSN:** 0883-4237. **DOI:** [10.1214/ss/1177011926](https://doi.org/10.1214/ss/1177011926).
- [28] Jerome Sacks et al. "Design and Analysis of Computer Experiments". In: *Statistical Science* 4.4 (Nov. 1989), pp. 409–423. **ISSN:** 0883-4237. **DOI:** [10.1214/ss/1177012413](https://doi.org/10.1214/ss/1177012413). **URL:** <http://projecteuclid.org/euclid.ss/1177012413>.
- [29] Andrea Saltelli, Stefano Tarantola, and Karen Chan. "A quantitative model-independent method for global sensitivity analysis of model output". In: *Technometrics* 41.1 (1999), p. 356.
- [30] Céline Scheidt. "Analyse statistique d'expériences simulées : Modélisation adaptative de réponses non-régulières par krigage et plans d'expériences, Application à la quantification des incertitudes en ingénierie des réservoirs pétroliers". PhD thesis. Louis Pasteur, 2006.
- [31] I.M Sobol'. *Sensitivity analysis for nonlinear mathematical models*. 1993.
- [32] Ronald E Walpole et al. *Probability and Statistics for Engineers and Scientists*. Vol. 9th. 2012, p. 816. **ISBN:** 0132047675.